PingFederate Integrations

July 1, 2025



Copyright

All product technical documentation is
Ping Identity Corporation
1001 17th Street, Suite 100
Denver, CO 80202
U.S.A.

Refer to https://docs.pingidentity.com for the most current product documentation.

Trademark

Ping Identity, the Ping Identity logo, PingAccess, PingFederate, PingID, PingDirectory, PingDataGovernance, PingIntelligence, and PingOne are registered trademarks of Ping Identity Corporation ("Ping Identity"). All other trademarks or registered trademarks are the property of their respective owners.

Disclaimer

The information provided in Ping Identity product documentation is provided "as is" without warranty of any kind. Ping Identity disclaims all warranties, either express or implied, including the warranties of merchantability and fitness for a particular purpose. In no event shall Ping Identity or its suppliers be liable for any damages whatsoever including direct, indirect, incidental, consequential, loss of business profits or special damages, even if Ping Identity or its suppliers have been advised of the possibility of such damages. Some states do not allow the exclusion or limitation of liability for consequential or incidental damages so the foregoing limitation may not apply.

Table of Contents

PingFederate and PingOne Integrations . . . 40
Agentless Integration Kit . . . 51
Overview of the identity provider integration . . . 54
Overview of the identity provider SSO flow . . . 55
Overview of the service provider SSO flow . . . 58
Custom application setup . . . 59
Developing your application . . . 59
Development considerations . . . 60
Authentication methods . . . 62
Configuring certificate authentication . . . 64
Attribute pickup process . . . 66
Attribute drop-off process . . . 68
External consent approval . . . 70
Deploying the integration files . . . 70
Configuring a Reference ID IdP Adapter instance . . . 71
Reference ID IdP Adapter settings reference . . . 72
Configuring a Reference ID SP Adapter instance . . . 75
Reference ID SP Adapter settings reference . . . 76
Attribute formatting . . . 79
Reference ID Adapter endpoints . . . 80
Sample application setup . . . 80
Deploying the integration files and sample applications . . . 80
Deploying the sample configuration archive . . . 81
Sample application user accounts . . . 82
Testing IdP-initiated SSO . . . 82
Testing SP-initiated SSO . . . 82
Troubleshooting
Enabling logging . . . 83
Release notes
Changelog . . . 83
Known issues and limitations . . . 87
Download manifest . . . 87
Amazon integrations . . . 88
Amazon Login Integration Kit . . . 90
Overview of the SSO flow . . . 91
Authentication API Support . . . 92
Setup
Deploying the integration files . . . 92
Registering PingFederate as a security profile in Amazon . . . 93

Configuring an adapter instance 94
Amazon IdP Adapter settings reference 94
Available user attributes 97
Integrating social sign-on into your application 97
Troubleshooting
Enabling debug logging 99
Release notes
Changelog 100
Known issues and limitations 100
Download manifest 101
AWS IAM Identity Center Provisioner 102
User and group management 103
Setup
Upgrading an existing deployment 105
Deploying the integration files 106
Enabling provisioning and single sign-on in PingFederate 106
Exporting SAML metadata from PingFederate 107
Registering PingFederate as an identity provider in AWS Single Sign-On . . 108
Creating a connection 108
Provisioning options reference 109
Supported attributes reference 111
Access control in AWS using session tags 113
Release notes
Changelog 113
Known issues and limitations 113
Download manifest 114
Amazon Web Services Connector 114
Setup
Upgrading an existing deployment 115
Create security credentials in Amazon Web Services 116
Deploying the integration files 117
Creating a provisioning connection 118
Provisioning options 119
Supported attributes 120
Configuring single sign-on 120
Exporting SAML metadata from PingFederate 121
Creating an identity provider in Amazon Web Services 121
Creating a federation role in Amazon Web Services 122
Creating a single sign-on connection 122
Troubleshooting
Show Amazon API errors in the server log 123
Release notes
Changelog 124
Known issues and limitations 124

Download manifest . . . 125
Amazon Web Services Session Tags Integrations Guide . . . 125
Session tags in PingFederate
Configuring AWS session tags for PingFederate SAML connections . . . 126
Configuring AWS session tags for PingFederate OIDC connections . . . 128
Session tags in PingOne
Configuring AWS session tags for PingOne SAML connections . . . 130
Configuring AWS session tags for PingOne OIDC connections . . . 132
Apache Linux Integration Kit . . . 133
Overview of the SSO flow . . . 136
Apache agent information . . . 137
Setup
Upgrading an existing deployment . . . 139
Updating the OpenToken Adapter . . . 141
Configuring an OpenToken SP Adapter instance . . . 141
Apache Agent setup . . . 142
Deploying the Apache agent . . . 142
Configuring the Apache agent . . . 144
Setting up virtual hosts . . . 144
Testing the Apache agent . . . 145
Release notes
Changelog . . . 145
Known issues and limitations . . . 149
Download manifest . . . 150
Apple Login Integration Kit . . . 151
Overview of the SSO flow . . . 154
Authentication API Support . . . 155
Setup
Deploying the integration files . . . 155
Registering an Apple app . . . 156
Registering an Apple service . . . 156
Creating an Apple private key . . . 157
Configuring an adapter instance . . . 158
Apple IdP Adapter settings reference . . . 158
Available user attributes . . . 161
Integrating Sign in with Apple into your application . . . 161
Troubleshooting
Enabling debug logging . . . 163
Release notes
Changelog . . . 164
Known issues and limitations . . . 165
Download manifest . . . 165
Aquera Provisioner . . . 166
User and group management . . . 169

Setup
Upgrading an existing deployment . . . 171
Deploying the integration files . . . 172
Enabling single sign-on in PingFederate . . . 173
Add an application in Aquera . . . 174
Creating a provisioning connection . . . 174
Aquera Connector settings reference . . . 175
Supported attributes reference . . . 178
Configuring single sign-on . . . 182
Adding SSO to a connection . . . 183
PingFederate SSO details for the service provider . . . 183
Release notes
Changelog . . . 184
Known issues and limitations . . . 184
Download manifest . . . 185
Atlassian integrations . . . 185
Atlassian Cloud Provisioner . . . 187
User and group management . . . 188
Setup
Upgrading an existing deployment . . . 190
Deploying the integration files . . . 191
Getting an Atlassian API key . . . 191
Creating a provisioning connection . . . 192
Provisioning options reference . . . 193
Supported attributes reference . . . 194
Configuring single sign-on . . . 195
Enabling single sign-on in PingFederate . . . 196
Enabling single sign-on in Atlassian . . . 196
Creating a single sign-on connection . . . 197
Release notes
Changelog . . . 198
Known issues and limitations . . . 199
Download manifest . . . 199
Atlassian Integration Kit . . . 200
Overview of the SSO flow . . . 201
Setup
Deploying the Agentless Integration Kit . . . 202
Upgrading instructions . . . 202
Defining the Authenticator configuration directory . . . 202
Uploading the plugin . . . 204
Configure the plugin . . . 205
Mutual SSL authentication . . . 207
Configuring PingFederate for mutual SSL authentication . . . 207

Configuring Atlassian . . . 209
Basic authentication. . . . 211
Configuring PingFederate for basic authentication . . . 211
Atlassian configuration . . . 211
Install the authenticator . . . 212
Configuring Jira . . . 212
Configuring Seraph . . . 214
Importing your self-signed certificate into Atlassian . . . 215
Troubleshooting . . . 216
Enabling logging . . . 217
Release Notes
Changelog . . . 217
Known issues and limitations . . . 218
Download manifest . . . 219
Azure AD integrations . . . 219
Azure AD Password Credential Validator . . . 221
Overview of the SSO flow . . . 222
Setup
Deploying the integration files . . . 223
Required Azure AD permissions. . . . 223
Configuring a password credential validator instance . . . 223
Configuring an IdP adapter instance . . . 226
Testing . . . 227
Release notes
Changelog . . . 227
Known issues and limitations . . . 228
Download manifest . . . 229
Azure AD and Office 365 Integration Guide . . . 229
Prerequisites
Getting ready to integrate Azure Active Directory . . . 229
Preparing Active Directory for federation . . . 230
Setup
Install and configure PingFederate . . . 235
Configuring PingFederate to omit line breaks in digital signatures 235
Enabling server protocols. . . . 236
Replacing the default SSL certificates . . . 237
Configuring an LDAP connection . . . 237
Creating a password credential validator . . . 238
Configuring an IdP adapter . . . 239
Configuring a username token processor instance . . . 240
Enable Kerberos authentication . . . 242

Creating a connection to Azure Active Directory 242
Configuring browser SSO 243
Configuring assertion creation 244
Configuring attribute source and user lookup for HTML Form Adapter instances 245
Mapping an HTML form adapter instance 245
Configuring WS-Trust STS 247
Configuring attribute source and user lookup for token processors 249
Troubleshooting
Testing your configuration 250
Performance considerations 250
Troubleshooting 251
Azure AD Identity Protection Integration Kit 251
Setup
Setting up the Microsoft Login Integration Kit 252
Deploying the integration files 253
Configuring an adapter instance 253
JSON Pointer syntax reference 254
Azure AD Identity Protection IdP Adapter settings reference . . . 255
Adding review statuses to your authentication policy 256
Authentication API support 260
Troubleshooting
Enabling debug logging 260
Release notes
Changelog 262
Known issues and limitations 262
Download manifest 263
Box Provisioner 263
User and group management 265
Setup
Obtain client ID and secret from Box 267
Generate OAuth access and refresh tokens 268
Download Box SAML 2.0 metadata file 270
Upgrade an existing connector 270
Install the connector 271
Enable outbound provisioning 272
Configure provisioning 272
Supported attributes reference 282
Sample OGNL expressions 283
Configure SAML SSO 284
Configure PingFederate for SSO 285
Configure Box for SSO 288

Troubleshooting . . . 288
Additional logging . . . 289
Updating Box OAuth tokens . . . 290
Release notes
Changelog . . . 290
Known issues and limitations . . . 292
Download manifest . . . 293
Citrix ADC Integration Guide . . . 293
Enabling single sign-on in PingFederate . . . 295
Exporting your PingFederate signing certificate . . . 296
Creating an authentication server in Citrix . . . 296
Creating a SAML policy in Citrix . . . 297
Creating an authentication virtual server in Citrix . . . 297
Enabling SAML policy on the Gateway virtual server . . . 298
Creating a single sign-on connection . . . 299
Configuring an adapter instance . . . 300
Code42 Integration Guide for PingFederate . . . 300
Updating an existing deployment . . . 303
Deploying the integration files . . . 304
Enabling provisioning and single sign-on in PingFederate . . . 304
Provisioning setup
Adding PingFederate as SCIM provider in Code42 . . . 305
Creating a provisioning connection . . . 306
SCIM Connector settings reference for Code42 . . . 307
Creating a channel . . . 308
Supported attributes reference . . . 309
Single sign-on setup
Exporting SAML metadata from PingFederate . . . 310
Adding PingFederate as an authentication provider in Code42 . . . 311
Creating a single sign-on connection . . . 311
Code42 Integration Guide for PingOne . . . 312
Provisioning setup
Adding PingOne as SCIM provider in Code42 . . . 314
Creating a provisioning connection in PingOne . . . 315
Testing user provisioning . . . 320
Single sign-on setup
Creating an SSO connection in PingOne . . . 320
Adding PingOne as an authentication provider in Code42 . . . 322
Testing single sign-on . . . 324
Concur Provisioner . . . 324
User management . . . 326
Setup
Obtain key and secret . . . 327
Generate OAuth access token . . . 329

Install the connector . . . 330
Enable outbound provisioning . . . 331
Configure provisioning . . . 332
Supported attributes reference . . . 335
Configure SAML SSO . . . 336
Configure PingFederate for SSO . . . 337
Configure Concur for SSO . . . 338
Release notes
Changelog . . . 339
Known issues and limitations . . . 339
Download manifest . . . 340
Contentful Integration Guide . . . 340
Upgrading an existing deployment . . . 343
Deploying the integration files . . . 344
Enabling provisioning and single sign-on in PingFederate . . . 345
Configuring provisioning . . . 346
Creating an access token in Contentful . . . 346
Creating a provisioning connection . . . 347
Creating a channel . . . 348
Configuring single sign-on . . . 349
Creating an HTML Form Adapter instance . . . 349
Exporting your PingFederate signing certificate . . . 350
Downloading the Contentful SAML metadata file . . . 351
Creating a single sign-on connection . . . 352
Configuring Browser SSO . . . 353
Exporting SAML metadata from PingFederate . . . 355
Configuring single sign-on in Contentful . . . 355
CoreBlox integrations . . . 356
CoreBlox Integration Kit . . . 358
Overview of the IdP SSO flow . . . 359
Overview of the SP SSO flow . . . 360
Setup
Deploying the integration files . . . 360
Configuring a CoreBlox IdP Adapter instance . . . 361
CoreBlox IdP Adapter settings reference . . . 362
Configuring a CoreBlox SP Adapter instance . . . 364
CoreBlox SP Adapter settings reference . . . 366
Troubleshooting
Enabling debug logging . . . 369
Release notes
Changelog . . . 370
Known issues and limitations . . . 372
Download manifest . . . 372

CoreBlox Token Translator . . . 372
Setup
Installing the CoreBlox token translator . . . 373
Token processor (IdP) processing overview . . . 373
Configuring the IdP token processor . . . 374
Token generator (SP) processing overview . . . 376
Configuring the SP token generator . . . 377
Deployment notes . . . 378
Release notes
Changelog . . . 379
Download manifest . . . 379
Coupa Provisioner . . . 379
Setup
Obtaining required information . . . 381
Obtaining your Coupa subdomain . . . 381
Obtaining your Coupa API key . . . 381
Obtaining your Coupa (SP) metadata file . . . 382
Synchronizing existing Coupa users . . . 382
Enabling provisioning and single sign-on (SSO) in PingFederate . . . 383
Creating a connection . . . 384
Completing setup of SAML SSO to Coupa . . . 385
Obtaining the EntityDescriptor's ID from your SP connection . . . 385
Preparing theidp-metadata.xmlfile . . . 386
Configuring your SP connection for IdP-initiated SSO . . . 386
Configuring Coupa for SAML SSO . . . 387
Attribute index . . . 388
Release notes
Changelog . . . 391
Known issues and limitations . . . 391
Download manifest . . . 392
CrowdStrike Integration Kit . . . 392
Overview of the SSO flow . . . 394
Setup
Deploying the integration files . . . 395
Configuring an adapter instance . . . 396
JSON Pointer syntax reference . . . 397
CrowdStrike IdP Adapter settings reference . . . 400
Using CrowdStrike Risk Level . . . 402
Troubleshooting
Enabling debug logging . . . 403
Release Notes
Changelog . . . 404
Known issues and limitations . . . 404
Download manifest . . . 405

Dropbox Provisioner . . . 405
User and group management . . . 407
Setup
Obtaining your OAuth access token . . . 409
Obtain an App Key and Secret from Dropbox . . . 409
Generate Your OAuth 2.0 Access Token . . . 410
Preparing the Dropbox SAML 2.0 metadata XML file . . . 411
Upgrading existing Dropbox connectors . . . 411
Deploying the integration files . . . 412
Enabling provisioning in PingFederate . . . 413
Creating a connection . . . 413
Supported attributes reference . . . 415
Completing setup of SAML SSO to Dropbox . . . 416
Release notes
Changelog . . . 417
Known issues and limitations . . . 418
Download manifest . . . 418
Duo Security integrations . . . 418
Duo Security Integration Kit 3.x . . . 420
Overview of the SSO flow . . . 421
Authentication API support . . . 423
Setup
Deploying the integration files . . . 423
Getting your Duo Security keys and API hostname . . . 423
Configuring an adapter instance . . . 424
JSON Pointer syntax reference . . . 425
Duo Security IdP Adapter settings reference . . . 427
Adding Duo Security to your authentication policy . . . 428
Troubleshooting
Enabling debug logging . . . 429
Release notes
Changelog . . . 430
Known issues and limitations . . . 432
Download manifest . . . 432
Duo Security Integration Kit 2.2 . . . 433
Overview of the SSO flow . . . 433
Setup
Deploying the integration files . . . 435
Getting your Duo Security keys and API hostname . . . 435
Configuring an adapter instance . . . 436
Duo Security Adapter settings reference . . . 437
Adding Duo Security to your authentication policy . . . 438
Troubleshooting
Enabling debug logging . . . 439

Release notes
Changelog . . . 440
Known issues and limitations . . . 441
Download manifest . . . 441
Dynamics CRM Integration Guide . . . 442
Prerequisites . . . 444
PingFederate configuration . . . 444
Create an SP connection . . . 445
Map an IdP adapter . . . 445
Configure protocol settings . . . 445
Configure WS-Trust STS . . . 446
Configure IdP token processor mapping . . . 446
Configure credentials . . . 447
Share the signing certificate with the Dynamics CRM server . . . 447
Select WS-Trust encryption algorithm . . . 447
Save and activate the connection . . . 448
Dynamics CRM configuration . . . 448
Change signing certificates . . . 449
Known issues . . . 449
Egnyte Provisioner . . . 450
User management . . . 452
Setup
Obtain API key . . . 453
Obtain access token . . . 454
Obtain Egnyte SAML 2.0 metadata . . . 456
Install the connector . . . 456
Enable outbound provisioning . . . 457
Configure provisioning . . . 458
Supported attributes reference . . . 461
Configure SAML SSO . . . 463
Configure Egnyte for SSO . . . 463
Configure PingFederate for SSO . . . 465
Release notes
Changelog . . . 468
Known issues and limitations . . . 468
Download manifest . . . 468
Entrust Identity Enterprise Integration Kit . . . 469
Overview of the SSO flow . . . 471
Authentication API Support . . . 472
Models, objects, and error codes . . . 472
Setup
Deploying the integration files . . . 476
Registering PingFederate as an application in Entrust . . . 477

Configuring an adapter instance . . . 478
Entrust IdP Adapter settings reference . . . 478
Troubleshooting
Enabling debug logging . . . 480
Release notes
Changelog . . . 481
Known issues and limitations . . . 482
Download manifest . . . 483
Evernote Provisioner . . . 484
User management . . . 486
Setup
Obtain PingFederate signing certificate . . . 487
Upgrade an existing connector . . . 488
Install the connector . . . 489
Enable outbound provisioning . . . 489
Configure PingFederate for SSO . . . 490
Configure Evernote for SSO . . . 493
Configure Evernote for provisioning . . . 494
Configure PingFederate for provisioning . . . 494
Supported attributes reference . . . 497
Release notes
Changelog . . . 497
Known issues and limitations . . . 497
Download manifest . . . 498
Facebook integrations . . . 498
Facebook Login Integration Kit . . . 500
Overview of the SSO flow . . . 500
Facebook permissions . . . 501
Setup
Upgrading an existing deployment . . . 502
Core contract attributes reference . . . 503
Deploying the integration files . . . 505
Creating an app in Facebook . . . 505
Configuring an adapter instance . . . 507
Facebook IdP Adapter settings reference . . . 508
Configuring SSO to an application within the PingFederate domain . . . 511
Configuring SSO to an SP outside the PingFederate domain . . . 511
Troubleshooting
Enabling debug logging . . . 512
Troubleshooting . . . 513
Release notes
Changelog . . . 514
Known issues and limitations . . . 515
Download manifest . . . 516

Workplace from Facebook Provisioner . . . 517
User management . . . 518
Setup
Upgrading an existing deployment . . . 519
Deploying the integration files . . . 520
Enabling provisioning and single sign-on in PingFederate . . . 521
Getting SAML details from Workplace . . . 522
Configuring provisioning . . . 523
Creating a custom integration and access token in Workplace . . . 523
App secret proof feature . . . 525
Creating a provisioning connection . . . 526
Provisioning options reference . . . 527
Supported attributes reference . . . 528
Configuring single sign-on . . . 531
Adding SSO to a connection . . . 532
Registering PingFederate as an identity provider in Workplace . . . 532
Release notes
Changelog . . . 533
Known issues and limitations . . . 535
Download manifest . . . 536
ForgeRock Intelligent Access Integration Kit . . . 536
Overview of the SSO flow . . . 538
Setup
Deploying the integration files . . . 539
Configuring an adapter instance . . . 540
ForgeRock Intelligent Access IdP Adapter settings reference . . . 541
Using ForgeRock Intelligent Access Journey . . . 542
Troubleshooting
Enabling debug logging . . . 543
Release notes
Changelog . . . 544
Known issues and limitations . . . 544
Download manifest . . . 544
GitHub integrations . . . 545
GitHub Login Integration Kit . . . 547
Overview of the SSO flow . . . 548
Authentication API Support . . . 549
Setup
Deploying the integration files . . . 549
Registering PingFederate as an OAuth application in GitHub . . . 550
Configuring an adapter instance . . . 550
Available user attributes . . . 551
JSON Pointer syntax reference . . . 552
GitHub IdP Adapter settings reference . . . 554

Integrating social sign-on into your application . . . 556
Troubleshooting
Enabling debug logging . . . 558
Release notes
Changelog . . . 559
Known issues and limitations . . . 559
Download manifest . . . 560
GitHub Provisioner . . . 561
User management . . . 561
Setup
Obtain client ID and secret from GitHub . . . 562
Generate OAuth access tokens . . . 566
Obtaining PingFederate SAML 2.0 metadata . . . 567
Obtain PingFederate signing certificate . . . 568
Install the provisioner . . . 568
Enable outbound provisioning . . . 569
Configure PingFederate for provisioning and SSO . . . 570
Supported attributes reference . . . 575
Configure GitHub for provisioning and SSO . . . 575
Release notes
Changelog . . . 579
Known issues and limitations . . . 579
Download manifest . . . 580
GitHub EMU Provisioner . . . 580
User and group management . . . 581
Setup
Creating a personal access token . . . 583
Obtaining PingFederate SAML 2.0 metadata . . . 583
Obtain PingFederate signing certificate . . . 584
Install the provisioner . . . 584
Enabling provisioning and single sign-on in PingFederate . . . 585
Configure PingFederate for provisioning and SSO . . . 586
Supported attributes reference . . . 594
Configure GitHub for provisioning and SSO . . . 595
Release notes
Changelog . . . 596
Known issues and limitations . . . 596
Download manifest . . . 597
Google integrations . . . 597
Google Login Integration Kit . . . 599
Overview of the SSO flow . . . 600
Setup
Upgrading an existing deployment . . . 601
Deploying the integration files . . . 601

Registering PingFederate in Google. 602
Get a refresh token 603
Configuring an adapter instance 605
Google IdP Adapter settings reference 606
Available user attributes reference 609
Configuring SSO to an application within the PingFederate domain 610
Configuring SSO to an SP outside the PingFederate domain 611
Troubleshooting
Enabling debug logging. 611
Release notes
Changelog 612
Known issues and limitations 614
Download manifest 614
Google Workspace Provisioner. 615
User and group management. 616
Setup
Obtain an application name, client ID, and secret 618
Generate authorized OAuth 2.0 tokens 619
Upgrade an existing deployment 621
Deploying the integration files 622
Enabling provisioning and single sign-on in PingFederate. 623
Creating a connection 624
Supported attributes reference 626
Complete setup of SAML SSO to Google Apps. 630
The Password Manager 631
IdP Adapter Setup 632
SP Adapter Setup 632
IdP - SP Adapter Mapping. 633
Deploying the Application 633
Configure and Deploy the Password Manager 634
Access the Password Manager 634
Release notes
Changelog 635
Known issues and limitations 636
Download manifest 637
Google Chrome Enterprise Integration Kit 637
Overview of the SSO flow 638
Setup
Deploying the integration files 639
Configuring an adapter instance 639
JSON Pointer syntax reference. 640
Google Chrome Enterprise Device Trust IdP Adapter settings reference 642
Using Chrome device signals 643

Troubleshooting
Enabling debug logging . . . 644
Release Notes
Changelog . . . 645
Known issues and limitations . . . 646
Download manifest . . . 646
Heroku Integration Guide . . . 646
Obtain the Heroku SAML 2.0 Metadata XML . . . 648
Configure Server Settings . . . 648
Configure a Connection . . . 649
Configure Heroku for SSO . . . 653
ID DataWeb Integration Kit . . . 655
Overview of ID DataWeb . . . 657
Overview of the SSO flow . . . 658
Device profiling methods . . . 659
Setup . . . 661
Deploying the integration files . . . 662
Integrating device profiling . . . 662
Adding device profiling to an authentication page . . . 663
Adding device profiling to a web application . . . 664
Configuring an adapter instance . . . 664
JSON Pointer syntax reference . . . 665
ID DataWeb IdP Adapter settings reference . . . 669
Adding ID DataWeb policy decisions to your authentication policy . . . 672
Troubleshooting
Enabling debug logging . . . 675
Troubleshooting information . . . 676
Release notes
Changelog . . . 677
Known issues and limitations . . . 678
Download manifest . . . 678
Intune Integration Kit . . . 679
Overview of the SSO flow . . . 682
Setup
Registering PingFederate as an application in Azure Active Directory . . . 683
Deploying and configuring the X.509 Certificate Integration Kit . . . 683
Deploying the integration files . . . 684
Configuring an adapter instance . . . 684
Intune IdP Adapter settings reference . . . 685
Adding Intune security posture results to your authentication policy . . . 685
Creating certificate profiles in Intune . . . 688
Testing the integration . . . 689
Release notes
Changelog . . . 689

Known issues and limitations . . . 690
Download manifest . . . 690
iovation Integration Kit . . . 690
Overview of iovation Device Risk . . . 692
Overview of the SSO flow . . . 693
Device profiling method . . . 694
Setup . . . 695
Deploying the integration files . . . 696
Adding device profiling to an authentication page . . . 696
Configuring an adapter instance . . . 697
Transaction insight parameters . . . 699
JSON Pointer syntax reference . . . 699
iovation IdP Adapter settings reference . . . 702
Adding risk results to your authentication policy . . . 705
Release notes
Changelog . . . 707
Known issues and limitations . . . 708
Download manifest . . . 708
Jamf Integration Kit . . . 709
Overview of the SSO flow . . . 711
Setup
Creating API credentials in Jamf Pro . . . 712
Certificate requirements . . . 713
Setting up the X.509 Certificate Integration Kit . . . 714
Deploying the integration files . . . 715
Configuring an adapter instance . . . 716
Jamf IdP Adapter settings reference . . . 717
JSON Pointer syntax reference . . . 718
Adding Jamf security posture results to your authentication policy . . . 723
Authentication API Support . . . 726
Release notes
Changelog . . . 727
Known issues and limitations . . . 727
Download manifest . . . 728
Java Integration Kit . . . 728
Overview of the OpenToken Adapter . . . 730
Overview of the SSO flow . . . 731
Setup
Updating the OpenToken Adapter . . . 732
Custom application setup . . . 733
Configuring an OpenToken SP Adapter instance . . . 733
Deploying the Java agent . . . 734
Reading and writing OpenTokens . . . 735

Identity provider applications . . . 737
IdP single sign-on (SSO) . . . 737
Encoding attributes . . . 738
Sending multi-value attributes . . . 738
IdP single logout (SLO) . . . 739
Processing logout requests . . . 740
Service provider applications . . . 741
SP single sign-on (SSO) . . . 741
Decoding attributes . . . 741
Receiving multi-value attributes . . . 742
SP single sign-on (using account linking) . . . 743
Sample code . . . 744
SP single logout (SLO) . . . 745
Testing . . . 746
Sample application setup . . . 747
Deploying the sample configuration archive . . . 747
Deploy the sample applications . . . 748
Using the sample applications . . . 749
Using the IdP sample application . . . 749
IdP sample application configuration reference . . . 751
Using the SP sample application . . . 751
SP sample application configuration reference . . . 753
Modifying sample source files . . . 754
Advanced installation and configuration . . . 755
Manual configuration settings . . . 755
Using separate IdP and SP servers . . . 757
Deploying the applications to separate servlet containers . . . 758
Release notes
Changelog . . . 758
Known issues and limitations . . . 762
Download manifest . . . 763
LinkedIn Login Integration Kit . . . 764
Overview of the SSO flow . . . 766
Authentication API Support . . . 768
Setup
Upgrading an existing deployment . . . 768
Deploying the integration files . . . 769
Register PingFederate as a Microsoft Application . . . 769
Configuring an adapter instance . . . 770
Core contract attributes . . . 771
JSON Pointer syntax reference . . . 772
LinkedIn IdP Adapter settings . . . 773
Integrating social sign-on into your application . . . 775

Troubleshooting
Troubleshooting . . . 776
Enabling debug logging . . . 778
Release notes
Changelog . . . 779
Known issues and limitations . . . 780
Download manifest . . . 780
Lucidchart Provisioner . . . 781
User management . . . 783
Setup
Obtain base URL and bearer token . . . 785
Install the connector . . . 785
Enable outbound provisioning . . . 785
Configure provisioning . . . 786
Supported attributes reference . . . 791
Configure SAML SSO . . . 791
Configure Lucidchart for SSO . . . 792
Configure PingFederate for SSO . . . 793
Release notes
Changelog . . . 796
Known issues and limitations . . . 796
Download manifest . . . 797
Microsoft EAM Integration Kit . . . 797
Overview of the SSO flow . . . 799
Setup
Deploying the integration files . . . 801
Configuring an adapter instance . . . 801
Microsoft EAM IdP Adapter settings reference . . . 802
Tracking HTTP request parameters . . . 802
Exporting amr as an additional attribute for PingID . . . 802
Configuring PingFederate to use the Microsoft EAM IdP adapter . . . 803
Troubleshooting
Enabling debug logging . . . 805
Release notes
Changelog . . . 806
Known issues and limitations . . . 806
Download manifest . . . 806
Internet Information Services (IIS) Integration Kit . . . 807
Overview of the OpenToken IIS Agent . . . 809
Overview of the SSO flow . . . 810
Session information and HTTP request headers . . . 811
Configuring the OpenToken IIS Agent to omit the HTTP header prefix . . . 811
Setup
Upgrading an existing deployment . . . 812

Updating the OpenToken Adapter . . . 815
Configuring an OpenToken SP Adapter instance . . . 815
Deploying and configuring the OpenToken IIS Agent . . . 816
Adding the OpenToken HTTP Module in IIS . . . 817
Release notes
Changelog . . . 818
Known issues and limitations . . . 820
Download manifest . . . 821
Microsoft Login Integration Kit . . . 821
Overview of the SSO flow . . . 823
Authentication API Support . . . 825
Setup
Deploying the integration files . . . 825
Registering PingFederate as an application in Azure . . . 826
Configuring an adapter instance . . . 827
Core contract attributes . . . 828
Microsoft IdP Adapter settings reference . . . 829
Integrating social sign-on into your application . . . 832
Troubleshooting
Enabling debug logging . . . 833
Release notes
Changelog . . . 834
Known issues and limitations . . . 835
Download manifest . . . 835
.NET Integration Kit . . . 836
Overview of the SSO flow . . . 838
Setup
Updating the OpenToken Adapter . . . 840
Deploying the OpenToken Agent . . . 840
Custom application setup . . . 840
Configuring an OpenToken SP Adapter instance . . . 841
Integrating the OpenToken agent into your .NET application . . . 841
IdP single sign-on integration . . . 842
IdP single logout integration . . . 843
SP single sign-on integration . . . 844
SP single sign-on integration using account linking . . . 845
SP single logout (SLO) . . . 847
Testing . . . 848
Sample applications setup . . . 848
Configuring PingFederate to use the sample applications . . . 849
Configuring IIS to use the sample applications . . . 850
Using the IdP sample application . . . 851
Using the SP sample application . . . 851

Release notes
Changelog . . . 852
Known issues and limitations . . . 854
Download manifest . . . 854
MobileIron Integration Kit . . . 855
Overview of the SSO flow . . . 857
Supported attributes reference . . . 858
Setup
Deploying the integration files . . . 859
Creating a MobileIron adapter instance . . . 860
Testing the adapter . . . 861
Sample use cases . . . 862
Configuration examples . . . 862
Release notes
Changelog . . . 863
Download manifest . . . 863
Office 365 integrations . . . 863
Office 365 Provisioner . . . 865
User and group management . . . 866
Setup
Setup overview diagram . . . 868
Getting started . . . 868
Choose an SSO configuration path . . . 869
Create an Azure account . . . 869
Download Office 365 SAML 2.0 metadata file . . . 870
Configuring PingFederate to omit line breaks in digital signatures 870
Upgrading an existing deployment . . . 870
Deploying the integration files . . . 872
Enabling provisioning and single sign-on in PingFederate . . . 873
Configuring provisioning . . . 874
Add application to Azure AD . . . 874
Configure outbound provisioning . . . 875
Supported attributes reference . . . 879
Configure license management . . . 882
Configure manager assignment . . . 882
Configuring single sign-on . . . 883
Install Powershell account connection software . . . 883
Add federated domain . . . 884
DNS updates . . . 885
Verify federated domain . . . 885
Configure federation settings . . . 886
Obtain PingFederate signing certificate . . . 886
Configure an SSO connection . . . 886
Add the signing certificate to Azure . . . 893

Troubleshooting . . . 894
Release notes
Changelog . . . 894
Known issues and limitations . . . 896
Download manifest . . . 896
OpenToken Token Translator . . . 897
WS-Trust STS processing . . . 899
Setup
IdP installation and setup . . . 900
OpenToken Token Processor Settings Reference . . . 901
SP installation and setup . . . 902
OpenToken Token Generator Settings Reference . . . 903
The OpenToken agent API . . . 904
Install the OpenToken agent API . . . 905
Read an OpenToken as a WSP . . . 905
Write an OpenToken as a WSC . . . 906
Release notes
Changelog . . . 907
Download manifest . . . 907
Oracle Access Manager (OAM) Integration Kit . . . 908
Setup
IdP implementation . . . 910
Upgrade instructions . . . 912
OAM IdP configuration . . . 912
Apache module installation . . . 913
Apache module configuration . . . 913
PingFederate IdP configuration . . . 914
IdP adapter instance configuration . . . 915
IdP adapter testing . . . 916
SP implementation . . . 917
Upgrade instructions . . . 918
OAM SP configuration . . . 919
PingFederate SP configuration . . . 920
SP adapter instance configuration . . . 920
SP adapter testing . . . 921
Release notes
Changelog . . . 922
Known issues and limitations . . . 922
Download manifest . . . 923
One-Time Passcode Integration Kit . . . 924
Overview of one-time passcodes . . . 927
Overview of the Twilio notification publishers . . . 927
Overview of the SSO flow . . . 928
PingFederate Authentication API support . . . 929

Setup
Deploying the integration files . . . 932
Configuring a Twilio notification publisher instance . . . 933
Getting your Twilio API credentials . . . 934
Twilio Programmable API Notification Publisher settings reference . . . 934
Twilio Verify API Notification Publisher settings reference . . . 936
Configuring an adapter instance . . . 937
Language property and template keys . . . 939
One-Time Passcode IdP Adapter settings reference . . . 941
Customizing templates . . . 944
Release notes
Changelog . . . 944
Known issues and limitations . . . 946
Download manifest . . . 946
PHP Integration Kit . . . 947
Overview of the SSO flow . . . 950
Setup
Updating the OpenToken Adapter . . . 951
Custom application setup . . . 951
Configuring an OpenToken Adapter instance . . . 952
Configuring your PHP server . . . 952
IdP single sign-on integration . . . 953
Sample code . . . 954
Passing multi-value attributes . . . 954
IdP single logout integration . . . 955
Sample code . . . 956
SP single sign-on integration . . . 957
Sample code . . . 958
Receiving multi-value attributes . . . 959
SP single sign-on integration using account linking . . . 959
Sample code . . . 961
SP single logout integration . . . 961
Testing . . . 962
Sample application setup . . . 962
Configuring PingFederate and the sample applications . . . 964
Deploying on Red Hat Linux . . . 964
Deploying on Linux or Unix . . . 965
Deploying on Windows . . . 966
Deploying on the PHP built-in web server . . . 967
Using the IdP sample application . . . 968
Using the SP sample application . . . 968
Release notes
Changelog . . . 969
Known issues and limitations . . . 970

Download manifest . . . 970
PingID Provisioner . . . 971
User management . . . 974
Device management . . . 975
Setup
Upgrading an existing deployment . . . 977
Deploying the integration files . . . 978
Enabling provisioning in PingFederate . . . 978
Get your PingID settings file . . . 979
Creating a provisioning connection . . . 979
PingID Connector settings reference . . . 980
Supported attributes reference . . . 982
Release notes
Changelog . . . 983
Known issues and limitations . . . 984
Download manifest . . . 984
PingID SDK Provisioner . . . 984
Setup
Get information from the PingID SDK . . . 987
Upgrade an existing connector . . . 987
Deploy the provisioning connector . . . 988
Configure provisioning . . . 989
Provisioning options . . . 990
Supported attributes reference . . . 991
Synchronize existing users . . . 992
Release notes
Changelog . . . 993
Known issues and limitations . . . 993
Download manifest . . . 995
PingOne Integrations . . . 995
PingOne Integration Kit . . . 998
Setup
Connecting PingFederate to PingOne . . . 999
Datastore and password credential validator . . . 999
Deploying the datastore and password credential validator files . 1000
Configuring the datastore . . . 1001
JSON Pointer syntax reference . . . 1002
Datastore configuration reference . . . 1004
Configuring the password credential validator . . . 1005
Customizing authentication error messages . . . 1006
Password credential validator configuration reference . 1007
Provisioning connector . . . 1008
User and group management . . . 1008
Authentication method management . . . 1011

Upgrading an existing deployment . . . 1012
Deploying the provisioning connector . . . 1013
Enabling provisioning in PingFederate . . . 1014
Creating a provisioning connection . . . 1014
Provisioning options reference . . . 1015
Configuring a channel . . . 1017
Supported attributes reference . . . 1019
Passing binary attributes to PingOne . . . 1022
Creating an OpenID Connect connection . . . 1023
Troubleshooting
Enabling debug logging . . . 1024
Release notes
Changelog . . . 1025
Known issues and limitations . . . 1028
Download manifest . . . 1028
PingOne MFA Integration Kit . . . 1029
Release Notes . . . 1031
Overview of the SSO flow . . . 1050
Overview of the automatic device pairing flow . . . 1051
Authentication method management . . . 1053
Setup
Deploying the integration files . . . 1055
Creating an organization and environment in PingOne . . . 1056
Creating a web or native OIDC application in PingOne . . . 1056
Connecting PingFederate to PingOne . . . 1059
Use cases . . . 1060
Enabling user and authentication method provisioning . . . 1060
Enabling the MFA setup prompt . . . 1061
Transaction approval setup . . . 1063
Using FIDO device with PingOne custom domain configuration . 1070
Enabling passwordless login using FIDO browser management . 1070
FIDO browser management use cases . . . 1072
Support for authentication code flow . . . 1075
Support for stateless one-time device OTP flow . . . 1080
Support for FIDO usernameless authentication flow . . . 1081
MFA setup . . . 1088
Creating an MFA authentication policy in PingOne . . . 1088
Configuring an adapter instance . . . 1089
PingOne MFA IdP Adapter settings reference . . . 1090
Adding PingOne MFA to your authentication policy . . . 1099
PingOne MFA status attributes reference . . . 1103
CIBA setup . . . 1106
CIBA request flow . . . 1106

CIBA prompt customizations . . . 1107
Simple CIBA prompt customizations . . . 1108
Advanced CIBA prompt customizations . . . 1109
Multiple PingOne application support . . . 1113
Creating a CIBA authentication policy in PingOne . . . 1113
Configuring a CIBA authenticator instance . . . 1115
PingOne MFA CIBA Authenticator settings reference . . . 1116
Authentication API Support . . . 1119
Authentication flows . . . 1119
Device pairing flows . . . 1122
Authorization flow . . . 1125
Models, objects, and error codes . . . 1127
Troubleshooting
Known issues and limitations . . . 1150
Enabling debug logging . . . 1151
Download manifest . . . 1152
PingOne Risk Integration Kit . . . 1154
Overview of PingOne Risk . . . 1155
Overview of the SSO flow . . . 1155
Device profiling methods . . . 1157
Setup
Deploying the integration files . . . 1158
Connecting PingFederate to PingOne . . . 1159
Integrating device profiling . . . 1160
Adding device profiling to a browser-based authentication page . 1161
Adding device profiling to an authentication API-based application . . . 1164
Configuring an adapter instance . . . 1170
JSON pointer syntax reference . . . 1171
PingOne Risk IdP adapter settings reference . . . 1176
Using custom risk predictors . . . 1178
Adding risk level results to your authentication policy . . . 1180
Configuring PingFederate to forward IP addresses . . . 1184
Troubleshooting
Enabling debug logging . . . 1184
Troubleshooting information . . . 1186
Release notes
Changelog . . . 1186
Known issues and limitations . . . 1188
Download manifest . . . 1188
PingOne Verify Integration Kit . . . 1189
Use cases . . . 1190
Overview of the verification flow . . . 1191

Setup
Deploying the integration files . . . 1191
Connecting PingFederate to PingOne . . . 1192
Configuring an adapter instance . . . 1193
PingOne Verify IdP Adapter settings reference . . . 1194
JSON Pointer syntax reference . . . 1199
Authentication setup . . . 1200
Adding PingOne Verify to your authentication policy . . . 1200
Self-service registration setup . . . 1204
Registration form modifications . . . 1205
Creating input and output policy contracts . . . 1205
Creating a policy fragment . . . 1206
Modifying your local identity profile . . . 1207
Adding PingOne Verify to your user registration flow . . . 1208
Authentication API Support . . . 1209
Verification flow . . . 1209
State models, action models, and attributes . . . 1211
Troubleshooting
Enabling debug logging . . . 1214
Release notes
Changelog . . . 1215
Known issues and limitations . . . 1217
Download manifest . . . 1218
PingOne Verify Integration Kit 1.1 . . . 1219
Use cases . . . 1220
Overview of the verification flow . . . 1221
Setup
Deploying the integration files . . . 1222
Connecting PingFederate to PingOne . . . 1223
Configuring an adapter instance . . . 1224
PingOne Verify IdP Adapter settings reference . . . 1225
JSON Pointer syntax reference . . . 1228
Authentication setup . . . 1230
Adding PingOne Verify to your authentication policy . . . 1230
Self-service registration setup . . . 1234
Registration form modifications . . . 1235
Creating input and output policy contracts . . . 1235
Creating a policy fragment . . . 1236
Modifying your local identity profile . . . 1237
Adding PingOne Verify to your user registration flow . . . 1238
Authentication API Support . . . 1239
Verification flow . . . 1239
State models, action models, and attributes . . . 1241

Troubleshooting
Enabling debug logging . . . 1243
Release notes
Changelog . . . 1244
Known issues and limitations . . . 1246
Download manifest . . . 1246
PingOne DaVinci Integration Kit . . . 1247
Overview of the SSO flow . . . 1249
Setup
Adding an application in DaVinci . . . 1249
Deploying the integration files . . . 1250
Configuring an adapter instance . . . 1251
PingOne DaVinci IdP Adapter settings reference . . . 1253
JSON Pointer syntax reference . . . 1256
Building a flow in DaVinci . . . 1257
Troubleshooting
Enabling debug logging . . . 1262
Release notes
Changelog . . . 1263
Known issues and limitations . . . 1264
Download manifest . . . 1264
PingAM Integration Kit . . . 1265
Overview of the SSO flow . . . 1266
Setup
Deploying the integration files . . . 1267
Configuring an adapter instance . . . 1267
PingAM IdP Adapter settings reference . . . 1269
Preparing to use a PingAM journey . . . 1271
Collecting session data from PingAM . . . 1272
Extracting data from PingFederate . . . 1274
Troubleshooting
Enabling debug logging . . . 1276
Tracking transactions . . . 1277
Release notes
Changelog . . . 1278
Known issues and limitations . . . 1279
Download manifest . . . 1279
Magic Link Integration Kit . . . 1279
Overview of the SSO flow . . . 1280
Authentication API Support . . . 1283
Models, attributes, and error codes . . . 1283
Setup
Deploying the integration files . . . 1284

Configuring an adapter instance . . . 1285
Magic Link IdP Adapter settings reference . . . 1286
Release Notes
Changelog . . . 1289
Known Issues and Limitations . . . 1289
Download Manifest . . . 1289
PingOne Protect Integration Kit . . . 1290
Overview of PingOne Protect . . . 1292
Overview of the SSO Flow . . . 1292
Device profiling methods . . . 1294
PingFederate setup . . . 1295
Deploying the integration files . . . 1295
Creating an organization and environment in PingOne . . . 1296
Integrating device profiling . . . 1296
Adding device profiling to a browser-based authentication page . 1297
Adding device profiling to an authentication API-based application . . . 1298
Configuring an adapter instance . . . 1300
JSON Pointer syntax reference . . . 1301
IdP Adapter settings reference . . . 1305
Configuring a provider instance . . . 1307
Setting risk evaluation for active sessions . . . 1308
Using custom risk predictors . . . 1308
Adding risk level results to your authentication policy . . . 1310
Configuring PingFederate to forward IP addresses . . . 1314
PingOne Protect Native SDKs . . . 1314
Troubleshooting . . . 1315
Release notes . . . 1318
PingOne Credentials Integration Kit . . . 1319
Overview of the SSO flow . . . 1322
Setup
Deploying the integration files . . . 1323
Connecting PingFederate to PingOne . . . 1324
Configuring an adapter instance . . . 1325
PingOne Credentials IdP Adapter settings reference . . . 1327
Extending the contract . . . 1330
Authentication API Support . . . 1332
Models, attributes, and error codes . . . 1332
Troubleshooting
Enabling debug logging . . . 1335
Release notes
Changelog . . . 1336
Known issues and limitations . . . 1336
Download manifest . . . 1336

RSA SecurID integrations 1338
RSA SecurID Cloud Authentication Integration Guide 1340
Integrating PingFederate with RSA SecurID 1341
Example OGNL expressions 1341
RSA SecurID Integration Kit 1342
Overview of the SSO flow 1344
Setup
Deploying the integration files 1345
Registering PingFederate as an agent in the RSA Security Console 1346
Configuring an adapter instance 1346
RSA SecurID IdP Adapter settings reference 1347
Customizing templates 1351
Authentication API Support 1352
Troubleshooting
Enabling debug logging 1352
Release notes
Changelog 1353
Known issues and limitations 1356
Download manifest 1357
Salesforce integrations 1358
Salesforce Login Integration Kit 1360
Overview of the SSO flow 1360
Setup
Install the Salesforce connector 1362
Configure the IdP adapter 1362
SSO to an SP partner 1363
SSO to an onsite application 1364
Define the SSO URL in Salesforce 1365
Add an SSO link in Salesforce (example) 1365
Release notes
Changelog 1367
Known issues and limitations 1367
Download manifest 1367
Salesforce Provisioner 1367
User and group management 1369
Salesforce Community users 1370
Setup
Upgrade an existing deployment 1371
Deploying the integration files 1372
Enabling provisioning and single sign-on (SSO) in PingFederate 1372
Registering PingFederate as a connected app in Salesforce 1373
Registering PingFederate as an SSO provider in Salesforce 1375
Getting an API access token from Salesforce 1376

Creating a connection . . . 1377
Provisioning options reference . . . 1379
Supported attributes reference . . . 1380
Release notes
Changelog . . . 1382
Known issues and limitations . . . 1385
Download manifest . . . 1386
Salesforce Contacts Provisioner . . . 1386
Lead and Contact management . . . 1387
Setup
Upgrading an existing deployment . . . 1388
Deploying the integration files . . . 1389
Enabling provisioning in PingFederate . . . 1389
Registering PingFederate as a connected app in Salesforce . . . 1390
Getting an API access token from Salesforce . . . 1391
Creating a provisioning connection . . . 1392
Provisioning options reference . . . 1394
Supported attributes reference . . . 1395
Release notes
Changelog . . . 1399
Known issues and limitations . . . 1399
Download manifest . . . 1401
SAP NetWeaver Integration Kit . . . 1401
Setup
Implementing IdP functionality . . . 1403
IdP installation and setup . . . 1404
SAP J2EE setup . . . 1405
Testing the IdP adapter . . . 1406
Implementing SP functionality . . . 1406
OpenToken adapter setup . . . 1407
SAP J2EE setup for NetWeaver 7.0 . . . 1408
SAP J2EE setup for NetWeaver 7.3 . . . 1410
SAP J2EE setup for NetWeaver 7.3 or 7.4 . . . 1411
Testing the SP adapter . . . 1413
Release notes
Changelog . . . 1414
Known issues and limitations . . . 1415
Download manifest . . . 1415
SCIM Provisioner . . . 1416
User and group management . . . 1420
Setup
Upgrading an existing deployment . . . 1422
Deploying the integration files . . . 1423
Enabling provisioning and single sign-on in PingFederate . . . 1423

Gathering information from the service provider . . . 1424
Creating a provisioning connection . . . 1425
SCIM provisioner settings reference . . . 1425
Supported attributes reference . . . 1431
Configuring single sign-on . . . 1436
Adding SSO to a connection . . . 1436
PingFederate SSO details for the service provider . . . 1436
Release notes
Changelog . . . 1437
Known issues and limitations . . . 1439
Download manifest . . . 1440
ServiceNow Provisioner . . . 1440
User management . . . 1443
Setup
Upgrading an existing deployment . . . 1444
Deploying the integration files . . . 1445
Configuring provisioning . . . 1445
Adding the Ping Identity provisioning role in ServiceNow . . . 1446
Creating a provisioning user in ServiceNow . . . 1447
Enabling provisioning in PingFederate . . . 1447
Creating a provisioning connection . . . 1448
Provisioning options reference . . . 1449
Supported attributes reference . . . 1450
Configuring single sign-on . . . 1452
Registering PingFederate as an identity provider in ServiceNow . . . 1453
Exchanging signing certificates . . . 1454
Creating a single sign-on connection . . . 1455
Release notes
Changelog . . . 1456
Known issues and limitations . . . 1457
Download manifest . . . 1458
SharePoint People Picker Integration Kit . . . 1459
System requirements . . . 1461
Assumptions and prerequisites . . . 1461
Setup
Installation and setup . . . 1462
Upgrade from a previous version . . . 1462
Add and deploy the solution . . . 1462
Activate features . . . 1463
Set the default claims provider . . . 1464
Configuration . . . 1464
People Picker configuration . . . 1464
Configuration storage . . . 1468

Supported search attributes with examples . . . 1470
Search by user . . . 1470
Search by group . . . 1471
Troubleshooting
Diagnostic logging . . . 1472
SharePoint log location . . . 1474
Uninstall the solution . . . 1474
Release notes
Changelog . . . 1477
Known issues and limitations . . . 1478
Download manifest . . . 1478
Slack Provisioner . . . 1478
User and group management . . . 1481
Setup
Upgrading an existing deployment . . . 1482
Deploying the integration files . . . 1483
Enabling provisioning and single sign-on in PingFederate . . . 1484
Registering PingFederate as an app in Slack . . . 1485
Getting a Slack API access token . . . 1486
Preparing your SAML metadata file . . . 1492
Creating a provisioning connection . . . 1492
Provisioning options reference . . . 1493
Supported attributes reference . . . 1494
Configuring single sign-on . . . 1495
Creating a single sign-on connection . . . 1496
Configuring Browser SSO . . . 1497
Configuring single sign-on in Slack . . . 1499
Release notes
Changelog . . . 1502
Known issues and limitations . . . 1503
Download manifest . . . 1504
Symantec VIP Integration Kit . . . 1504
Overview of the SSO flow . . . 1506
PingFederate Authentication API support . . . 1508
Setup
Deploying the integration files . . . 1518
Importing the Symantec certificate into PingFederate . . . 1519
Configuring an adapter instance . . . 1519
Symantec VIP IdP adapter settings reference . . . 1520
Customizing templates . . . 1524
Customizing user messages . . . 1525
User credential management . . . 1525
Adding credentials as an administrator . . . 1525
Adding credentials as a user . . . 1526

Resetting credentials as a user . . . 1528
Troubleshooting . . . 1529
Release notes
Changelog . . . 1529
Known issues and limitations . . . 1532
Download manifest . . . 1532
ThreatMetrix Integration Kit . . . 1533
Overview of ThreatMetrix . . . 1536
Overview of the SSO flow . . . 1537
Device profiling methods . . . 1538
Setup
Deploying the integration files . . . 1541
Configuring ThreatMetrix . . . 1541
Integrating device profiling . . . 1543
Integrating device profiling - Web app (basic) . . . 1543
Integrating device profiling - Web app (enhanced) . . . 1544
Integrating device profiling - Mobile or native app (enhanced) . . . 1545
Configuring an adapter instance . . . 1546
ThreatMetrix IdP Adapter settings reference . . . 1547
Adding review statuses to your authentication policy . . . 1551
Authentication API Support . . . 1555
Troubleshooting
Enabling debug logging . . . 1556
Release notes
Changelog . . . 1557
Known issues and limitations . . . 1558
Download manifest . . . 1559
Twitter Login Integration Kit . . . 1560
Overview of the SSO flow . . . 1563
Authentication API Support . . . 1564
Setup
Deploying the integration files . . . 1564
Registering PingFederate in Twitter . . . 1565
Configuring an adapter instance . . . 1565
Twitter IdP Adapter settings reference . . . 1566
Integrating social sign-on into your application . . . 1569
Troubleshooting
Enabling debug logging . . . 1570
Release notes
Changelog . . . 1571
Known issues and limitations . . . 1572
Download manifest . . . 1572

Web Access Management integrations 1573
Web Access Management (WAM) Integration Kit 1575
Setup
WAM plug-in installation for RSA 1576
WAM plug-in installation for OAM 1576
OAM-specific configuration. 1577
Custom WAM plug-in installation 1577
Install or upgrade the adapter 1578
IdP process overview 1579
Configuring an IdP adapter instance 1580
IdP deployment note 1587
Test the IdP adapter. 1588
SP process overview 1589
Configuring an SP adapter instance. 1589
Creating a Custom Authentication Scheme for OAM . . 1597
SP deployment notes 1598
Test the SP adapter 1598
Release notes
Changelog 1599
Download manifest 1600
Web Access Management (WAM) Token Translator 1601
Setup
WAM plug-ins 1602
Custom WAM plug-in installation 1602
WAM plug-in installation for RSA 1602
WAM plug-in installation for OAM 1603
Token translator installation 1603
WS-Trust STS processing 1604
Configuring the IdP token processor 1605
Configuring the SP token generator 1608
Creating a custom authentication scheme for OAM 1612
Using the STS client SDK 1613
Token processor sample code 1613
Token generator sample code 1613
Release notes
Changelog 1614
Known issues and limitations 1614
Download manifest 1614
Webex Provisioner. 1615
User management 1617
Setup
Upgrade an existing connector 1619
Deploy the provisioning connector. 1619

Configuring provisioning and single sign-on . . . 1620
Downloading your Webex SAML metadata file . . . 1622
Provisioning options . . . 1623
Supported attributes reference . . . 1623
Configuring single sign-on in Webex . . . 1624
Release notes
Changelog . . . 1626
Known issues and limitations . . . 1627
Download manifest . . . 1628
WebSphere Integration Guide . . . 1628
Setup
Enabling single sign-on in PingFederate . . . 1630
Exporting SAML metadata from PingFederate . . . 1631
Configuring single sign-on in WebSphere . . . 1631
Creating a single sign-on connection. . . . 1633
Configuring service provider-initiated SSO . . . 1633
Workspace ONE UEM Integration Kit . . . 1634
Overview of the SSO flow . . . 1637
Supported attributes reference . . . 1638
Setup
Deploying the integration files . . . 1639
Configuring a Workspace ONE IdP Adapter instance . . . 1640
Restrict device access based on criteria . . . 1640
Create an X.509 Certificate Adapter instance . . . 1641
Create a policy contract . . . 1642
Adding device postures to your authentication policy. . . . 1642
Configure the SP connection to use the policy contract . . . 1643
Test the integration . . . 1644
Release notes
Changelog . . . 1645
Known issues and limitations . . . 1645
Download manifest . . . 1646
X.509 integrations . . . 1646
X.509 Certificate Integration Kit . . . 1648
Overview of the SSO flow . . . 1648
Setup
Deploying the integration files . . . 1649
Configuring an adapter instance . . . 1650
X.509 Certificate IdP Adapter settings reference . . . 1651
Sample OGNL expressions . . . 1653
Proxy client authentication . . . 1654
Release notes
Changelog . . . 1654
Known issues and limitations . . . 1655

Download manifest . . . 1656
X.509 Token Translator . . . 1656
WS-Trust STS processing . . . 1657
Setup
Deploying the integration files . . . 1657
Configuring an X.509 Token Translator instance . . . 1658
Using the Java Client SDK . . . 1659
Release notes
Changelog . . . 1659
Known issues and limitations . . . 1660
Download manifest . . . 1660
Zendesk Provisioner . . . 1660
User and group management . . . 1662
Setup
Upgrading an existing deployment . . . 1664
Obtain required information . . . 1665
Enabling provisioning and single sign-on in PingFederate . . . 1667
Deploying the integration files . . . 1668
Creating a connection . . . 1669
Supported attributes reference . . . 1670
Configuring SAML SSO in Zendesk . . . 1671
Release notes
Changelog . . . 1672
Known issues and limitations . . . 1673
Download manifest . . . 1673
Zoom Provisioner . . . 1673
User management . . . 1676
Setup
Upgrading an existing deployment . . . 1677
Deploying the integration files . . . 1678
Getting a vanity URL for Zoom . . . 1678
Creating a Zoom app . . . 1679
Enabling provisioning and single sign-on in PingFederate . . . 1681
Creating a provisioning connection . . . 1682
Provisioning options reference . . . 1683
Supported attributes reference . . . 1684
Configuring single sign-on . . . 1685
Configuring single sign-on in Zoom . . . 1685
Add the Zoom metadata URL in PingFederate . . . 1686
Creating a single sign-on connection . . . 1686
Release notes
Changelog . . . 1687
Known issues and limitations . . . 1688
Download manifest . . . 1688

Zscaler integrations 1689
Zscaler Internet Access Provisioner 1691
User and group management. 1692
Setup
Upgrading an existing deployment 1694
Deploying the integration files 1695
Enabling provisioning and single sign-on in PingFederate 1696
Getting SAML details from Zscaler 1697
Configuring provisioning 1697
Getting a base URL and bearer token from Zscaler 1697
Creating a provisioning connection 1698
Provisioning options reference 1699
Supported attributes reference 1700
Configuring single sign-on 1700
Creating a single sign-on connection 1701
Registering PingFederate as an identity provider in Zscaler 1702
Release notes
Changelog 1703
Known issues and limitations 1704
Download manifest 1705
Zscaler Private Access Provisioner. 1705
User and group management. 1707
Setup
Upgrading an existing deployment 1708
Deploying the integration files 1709
Enabling single sign-on in PingFederate 1710
Exporting SAML metadata from PingFederate 1710
Enabling provisioning and single sign-on in Zscaler 1711
Creating a provisioning connection. 1712
Provisioning options reference 1713
Supported attributes reference 1714
Creating a single sign-on connection 1715
Release notes
Changelog 1716
Known issues and limitations 1716
Download manifest 1717

PingFederate and PingOne Integrations

Documentation for PingFederate and select PingOne Integrations. Find 1800+ more in the Integration Directory.

A-E

Agentless Integration Kit

Integrate any application with PingFederate for SSO. Uses API calls with no need for agent software.

Amazon Integration Kit and Provisioners

Integrate with Amazon Web Services (AWS).

Apache Integration Kit

Integrate your Apache web application with PingFederate for SSO.

Apple Login Integration Kit

Allow users to sign on with Apple for PingFederate SSO.

Aquera Provisioner

Integrate any application supported by Aquera with PingFederate for user provisioning and SSO.

Atlassian Provisioner and Integration Kit

Integrate Atlassian Cloud and on-premises solutions with PingFederate for user provisioning and SSO.

Copyright © 2025 Ping Identity Corporation

Azure Integration Kits

Integrate Azure AD services with PingFederate.

Box Provisioner

Integrate Box with PingFederate for user provisioning and SSO.

Citrix ADC Integration Guide

Integrate Citrix ADC and Gateway with PingFederate for SSO.

Code42 for PingFederate

Integrate Code42 with PingFederate for user provisioning and SSO.

Code42 for PingOne

Integrate Code42 with PingOne for user provisioning and SSO.

Concur Provisioner

Integrate Concur with PingFederate for user provisioning and SSO.

Contentful Integration Guide

Integrate Contentful with PingFederate for user provisioning and SSO.

CoreBlox Integration Kits

Integrate CoreBlox and CoreBlox Token Translator with PingFederate.

Copyright © 2025 Ping Identity Corporation

Coupa Provisioner

Integrate Coupa with PingFederate for user provisioning and SSO.

CrowdStrike Integration Kit

When paired with the Google Chrome Enterprise Integration Kit in PingFederate, the CrowdStrike Integration Kit enables organizations to strike the right balance between security and user experience.

Dropbox Provisioner

Integrate Dropbox with PingFederate for user provisioning and SSO.

Duo Security Integration Kits

Use Duo Security MFA in your PingFederate sign-on flow, Using the Duo Universal Prompt experience or an iFrame-based experience.

Dynamics CRM Integration Guide

Integrate Dynamics CRM with PingFederate for SSO.

Egnyte Provisioner

Integrate Egnyte with PingFederate for user provisioning and SSO.

Entrust Identity Enterprise Integration Kit

Use Entrust Identity Enterprise MFA and risk intelligence in your PingFederate sign-on flow.

Evernote Provisioner

Integrate Evernote with PingFederate for user provisioning and SSO.

Copyright © 2025 Ping Identity Corporation

F-J

Facebook Integration Kit and Provisioner

Allow users to sign on with Facebook for PingFederate SSO, or integrate Workplace from Meta with PingFederate for user provisioning and SSO.

ForgeRock Intelligent Access Integration Kit

The ForgeRock Intelligent Access Integration Kit allows PingFederate to communicate with ForgeRock Intelligent Access and provide a rich sign on orchestration experience.

GitHub Integration Kit and Provisioners

Integrate GitHub and GitHub Enterprise with PingFederate for user provisioning and SSO.

Google

Integrate Google and Google Workspace with PingFederate for user provisioning and SSO.

Heroku Integration Guide

Integrate Heroku with PingFederate for SSO.

ID DataWeb Integration Guide

Use ID DataWeb's ThreatMetrix risk management in your PingFederate sign-on flow.

Copyright © 2025 Ping Identity Corporation

Intune Integration Kit

Use Intune secure device management in your PingFederate sign-on flow.

iovation Integration Kit

Use iovation risk management in your PingFederate sign-on flow.

Jamf Integration Kit

Use Jamf secure device management in your PingFederate sign-on flow.

Java Integration Kit

Integrate your Java application with PingFederate for SSO.

K-O

LinkedIn Login Integration Kit

Allow users to sign on with LinkedIn for PingFederate SSO.

Lucidchart Provisioner

Integrate LucidChart with PingFederate for user provisioning and SSO.

Copyright © 2025 Ping Identity Corporation

Microsoft EAM Integration Kit

Process the `id_token_hint` and claims sent by Microsoft Entra ID's external authentication mechanism.

Internet Information Services (IIS) Integration Kit

Integrate your IIS web application with PingFederate for SSO.

Microsoft Login Integration Kit

Allow users to sign on with Microsoft for PingFederate SSO.

NET Integration Kit

Integrate your .NET application with PingFederate for SSO.

MobileIron Integration Kit

Use MobileIron secure device management in your PingFederate sign-on flow.

Office 365 Provisioner

Integrate Office 365 with PingFederate for user provisioning and SSO.

OpenToken Token Translator

Integrate OpenToken authentication sessions with the PingFederate WS-Trust Security Token Service (STS).

Oracle Access Manager (OAM) Integration Kit

Integrate OAM with PingFederate for SSO.

Copyright © 2025 Ping Identity Corporation

One-Time Passcode Integration Kit

Use one-time passcode for MFA in your PingFederate sign-on flow.

P-T

PHP Integration Kit

Integrate your PHP application with PingFederate for SSO.

PingID Provisioner

Integrate PingID with PingFederate for user provisioning.

PingID SDK Provisioner

Integrate PingID SDK with PingFederate for user provisioning.

PingOne Integration Kits

Integrate PingOne with PingFederate for user provisioning and SSO or use PingOne Credentials in the PingFederate sign-on flow, including PingOne Verify, PingOne Protect, and PingOne MFA.

Copyright © 2025 Ping Identity Corporation

PingOne Credentials Integration Kit

Integrate PingOne Credentials in the PingFederate sign-on flow.

RSA SecurID Integration Kits

Use RSA SecurID Cloud Authentication MFA or RSA SecurID on-premises MFA in your PingFederate sign-on flow.

Salesforce Integration Kit and Provisioners

Integrate Salesforce with PingFederate for user provisioning and SSO.

SAP NetWeaver Integration Kit

Integrate SAP NetWeaver with PingFederate for SSO.

SCIM Provisioner

Integrate your SCIM-capable application with PingFederate for user provisioning and SSO.

ServiceNow Provisioner

Integrate ServiceNow with PingFederate for user provisioning and SSO.

SharePoint People Picker Integration Kit

Allow the SharePoint people picker to search user stores connected to PingFederate.

Slack Provisioner

Integrate Slack with PingFederate for user provisioning and SSO.

Copyright © 2025 Ping Identity Corporation

Symantec VIP Integration Kit

Use Symantec VIP MFA in your PingFederate sign-on flow.

ThreatMetrix Integration Kit

Use ThreatMetrix risk management in your PingFederate sign-on flow.

Twitter Login Integration Kit

Allow users to sign on with Twitter for PingFederate SSO.

U-Z

Web Access Management (WAM) Integration Kits

Integrate your WAM application with PingFederate for SSO.

Webex Provisioner

Integrate Webex with PingFederate for user provisioning and SSO.

Copyright © 2025 Ping Identity Corporation

WebSphere Integration Guide

Integrate WebSphere with PingFederate for SSO.

Workspace ONE UEM Integration Kit

Use Workspace ONE UEM secure device management in your PingFederate sign-on flow. Formerly known as the AirWatch Integration Kit.

X.509 Certificate Integration Kits

Integrate X.509 certificate authentication with PingFederate for SSO.

Zendesk Provisioner

Integrate Zendesk with PingFederate for user provisioning and SSO.

Zoom Provisioner

Integrate Zoom with PingFederate for user provisioning and SSO.

Zscaler Provisioners

Integrate Zscaler Internet Access and Private Access with PingFederate for user provisioning and SSO.

Copyright © 2025 Ping Identity Corporation

Agentless Integration Kit

The Agentless Integration Kit allows PingFederate to integrate with identity provider (IdP) and service provider (SP) applications for single sign-on (SSO).

This integration uses a RESTful approach that doesn't require you to integrate agent software into your application. Instead, your application exchanges user-session attributes with PingFederate through direct HTTP calls to the adapter endpoints.

The Reference ID IdP Adapter can also redirect OAuth authorization consent requests to an external application, as described in External consent approval.

Components

Reference ID IdP Adapter

Allows PingFederate to integrate with IdP applications for SSO.

Reference ID SP Adapter

Allows PingFederate to integrate with SP applications for SSO.

Agentless IdP Java sample application

An example IdP application that you can use to test the IdP adapter.

Agentless SP Java sample application

An example SP application that you can use to test the SP adapter.

> **Note**
>
> The sample applications included in the integration `.zip` archive are built with Java. You can find sample applications built with other languages, such as .NET Core, PHP, and Python, and get the latest version of the Java sample applications in the Ping Identity GitHub repository.

Intended audience

This document is intended for PingFederate administrators.

If you need help during the setup process, you can find more information in the following sections of the PingFederate documentation:

- Managing IdP adapters
- Managing SP adapters
- Authentication policies

System requirements

- PingFederate 11.3 or later

Copyright © 2025 Ping Identity Corporation

- Your application must be web-based and able to do the following:
  - Make REST API calls to PingFederate
  - Store a reference ID, resume path, and other values

Overview of the identity provider integration

You can use the Agentless Integration Kit to extend your sign-on flow to include a custom authentication application.

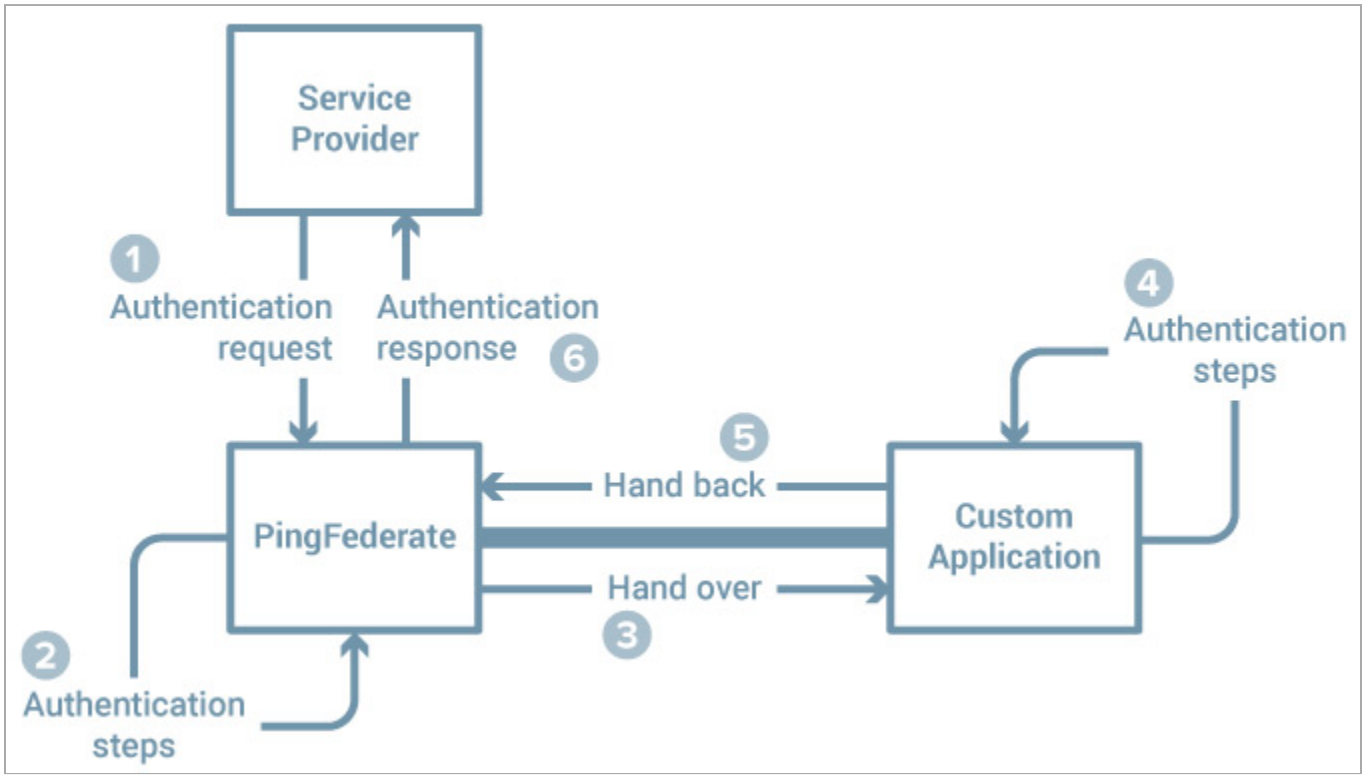
Use case

Some proprietary, third-party, or complex authentication applications might not be covered by the ready-made solutions that are included with PingFederate or available in the Ping Identity Integration Directory.

The Agentless Integration Kit offers a flexible way to integrate any application into the PingFederate authentication flow. This allows you to use your existing application without needing to develop a custom adapter for PingFederate. The only requirement is that your application can make REST API calls.

High-level view of the sign-on flow

Your application gets information from PingFederate, performs any number of authentication steps, then passes the user back to PingFederate to complete the sign-on process.



1. The user initiates the sign-on process, either at the identity provider (IdP) or service provider (SP).
2. PingFederate starts the authentication policy, which can include any number of adapters or authentication steps.

Copyright © 2025 Ping Identity Corporation

3. When the Reference ID IdP Adapter is triggered in the authentication policy, PingFederate sends a set of values to the application's authentication endpoint, including the following:

   REF

   The unique reference ID that the application uses to pick up user attributes.

   `resumePath`

   The PingFederate URI that the application redirects the user to after authentication. This includes a random string that is unique to the user session.

   The application stores these values.

4. The application makes a call to the Reference ID IdP Adapter to pick up any user attributes that PingFederate has associated with the reference ID, such as the user ID, LDAP attributes, tracked HTTP parameters, and any claims.

   The application stores these attributes.

5. The application completes any number of authentication steps, then drops off resulting attributes to the Reference ID IdP Adapter using an HTTP POST call.

6. PingFederate continues executing the authentication policy, now with access to the user attributes provided by the application. PingFederate provides a response to the service provider.

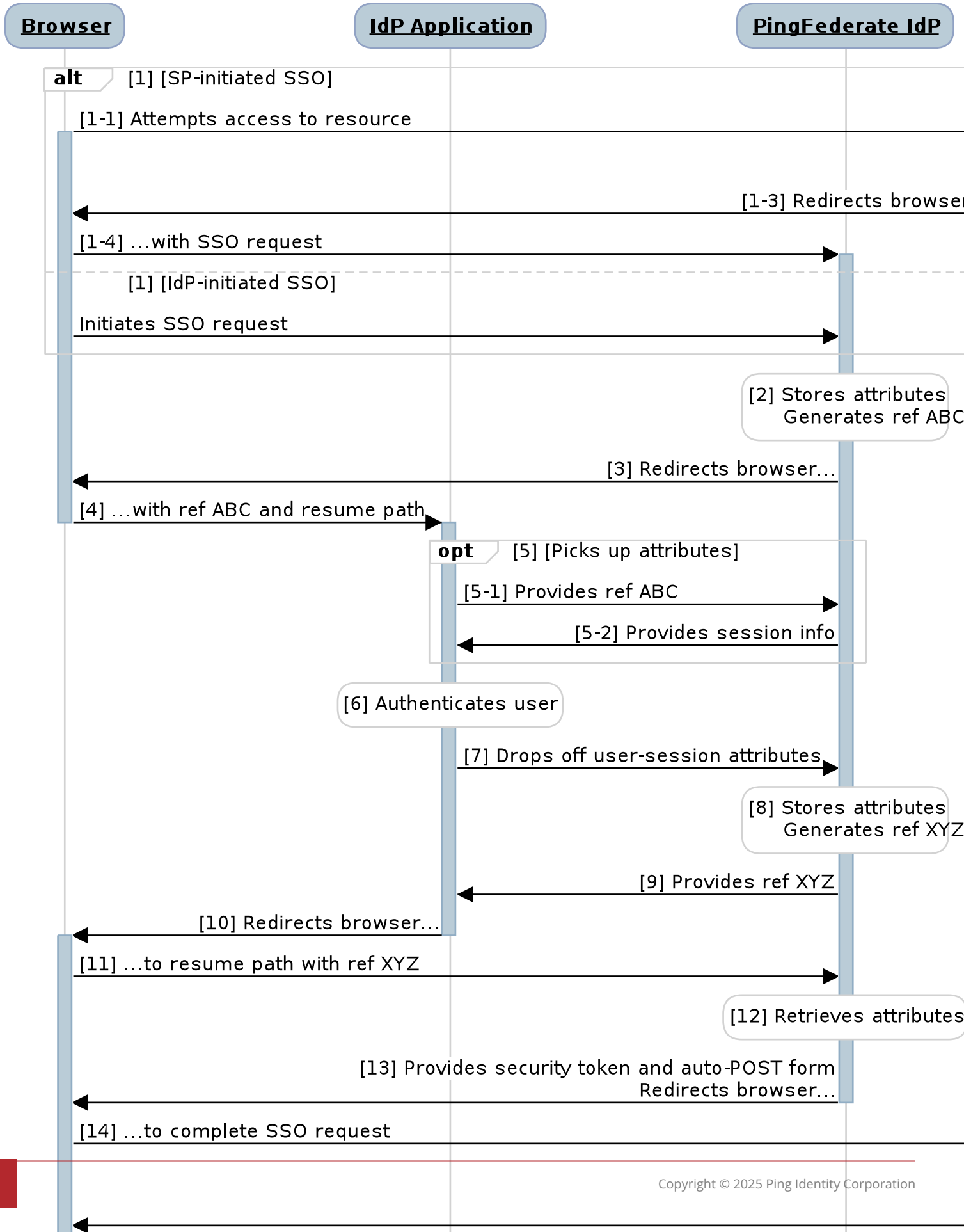
For a detailed description of the flow, see Overview of the identity provider SSO flow.

Overview of the identity provider SSO flow

With the Agentless Integration Kit, PingFederate includes your custom identity provider (IdP) application in the sign-on flow.

The following figure shows how your custom application is integrated into the sign-on process using the Reference ID IdP Adapter:

Copyright © 2025 Ping Identity Corporation



Copyright © 2025 Ping Identity Corporation

Processing steps

1. Single sign-on (SSO) is initiated by the service provider (SP) or IdP.
    - SP-initiated SSO:
        1. Through the browser, the user tries to access a protected resource.
        2. The SP application starts an SP-initated SSO request.
        3. The SP sends a redirect action to the browser.
        4. The browser redirects to PingFederate with an authentication request to start an SP-initiated SSO request.
    - IdP-initiated SSO:
        - Through the browser, the user starts an IdP-initiated SSO request.
2. PingFederate stores user-session attributes, including values from tracked HTTP parameters, and generates a reference value (ABC). Learn more in Development considerations.
3. PingFederate sends a redirect action to the browser.
4. The browser redirects to the IdP application with the reference value (ABC) and a resume path.
5. The IdP application picks up the user-session attributes.
    1. The IdP application sends the reference value (ABC) to PingFederate and requests the user-session attributes.
    2. PingFederate verifies the reference value (ABC) and returns the user-session attributes, including values from tracked HTTP parameters, to the IdP application.

    > **Note**
    >
    > You can find a list of all possible user-session attributes in the **IdpAuthenticationAdapterV2** section of the PingFederate Javadocs at `<pf_install>/PingFederate/sdk/doc/index.html` .
    > If you select **Include Null Attributes** in the adapter configuration, null attributes are included in the response from the pickup endpoint. Learn more in Reference ID IdP Adapter settings reference.
6. The IdP application authenticates the user.
7. The IdP application uses a back-channel call to authenticate to PingFederate and drop off the user-session attributes. Learn more in Authentication methods.
8. PingFederate stores the user-session attributes and generates a second reference value (XYZ).
9. PingFederate returns the second reference value (XYZ) to the IdP application using the back-channel call.
10. The IdP application sends a redirect action to the browser.
11. The browser redirects to the resume path with the second reference value (XYZ).
12. PingFederate retrieves the user-session attributes associated with the second reference value (XYZ) and processes the request.
13. PingFederate sends a redirect action to the browser and provides a security token and auto-POST form.

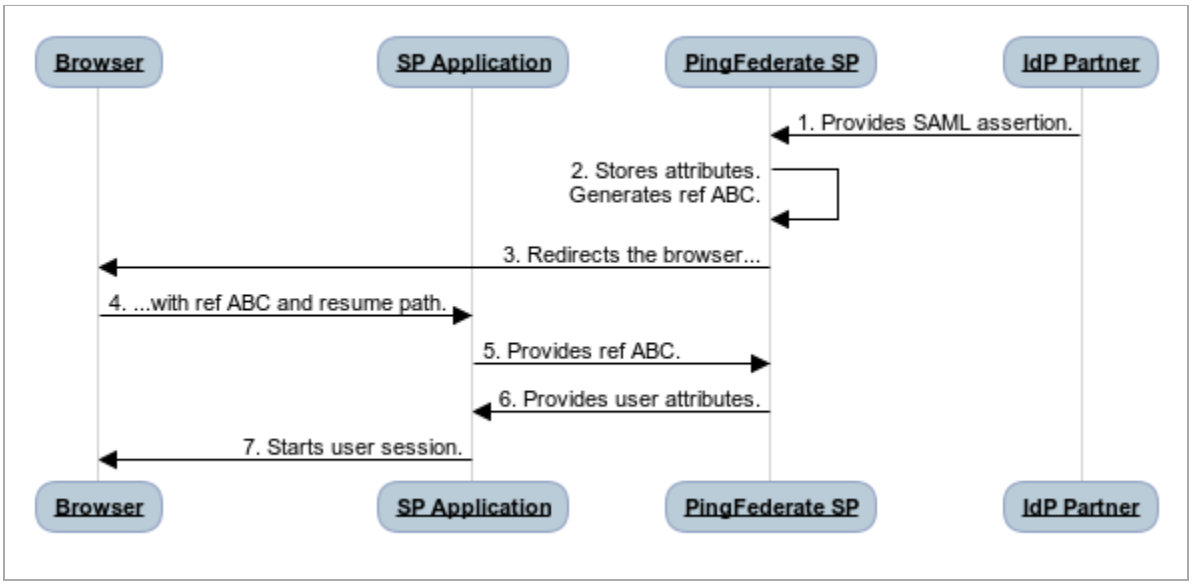
Copyright © 2025 Ping Identity Corporation

14. The browser redirects to the SP to complete the SSO request.
15. The SP completes the SSO request.
16. The SP application returns the protected resource that the user requested.

Overview of the service provider SSO flow

With the Agentless Integration Kit, PingFederate includes your custom service provider (SP) application in the sign-on flow.

The following figure shows how your custom application is integrated into the sign-on process using the Reference ID SP Adapter:



Description

1. The identity provider (IdP) sends a SAML assertion to PingFederate.
2. PingFederate validates the assertion and temporarily captures the user-session attributes. The Reference ID SP Adapter generates a reference value (ABC). Learn more in Development considerations.
3. PingFederate redirects the browser...
4. ...to the SP application with the reference value (ABC).

   The reference is included in the URL query string: https://target.example.com?REF=<*referenceValue*>
5. The SP application sends the reference value (ABC) to PingFederate and requests the user-session attributes.

   The application makes the request through an authenticated direct HTTP call to the pickup endpoint: https://pingfederate.example.com:9031/ext/ref/pickup?REF=<*referenceValue*>

Copyright © 2025 Ping Identity Corporation

> **Note**
>
> The applications must authenticate to PingFederate using one of the three methods described in Authentication methods. If the authentication fails, the HTTP request results in an HTTP response `401 – Unauthorized` status code message.
> If you select **Include Null Attributes** in the adapter configuration, null attributes are included in the response from the pickup endpoint. Learn more in Reference ID SP Adapter settings reference.

6. PingFederate verifies the reference value (ABC) and returns the user-session attributes to the SP application in the HTTP response.
7. The target application uses the attributes associated with the reference value to create a user session, which allows the user to access the resource.

Custom application setup

You can integrate PingFederate with your identity provider or service provider application by modifying your application and configuring a Reference ID Adapter instance.

> **Tip**
>
> If you would like to see a working demonstration of the Agentless Integration Kit before integrating your own applications, see Sample application setup.

Developing your application

The Agentless Integration Kit doesn't require you to install special agent software, but you will have to modify your application to communicate with the PingFederate Reference ID Adapter through HTTP calls. The following information and recommendations should guide your development process.

About this task

> **Important**
>
> This section provides example code in Java. You can find the latest example code for a variety of languages (including .NET, PHP, and Python) in the Agentless Integration Kit projects in the Ping Identity GitHub repository. Choose the language you use and see the `example-code` directory to help you with the following steps.

Steps

1. Determine whether your application will need to pick up attributes, drop off attributes, or both.

   Typically, if your application acts as an identity provider, it will drop off attributes with PingFederate. If your application acts as a service provider, it will pick up attributes from PingFederate.

   Learn more about the two flows in Overview of the identity provider SSO flow and Overview of the service provider SSO flow.

2. Configure an endpoint in your application to receive `REF` and `resumePath` values from PingFederate.

Copyright © 2025 Ping Identity Corporation

> **Note**
>
> Note the endpoint URL. You'll enter it in the **Authentication Endpoint** field when you configure a Reference ID Adapter instance later in the setup process.

3. Decide how your application will authenticate with PingFederate.

   Learn more in Authentication methods.

4. Configure your application to do one or both of the following:

   - Request attributes from PingFederate and parse the response as shown in Attribute pickup process.
   - Encode attributes and provide them to PingFederate as shown in Attribute drop-off process.

5. Configure your application to redirect the browser to the URL provided in the `resumePath` parameter, which PingFederate provided in the attribute drop-off process.

   Redirect URL format:

   ```
   https://pf_host:pf_port/resume_path?REF=reference_ID
   ```

   Example code:

   ```
   String resumeURL = String.format("https://localhost:9031%s?REF=%s", resumePath, ref);

   response.sendRedirect(resumeURL);
   ```

Development considerations

The Agentless Integration Kit does not require you to install special agent software in your application, but it does require your application to be modified to communicate with the PingFederate Reference ID Adapter. The following information and recommendations should guide your modifications.

Reference ID length

The reference ID is a long hexadecimal string. Length is determined by the **Reference Length** setting in the Reference ID Adapter instance configuration. The default length is 30 bytes.

For example:

```
A9C020F7CF8C21002CDC774B48A7CFE6B3ECA5FC6CCA507EE419B4432DB
```

Each reference ID is specific to the instance of the Reference ID Adapter that issued it.

Error handling

During the attribute pickup or drop-off processes, PingFederate returns any errors as HTTP status codes. For example when an incorrect client ID or password is provided, PingFederate returns a 401 HTTP error.

Copyright © 2025 Ping Identity Corporation

If an error occurs during an authentication attempt, then the authentication form should handle the error (that is, display a message to the user) and abort the sign on process. Your application will only receive a sign-in request from a user that was authenticated by the identity provider (IdP). Failed authentications are not redirected back to the application. Any errors received during the authentication process should be handled with the assumption that the user has already completed the authentication process at the IdP.

If your application presents an invalid reference ID to PingFederate, the Reference ID Adapter returns an empty set of attributes.

> **Tip**
>
> Locate error screens outside the protected content. When a user receives an error during the sign-on or authorization stages, they need to be able to see the error without being sent back through the authentication process.

Timeouts

A given reference value can only be used once and expires after a set duration. This improves security by preventing "replay" attacks. The duration is determined by the **Reference Duration** field in the Reference ID Adapter instance configuration. The default duration is three seconds.

To prevent unexpected timeouts, make sure the clock is reasonably synchronized between the application server and the federation server. You can also adjust the value of the **Reference Duration** configuration setting can also be used to allow for a larger tolerance for time skew.

Session management

The IdP adapter should maintain a session for the user so that subsequent calls for authentication can re-use the existing sign-on session. This creates a single sign-on experience for the user.

The IdP adapter should honor the `forceAuthn` parameter to force the user to enter credentials even if the user has an existing session.

Deep linking

Deep linking (also called direct linking) is the ability to link to a page nested inside the protected content. This saves the user from having to navigate to the content from a generic landing page.

Your application receives a deep link in the `TargetResource` parameter. This parameter is included in the initial redirect to the sign-on URL.

Your application should honor this URL during the sign-on process by redirecting the user to the deep link after the application session is created. When recovering from a timeout or other scenario that might require re-authentication, your application should include the requested URL in the request.

> **Tip**
>
> When passing the `TargetResource` parameter, or any URL parameter, remember to URL-encode the value.

Application authorization

Authorization includes determining whether the user has access to the application and what roles the user has when they are inside the application.

Copyright © 2025 Ping Identity Corporation

Your service provider application will only receive users that have already been authenticated. Your application should perform any authorization decisions before creating an application session.

> **Tip**
>
> Locate error pages outside the protected content. When a user receives an error during the sign-on or authorization stages, they need to be able to see the error without being sent back through the authentication process.

Application user profile management

Your application will receive an already authenticated user from a trusted identity provider that can contain additional attributes. This authenticated identity can be used to provision new accounts into the application store or update existing user profile information.

Authentication methods

When picking up and dropping off attributes, your application has to authenticate with PingFederate. There are four authentication methods that you can use:

1. Bearer token authentication (IETF RFC 6750)
2. Certificate authentication (mutual TLS)
3. Custom HTTP header variables (`ping.uname` and `ping.pwd`)
4. HTTP Basic authentication (IETF RFC 7617)

> **Note**
>
> If you enable both HTTP Basic authentication and certificate authentication methods, the application can authenticate with either.

If authentication fails, PingFederate responds with a `401` HTTP response to the application.

Bearer token authentication

The application depends on the PingFederate authorization server for authentication and authorization. If the adapter is configured for bearer token-based authentication, the resulting OAuth access token is included in the Authorization header of the HTTP requests for the pickup and drop-off APIs.

```
"Authorization: Bearer " + <access token>
```


Copyright © 2025 Ping Identity Corporation

```
POST https://pf.example.com:9031/ext/ref/dropoff HTTP/1.1
Content-Length: 20
Content-Type: application/json
Authorization: Bearer
eyJhbGciOiJSUzI1NiIsImtpZCI6ImsxIiwicGkuYXRtIjoiNnBhYyJ9.eyJzY29wZSI6WyJvcGVuaWQiXSwiYXV0aG9yaXphdGlvbl9kZXRhaWxzIjpb
g0NhLzC03oV0I4-PxMZWffSAn6UhvPyBwlAQ1KH0whdbQm84oSM93u2gffyF9qtf-34PycLIbhiR7syGS-
uaTGSnGVKIoOrNp5GeKJ6gBA48sInSKzO9LnqTzvIeN-
vDpU3SVb16EBCx5UjJRIHxTPHonzQLld7Au_FAKGpG6eQfzUbPt0DvJyealzWLdBsn4VgdegtZJNQbnF9UhmgS5ead2wn_skAG-
g_dekkePUN44LMd5B5Yf0V-xSAJmU8LYMqyB8ZedLwH-9ObUKig4kJZwGGlsGyGALVnU9f60nOfXLUZSPb8H6YGmoy-Fw
ping.instanceId: sample_adapter

{
"subject":"jsmith"
}
```

To use this method, complete the **Access Token Manager**, **Allowed Bearer Access Token Client IDs**, and **Required Bearer Access Token Scopes** fields in the Reference ID Adapter instance configuration. Learn more in Reference ID IdP Adapter settings reference and Reference ID SP Adapter settings reference.

Certificate authentication

Authentication relies on a client SSL private key and the corresponding public certificate. The application sends a request to the Reference ID Adapter endpoints using the back-channel port.

> **Note**
>
> The certificate is transported during the SSL/TLS negotiation and does not appear in the HTTP request.

```
POST https://pf.example.com:9032/ext/ref/dropoff HTTP/1.1
Content-Length: 20
Content-Type: application/json
ping.instanceId: sample_adapter

{
  "subject":"jsmith"
}
```

To use this method, complete the steps in Configuring certificate authentication.

Custom HTTP header variables

The application includes the **User Name** and **Pass Phrase** as the value of the `ping.uname` and `ping.pwd` HTTP headers in the HTTP request.

> **Note**
>
> Use this method if your application does not support Base64 encoding or certificate authentication.

Copyright © 2025 Ping Identity Corporation

```
POST https://pf.example.com:9031/ext/ref/dropoff HTTP/1.1
Content-Length: 20
Content-Type: application/json
ping.uname: sample_id ping.pwd: sample_password
ping.instanceId: sample_adapter

{
  "subject":"jsmith"
}
```

To use this method, complete the **User Name** and **Pass Phrase** fields in the Reference ID Adapter instance configuration. Learn more in Reference ID IdP Adapter settings reference and Reference ID SP Adapter settings reference.

HTTP Basic authentication

The application encodes the **User Name** and **Pass Phrase** together using Base64. It includes the result as the value of the `Authorization` header in the HTTP request.

```
"Authorization: BASIC " + Base64_Encode(  <username>  + ":" +  <pass phrase> )

"Authorization: BASIC " + Base64_Encode("myportal:q6^&2dR!Vc7PtA")

"Authorization: BASIC c2FtcGxlX2lkOnNhbXBsZV9wYXNzd29yZA=="
```

```
POST https://pf.example.com:9031/ext/ref/dropoff HTTP/1.1
Content-Length: 20
Content-Type: application/json
Authorization: BASIC c2FtcGxlX2lkOnNhbXBsZV9wYXNzd29yZA==
ping.instanceId: sample_adapter

{
  "subject":"jsmith"
}
```

To use this method, complete the **User Name** and **Pass Phrase** fields in the Reference ID Adapter instance configuration. Learn more in Reference ID IdP Adapter settings reference and Reference ID SP Adapter settings reference.

Configuring certificate authentication

You can configure certificate authentication between your application and PingFederate.

Steps

1. Check that the client certificate issuer is a trusted root certificate authority (CA) in PingFederate. If not, add the intermediate and root CA certificates.

   Learn more in Manage trusted certificate authorities in the PingFederate documentation.

Copyright © 2025 Ping Identity Corporation

2. In PingFederate, export your signing certificate.
    1. On the PingFederate admin console, go to **Security > Signing & Decryption Keys & Certificates**.
    2. For the certificate that you want to use, in the **Action** column, click **Export**.
    3. On the **Export Certificate** screen, click **Next**.
    4. On the **Export & Summary** screen, click **Export**.
    5. Open the `*.crt` file in a text editor.
3. Import your PingFederate signing certificate into your application.

> **Tip**
>
> You can use OpenSSL to convert the PCKS12 certificate and key to PEM format. Use one of the following commands.
>
> **PEM certificate only**
>
> ```
> openssl pkcs12 -in  <certname>.p12-passin pass:<password>  -nokeys -out
> <certname>.cert.pem
> ```
>
> **PEM key only**
>
> ```
> openssl pkcs12 -in  <certname>.p12 -passin pass:<password>  -nocerts -out
> <certname>.key.pem
> ```
>
> **PEM certificate and key**
>
> ```
> openssl pkcs12 -in  <certname>.p12 -passin pass:<password>  -out
> <certname>.certandkey.pem
> ```

4. If you have already configured a Reference ID Adapter instance, update it by setting the **Allowed Subject DN** field, the **Allowed Issuer DN** field, or both to match the client certificate.
5. Configure a second port for PingFederate to receive back-channel calls.
    1. Stop PingFederate.
    2. Open the `<pf_install>/pingfederate/bin/run.properties` file for editing.
    3. Change the value of the `pf.secondary.https.port` property to a valid port number, such as 9032.

       You can find information about this property in Configuring PingFederate properties in the PingFederate documentation.
    4. Save the file.
    5. Start PingFederate.

Copyright © 2025 Ping Identity Corporation

6. Configure your application to send requests to the Reference ID Adapter endpoints using the back-channel port:

```
POST https://pf.example.com:9032/ext/ref/dropoff HTTP/1.1
Content-Length: 20
Content-Type: application/json
ping.instanceId: sample_adapter

{
  "subject":"jsmith"
}
```

7. Configure your application to send the client certificate with the request.

> **Tip**
>
> Learn more about exporting your PingFederate certificate in Manage SSL server certificates in the PingFederate documentation.

The following code uses PHP to send the client certificate:

```
$client_cert = dirname(__FILE__).'/sample_cert.cert.pem';
$client_key = dirname(__FILE__).'/sample_cert.key.pem';
$client_key_password = 'sample_key_password';
$http_headers[] = 'ping.instanceId: '.$adapter_instance_id;

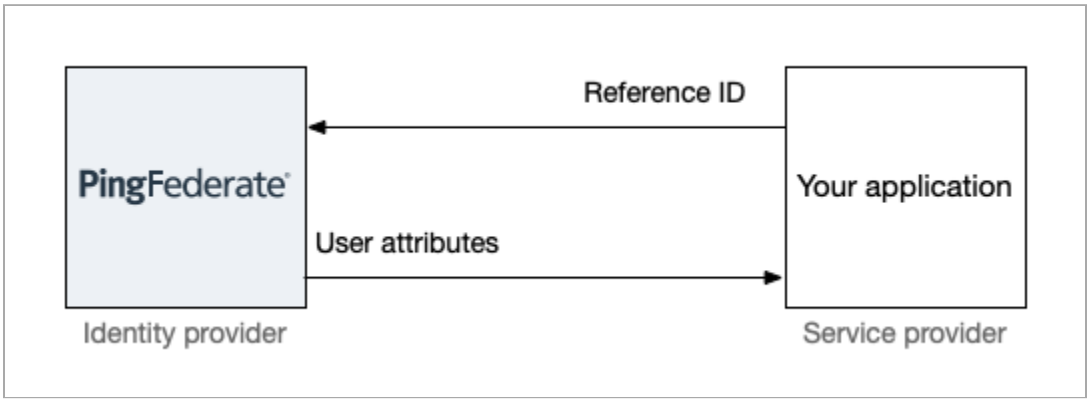
// PHP can use curl to make the HTTP calls to the pickup endpoint
$crl = curl_init();

// Dropoff URL
curl_setopt($crl, CURLOPT_URL, $dropoff_loc);
curl_setopt($crl, CURLOPT_SSLCERT, $client_cert);
curl_setopt($crl, CURLOPT_SSLKEYTYPE, 'PEM');
curl_setopt($crl, CURLOPT_SSLKEY, $client_key);
curl_setopt($crl, CURLOPT_SSLKEYPASSWD, $client_key_password);
...
$result = curl_exec($crl);
```

Attribute pickup process

To pick up user attributes from PingFederate, configure your application to authenticate itself, provide the reference ID, and store the user attributes from the response. This lets you get the user information needed to create a session in your application.

Copyright © 2025 Ping Identity Corporation



Requesting the user attributes

Configure your application to make an HTTP GET call to the Reference ID Adapter pickup endpoint:

```
https://pf_host:pf_port/ext/ref/pickup
```

In the call, provide the following:

- The `REF` value.
- Authentication credentials, if you are using the HTTP Basic or HTTP header authentication. Learn more in Authentication methods.
- The ID of the Reference ID Adapter instance.

Example code:

```
// Call back to PF to get the attributes associated with the reference
String pickupLocation = "https://localhost:9031/ext/ref/pickup?REF=" + referenceValue;
System.out.println(pickupLocation);
URL pickUrl = new URL(pickupLocation);
URLConnection urlConn = pickUrl.openConnection();
HttpsURLConnection httpsURLConn = (HttpsURLConnection)urlConn;
httpsURLConn.setSSLSocketFactory(socketFactory);
urlConn.setRequestProperty("ping.uname", "changeme"); urlConn.setRequestProperty("ping.pwd", "please change me before you go into production!"); urlConn.setRequestProperty("ping.instanceId", "spadapter");
```

Parsing the attributes from the response

The response from PingFederate includes user attributes from earlier in the authentication flow. They are encoded as a JSON object or as properties using the `java.util.Properties` class.

> **Tip**
>
> Your adapter instance configuration determines which attributes are included in the response and how they are formatted. See the **Send Request Parameters** and **Outgoing Attribute Format** settings.

Example response:

Copyright © 2025 Ping Identity Corporation

```
HTTP/1.1 200 OK
{
"authnCtx":"urn:oasis:names:tc:SAML:2.0:ac:classes:unspecified",
"partnerEntityID":"company:saml20:idp",
"subject":"jsmith",
"instanceId":"sample_adapter",
"sessionid":"sFMcTOaropYv5gYQZi1ZOpX7DZ8",
"authnInst":"2013-03-28 20:42:10-0500"
}
```

Configure your application to parse the response and store the user attributes.

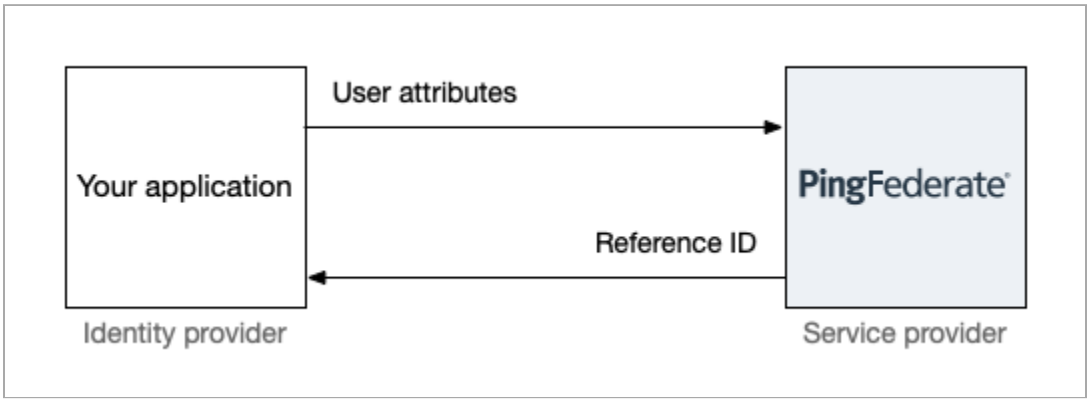
Example code:

```
// Get the response and parse it into another JSON object which are the 'user attributes'.
// This example uses UTF-8 if encoding is not found in request.
String encoding = urlConn.getContentEncoding();
InputStream is = urlConn.getInputStream();
InputStreamReader streamReader = new InputStreamReader(is, encoding != null ? encoding : "UTF-8");

JSONParser parser = new JSONParser();
JSONObject spUserAttributes = (JSONObject)parser.parse(streamReader);
System.out.println("User Attributes received = " + spUserAttributes.toString());
```

Attribute drop-off process

To drop off user attributes to PingFederate, configure your application to authenticate itself, provide the encoded attributes, and store a reference ID from the response.



Providing the user attributes

Configure your application to make an HTTP POST call to the Reference ID Adapter dropoff endpoint:

```
https://pf_host:pf_port/ext/ref/dropoff
```


Copyright © 2025 Ping Identity Corporation

In the call, provide the following:

- Authentication credentials, if you are using the HTTP Basic or HTTP header authentication. Learn more in Authentication methods.
- The ID of the Reference ID Adapter instance.
- The user attributes.

PingFederate expects the attributes to be encoded as a JSON object or as query parameters in the URL, depending on the **Incoming Attribute Format** setting in your adapter instance configuration.

Example code:

```
// Drop the attributes into PingFederate
String dropoffLocation = "https://localhost:9031/ext/ref/dropoff";
System.out.println(dropoffLocation);
URL dropUrl = new URL(dropoffLocation);
URLConnection urlConnection = dropUrl.openConnection();
HttpsURLConnection httpsURLConnection = (HttpsURLConnection)urlConnection;
httpsURLConnection.setSSLSocketFactory(socketFactory);
urlConnection.setRequestProperty("ping.uname", "changeme"); urlConnection.setRequestProperty("ping.pwd",
"this is a default example and should not be used in production");
urlConnection.setRequestProperty("ping.instanceId", "idpadapter");

// Write the attributes in URL Connection, this example uses UTF-8 encoding
urlConnection.setDoOutput(true);
OutputStreamWriter outputStreamWriter = new OutputStreamWriter(urlConnection.getOutputStream(),
StandardCharsets.UTF_8);
idpUserAttributes.writeJSONString(outputStreamWriter);
outputStreamWriter.flush();
outputStreamWriter.close();
```

Parsing the reference ID from the response

The response from PingFederate includes a newly-generated reference ID associated with the attributes it received. You will use this `REF` value later in the authentication flow.

Example response:

```
HTTP/1.1 200 OK

{
“REF”:”54321”
}
```

Configure your application to parse the response and store the reference ID.

Example code:

Copyright © 2025 Ping Identity Corporation

```
// Get the response and parse it into a JSON object
InputStream is = urlConnection.getInputStream();
InputStreamReader streamReader = new InputStreamReader(is, StandardCharsets.UTF_8);

JSONParser parser = new JSONParser();
JSONObject jsonRespObj = (JSONObject)parser.parse(streamReader);

// Grab the value of the reference Id from the JSON Object. This value
// must be passed to {pingfed} on resumePath as the parameter 'REF'
String referenceValue = (String)jsonRespObj.get("REF");
System.out.println("Reference ID = " + referenceValue);
```

External consent approval

During the OAuth authorization flow, after a user authenticates, PingFederate directs the user to an authorization consent screen. On this screen, users can consent to a scope of privileges that the client has requested.

Instead of using the consent screen that is provided with PingFederate, you can direct users to your own application to process the consent requests. This gives you more control over the presentation of the consent request.

Learn more in Consent approval in the PingFederate documentation.

Identifying consent requests

PingFederate sends authentication requests and authorization consent requests to your application through the same endpoint URL. To process the request and display the appropriate page to the user, your application must be able to differentiate authentication requests from consent requests.

All consent requests contain the following attribute:

Key: `com.pingidentity.adapter.input.parameter.adapter.action`

Value: `com.pingidentity.adapter.action.external.consent`

Because authentication requests do not contain this attribute, your application can differentiate authentication and consent requests based on whether this attribute is present.

Passing "consent declined" results to PingFederate

If a user rejects the scope of privileges requested by a client, your application needs to signal that rejection to PingFederate.

In this case, your application must drop off the following attribute to PingFederate:

Key: `com.pingidentity.adapter.refid.external.application.failure.message`

Value: An optional error message.

Deploying the integration files

To get started with the integration, deploy the Agentless Integration Kit files to your PingFederate directory.

Copyright © 2025 Ping Identity Corporation

About this task

> **Note**
>
> If you operate PingFederate in a cluster, the following steps refer to the console node.

Steps

1. Download the Agentless Integration Kit `.zip` archive from the **Add-ons** tab of the PingFederate download page or the Ping Identity Integration Directory.
2. Stop PingFederate.
3. If you are upgrading an existing deployment, delete `pf-referenceid-adapter-<version>.jar` from `<pf_install>/pingfederate/server/default/deploy`.
4. Extract the `.zip` archive and copy the contents of the `dist` directory to your `<pf_install>/pingfederate/server/default` directory.
5. If there is more than one version of the `pf-authn-api-sdk-<version>.jar` file in your `<pf_install>/pingfederate/server/default/lib` directory, delete all but the latest version of the file.
6. Start PingFederate.
7. If you operate PingFederate in a cluster, repeat steps 2 - 6 for each engine node.

Configuring a Reference ID IdP Adapter instance

Configure the Reference ID IdP Adapter to determine how PingFederate communicates with your identity provider application.

About this task

You can configure the adapter to direct authentication requests, authorization consent requests, or both to your application.

Steps

1. In the PingFederate administrative console, create a new IdP adapter instance:

   Choose from:

   - For PingFederate 10.1 or later: go to **Authentication > Integration > IdP Adapters**. Click **Create New Instance**.
   - For PingFederate 10.0 or earlier: go to **Identity Provider > Adapters**. Click **Create New Instance**.

2. On the **Type** tab, set the basic adapter instance attributes.

   1. In the **Instance Name** field, enter a name for the adapter instance.
   2. In the **Instance ID** field, enter a unique identifier for the adapter instance.
   3. In the **Type** list, select **Reference ID IdP Adapter**. Click **Next**.

3. On the **IdP Adapter** tab, configure the adapter instance by referring to Reference ID IdP Adapter settings reference. Click **Next**.

Copyright © 2025 Ping Identity Corporation

4. **Optional:** On the **Actions** tab, click **Show Pass Phrase** and copy the pass phrase to your application. Click **Next**.
5. On the **Extended Contract** screen, add any attributes that you expect to retrieve in addition to the SAML subject (user ID). If you have an external consent application, add the attribute that contains the authorization scopes. Click **Next**.
6. On the **Adapter Attributes** tab, set pseudonym and masking options as shown in Set pseudonym and masking options in the PingFederate documentation. Click **Next**.
7. On the **Adapter Contract Mapping** tab, configure the contract fulfillment details for the adapter as shown in Define the IdP adapter contract in the PingFederate documentation. Click **Next**.
8. On the **Summary** tab, check and save your configuration:

   Choose from:

   - For PingFederate 10.1 or later: click **Save**.
   - For PingFederate 10.0 or earlier: click **Done**. On the **Manage IdP Adapter Instances** tab, click **Save**.
9. Create an SP connection using this Reference ID IdP Adapter instance. Learn more in Identity provider SSO configuration in the PingFederate documentation.

Reference ID IdP Adapter settings reference

Field descriptions for the Reference ID IdP Adapter configuration page.

Field Name	Description
Authentication Endpoint	The application endpoint URL for user authentication requests, user authorization consent requests, or both.
User Name	The ID that the application uses to authenticate to the PingFederate server. This field is required to enable Basic HTTP authentication for the application.
Pass Phrase	The password that the application uses to authenticate to the PingFederate server. This field is required to enable Basic HTTP authentication for the application.
Allowed Subject DN	If your application uses certificate authentication, set this to an acceptable subject distinguished name (DN) of the client certificate. You can use the asterisk (`*`) wildcard character to match variances in the value for components that are allowed to be variable, like the common name (CN). A maximum of one wildcard character can be used per DN. Separate multiple subject DNs with a pipe (`|`). If this field is blank, any subject DN is allowed.

Copyright © 2025 Ping Identity Corporation

Field Name	Description
Allowed Issuer DN	If your application uses certificate authentication, set this to an acceptable issuer DN of the incoming client certificate. You can use the asterisk (`*`) wildcard character to match variances in the value for components that are allowed to be variable, like the common name (CN). A maximum of one wildcard character can be used per DN. Separate multiple subject DNs with a pipe (`|`). If this field is blank, any issuer DN is allowed.
Access Token Manager	To enable bearer token authentication for applications, select an Access token management instance to issue and validate access tokens.
Allowed Bearer Access Token Client IDs	A list of client IDs allowed in the bearer access token. Separate values with the pipe character (`|`). If this field is blank, all client IDs are accepted.
Required Bearer Access Token Scopes	A list of required scopes that must be included in the bearer access token. Separate values with the pipe character (`|`). If this field is blank, all scopes are accepted.
Logout Service Endpoint	The URL of your application's logout service endpoint, such as https://portal.example.com/logout. When **Logout Mode** is set to **Front Channel**, PingFederate uses this URL as part of the single logout (SLO) flow. For details, see the description below.
Logout Mode	Determines how the adapter handles application logout. ***Front Channel*** 1. During the SLO flow, PingFederate redirects the browser to your application's **Logout Service Endpoint** URL and provides the reference ID and resume path values. 2. Your application uses the reference ID or a session cookie to identify and end the user session, then redirects the browser back to the PingFederate resume path. 3. PingFederate completes the SLO process. ***Back Channel*** The adapter sends a direct HTTP request to the IdP application. To include an attribute in a dynamic URL, use the `${attribute-name}` variable. ***None*** Select this option if your application does not maintain user sessions. The default selection is **None**.

Copyright © 2025 Ping Identity Corporation

Field Name	Description
Prefix Referenced Attributes	When selected, the adapter adds a prefix to attribute keys to identify their source. • Attributes from previous authentication sources (also known as chained attributes) are prefixed with `chainedattr.`. • Signed request object attributes are prefixed with `signedreqattr.`. • HTTP parameters are prefixed with `httpparam.`. Applicable only if the **Ignore Untracked HTTP Parameters** checkbox is not selected. • Tracked HTTP parameters are prefixed with `trackedparam.`. This feature requires PingFederate Server 9.2 or later. This check box is selected by default.
Preserve JSON	When selected, the adapter automatically preserves the incoming chained attribute value and tracked HTTP parameters as a JSON data type if the adapter can detect a valid JSON format. **Note** The adapter always passes the tracked HTTP parameter values as an array. This checkbox is cleared by default.
Ignore Untracked HTTP Parameters	When selected, the adapter ignores parameters from the initial sign-on HTTP request that aren't included on the **Tracked HTTP Parameters** tab of the **Authentication Policies** window. This check box is selected by default.
Send Request Parameters	Determines which parameters from the original sign-on HTTP request the adapter passes to the identity provider (IdP) application. Applies in query parameter mode. The default selection is **None**.
Include Null Attributes	When selected, the pickup API endpoint response includes any existing attributes with null values in the payload it drops off. Additionally, the Reference ID IdP Adapter includes null claims in the signed request object. This checkbox is cleared by default.
Transport Mode	This field defines the method that the adapter uses for front-channel communication with the application. ***Form POST*** The adapter sends data using a POST request. Data is contained within the body of the request. ***Query Parameter*** The adapter sends data as part of the URL string. Some data, such as the reference value, is exposed with this method. The default selection is **Form POST**.
Reference Duration	The amount of time (in seconds) that the PingFederate server keeps the referenced attributes in memory. Increase this value to accommodate network delays. Learn more in Development considerations. The default value is `3`.

Copyright © 2025 Ping Identity Corporation

Field Name	Description
Reference Length	The number of bytes used for the pseudo-randomly-generated reference ID. Increase this value to make the reference ID more difficult to replicate. Learn more in Development considerations. The default value is `30`.
Require SSL/TLS	This checkbox controls whether adapter requires a secure connection for calls made to the Reference ID IdP Adapter pickup and dropoff endpoints. This check box is selected by default.
Outgoing Attribute Format	The format that the adapter uses to encode attribute values in HTTP responses it sends to the application. The application must be able to parse this format. Learn more in Attribute pickup process. The default selection is **JSON**.
Incoming Attribute Format	The format that the application uses to encode attribute values in HTTP requests it sends to the adapter. Learn more in Attribute drop-off process. The default selection is **JSON**.
Skip Host Name Validation	When a connection is established with the application, this setting determines whether PingFederate matches the target host name against the names stored inside the server certificate presented by the application. This can be useful during development or testing. Applies when **Logout Mode** is set to **Back Channel**. This check box is cleared by default.
Relax Pass Phrase Requirements	When selected, the adapter does not enforce requirements for the application credentials entered in the **Pass Phrase** field. When cleared, the adapter enforces strong password requirements for better security. Use this for development, testing, or upgrading from previous versions of the adapter that did not enforce password requirements. This check box is cleared by default.

Configuring a Reference ID SP Adapter instance

Configure the Reference ID IdP Adapter to determine how PingFederate communicates with your service provider (SP) application.

Steps

1. In the PingFederate administrative console, create a new SP adapter instance.

   Choose from:

   - For PingFederate 10.1 or later: go to **Applications > Integration > SP Adapters**. Click **Create New Instance**.
   - For PingFederate 10.0 or earlier: go to **Service Provider > Adapters**. Click **Create New Instance**.

Copyright © 2025 Ping Identity Corporation

2. On the **Type** tab, set the basic adapter instance attributes.
    1. In the **Instance Name** field, enter a name for the adapter instance.
    2. In the **Instance ID** field, enter a unique identifier for the adapter instance.
    3. In the **Type** list, select **Reference ID SP Adapter**. Click **Next**.
3. On the **Instance Configuration** tab, configure the adapter instance by referring to Reference ID SP Adapter settings reference. Click **Next**.
4. **Optional:** On the **Actions** tab, click **Show Pass Phrase** and copy the pass phrase to your application. Click **Next**.
5. On the **Extended Contract** tab, add any attributes that you expect to retrieve other than the SAML subject. Click **Next**.
6. On the **Target App Info** tab, enter the basic information about your SP application. Click **Next**.
7. On the **Summary** tab, check and save your configuration.

    Choose from:

    - For PingFederate 10.1 or later: click **Save**.
    - For PingFederate 10.0 or earlier: click **Done**. On the **Manage SP Adapter Instances** tab, click **Save**.
8. Create an IdP connection using this Reference ID SP Adapter instance. Learn more in Service provider SSO configuration in the PingFederate documentation.

Reference ID SP Adapter settings reference

Field descriptions for the Reference ID SP Adapter configuration page.

Field Name	Description
Authentication Endpoint	The application endpoint URL for user authentication requests.
User Name	The ID that the application uses to authenticate to the PingFederate server. This field is required to enable Basic HTTP authentication for the application.
Pass Phrase	The password that the application uses to authenticate to the PingFederate server. This field is required to enable Basic HTTP authentication for the application.
Allowed Subject DN	For certificate authentication, this field specifies an acceptable subject DN of the client certificate. You can use the asterisk (`*`) wildcard character to match variances in the value for components that are allowed to be variable, like the common name (CN). A maximum of one wildcard character can be used per DN. Separate multiple subject DNs with a pipe (`|`). If this field is blank, any subject DN is allowed.

Copyright © 2025 Ping Identity Corporation

Field Name	Description
Allowed Issuer DN	For certificate authentication, this field specifies an acceptable issuer DN of the incoming client certificate. You can use the asterisk (`*`) wildcard character to match variances in the value for components that are allowed to be variable, like the common name (CN). A maximum of one wildcard character can be used per DN. Separate multiple subject DNs with a pipe (`|`). If this field is blank, any issuer DN is allowed.
Access Token Manager	To enable bearer token authentication for applications, select an Access token management instance to issue and validate access tokens.
Allowed Bearer Access Token Client IDs	A list of client IDs allowed in the bearer access token. Separate values with the pipe character (`|`). If this field is blank, all client IDs are accepted.
Required Bearer Access Token Scopes	A list of required scopes that must be included in the bearer access token. Separate values with the pipe character (`|`). If this field is blank, all scopes are accepted.
Logout Service Endpoint	The URL of your application's logout service endpoint, such as https://portal.example.com/logout. When **Logout Mode** is set to **Front Channel**, PingFederate uses this URL as part of the single logout (SLO) flow. You can find more details in the following description.
Logout Mode	Determines how the adapter handles application logout. ***Front Channel*** 1. During the SLO flow, PingFederate redirects the browser to your application's **Logout Service Endpoint** URL and provides the reference ID and resume path values. 2. Your application uses the reference ID or a session cookie to identify and end the user session, then redirects the browser back to the PingFederate resume path. 3. PingFederate completes the SLO process. ***Back Channel*** The adapter sends a direct HTTP request to the IdP application. To include an attribute in a dynamic URL, use the `${attribute-name}` variable. ***None*** Select this option if your application does not maintain user sessions. The default selection is **None**.
Account Linking Authentication Endpoint	The SP application URL where incoming SSO users can access IDs for local accounts, via SAML account linking. Learn more about account linking in About account linking in the PingFederate documentation.

Copyright © 2025 Ping Identity Corporation

Field Name	Description
Include Null Attributes	When selected, the pickup API endpoint response includes any existing attributes with null values in the payload it drops off. This checkbox is cleared by default.
Transport Mode	This field defines the method that the adapter uses for front-channel communication with the application. ***Form POST*** The adapter sends data using a POST request. Data is contained within the body of the request. ***Query Parameter*** The adapter sends data as part of the URL string. Some data, such as the reference value, is exposed with this method. The default selection is **Form POST**.
Reference Duration	The amount of time (in seconds) that the PingFederate server keeps the referenced attributes in memory. Increase this value to accommodate network delays. Learn more in Development considerations. The default value is `3`.
Reference Length	The number of bytes used for the pseudo-randomly-generated reference ID. Increase this value to make the reference ID more difficult to replicate. Learn more in Development considerations. The default value is `30`.
Require SSL/TLS	This checkbox controls whether PingFederate requires a secure connection for calls made to the attribute-retrieval endpoint. We recommend using the secure transport protocol unless a secure, dedicated network segment exists between the application server and PingFederate. This check box is selected by default.
Outgoing Attribute Format	The format that the adapter uses to encodes attribute values in HTTP responses it sends to the application. The application must be able to parse this format. Learn more in Attribute formatting. The default selection is **JSON**.
Incoming Attribute Format	The format that the application uses to encode attribute values in HTTP requests it sends to the adapter. Learn more in Attribute formatting. The default selection is **JSON**.
Skip Host Name Validation	When a connection is established with the application, this setting determines whether PingFederate matches the target hostname against the names stored inside the server certificate presented by the application. This can be useful during development or testing. Applies when **Logout Mode** is set to **Back Channel**. This check box is cleared by default.

Copyright © 2025 Ping Identity Corporation

Field Name	Description
Relax Pass Phrase Requirements	When selected, the adapter does not enforce requirements for the application credentials entered in the **Pass Phrase** field. When cleared, the adapter enforces strong password requirements for better security. Use this for development, testing, or upgrading from previous versions of the adapter that did not enforce password requirements. This check box is cleared by default.

Attribute formatting

Attribute formatting specifies how attribute names and values are passed between the application and PingFederate. To set the incoming and outgoing attribute formats, see the **Advanced Fields** section of the adapter instance configuration.

By default, PingFederate uses JSON to format incoming and outgoing attributes.

Incoming attributes

Incoming attribute formatting specifies how attributes are encoded into the HTTP request from the application to the PingFederate dropoff endpoint. Learn more in Reference ID Adapter endpoints.

Incoming attribute formats

Format	Details
JSON	Attributes are encoded according to the JSON data model.
Query parameter	Attributes are encoded as query string parameters in the URL.

Outgoing attributes

Outgoing attribute formatting specifies how attributes are encoded into the HTTP response from PingFederate to the application at the pickup endpoint.

Outgoing attribute formats

Format	Details
JSON	Attributes are encoded according to the JSON data model.
Properties	Attributes are encoded by using the `java.util.Properties` class.

Copyright © 2025 Ping Identity Corporation

Reference ID Adapter endpoints

/ext/ref/dropoff

The PingFederate URL that an application uses to drop off user-session attributes.

https://*<pf_host>*:*<pf_port>*/ext/ref/dropoff

/ext/ref/pickup

The PingFederate URL that an application uses to pick up user-session attributes.

https://*<pf_host>*:*<pf_port>*/ext/ref/pickup

Sample application setup

You can configure the Agentless Integration Kit to work with the included sample applications.

> **Note**
>
> This section discusses the Java sample applications that included in the integration `.zip` archive. For sample applications built with other languages, such as .NET Core, PHP, and Python, or to get the latest version of the Java sample applications, see the Ping Identity GitHub repository.

The Java sample applications allow you to try a working demonstration of the Agentless Integration Kit with identity-provider (IdP)-initiated single sign on (SSO), service provider (SP)-initiated SSO, and single logout.

Components

- IdP and SP Java web applications
- A PingFederate configuration archive
  - Configures a single instance of PingFederate with an example integration that uses both the IdP and SP sample applications.

Deploying the integration files and sample applications

To get started with the integration, deploy the Agentless Integration Kit files to your PingFederate directory.

Copyright © 2025 Ping Identity Corporation

About this task

> **Note**
>
> If you operate PingFederate in a cluster, the following steps refer to the console node.

Steps

1. Download the Agentless Integration Kit `.zip` archive from the **Add-ons** tab of the PingFederate download page or the Ping Identity Integration Directory.
2. Stop PingFederate.
3. If you are upgrading from a previous deployment, delete the following directories and files from `<pf_install>/pingfederate/server/default/deploy`.
    - `/AgentlessIntegrationKitSampleIdP`
    - `/AgentlessIntegrationKitSampleSP`
    - `AgentlessIdPSample.war`
    - `AgentlessSPSample.war`
    - `pf-referenceid-adapter-<version>.jar`
4. Extract the `.zip` archive and copy the contents of the `dist` directory to your `<pf_install>/pingfederate/server/default` directory.
5. From the `.zip` archive, copy the following files from `sample` to `<pf_install>/pingfederate/server/default/deploy`.
    - `AgentlessIdPSample.war`
    - `AgentlessSPSample.war`
6. If there is more than one version of the `pf-authn-api-sdk-<version>.jar` file in your `<pf_install>/pingfederate/server/default/lib` directory, delete all but the latest version of the file.
7. Start PingFederate.

Deploying the sample configuration archive

About this task

The sample configuration archive configures a single instance of PingFederate with an example integration that uses both the IdP and SP sample applications. It automatically creates instances of the Reference ID IdP Adapter and Reference ID SP Adapter.

> **Caution**
>
> Deploying the sample application configuration archive will destroy your existing PingFederate configuration. We recommend that you test it on a fresh installation of PingFederate, or back up your current configuration as shown in Exporting an archive in the PingFederate documentation.

Copyright © 2025 Ping Identity Corporation

Steps

1. Start PingFederate.
2. From the Agentless Integration Kit `.zip` archive, copy `config/data.zip` to `<pf_install>/pingfederate/server/default/data/drop-in-deployer`.

Sample application user accounts

Sample application user accounts

Username	Password
joe	test
scott	test
sarah	test

Testing IdP-initiated SSO

Steps

1. In a browser, go to https://pf_host:pf_port/AgentlessIdPSample/app.
2. Log in as one of the Sample application user accounts.
3. On the **Identity Provider Sample** screen, review the attributes that the identity provider (IdP) application will send to the service provider (SP) application. Click **OK**.
4. On the **Identity Provider Sample** screen, review the attributes that the SP application received from the IdP application.
5. **Optional:** In the **Sign On** area, try the demonstration again.

Testing SP-initiated SSO

Steps

1. In a browser, go to https://*pf_host*:*pf_port*/AgentlessSPSample/app.
2. On the **Service Provider Sample** screen, click **Single Sign On**.
3. On the **Identity Provider Sample** screen, sign on as one of the Sample application user accounts.
4. Review the attributes that the identity provider (IdP) application will send to the service provider (SP) application. Click **OK**.
5. On the **Identity Provider Sample** screen, review the attributes that the SP application received from the IdP application.
6. Click **Single Logout**.

Copyright © 2025 Ping Identity Corporation

7. Click **OK**.
8. On the **Identity Provider Sample** screen, click **OK**.

Troubleshooting

Enabling logging

You can choose to show Reference ID adapter events in the PingFederate server log file.

About this task

For more information about using Log4j, see PingFederate Log4J articles in the Knowledge Base, and Log4j 2 logging service and configuration in the PingFederate documentation.

Steps

1. In your Java Virtual Machine options, set the `javax.net.debug` property to `ssl` or `all`.
2. Restart PingFederate.
3. See the debug messages in the `<pf_install>/pingfederate/log/server.log` file.

Release notes

Changelog

The following is the change history for the Agentless Integration Kit.

Agentless Integration Kit 2.2.1 - May 2025

- Fixed an issue that prevented the adapter configuration from supporting exclusive and dynamic scopes in Bearer access token authentication flow.
- Fixed an issue where certificate details were not correctly logged during certificate authentication failures.
- Updated the adapter to ensure proper validation behavior when only Bearer token authentication is configured.

Agentless Integration Kit 2.2 - April 2025

- Fixed an issue that caused null attributes to be excluded from the response of the pickup API endpoint and added the **Include Null Attributes** checkbox to the adapter configuration. Learn more in Overview of the service provider SSO flow, Overview of the identity provider SSO flow, Reference ID IdP Adapter settings reference, and Reference ID SP Adapter settings reference.
- Added the ability to preserve the JSON format of chained attribute and tracked HTTP parameter values. Learn more in Reference ID IdP Adapter settings reference.

Copyright © 2025 Ping Identity Corporation

Agentless Integration Kit 2.1 - March 2025

- Added the ability to use Bearer token authentication as an authentication method.

  Learn more in Authentication methods and refer to the **Access Token Manager** - **Required Bearer Access Token Scopes** fields in Reference ID IdP Adapter settings reference and Reference ID SP Adapter settings reference.

Agentless Integration Kit 2.0.4 – November 2021

- Fixed an issue that caused the adapter to incorrectly handle integers in JSON payloads.
- Fixed an issue that caused the adapter to incorrectly handle multi-value attributes in PingFederate 10 and later.
- Verified that the Reference ID IdP Adapter is compliant with US Federal Information Processing Standards (FIPS). Learn more in Integrating with Bouncy Castle FIPS provider in the PingFederate documentation.

Agentless Integration Kit 2.0.3 – June 2021

- Fixed an issue with client certificate authentication handling. Learn more in security bulletin SECADV025.

Agentless Integration Kit 2.0.2 – May 2021

- Improved the description for the **Relax Pass Phrase Requirements** setting.
- Fixed an issue that caused the adapter to format multi-value attributes incorrectly.
- Fixed an issue that could cause a validation error in the **Logout Service Endpoint** field when upgrading the adapter.

Agentless Integration Kit 2.0.1 – September 2020

- Fixed an issue that prevented an error from triggering when the wrong REF ID was provided.

Agentless Integration Kit 2.0 – August 2020

- Added the **Send Request Parameters** setting to control the information sent in query parameter mode.
- Added validation to the **Pass Phrase** field to enforce strong passwords.
- Added support for the JavaScript Widget for the PingFederate Authentication API.
- Added support for the PingFederate authentication API.
- Improved support for applications that do not maintain sessions. In the adapter configuration, the **Logout Mode** setting includes a **None** option.
- Improved debug log entries to show the adapter instance ID more clearly.
- Improved info log entries to show invalid reference ID errors.
- Created new Java, .NET Core, PHP, and Python sample applications and example code. Learn more in the Ping Identity GitHub repository.

Agentless Integration Kit 1.5.4 – March 2020

- Added support for header names with dashes at the attribute pick up and dropoff endpoints.

Copyright © 2025 Ping Identity Corporation

- Fixed an issue that could cause a PingFederate engine node to stop responding on configuration reload.
- Fixed an issue that caused multi-valued attributes to be treated as a single string.
- Fixed an issue that caused the drop-off endpoint to return HTML when an error occurred.

Agentless Integration Kit 1.5.3 – January 2020

- Fixed an issue that prevented the adapter from correctly linking federated users to application users.
- Removed `date` from the adapter response.
- Improved error messages.
- Improved default values for some adapter configuration fields.

Agentless Integration Kit 1.5.2 – July 2019

- Improved the query parameter transport mode. It now passes query parameters to the authentication endpoint.

Agentless Integration Kit 1.5.1 – February 2019

- Included all-new sample applications in the distribution file. You can find the latest source code in PingFederate Agentless Integration Kit Sample on the PingIdentity GitHub page.
- Fixed a bug that caused an invalid content-type header value in some API responses.

Agentless Integration Kit 1.5 – December 2018

- Added the ability to direct authorization consent requests to an external application. Learn more in External consent approval.
- Added support for virtual host names in the Reference ID Adapter. Learn more in the Configuring virtual host names section of the PingFederate documentation.
- Fixed a cross-site scripting (XSS) issue. Learn more in security bulletin SECBL013.

Agentless Integration Kit 1.4.1 – November 2018

- Improved the way Reference ID Adapter instance configurations are applied. Configuration changes now take effect without the need to restart PingFederate.

Agentless Integration Kit 1.4 – September 2018

- Reference ID IdP Adapter
    - Added the capability to prefix attribute keys with source information.
    - Added support for tracked parameters, a feature in the upcoming release ofPingFederate Server 9.2.
    - Added the capability to ignore untracked HTTP parameters.
    - Added support for JSON object based claims inside signed JWT request object.

Copyright © 2025 Ping Identity Corporation

- Reference ID IdP and SP Adapter
    - Improved logging for back-channel authentication failures.
    - Improved support for authentication method based on configured values.

Agentless Integration Kit 1.3.3 – August 2018

- Fixed issue with HTTP request parameters overriding signed JWT request parameter values.
- Target resource validation and error resource validation are now enabled in the sample configuration archive.

Agentless Integration Kit 1.3.2 – April 2018

- Fixed issue processing JSON-based request object claims.
- Fixed incorrect Content-Type header in Reference ID Adapter endpoint responses.
- Fixed data encoding issue with Form POST-based front-channel communication.

Agentless Integration Kit 1.3.1 – September 2017

- Fixed issue with exposing attributes to the IdP authentication endpoint.

Agentless Integration Kit 1.3 – August 2017

- Compatibility updates for JDK 8.
- Added support for exposing attributes to the IdP authentication endpoint.
- Added support for multi-value attributes.

Agentless Integration Kit 1.2.1 – January 2016

- Updated sample application data archive

Agentless Integration Kit 1.2 – September 2012

- Added two Transport Mode options (Query Parameter or Form Post) for Front Channel Communication for both the IdP and SP Reference ID Adapter.
- Added an authentication endpoint for the SP Adapter, which enables end-user redirection for authentication.
- Added Sample IdP/SP applications to the distribution. Sample applications provide a way to test an end-to-end identity provider (IdP) and service provider (SP) integration with PingFederate using the Agentless Integration Kit.
- Added the Certificate DN Compare Fix in the Agentless Integration Kit.

Agentless Integration Kit 1.1 – November 2011

- Support for multiple client certificates in a single Reference ID Adapter instance for client-certificate authentication.
- Allowed Subject DN and Allowed Issuer DN can be configured independently of each other.
- Basic authentication no longer required when using client-certificate authentication.

Copyright © 2025 Ping Identity Corporation

- HTTP Header ping.instanceId is optional when a single Reference ID Adapter instance is configured.
- (Bug Fix) Unfound reference attribute association no longer returns Null Pointer Exception.
- (Bug Fix) Account Linking with Reference ID Adapter corrected.

Agentless Integration Kit 1.0 – October 2010

- Initial Release.

Known issues and limitations

The following are known issues or limitations for the Agentless Integration Kit.

Known issues

- Using the PingFederate admin API to create or import a Reference ID Adapter instance using the adapter descriptor from Agentless Integration Kit 1.5.3 or earlier can cause an invalid single logout (SLO) configuration when you deploy a later version of the Agentless Integration Kit.

Known limitations

There are no known limitations.

Download manifest

The following files are included in the Agentless Integration Kit `.zip` archive:

- `Legal.pdf` : Copyright and license information.
- `config/data.zip` : A PingFederate configuration archive to use with the sample applications.
- `dist/pingfederate/server/default` : Contains the integration files.
    - `lib/pf-authn-api-sdk-<version>.jar` : A JAR file that contains the PingFederate Authentication API SDK.
    - `deploy` : Contains the Java libraries.
        - `pf-referenceid-adapter-<version>.jar` : `.jar` file that contains the Reference ID IdP Adapter and Reference ID SP Adapter.
    - `conf` : Contains the HTML template that presents the application sign-on form.
        - `language-packs` : Contains files with customizable user-facing messages.
            - `authn-api-messages.properties` : A version of the common PingFederate authentication API variable file that includes adapter-specific messages.
- `sample` : Contains files needed to set up a working demonstration of this integration kit.
    - `AgentlessIdPSample.war` : The compiled Agentless identity provider Java sample application.
    - `AgentlessSPSample.war` : The compiled Agentless service provider Java sample application.

Copyright © 2025 Ping Identity Corporation

> **Note**
>
> You can find the latest Java, .NET Core, PHP, and Python sample applications in the Ping Identity GitHub repository.

Copyright © 2025 Ping Identity Corporation

Amazon integrations

Amazon Cloud Identity Connector (Amazon Login IK)

The Amazon Cloud Identity Connector allows PingFederate to use Amazon as an identity provider (IdP). This allows users to access service provider (SP) applications by signing into Amazon.

Amazon Web Services Connector

The Amazon Web Services (AWS) Connector allows PingFederate to integrate with Amazon's AWS Identity and Access Management service for provisioning and single sign-on (SSO).

AWS Single Sign-On Connector (AWS IAM Identity Center Provisioner)

The AWS Single Sign-On Connector allows PingFederate to integrate with Amazon's AWS Single Sign-On service for provisioning and SSO.

Amazon Web Services session tags integrations for PingFederate and PingOne

AWS session tags allow enterprises to use Ping Identity products for attribute-based access control (ABAC) in AWS. Includes instructions for PingFederate and PingOne for Customers using SAML or OIDC.

Amazon Login Integration Kit

The Amazon Login Integration Kit allows PingFederate to use Amazon as an identity provider (IdP). This allows users to access service provider (SP) applications by signing into Amazon.

Features

- Supports the PingFederate PingFederate authentication API.
- Supports the JavaScript Widget for the PingFederate Authentication API.

Components

Amazon IdP Adapter

- Allows PingFederate to communicate with the Amazon API to process sign-on requests and get user information.

Templates

- Allows the adapter to prompt the user to sign on. The template can be presented with a browser redirect or as a pop-up window.
- Allows you to modify the appearance of the sign-on prompt.

Copyright © 2025 Ping Identity Corporation

Language packs

- Allows you to customize or localize the messages returned by the PingFederate authentication API and shown on the templates during authentication. For help, see Localizing messages for end users in the PingFederate documentation.

Intended audience

This document is intended for PingFederate administrators.

If you need help during the setup process, see the following sections of the PingFederate documentation:

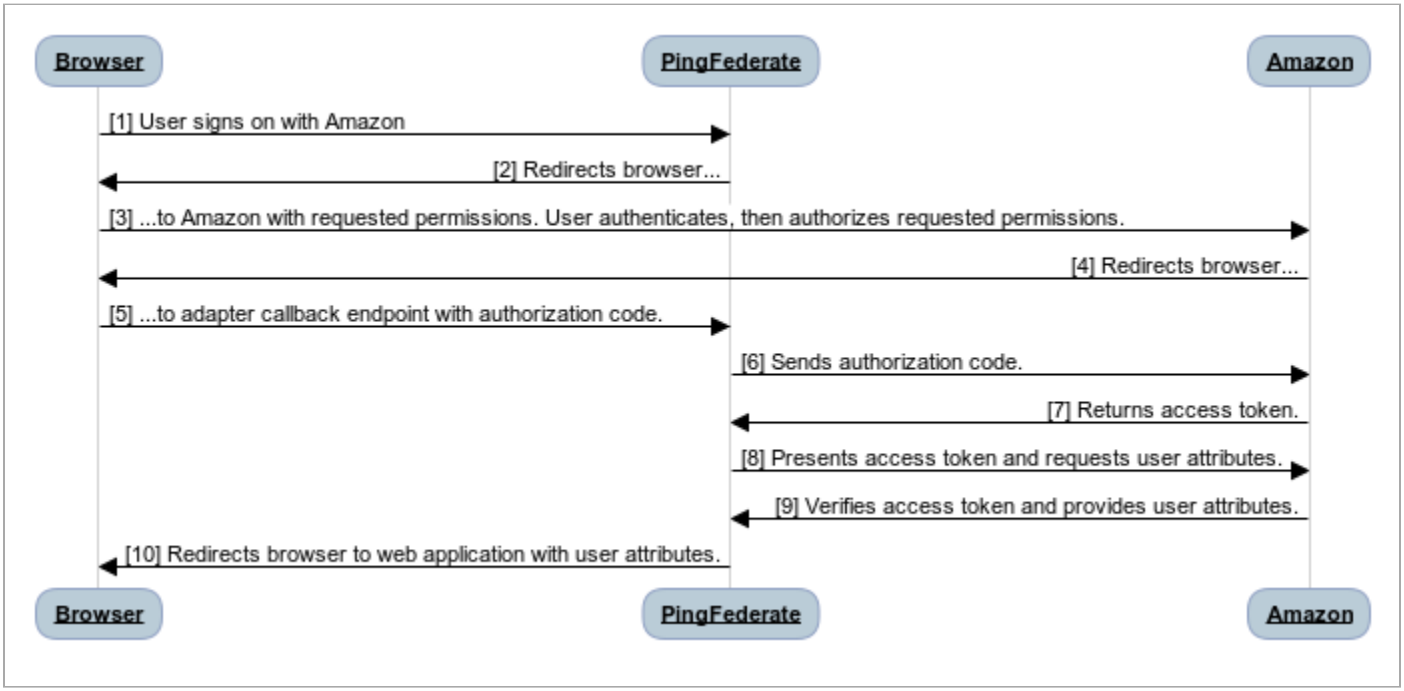
System requirements

- PingFederate 9.0 or later, or 9.3 to use the authentication API
- An Amazon developer account

Overview of the SSO flow

With the Amazon Cloud Identity Connector, PingFederate includes a Amazon authentication server in the sign-on flow.

The following diagram shows a service provider (SP)-initiated single sign-on (SSO) scenario in which PingFederate authenticates users to an SP application using the Amazon IdP Adapter.



Amazon Cloud Identity Connector - Overview of the SSO flow

Processing Steps

1. The user opens a web application and selects the Amazon sign-on option.
2. The sign-on link points to the PingFederate Amazon IdP Adapter, which redirects the browser...

Copyright © 2025 Ping Identity Corporation

3. ...to Amazon with a list of requested permissions and the authorization callback endpoint. On Amazon, the user authenticates their identity and then authorizes the requested permissions.
4. Amazon redirects the browser...
5. ...to the PingFederate Amazon IdP Adapter authorization callback endpoint with an authorization code.

   If the user fails to authenticate or does not authorize the request, the response includes an error code instead.
6. PingFederate sends Amazon the client ID, client secret, authorization code, and the PingFederate authorization callback URL.
7. Amazon returns an access token.
8. PingFederate sends Amazon a request for user attributes and presents the access token.
9. Amazon verifies the access token and provides the user information.
10. PingFederate redirects the user to the web application with the user attributes.

Authentication API Support

You can use the PingFederate authentication API to integrate the Amazon IdP Adapter into your application.

The PingFederate Authentication API provides access to the current state of the authentication flow as a user steps through the PingFederate authentication policy. For more information, see PingFederate authentication API in the PingFederate documentation. To integrate the Amazon IdP Adapter into your authentication flow, configure your application based on the states, actions, and models available in the PingFederate Authentication API Explorer. For help, see Exploring the Authentication API in the PingFederate documentation.

Setup

Deploying the integration files

To get started with the integration, deploy the Amazon Login Integration Kit files to your PingFederate directory.

About this task

> **Note**
>
> If you operate PingFederate in a cluster, the following steps refer to the console node.

Steps

1. Download the Amazon Login Integration Kit `.zip` archive from the Ping Identity Integration Directory.
2. Stop PingFederate.

Copyright © 2025 Ping Identity Corporation

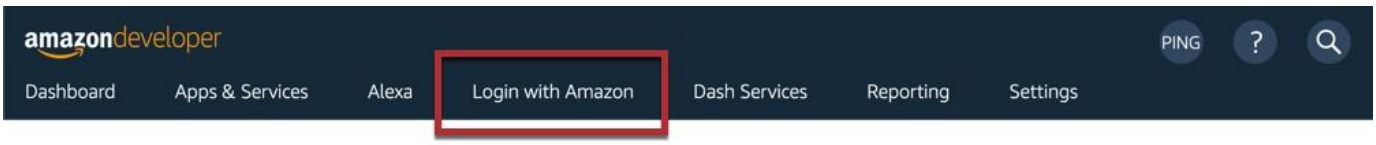
3. If you're upgrading an existing deployment, back up your customizations and delete earlier versions of the integration files:
    1. Back up any Amazon Login Integration Kit files that you customized in the `<pf_install>/pingfederate/server/default/conf/` directory.
    2. Delete the `pf-amazon-idp-adapter-<version>.jar` file from your `<pf_install>/pingfederate/server/default/deploy` directory.
4. Extract the `.zip` archive and merge the contents of the `dist` directory with your `<pf_install>/pingfederate` directory.
5. If there is more than one version of the `pf-authn-api-sdk-<version>.jar` file in your `<pf_install>/pingfederate/server/default/lib` directory, delete all but the latest version of the file.
6. If you backed up any customized files, modify the new files with your customizations.
7. Start PingFederate.
8. If you operate PingFederate in a cluster, repeat steps 2-8 for each engine node.

Registering PingFederate as a security profile in Amazon

To allow PingFederate to process social sign-on requests with Amazon, add PingFederate as an OAuth application in the Amazon Developer Console.

Steps

1. Sign on to the Amazon Developer Console.
2. On the navigation menu, click **Login with Amazon**.



3. Click **Create a New Security Profile**.
4. On the **Name your new Security Profile** screen:
    1. In the **Security Profile Name** field, enter a name, such as `PingFederate`.
    2. In the **Security Profile Description** field, enter a description, such as `For PingFederate Social Login with Amazon`.
    3. In the **Consent Privacy Notice URL** field, enter a URL that contains a privacy notice, such as `http://www.example.com/privacy.html`, then click **Save**.
5. On the **Login with Amazon** screen, locate the security profile that you just created, hover over the **Settings** icon, then click **Web Settings**.
6. On the **Security Profile Management** screen, click **Edit**:
    1. In the **Allowed Origins** field, enter the hostname or virtual hostname and port of your PingFederate server.

Copyright © 2025 Ping Identity Corporation

2. In the **Allowed Return URLs** field, enter the hostname or virtual hostname and port of your PingFederate server followed by the Amazon adapter instance endpoint.

   > **Note**
   >
   > The default endpoint is `/amazon-authn`. If you set a custom endpoint here, enter the matching value in the **Authorization Callback Endpoint** field of your adapter instance configuration when you complete the steps in Configuring an adapter instance.

3. Click **Save**.

7. Click **Show Secret** and note the Client ID and Client Secret to use in Configuring an adapter instance.

Configuring an adapter instance

Configure the Amazon IdP Adapter to determine how PingFederate communicates with the Amazon authentication service.

Steps

1. Sign on to the PingFederate administrative console.
2. On the **Identity Provider > Adapters** screen, click **Create New Instance**.
3. On the **Type** screen, set the basic adapter instance attributes:
   1. In the **Instance Name** field, enter a name for the adapter instance.
   2. In the **Instance ID** field, enter a unique identifier for the adapter instance.
   3. From the **Type** list, select **Amazon IdP Adapter 1.0**. Click **Next**.
4. On the **IdP Adapter** screen, configure the adapter instance. For more information, see the Amazon IdP Adapter settings reference. Click **Next**.
5. On the **Actions** screen, click **Test Connection**. Resolve any issues and then click **Next**.
6. On the **Extended Contract** screen, add any non-core attributes that you want to include in the contract. Click **Next**.

   For a list of core contract attributes, see Available user attributes.
7. On the **Adapter Attributes** screen, set pseudonym and masking options as shown in Set pseudonym and masking options in the PingFederate documentation. Click **Next**.
8. On the **Adapter Contract Mapping** screen, configure the contract fulfillment details for the adapter as shown in Define the IdP adapter contract in the PingFederate documentation. Click **Next**.
9. On the **Summary** screen, verify your configuration. Click **Done**.
10. On the **Manage IdP Adapter Instances** screen, click **Save**.

Amazon IdP Adapter settings reference

Field descriptions for the Amazon IdP Adapter configuration screen.

Copyright © 2025 Ping Identity Corporation

Standard fields

Field	Description
Client ID	The client ID that you noted in Registering PingFederate as a security profile in Amazon.
Client Secret	The client secret that you noted in Registering PingFederate as a security profile in Amazon.
Error Redirect URL	When an error occurs in the adapter, PingFederate redirects the browser to this URL instead of the default error page. This field is blank by default.
Unauthorized Redirect URL	When a user denies the requested Amazon permissions, PingFederate redirects the browser to this URL instead of the default error page. This field is blank by default.
Authorization Callback Endpoint	The PingFederate endpoint that Amazon uses to respond to authorization requests. If you set a custom endpoint in the **Authorization callback URL** field in Registering PingFederate as a security profile in Amazon, change this field to match. This default value is `/amazon-authn`.

Advanced fields

Field	Description
Amazon Authorization URL	The URL that PingFederate uses to send authentication requests to Amazon. If Amazon changes this endpoint, enter the new URL. The default value is `https://amazon.com/ap/oa`.
Amazon Access Token URL	The URL that PingFederate uses to retrieve access tokens from Amazon. If Amazon changes this endpoint, enter the new URL. The default value is `https://amazon.com/auth/02/token`.
Amazon User Info URL	The URL that PingFederate uses to retrieve user data from Amazon. The default value is `https://api.amazon.com/user/profile`.

Copyright © 2025 Ping Identity Corporation

Field	Description
Scopes	The scopes that you want to request from Amazon. Separate scopes with a space. The default value is `profile`.
Amazon Sign-On Presentation	Determines how the adapter presents the Amazon sign-on form. ***Redirect (default)*** The adapter redirects the browser to the Amazon sign-on form. ***Pop-up window*** The adapter opens a new window with the Amazon sign-on form on a PingFederate template. Use this option if automatic redirects are blocked by your users' browsers. This setting has no effect when using the adapter through the PingFederate authentication API.
Amazon Pop-Up Template	The template file that presents the Amazon sign-on form. Applies only when **Amazon Login Presentation** is set to **Pop-up window**. The default value is `amazon-pop-up-template.html`.
Amazon Post-Auth Template	The template file that the adapter presents after the user signs on. Applies only when **Amazon Login Presentation** is set to **Pop-up window**. The default value is `amazon-post-auth-template.html`.
Amazon Messages File	The language-pack file associated with the Amazon pop-up template. The default value is `pingfederate-amazon-adapter-messages`.
Retry Request	Determines whether PingFederate will retry requests after it receives a response with a failure code. This check box is cleared by default.
Maximum Retries Limit	Determines how many times PingFederate retries a request. The default value is `5`.
Retry Error Codes	A list of response codes that should trigger a retry. Separate response codes with a comma. The default value is `403`.

Copyright © 2025 Ping Identity Corporation

Field	Description
API Request Timeout	The amount of time in milliseconds that PingFederate waits for Amazon to respond to requests. A value of `0` disables the timeout. The default value is `5000`.
Connection Timeout	The amount of time in milliseconds that PingFederate allows to establish a connection with Amazon. A value of 0 disables the timeout. The default value is `5000`.
Proxy Settings	Defines proxy settings for outbound HTTP requests. The default value is **System Defaults**.
Custom Proxy Host	The proxy server host name to use when **Proxy Settings** is set to **Custom**. This field is blank by default.
Custom Proxy Port	The proxy server port to use when **Proxy Settings** is set to **Custom**. This field is blank by default.

Available user attributes

Lists the attributes supported by Amazon.

Attribute	Description
access_token	The access token that Amazon provides to your application.
email	The user's email address.
name	The user's name.
postal_code	The user's shipping postal code for the Amazon account.
user_id	The user's Amazon ID.

Integrating social sign-on into your application

To complete your Amazon sign-on integration, add a sign-on hyperlink to your application.

Copyright © 2025 Ping Identity Corporation

Steps

1. Create an SP connection to your application as shown in SP connection management in the PingFederate documentation. Use your IdP adapter instance and Mapping an adapter instance in the PingFederate documentation.
    - Use the Amazon IdP Adapter instance as an authentication source. For help, see Mapping an adapter instance in the PingFederate documentation.
2. If your application is outside the PingFederate domain, configure a service provider (SP) connection.
    1. Create an SP connection that uses your IdP adapter instance as shown in SP connection management and Mapping an adapter instance in the PingFederate documentation.
    2. In your web application, create a hyperlink to allow users to sign on to the SP application. Use the following URL and replace the variables based on the descriptions in the following table.

```
https://<pf_host>:<pf_port>/idp/startSSO.ping?PartnerSpId=<ConnectionId>
```

Variable	Description
<pf_host>	The host name or IP address of the PingFederate server.
<pf_port>	The port number for PingFederate.
<ConnectionId>	The federation identifier of the SP for the connection that uses the GitHub IdP Adapter instance.

3. If your application is inside the PingFederate domain, configure an adapter-to-adapter mapping.
    1. On the **System > Protocol Settings > Roles & Protocols** screen, select **Enable Identity Provider (IdP) role and support for the following** and **Enable Service Provider (SP) role and support for the following**.
    2. In both the **Enable Identity Provider** and **Enable Service Provider** sections, select any protocol, such as **SAML 2.0**. Click **Save**.

> **Note**
>
> PingFederate requires a protocol selection to activate the roles. The protocol that you select is not used for this integration.

    3. On the **Service Provider > Adapters** screen, create or select an adapter instance that is integrated with the application as shown in SP application integration settings in the PingFederate documentation.
    4. On the **Identity Provider > Adapter-to-Adapter Mappings** screen, configure the IdP-to-SP adapter mapping as shown in Adapter-to-adapter mappings in the PingFederate documentation.
    5. In your web application, create a hyperlink to allow users to sign on to the SP application. Use the following URL and replace the variables based on the descriptions in the table below:

Copyright © 2025 Ping Identity Corporation

`https://<pf_host>:<pf_port>/pf/adapter2adapter.ping?IdpAdapterId=<IdpAdapterId>&SpSessionAuthnAdapterId=<SpAdapterId>`

Variable	Description
<pf_host>	The host name or IP address of the PingFederate server.
<pf_port>	The port number for PingFederate.
<IdpAdapterId>	The instance ID of the GitHub IdP Adapter instance.
<SpAdapterId>	The instance ID of the SP adapter instance that has been integrated with the application.

Troubleshooting

Enabling debug logging

To help with troubleshooting or monitoring, you can turn on activity logging for the adapter.

About this task

These steps are optional. For general information about logging, see Enabling debug messages and console logging in the PingFederate documentation.

Steps

1. Open the `<pf_install>/pingfederate/server/default/conf/log4j2.xml` file for editing.
2. If you want to log activity for PingFederate and all adapters, do the following.
    1. Find the following section.

```
<AsyncRoot level="INFO" includeLocation="false">
        <!-- <AppenderRef ref="CONSOLE" /> -->
        <AppenderRef ref="FILE" />
</AsyncRoot>
```

    2. Change `INFO` to `DEBUG`.

```
<AsyncRoot level="DEBUG" includeLocation="false">
        <!-- <AppenderRef ref="CONSOLE" /> -->
        <AppenderRef ref="FILE" />
</AsyncRoot>
```

    3. If you want to see the adapter activity in the console, remove the comment tags.

Copyright © 2025 Ping Identity Corporation

```
<AsyncRoot level="INFO" includeLocation="false">
        <AppenderRef ref="CONSOLE" />
        <AppenderRef ref="FILE" />
</AsyncRoot>
```

3. If you want to log activity just for the Amazon IdP Adapter, add the following line.

```
<Logger name="com.pingidentity.adapter.idp.amazon" level="DEBUG"/>
```

4. Save the file.

Release notes

Changelog

The following is the change history for the Amazon Login Integration Kit.

Amazon Cloud Identity Connector 1.1 – November 2020

- Added a setting to use browser redirect or pop-up window for the sign-on presentation.
- Added customizable sign-on templates for the pop-up window presentation.
- Added customizable user-facing language-pack messages.
- Added support for the PingFederate authentication API.
- Added support for the JavaScript Widget for the PingFederate Authentication API.

Amazon Cloud Identity Connector 1.0 – October 2019

- Initial release.
- Added support for social login using Amazon credentials.
- Added support for retrieving Amazon user information.
- Added settings for retrying failed request and maximum number of retries.
- Added an adapter setting to override system-default proxy settings.
- Added API connection timeout settings.
- Added Amazon API endpoints override settings.
- Added custom Error Redirect and Unauthorized Redirect URL settings.

Copyright © 2025 Ping Identity Corporation

Known issues and limitations

The following are known issues or limitations for the Amazon Cloud Identity Connector.

Known issues

There are no known issues.

Known limitations

There are no known limitations.

Download manifest

The following files are included in the Amazon Login Integration Kit `.zip` archive:

- `ReadMeFirst.pdf` – contains links to this online documentation
- `legal/Legal.pdf` – copyright and license information.
- `dist` – contains the integration files
  - `deploy` – contains the Java libraries
    - `pf-amazon-idp-adapter-<version>.jar` – JAR file that contains the Amazon IdP Adapter.
  - `conf` – contains the HTML template that presents the Amazon sign-on form.
    - `language-packs` – contains files with customizable user-facing messages
      - `authn-api-messages.properties` – a version of the common PingFederate authentication API variable file that includes adapter-specific messages
      - `pingfederate-amazon-adapter-messages.properties` – a variable file that customizes the messages that appear on the template files.
    - `template` – contains user-facing HTML template files
      - `amazon-pop-up-template.html` – a template that opens a pop-up window to prompt the user to sign on.
      - `amazon-post-auth-template.html` – a template that returns the user to the PingFederate sign-on flow after they sign on with Amazon.
      - `assets` – contains functional scripts and files used by the template
        - `css/amazon.css` – a CSS file that customizes the appearance of the template files.
        - `fonts/end-user` – contains template fonts and icons
          - `icons` – contains template icons
          - `proxima-nova` – contains template fonts

Copyright © 2025 Ping Identity Corporation

        - `images` – contains template image files
            - `ping-logo.svg` – an image file with company branding
    - `lib/pf-authn-api-sdk-<version>.jar` – a JAR file that contains the PingFederate Authentication API SDK

AWS IAM Identity Center Provisioner

The AWS IAM Identity Center Provisioner allows PingFederate to integrate with Amazon's AWS IAM Identity Center service for provisioning and single sign-on (SSO).

Features

- Manages users in AWS IAM Identity Center based on changes in a datastore that is attached to PingFederate.
    - Creates, updates, disables, and deletes users.
    - Allows you to enable the create, update, disable, and delete capabilities independently.
    - Allows you to provision disabled users.
    - Allows you to choose whether to disable or delete users when deprovisioning.
- Manages groups in AWS IAM Identity Center based on changes in an external data store that is attached to PingFederate.
    - Creates and deletes groups.
    - Updates group memberships.
- Enables browser-based SSO initiated by the service provider (SP) or identity provider (IdP).
- Pre-populates some connection settings with the included quick connection template.

Intended audience

This document is intended for PingFederate administrators.

If you need help during the setup process, see the following resources:

- AWS IAM Identity Center documentation:
    - What Is AWS Single Sign-On?
    - Understanding Key AWS Single Sign-On Concepts
    - Users, Groups, and Provisioning
    - SCIM Profile and SAML 2.0 Implementation
    - Automatic Provisioning
- PingFederate documentation:
    - SP connection management

Copyright © 2025 Ping Identity Corporation

- Identity provider SSO configuration
- Managing digital signing certificates and decryption keys
- Datastores
- Configuring outbound provisioning
- Configuring outbound provisioning settings

System requirements

- PingFederate 9.0 or later.
- An AWS IAM Identity Center administrator account.
- To allow PingFederate to make outbound connections to the AWS IAM Identity Center API, you might need to allow the following domain in your firewall:
  - https://aws.amazon.com

User and group management

The AWS IAM Identity Center Provisioner synchronizes users and groups from your datastore to AWS IAM Identity Center. The following describes the behavior of each provisioning capability.

> **Tip**
>
> You can configure the following capabilities and specify which users to provision when you get to the Creating a connection part of the setup process.

Synchronizing existing users

PingFederate synchronizes users based on the `userName` attribute in AWS IAM Identity Center. If a user already exists in your datastore and AWS IAM Identity Center, mapping this attribute correctly links the two records together.

For example:

- In AWS IAM Identity Center, Janet's `userName` is `jsmith@example.com`.
- In your datastore, Janet's `sAMAccountName` is `jsmith@example.com`.
- On the **Attribute Mapping** tab of your provisioning connection configuration, map the `userName` attribute to `sAMAccountName`.
- When the provisioning connector runs, the datastore user is provisioned with a `userName` of `jsmith@example.com`. That matches Janet's existing `userName` in AWS IAM Identity Center, so her information in the datastore is synchronized to her AWS IAM Identity Center account.

User provisioning

PingFederate provisions users when any of the following happens:

- A user is added to the datastore group or filter that is targeted by the provisioning connector.

Copyright © 2025 Ping Identity Corporation

- A user with `disabled` status is added to the datastore group or filter that is targeted by the provisioning connector, and the **Provision disabled users** provisioning option is enabled. This feature is not available in all provisioning connector versions.

You can define which users PingFederate targets for provisioning on the **Source Location** tab of your provisioning connection configuration.

User updates

PingFederate updates users when a user attribute changes in your datastore.

You can define which attributes PingFederate monitors for changes on the **Attribute Mapping** tab of your provisioning connection configuration.

User deprovisioning

PingFederate deprovisions users when any of the following happens:

- A user is deleted from the user store.
- A user is disabled in the user store.
- A user is removed from the datastore group or filter that is targeted by the provisioning connector.

The **Remove User Action** setting in the connection configuration determines whether the deprovisioning action disables or deletes the user.

Synchronizing existing groups

PingFederate synchronizes groups from the datastore to the target service based on the group name.

For example:

- In AWS IAM Identity Center, there is a group is named `Accounting`.
- In your datastore, there is a group with a `CN` of `Accounting`.
- When the provisioning connector runs, the two groups are synchronized.

Group provisioning

PingFederate provisions groups when a group is added to the datastore filter that is targeted by the provisioning connector.

You can define which groups PingFederate targets for provisioning and monitors for changes on the **Source Location** tab in your provisioning connection configuration.

Group name updates

PingFederate renames groups when they are renamed in the datastore.

Group membership updates

PingFederate updates group memberships when memberships change in the datastore, whether the change is in the group's properties or a user's properties.

Group memberships in the datastore overwrite the group memberships in AWS IAM Identity Center.

Copyright © 2025 Ping Identity Corporation

Group deletion

PingFederate deletes groups when any of the following happens:

- The group is deleted in the datastore.
- The group is removed from the datastore group or filter that is targeted by the provisioning connector.

> **Important**
>
> Group deletions are permanent and cannot be undone.

Setup

Upgrading an existing deployment

If you are upgrading from a previous version of the AWS IAM Identity Center Provisioner, note your existing service provider (SP) connection configuration and create a new connection.

Steps

1. Back up your current PingFederate configuration as shown in Configuration archive in the PingFederate documentation.
2. In the PingFederate administrative console, open your existing SP connection.

   Choose from:

   - For PingFederate 10.1 or later: go to **Applications > Integration > SP Connections** and select your connection.
   - For PingFederate 10.0 or earlier: go to **Identity Provider > SP Connections** and select your connection.
3. Note the attribute mappings for your existing SP connection. For help, see Mapping attributes in the PingFederate documentation.
4. Delete your existing SP connection.

   Choose from:

   - For PingFederate 10.1 or later:
     1. Go to **Applications > Integration > SP Connections**.
     2. For your existing connection, click **Select action**, and then click **Delete**. Click **Confirm.**
   - For PingFederate 10.0 or earlier:
     1. Go to **Identity Provider > SP Connections > Manage All**.
     2. For your existing connection, click **Select action**, and then click **Delete**. Click **Save.**
5. Complete the steps in Deploying the integration files.

Copyright © 2025 Ping Identity Corporation

6. Complete the steps in Creating a connection.
    - From **Outbound Provisioning > Manage Channels > Channel**, on the **Attribute Mapping** tab, configure the attribute mappings based on your notes.

Deploying the integration files

To get started with the integration, deploy the AWS IAM Identity Center Provisioner files to your PingFederate directory.

Steps

1. Download the AWS IAM Identity Center Provisioner `.zip` archive from the Ping Identity Integration Directory.
2. Stop PingFederate.
3. If you're upgrading an existing deployment, delete the `pf-aws-sso-quickconnection-<version>.jar` file from your `<pf_install>/pingfederate/server/default/deploy` directory.
4. Extract the `.zip` archive and merge the contents of the `dist` directory with your `<pf_install>/pingfederate/server/default/deploy` directory.
5. Enable the PingFederate provisioning engine:
    1. Open your `<pf_install>/pingfederate/bin/run.properties` file.
    2. Change `pf.provisioner.mode` to `STANDALONE`. Save the file.

    > **Note**
    >
    > To configure the `FAILOVER` mode instead, see Deploying provisioning failover in the PingFederate Server Clustering Guide.

6. Start PingFederate.
7. If you operate PingFederate in a cluster, repeat steps 2-4 and step 6 for each engine node.

Enabling provisioning and single sign-on in PingFederate

To use PingFederate for provisioning and single sign-on, configure an external datastore and set a SAML entity ID.

About this task

Your external datastore acts as the source of data for provisioning. PingFederate also uses an internal datastore to store the state of synchronization between the source datastore and the target datastore.

For more information, see Datastores and Configuring outbound provisioning settings in the PingFederate documentation.

Enabling provisioning and single sign-on in PingFederate 10.1 or later

Steps

1. In the PingFederate administrative console, configure the datastore for PingFederate to use as the source of user data.

Copyright © 2025 Ping Identity Corporation

For help, see Datastores in the PingFederate documentation.

2. Go to **System > Server > Protocol Settings > Federation Info**.
3. In the **SAML 2.0 Entity ID** field, enter a name for PingFederate to use when identifying itself to SAML partners.
4. On the **Outbound Provisioning** tab, in the **Provisioner Data Store** list, select the internal database that will store the synchronization state. Click **Save**.

   For more information, see Configuring outbound provisioning settings in the PingFederate documentation.

Enabling provisioning and single sign-on in PingFederate 10.0 or earlier

Steps

1. In the PingFederate administrative console, configure the datastore for PingFederate to use as the source of user data.

   For help, see Datastores in the PingFederate documentation.

2. Go to **System > Protocol Settings > Roles & Protocols**.
3. Select the **Enable Identity Provider IdP Role and Support the Following** check box.
4. Select the **SAML 2.0** and **Outbound Provisioning** check boxes. Click **Next**.
5. Click the **Federation Info** tab.
6. In the **SAML 2.0 Entity ID** field, enter a name for PingFederate to use to identify itself to SAML partners.
7. On the **Outbound Provisioning** tab, in the **Provisioner Data Store** list, select the internal database that will store the synchronization state. Click **Save**.

   For more information, see Configuring outbound provisioning settings in the PingFederate documentation.

Exporting SAML metadata from PingFederate

Export a metadata file that describes your PingFederate identity provider configuration.

About this task

For general information about these steps, see Metadata export in the PingFederate documentation.

Steps

1. In the PingFederate administrative console, go to the **Metadata Export** window.

   Choose from:

   - For PingFederate 10.1 or later: go to **System > Protocol Metadata > Metadata Export**.
   - For PingFederate 10.0 or earlier: go to **System > Metadata Export**.

2. If you see the **Metadata Role** tab, select **I am the identity provider (IdP)**. Click **Next**.
3. On the **Metadata Mode** tab, select **Select information to include in metadata manually**. Click **Next**.

Copyright © 2025 Ping Identity Corporation

4. On the **Protocol** tab, click **Next**.
5. On the **Attribute Contract** tab, click **Next**.
6. On the **Signing Key** tab, select a signing certificate. Click **Next**.
7. **Optional:** On the **Metadata Signing** tab, select a certificate to sign the metadata XML file. Click **Next**.
8. On the **XML Encryption Certificate** tab, click **Next**.
9. On the **Export & Summary** tab, click **Export**.
10. Save `metadata.xml`.
11. Click **Done**.

Registering PingFederate as an identity provider in AWS Single Sign-On

To allow PingFederate to communicate with AWS IAM Identity Center, exchange the two metadata files between the two systems and note the provisioning details from AWS.

Steps

1. Sign on to the AWS SSO Console as the root user for the AWS account.
2. Go to **Settings**. In the **Identity source** section, on the **Identity source** row, click **Change**.
3. On the **"Choose where your identities are sourced** page, click **External identity provider**.
4. In the **Service provider metadata** section, click **Download metadata file**. Save the sp-saml-metadata.xml file.
5. In the **Identity provider metadata** section, upload the PingFederate metadata file that you exported in Exporting SAML metadata from PingFederate.
6. Click **Next: Review**.
7. In the **Review and confirm** section, enter `ACCEPT`. Click **Save identity source**.
8. Note your provisioning SCIM URL and access token ID.
    1. On the **Settings** window, in the **Identity source** section, on the **Provisioning** row, click **View details**.
    2. On the **Automatic Provisioning** window, note the **SCIM endpoint** and **Access token ID**. You will use these in Creating a connection.

Creating a connection

To allow PingFederate to act as an identity provider and manage users in AWS IAM Identity Center, create a service provider (SP) connection.

Steps

1. In the PingFederate administrator console, create a new SP connection:

Copyright © 2025 Ping Identity Corporation

Choose from:

- For PingFederate 10.1 or later: go to **Applications > Integration > SP Connections**. Click **Create Connection**.
- For PingFederate 10.0 or earlier: go to **Identity Provider > SP Connections**. Click **Create Connection**.

2. Configure the basic connection details with the AWS IAM Identity Center quick connection template.
    1. On the **Connection Template** tab, select **Use a template for this connection**.
    2. From the **Connection Template** list, select **AWS SSO Cloud Connector**.
    3. On the **Metadata File** row, upload the sp-saml-metadata.xml file that you saved in Registering PingFederate as an identity provider in AWS Single Sign-On. Click **Next**.
    4. On the **Connection Type** tab select **Browser SSO Profiles** and **Outbound Provisioning**. Click **Next**.
    5. On the **Connection Options** tab, click **Next**.
    6. On the **General Info** tab, in the **Connection Name** field, enter a name of your choosing. Click **Next**.
3. On the **Browser SSO** tab, configure your assertion creation settings and customize the defaults set by the metadata file.

    For help, see Configuring IdP Browser SSO in the PingFederate documentation.
4. On the **Credentials** tab, configure the connection credentials as shown in Configuring credentials in the PingFederate documentation. Click **Next**.
5. On the **Outbound Provisioning** tab, configure provisioning with the following details.

    For help, see Configuring outbound provisioning in the PingFederate documentation.
    1. On the **Target** tab, complete the **SCIM URL** and **Access Token** fields with the values that you noted in Registering PingFederate as an identity provider in AWS Single Sign-On.
    2. Under **Provisioning Options**, customize the provisioning connector behavior by referring to Provisioning options reference. Click **Next**.
    3. On the **Manage Channels > Attribute Mapping** tab, at the bottom of the attribute list, click **Refresh Fields** to get fields and specifications from your AWS IAM Identity Center site. Complete the attribute mappings by referring to Supported attributes reference.

        For help, see Managing channels in the PingFederate documentation.
6. On the **Activation and Summary** tab, above the **Summary** section, click the toggle to turn on the connection. Click **Save**.

Provisioning options reference

The AWS IAM Identity Center Provisioner allows you to enable actions independently and customize other provisioning options.

Copyright © 2025 Ping Identity Corporation

Field Name	Description
User Create	***Selected (default)*** PingFederate creates users in AWS IAM Identity Center. ***Cleared*** PingFederate does not create users in AWS IAM Identity Center.
User Update	***Selected (default)*** PingFederate updates existing users in AWS IAM Identity Center. ***Cleared*** PingFederate does not update existing users in AWS IAM Identity Center.
User Disable / Delete	***Selected (default)*** PingFederate removes users from AWS IAM Identity Center according to the **Remove User Action** setting. ***Cleared*** PingFederate does not remove users from AWS IAM Identity Center.
Provision Disabled Users	***Selected*** PingFederate creates users in AWS IAM Identity Center with a "disabled" status. ***Cleared (default)*** If a user has a "disable" status, PingFederate does not create the user in AWS IAM Identity Center.
	Note If any of the above options are cleared, PingFederate logs a warning in the user workflow section of `provisioner.log` when the related action fails.

Copyright © 2025 Ping Identity Corporation

Field Name	Description
Remove User Action	This option applies when: • **User Disable / Delete** is selected, and • a previously-provisioned user no longer meets the condition set on the **Source Location** tab, or • a user has been disabled or deleted from the data store. ***Delete*** PingFederate deletes the user from AWS IAM Identity Center. ***Disable (default)*** PingFederate disables the user in AWS IAM Identity Center.

Supported attributes reference

The following attributes can be mapped for user provisioning to AWS IAM Identity Center.

For more information about these attributes, see SCIM: Core Schema on ietf.org and Supported User and Group Attributes in the AWS IAM Identity Center documentation.

Attribute	Description
`userName`	The user's unique identifier in AWS IAM Identity Center. This attribute cannot be updated. This attribute is required.
`displayName`	The name of the user, suitable for display to end-users. This attribute is required.
`familyName`	The family name of the user, or last name in most Western languages. For example, "Jensen" given the full name "Ms. Barbara Jane Jensen, III". This attribute is required.
`givenName`	The given name of the user, or first name in most Western languages. For example, "Barbara" given the full name "Ms. Barbara Jane Jensen, III". This attribute is required.
`workEmail`	The user's work email address, formatted as "bjenson@example.com". This attribute is required.
`costCenter`	The user's cost center.
`country`	The user's country.

Copyright © 2025 Ping Identity Corporation

Attribute	Description
`department`	The user's department.
`division`	The user's division.
`employeeNumber`	A string identifier, typically numeric or alphanumeric, assigned to a person, often based on order of hire or association with an organization.
`externalId`	A string that is an identifier for the resource as defined by the provisioning client.
`formattedAddress`	The user's complete, formatted address.
`formattedName`	The user's complete, formatted name.
`locale`	Used to indicate the User's default location for purposes oflocalizing such items as currency, date time format, or numericalrepresentations. A valid value is a language tag as defined in RFC-5646.
`locality`	The city or locality component of the user's address. For example, "Hollywood".
`middleName`	The user's middle name or names. For example, "Jane" given the full name "Ms. Barbara Jane Jensen, III".
`nickName`	The user's nickname. For example, "Barb" given the full name "Ms. Barbara Jane Jensen, III".
`organization`	The user's organization.
`portalCode`	The user's zip code or postal code.
`preferredLanguage`	Indicates the user's preferred written or spoken languages and is generally used for selecting a localized user interface.
`prefixName`	The honorific prefix or prefixes of the user, or title in most Western languages. For example, "Ms." given the full name "Ms. Barbara Jane Jensen, III".
`region`	The state or region component of the user's address. For example, "CA".
`streetAddress`	The user's street address, which may include house number, street name, P.O. box, and multi-line extended street address information.
`suffixName`	The honorific suffix(es) of the user, or title in most Western languages. For example, "III" given the full name "Ms. Barbara Jane Jensen, III".
`timezone`	The user's time zone, in IANA Time Zone database format. For example, "America/Los_Angeles".
`title`	The user's title, such as "Vice President".
`userType`	The type of user. For example, "employee".

Copyright © 2025 Ping Identity Corporation

Attribute	Description
`workPhone`	The user's work phone number, formatted as "+1-201-555-0123".

Access control in AWS using session tags

Amazon Web Services (AWS) session tags allow enterprises to use Ping Identity products for attribute-based access control (ABAC) in the AWS Single Sign-On (SSO) service.

For information and configuration steps, see Amazon Web Services session tags integrations in the Ping Identity integration documentation.

Release notes

Changelog

The following is the change history for the AWS IAM Identity Center Provisioner.

AWS IAM Identity Center Provisioner 1.1 – October 2024

- Fixed an issue that prevented updates made to an existing user's `sAMAccountName` in the LDAP source from synchronizing with the nickName attribute in the AWS platform.

AWS IAM Identity Center Provisioner 1.0 – October 2020

- Initial release.
- Included support for user and group provisioning.
- Included support for AWS IAM Identity Center attributes.
- Included support for provisioning disabled users.
- Included configuration options for create, update, and disable/delete capabilities.
- Included configuration options for deprovisioning actions.

Known issues and limitations

The following are known issues or limitations with the AWS IAM Identity Center Provisioner.

Known issues

There are no known issues.

Copyright © 2025 Ping Identity Corporation

Known limitations

- Attributes
  - After an attribute is set, it can be updated but not cleared.
- Deprovisioning
  - When an LDAP user is deleted in a targeted group distinguished name (DN), the provisioning connector does not propagate the deletion until a new user is added to the group. This limitation is compounded when the **User Create** provisioning option is disabled. For solutions, see SaaS provisioner does not remove the user in the Knowledge Base.

Download manifest

The following files are included in the AWS IAM Identity Center Provisioner `.zip` archive:

- `Legal.pdf` – copyright and license information
- `dist` – contains the integration files
  - `pf-aws-sso-quickconnection-<version>.jar` – The AWS IAM Identity Center Provisioner quick connection template.

Amazon Web Services Connector

The Amazon Web Services (AWS) Connector allows PingFederate to integrate with Amazon's AWS Identity and Access Management service for provisioning and single sign-on (SSO).

Features

- Manages users in Amazon Web Services based on changes in an external data store that is attached to PingFederate.
  - Creates, updates, and deletes users.
  - Allows you to enable the create, update, and delete capabilities independently.
- Manages groups
  - Creates groups
  - Updates group memberships and names.
- Enables browser-based SSO initiated by the identity provider (IdP).
- Pre-populates some connection settings with the included quick connection template.
- Supports the following attributes: `UserName`, `Password`, and `PasswordResetRequired`.
- Pre-populates some connection settings with the included quick connection template.

Copyright © 2025 Ping Identity Corporation

Intended audience

This document is intended for PingFederate administrators.

Before you start, you should be familiar with the following:

- The following sections of the Amazon Web Services documentation:
  - AWS Identity and Access Management (IAM)
  - IAM Users
- The following sections of the PingFederate documentation:
  - Identity provider SSO configuration
  - Datastores
  - Managing IdP adapters
  - Managing digital signing certificates and decryption keys
  - SP connection management
  - Configuring outbound provisioning
  - Configuring outbound provisioning settings

System requirements

- PingFederate 8.0 or later.
- To allow PingFederate to make outbound connections to Amazon Web Services, you might need to allow the following endpoint in your firewall:
  - https://iam.amazonaws.com

Setup

Upgrading an existing deployment

If you're upgrading from a previous version of the Amazon Web Services Connector, note your existing service provider (SP) connection configuration and create a new connection.

Upgrading an existing deployment in PingFederate 10.1 or later

Steps

1. Back up your current PingFederate configuration as shown in Configuration archive in the PingFederate documentation.

Copyright © 2025 Ping Identity Corporation

2. In the PingFederate administrative console, go to **Applications > Integration > SP Connections** and select your connection.
3. Note the attribute mappings for your existing SP connection. For help, see Mapping attributes in the PingFederate documentation.
4. Delete your existing SP connection.
    1. Go to **Applications > Integration > SP Connections**.
    2. For your existing connection, click **Select action**, and then click **Delete**. Click **Confirm.**
5. Complete the steps in Deploying the integration files.
6. Complete the steps in Creating a provisioning connection.
    - From **Outbound Provisioning > Manage Channels > Channel**, on the **Attribute Mapping** tab, configure the attribute mappings based on your notes.
7. **Optional**: Complete the steps in Creating a single sign-on connection.

Upgrading an existing deployment in PingFederate 10.0 or earlier

Steps

1. Back up your current PingFederate configuration as shown in Configuration archive in the PingFederate documentation.
2. In the PingFederate administrative console, go to **Identity Provider > SP Connections** and select your connection.
3. Note the attribute mappings for your existing SP connection. For help, see Mapping attributes in the PingFederate documentation.
4. Delete your existing SP connection.
    1. Go to **Identity Provider > SP Connections > Manage All**.
    2. For your existing connection, click **Select action**, and then click **Delete**. Click **Save.**
5. Complete the steps in Deploying the integration files.
6. Complete the steps in Creating a provisioning connection.
    - From **Outbound Provisioning > Manage Channels > Channel**, on the **Attribute Mapping** tab, configure the attribute mappings based on your notes.
7. **Optional**: Complete the steps in Creating a single sign-on connection.

Create security credentials in Amazon Web Services

About this task

Copyright © 2025 Ping Identity Corporation

PingFederate needs credentials to manage users and groups in Amazon Web Services (AWS). You can create these credentials in the AWS management console. For more details, see Managing Access Keys for Your AWS Account Root User in the AWS documentation.

Steps

1. Sign on to the AWS Management Console as the root user for the AWS account.
2. On the navigation menu, click your account name, and then click **My Security Credentials**.
3. If you receive a warning about accessing the security credentials for your AWS account, click **Continue to Security Credentials**.
4. Expand the **Access keys (access key ID and secret access key)** section.
5. Click **Create New Access Key**.

> **Note**
>
> You might need to delete an existing access key before you can create a new key. See IAM Object Limits in the AWS documentation.

6. Note the access key ID and the access key secret.

> **Important**
>
> This is your only opportunity to view or download the secret access key.

Deploying the integration files

To get started with the integration, deploy the Amazon Web Services Connector files to your PingFederate installation directory. This must be done on the administrative node installation first, regardless of whether you use PingFederate in `STANDALONE` mode or as part of a cluster.

Steps

1. Download the Amazon Web Services Connector `.zip` archive from the Ping Identity Integration Directory.
2. Stop PingFederate.
3. If you're upgrading an existing deployment, delete the `pf-aws-quickconnection-<version>.jar` file from your `<pf_install>/pingfederate/server/default/deploy` directory.
4. Extract the `.zip` archive and copy the `pf-aws-quickconnection-<version>.jar` file from the `dist` directory to the `<pf_install>/pingfederate/server/default/deploy` directory.

> **Note**
>
> You will use the supplied `aws-saml-metadata.xml` file later in the Creating a provisioning connection procedure.

5. Enable the PingFederate provisioning engine:
    1. Open your `<pf_install>/pingfederate/bin/run.properties` file.

Copyright © 2025 Ping Identity Corporation

2. Change `pf.provisioner.mode` to `STANDALONE`. Save the file.

> **Note**
>
> To configure the `FAILOVER` mode instead, see Deploying provisioning failover in the PingFederate Server Clustering Guide.

6. Start PingFederate.

Result:

If you are using the PingFederate provisioning engine in `STANDALONE` mode on the administrative node, you have completed the deploying the integration files procedure. Otherwise, continue to step 7.

7. If you are using the PingFederate provisioning engine in any mode other than `STANDALONE` mode, including `FAILOVER` mode in a clustered environment, repeat steps 2-4 and step 6 for each engine node.

Creating a provisioning connection

To allow PingFederate to manage users in Amazon Web Services (AWS), create a service provider (SP) connection.

About this task

> **Note**
>
> You can follow these steps to create a new SP connection, or you can modify an existing connection.

Steps

1. In the PingFederate administrator console, configure the data store that PingFederate will use as the source of user data. For instructions, see Datastores in the PingFederate documentation.
    - When targeting users and groups for provisioning, exclude the user account that you will use to administer users in your connection to AWS. This prevents the PingFederate provisioning engine from interfering with the account that provisions users and groups.
2. Enable provisioning.
    1. On the **System > Protocol Settings > Roles & Protocols** screen, select **Enable Identity Provider IdP Role and Support the Following**.
    2. Select **Outbound Provisioning**. Click **Save**.
3. Download the latest SAML metadata file from Amazon. Save it as `aws-saml-metadata.xml`.
4. Create an SP connection with the Amazon Web Services quick connection template.
    1. On the **Identity Provider** screen, in the **SP Connections** area, click **Create new**.
    2. On the **Connection Template** screen, select **Use a template for this connection**.
    3. In the **Connection Template** list, select **Amazon Web Services Connector**.
    4. Click **Choose File**, select the `aws-saml-metadata.xml` that you downloaded, and then click **Open**. Click **Next**.

Copyright © 2025 Ping Identity Corporation

5. On the **Connection Type** screen, clear **Browser SSO Profiles** and select **Outbound Provisioning**. Click **Next**.
6. On the **General Info** screen, the basic connection information is populated by the metadata XML file. Click **Next**.
7. On the **Outbound Provisioning** screen, configure the provisioning target and channel.

   See Configuring outbound provisioning in the PingFederate *Administrator's Manual*.

   1. Click **Configure Provisioning**.
   2. On the **Target** screen, complete the **Access Key ID** and **Access Key Secret** fields with the values that you noted in Create security credentials in Amazon Web Services.

      > **Note**
      >
      > PingFederate verifies the credentials when you activate the channel and SP connection.

   3. Under **Provisioning Options**, customize the provisioning connector actions. Click **Next**.

      See Provisioning options.

   4. On the **Manage Channels** screen, create a channel. Click **Done**.

      See Managing channels in the PingFederate documentation.

      > **Tip**
      >
      > If you want to force a one-time password change for new users:
      > 1. On the **Manage Channels > Attribute Mapping** screen, on the **PasswordResetRequired** line, click **Edit**.
      > 2. On the **PasswordResetRequired** mapping screen, under **Options**, click **Create only**. Click **Save**.

   5. On the **Outbound Provisioning** screen, click **Next**.
8. On the **Activation and Summary** screen, above the **Summary** section, turn on the connection. Click **Save**.

Provisioning options

Field Name	Description
User Create	**Selected** (default) – PingFederate creates users in Amazon Web Services (AWS). **Cleared** – PingFederate does not create users in AWS.
User Update	**Selected** (default) – PingFederate updates existing users in AWS. **Cleared** – PingFederate does not update existing users in AWS.
User Delete	**Selected** (default) – PingFederate deletes users in AWS. **Cleared** – PingFederate does not delete users in AWS.

Copyright © 2025 Ping Identity Corporation

Field Name	Description

> **Note**
> If any of the above options are cleared, PingFederate logs a warning in the user workflow section of `provisioner.log` when the related action fails.

Supported attributes

The following table lists the attributes that can be mapped for user provisioning.

Attribute	Description
UserName	The unique user identifier. This attribute is required.
Password	The user password.
PasswordResetRequired	If `true`, the user must reset their password the next time they log in. If any other value, Amazon Web Services treats this as `false`. **Important** You also need to give the user permission to change their password in AWS. See Permitting IAM Users to Change Their Own Passwords in the AWS documentation.

Configuring single sign-on

To allow PingFederate to act as an identity provider for Amazon Web Services (AWS), enable single sign-on (SSO) in PingFederate and AWS, and create a connection.

About this task

> **Note**
> Single sign-on integration is an optional part of this integration. If you only want to use the Atlassian Connector for provisioning, skip these steps.

Steps

1. Complete the steps in Exporting connection-specific SAML metadata.
2. Create an identity provider profile to represent PingFederate in Amazon Web Services (AWS).

   Complete the steps in Creating and Managing an IAM SAML Identity Provider (Console) in the AWS documentation.
3. Create a federation role in Amazon Web Services (AWS) that uses PingFederate as an identity provider.

   Complete the steps in Creating a Role for SAML 2.0 Federation (Console) in the AWS documentation.

Copyright © 2025 Ping Identity Corporation

4. Complete the steps in Creating a single sign-on connection.

Exporting SAML metadata from PingFederate

Export a metadata file that describes your PingFederate identity provider configuration.

About this task

For general information about these steps, see Metadata export in the PingFederate documentation.

Steps

1. In the PingFederate administrative console, go to the **Metadata Export** window.

   Choose from:

   - For PingFederate 10.1 or later: go to **System > Protocol Metadata > Metadata Export**.
   - For PingFederate 10.0 or earlier: go to **System > Metadata Export**.

2. If you see the **Metadata Role** tab, select **I am the identity provider (IdP)**. Click **Next**.
3. On the **Metadata Mode** tab, select **Select information to include in metadata manually**. Click **Next**.
4. On the **Protocol** tab, click **Next**.
5. On the **Attribute Contract** tab, click **Next**.
6. On the **Signing Key** tab, select a signing certificate. Click **Next**.
7. **Optional:** On the **Metadata Signing** tab, select a certificate to sign the metadata XML file. Click **Next**.
8. On the **XML Encryption Certificate** tab, select the certificate that you want to use to encrypt the XML content. Click **Next**.
9. On the **Export & Summary** tab, click **Export**.
10. Save `metadata.xml`.
11. Click **Done**.

Creating an identity provider in Amazon Web Services

About this task

Create an identity provider profile to represent PingFederate in Amazon Web Services (AWS).

For detailed configuration information, see Creating and Managing an IAM SAML Identity Provider (Console) in the AWS documentation.

Steps

1. In the Amazon Web Services IAM console, on the **Identity Providers** screen, click **Create Provider**.
2. On the **Configure Provider** screen, in the **Provider Type** list, select **SAML**.
3. In the **Provider Name** field, type a name, such as `PingFederate`.

Copyright © 2025 Ping Identity Corporation

4. In the **Metadata Document** line, click **Choose File**. Select the `metadata.xml` file that you exported in Exporting SAML metadata from PingFederate, and then click **Open**. Click **Next Step**.
5. On the **Verify** screen, click **Create**.
6. On the **Identity Providers** screen, select the identity provider that you created, and note the **Provider ARN**.

Creating a federation role in Amazon Web Services

About this task

Create a federation role in Amazon Web Services (AWS) that uses PingFederate as an identity provider.

For detailed configuration information, see Creating a Role for SAML 2.0 Federation (Console) in the AWS documentation.

Steps

1. In the Amazon Web Services IAM console, on the **Roles** screen, click **Create role**.
2. On the **Create Role** screen, click **Saml 2.0 federation**.
3. In the **SAML Provider** list, select the identity provider that you created in Creating an identity provider in Amazon Web Services.
4. Select one of the following SAML 2.0 access methods, and then click **Next: Permissions**.
   - To create a role that can be assumed programmatically from the AWS API or AWS CLI, select **Allow programmatic access only**.
   - To create a role that can be assumed programmatically and from the console, select **Allow programmatic and AWS Management Console access**.
5. On the **Attach permissions policy** screen, select or create a permissions policy. Click **Next: Tagging**.

   For more information about creating policies, see Creating IAM Policies in the AWS documentation.
6. On the **Tagging** screen, click **Next: Review**.
7. On the **Review** screen, in the **Role name** field, type a unique name for the federation role. Click **Create role**.
8. On the **Roles** screen, select the role that you created, and note the **Role ARN**.

Creating a single sign-on connection

Create an SP connection in PingFederate to communicate single sign-on (SSO) information to Amazon Web Services (AWS).

About this task

> **Note**
>
> You can follow these steps to create a new SP connection, or you can modify an existing connection.

Steps

1. Download the latest SAML metadata file from Amazon. Save it as `aws-saml-metadata.xml`.

Copyright © 2025 Ping Identity Corporation

2. In the PingFederate administrator console, configure an SP connection.
    1. On the **Identity Provider** screen, in the **SP Connections** area, click **Create new**.
    2. On the **Connection Template** screen, select **Use a template for this connection**.
    3. In the **Connection Template** list, select **Amazon Web Services Connector**.
    4. Click **Choose File**, select the `aws-saml-metadata.xml` that you downloaded, and then click **Open**. Click **Next**.
3. On the **Connection Type** screen, select **Browser SSO Profiles** and clear **Outbound Provisioning**. Click **Next**.
4. On the **Connection Options** screen, click **Next**.
5. On the **General Info** screen, the basic connection information is populated by the metadata XML file. Click **Next**.
6. On the **Browser SSO** screen, configure browser SSO.

    For a complete guide, see Configuring IdP Browser SSO in the PingFederate documentation.
    1. On the **Browser SSO > Assertion Creation > IdP Adapter Mapping > Attribute Contract Fulfillment** screen, on the **SAML_SUBJECT** line, select a source.
    2. On the **https://aws.amazon.com/SAML/Attributes/Role** line, select **Text**.
    3. In the **Value** field, type the role ARN and provider ARN that you noted in Creating an identity provider in Amazon Web Services, and Creating a federation role in Amazon Web Services. Separate the ARNs with a comma, as follows:

        `<role ARN>,<provider ARN>`
    4. On the **https://aws.amazon.com/SAML/Attributes/RoleSessionName** line, select a value to use as the user's display name in AWS.
7. On the **Credentials** screen, configure the connection credentials.

    See Configuring credentials in the PingFederate documentation.
8. On the **Activation and Summary** screen, above the **Summary** section, click the toggle button to enable the connection. Click **Save**.

Troubleshooting

Show Amazon API errors in the server log

About this task

You can show error responses from the Amazon API in the PingFederate server log.

Steps

1. Open `<pf_install>/pingfederate/server/default/conf/log4j2.xml` for editing.
2. In the `<Loggers>` section, add the following lines. Save the file.

Copyright © 2025 Ping Identity Corporation

```
<Logger name="com.pingidentity.shaded.aws.apache.wire" level="DEBUG" />
<Logger name="software.amazon.awssdk" level="DEBUG" />
```

Release notes

Changelog

Amazon Web Services (AWS) Connector 2.0 – April 2019 (current release)

- Added support for the AWS 2.0 API.
- Added support for the `Password` and `PasswordResetRequired` attributes.
- Added the ability to update group names.
- Added the ability to disable the "create", "update", and "delete" capabilities individually.
- Added the ability to show debug information from the AWS API in the PingFederate server log. See Show Amazon API errors in the server log.
- Added support for proxy connections.
- Improved error-handling and reporting behavior.

Amazon Web Services Connector 1.0 – January 2014

- Initial release.
- Added support for user and group provisioning.
- Added support for browser-based single sign-on.

Known issues and limitations

Known issues

There are no known issues.

Known limitations

- Groups
  - Before deleting a group, Amazon Web Services (AWS) requires you to remove all users from the group. Otherwise, an error will occur in the AWS API.
  - AWS does not allow a user to be a member of more than 10 groups.
- Deprovisioning
  - AWS does not support disabled users. These users are deleted instead.

Copyright © 2025 Ping Identity Corporation

    - When an LDAP user is deleted in a targeted Group DN, the provisioning connector does not propagate the deletion until a new user is added to the group. This limitation is compounded when the **User Create** provisioning option is disabled. For solutions, see SaaS provisioning connector does not remove the user in the Knowledge Base.

- When a user is created with a `passwordResetRequired` value other than `true` or `TRUE`, the provisioning connector sets the value to `false` in AWS.
- Attributes
    - Due to limitations in PingFederate, user attributes cannot be cleared after they are set.
    - Due to limitations in PingFederate, the "Create only" attribute mapping option is not available for the `UserName` attribute.
- In PingFederate 8.1 and earlier, when you configure a second SP connection with the provisioning connector, the second connection may be pre-populated with the channel from the first connection. To avoid conflicts, remove this pre-populated channel and create a unique channel for each connection.

Download manifest

The following files are included in the Amazon Web Services Connector `.zip` archive:

- `legal/Legal.pdf` – copyright and license information.
- `dist` – contains the integration files
    - `pf-aws-quickconnection-<version>.jar` – The Amazon Web Services Connector quick connection template.
- `aws-saml-metadata.xml` – a modifiable metadata file used to create the connection.

Amazon Web Services Session Tags Integrations Guide

Amazon Web Services (AWS) session tags allow enterprises to use Ping Identity products for attribute-based access control (ABAC) in the AWS Identity and Access Management (IAM) and AWS IAM Identity Center services.

ABAC helps enterprises simplify fine-grained access to AWS resources by using attributes from their corporate directories in permissions rules.

When an employee federates into AWS from a standards-compliant identity provider (IdP) such as PingFederate or PingOne for Enterprise, the administrator can include attributes such as cost center, job title, and email address in the AWS session. These attributes function as session tags and can be matched to tags on AWS resources to control their access to resources during their AWS session.

For example, an employee with job title “Systems Engineer” and cost center “Stratford” could be granted write access to an Amazon EC2 table that is also tagged with cost center “Stratford.”

You can configure PingFederate or PingOne for Enterprise to send attributes in the AWS sessions when your users federate into AWS. Then, you can implement a policy in AWS that evaluates the attributes from PingFederate or PingOne for Enterprise to control access to AWS resources.

Copyright © 2025 Ping Identity Corporation

For more information, see the following:

- Attributes for Access Control in the AWS IAM Identity Center documentation.
- IAM Tutorial: Use SAML session tags for ABAC in the AWS IAM documentation.

To use AWS session tags for ABAC, complete the steps in one or more of the following topics.

Session tags in PingFederate

Configuring AWS session tags for PingFederate SAML connections

You can configure AWS Identity and Access Management (IAM) and AWS IAM Identity Center session tag support for SAML connections in PingFederate.

Before you begin

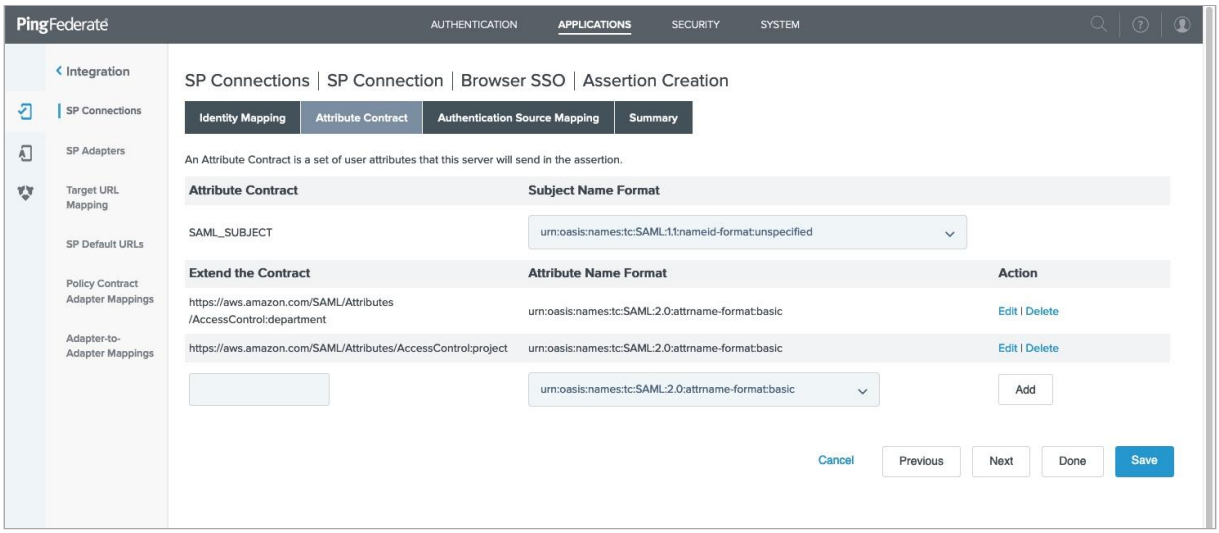
- If you want to use OGNL expressions to populate the values of the AWS session tags, see Enabling and disabling expressions in the PingFederate documentation.
- Create an Amazon Web Services (AWS) console account and policy that uses session tags. For help, see AWS prerequisites (page 241) in the PingAccess documentation.

Steps

1. Open your service provider (SP) connection. Go to **SP Connection > Browser SSO > Assertion Creation > Attribute Contract**.
2. Extend the contract of the AWS SP connection.

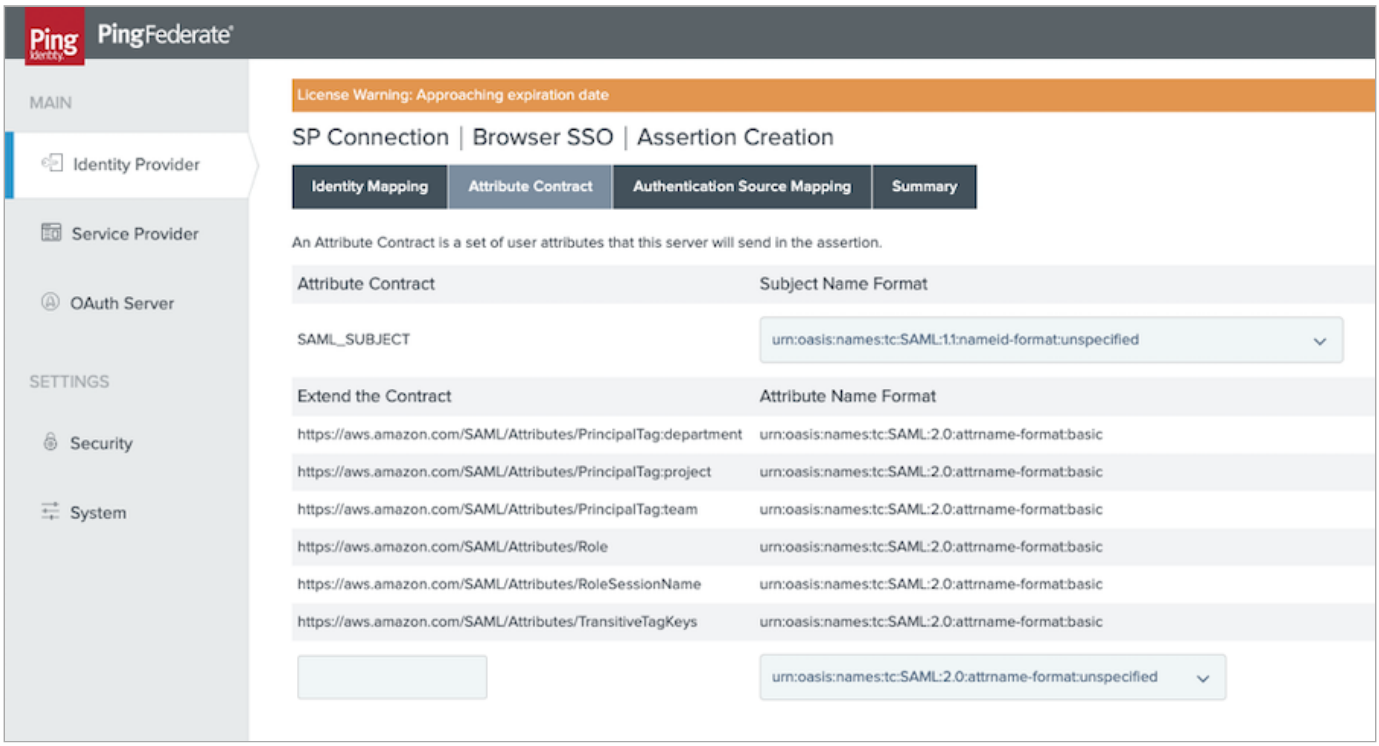
   Choose from:

   - If you are using AWS SSO, include the access control tags based on the following format:
     - `https://aws.amazon.com/SAML/Attributes/AccessControl:{attribute}`



Copyright © 2025 Ping Identity Corporation

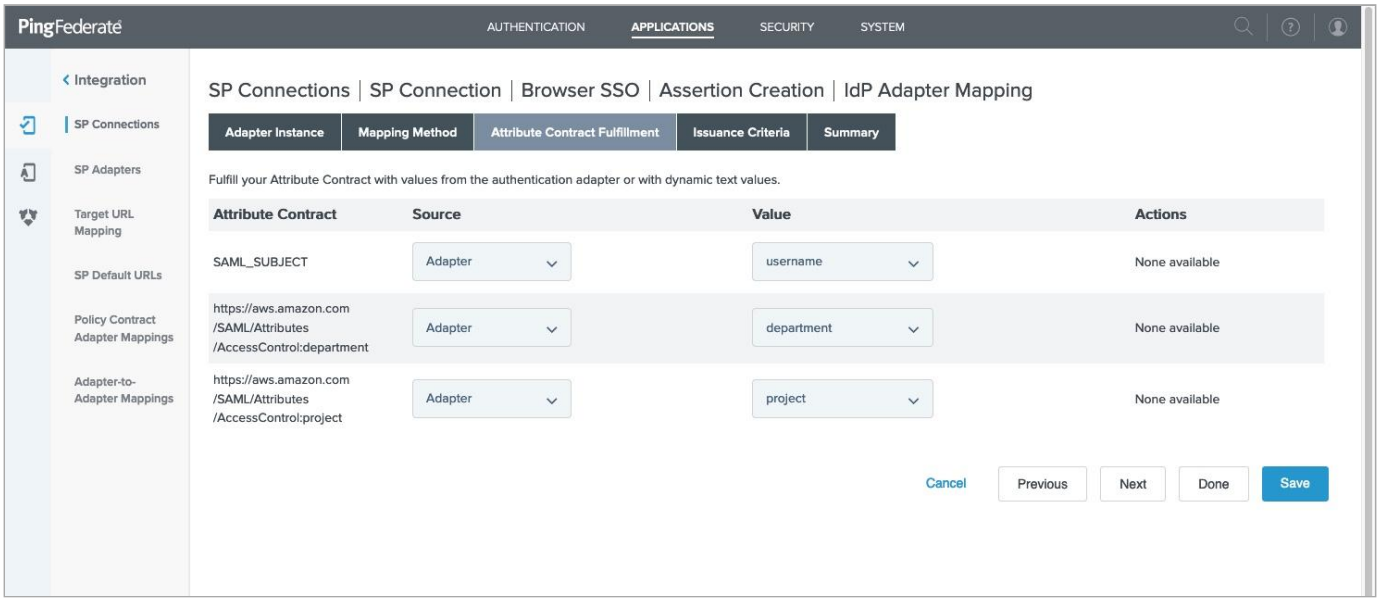
- If you are using AWS IAM, include the AWS Principal Tags and TransitiveTagKeys, based on the following examples:
  - `https://aws.amazon.com/SAML/Attributes/AccessControl:{attribute}`
  - `https://aws.amazon.com/SAML/Attributes/PrincipalTag:project`
  - `https://aws.amazon.com/SAML/Attributes/Role`
  - `https://aws.amazon.com/SAML/Attributes/RoleSessionName`
  - `https://aws.amazon.com/SAML/Attributes/TransitiveTagKeys`



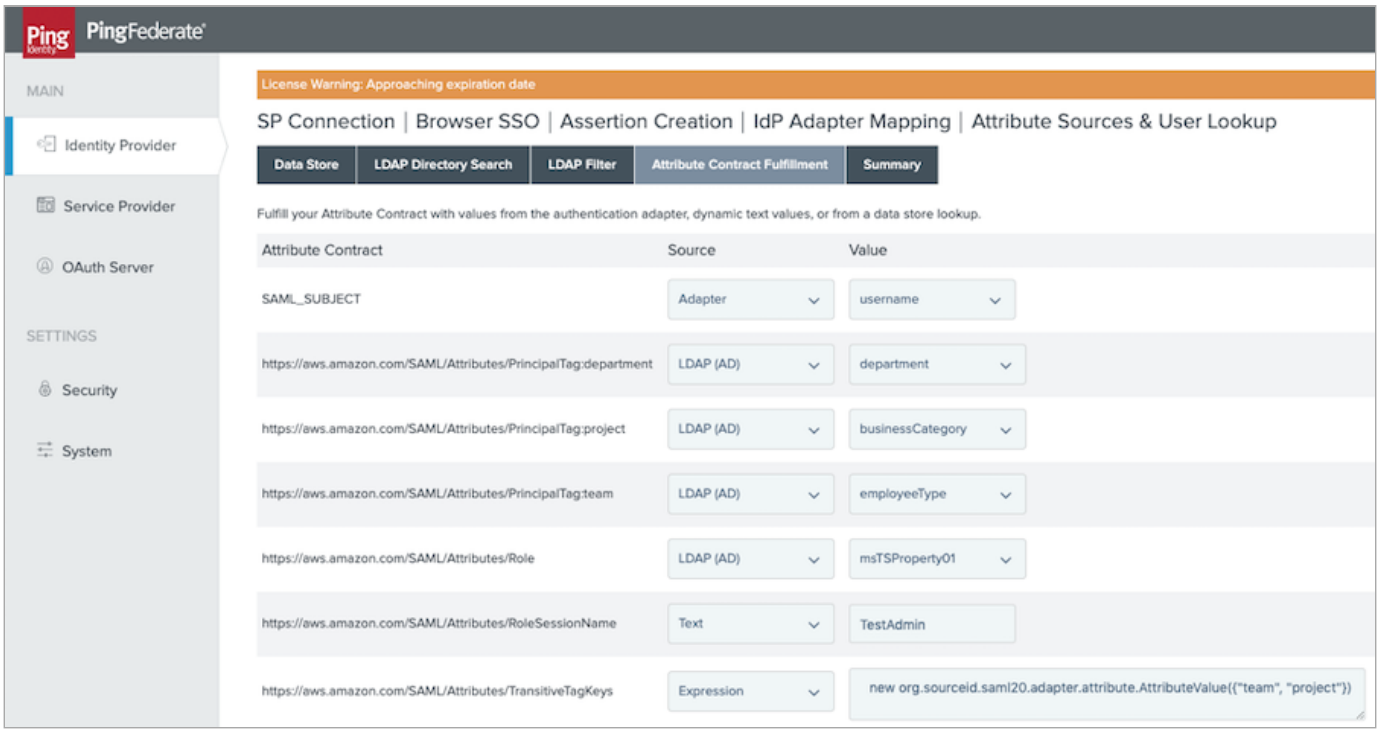
3. Go to **SP Connection > Browser SSO > Assertion Creation > IdP Adapter Mapping > Attribute Sources & User Lookup > Attribute Contract Fulfillment**.
4. Configure the attribute contract fulfillment for the AWS attributes.

   Example 1: This example shows AWS IAM Identity Center attributes mapped directly from an HTML Form Adapter instance.

Copyright © 2025 Ping Identity Corporation



Example 2: This example shows AWS IAM attributes mapped from a data source and manipulated by the OGNL expression language available in PingFederate.



5. Click **Save**.

Result

The AWS session tags are now included in the SAML assertion created by PingFederate.

Copyright © 2025 Ping Identity Corporation

Configuring AWS session tags for PingFederate OIDC connections

You can configure AWS Session Tag support for OpenID Connect (OIDC) connections in PingFederate.

Before you begin

- If you want to use OGNL expressions to populate the values of the AWS session tags, see Enabling and disabling expressions in the PingFederate documentation.
- Create an Amazon Web Services (AWS) console account and policy that uses session tags. For help, see AWS prerequisites (page 241) in the PingAccess documentation.
- Configure your PingFederate OAuth client for AWS console authentication.
- Define a PingFederate OpenID Connect policy. For help, see Configuring OpenID Connect policies in the PingFederate documentation.

Steps

1. Open your OpenID Connect policy.

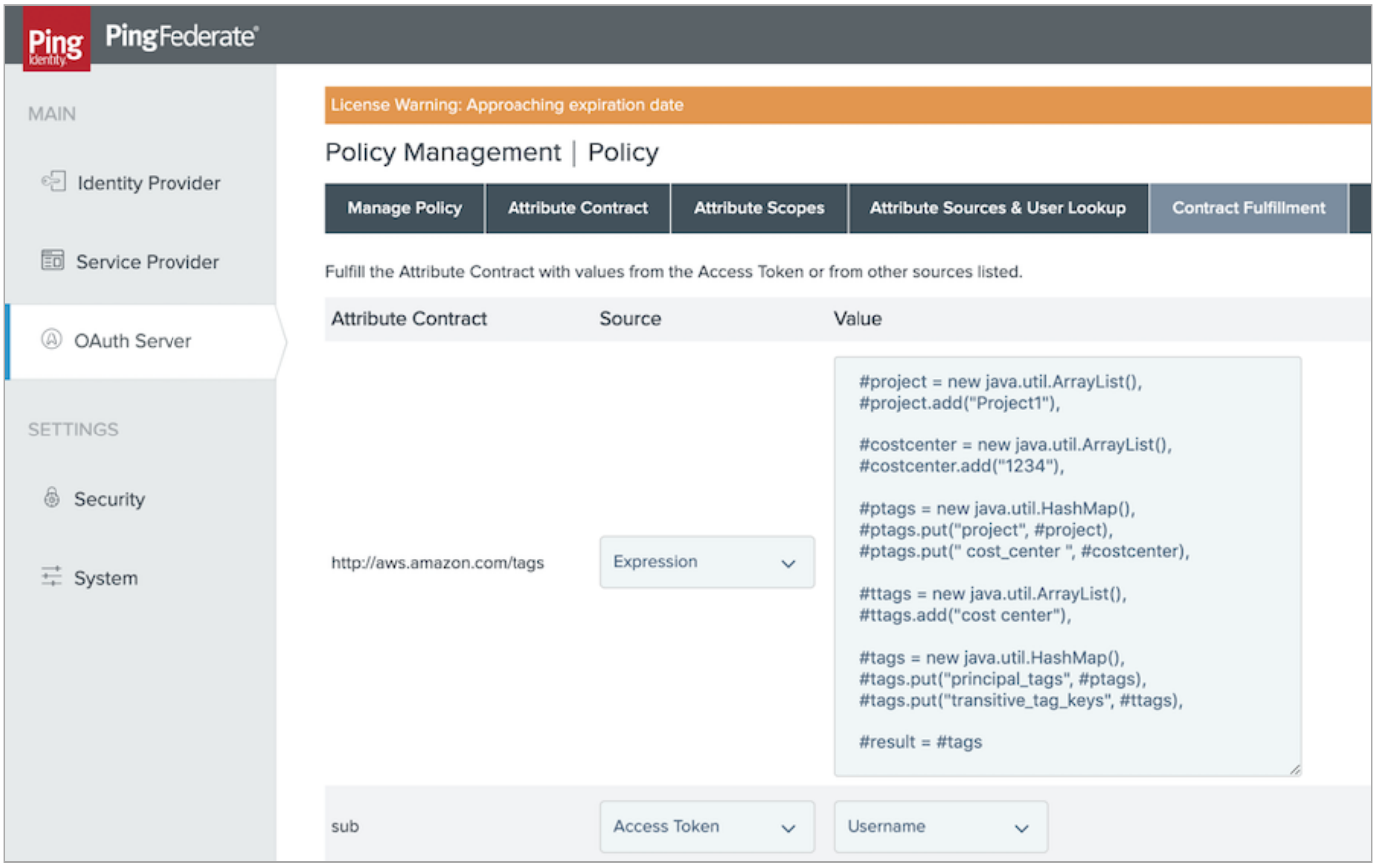
   Choose from:

   - For PingFederate 10.1 or later: go to **Applications > OAuth > OpenID Connect Policy Management**.
   - For PingFederate 10.0 or earlier: go to **OAuth Server > OpenID Connect Policy Management**.

2. Select the client that you want to edit. Click **Attribute Contract**.
3. Create a new attribute and name it `http://aws.amazon.com/tags`. Click **Add**.
4. Click **Contract Fulfillment** and enter the required OGNL expression for the session tag.

> **Note**
>
> You must construct the OGNL expression for the specific source data structure, as shown in the following example.

Copyright © 2025 Ping Identity Corporation



5. Click **Save**, then from **Policy** drop-down list, select the OpenID Connect policy you just created.

Result

The ID token generated by PingFederate will include the following AWS Tags value:

```
https://aws.amazon.com/Tags:
{
"principal_tags ": {
          "project  ":["Project1],
          "cost_center": ["1234"]
        }
"transitive_tags": ["cost_center"]
}
```

Session tags in PingOne

Configuring AWS session tags for PingOne SAML connections

You can create a custom SAML application to support AWS Identity and Access Management (IAM) and AWS IAM Identity Center session tags for SAML connections in PingOne.

Copyright © 2025 Ping Identity Corporation

Before you begin

- Create an Amazon Web Services (AWS) console account and policy that uses session tags. For help, see AWS prerequisites (page 241) in the PingAccess documentation.
- Sign on to your PingOne account as an administrator.
- Configure an external identity provider, such as PingFederate, that will provide the values for the AWS attributes.

About this task

> **Note**
>
> In the PingOne App Catalog, PingOne provides a ready-made AWS application template. That template uses static SAML attributes, and cannot be used for session tags. The following steps allow you to create a custom SAML application to use with AWS session tags.

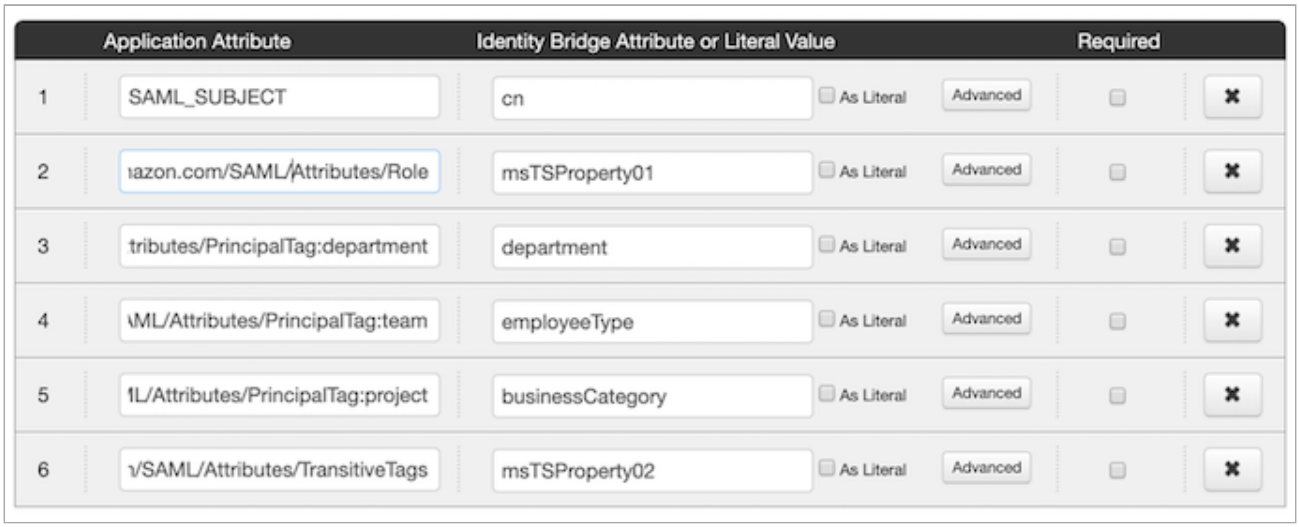
Steps

1. On the PingOne console, go to **Applications > My Applications > Add Application > New SAML Application**.
2. Enter an application name, such as `AWS with Session Tags`.
3. Enter the application description, category, and application icon and then click **Continue to Next Step**.
4. In the **Application Configuration** section, enter the following:
    1. In the **Assertion Consumer Service (ACS)**field, enter `https://signin.aws.amazon.com/saml`.
    2. In the **Entry ID** field, enter `urn:amazon:webservices`.
5. Click **Continue to Next Step**.
6. In the **SSO Mapping Attributes**section, click **Add new attribute**. Enter the session tags attributes that you plan to use.

    Choose from:

    - If you are using AWS IAM Identity Center, include the access control tags based on the following format: `https://aws.amazon.com/SAML/Attributes/AccessControl:{attribute}`
    - If you are using AWS IAM, enter the AWS Principal Tags and TransitiveTagKeys, based on the following examples:
    - `https://aws.amazon.com/SAML/Attributes/AccessControl:{attribute}`
    - `https://aws.amazon.com/SAML/Attributes/PrincipalTag:project`
    - `https://aws.amazon.com/SAML/Attributes/Role`
    - `https://aws.amazon.com/SAML/Attributes/RoleSessioName`
    - `https://aws.amazon.com/SAML/Attributes/TransitiveTagKeys`

Copyright © 2025 Ping Identity Corporation



7. Click **Continue to Next Step** twice and then click **Finish** to create the AWS Session Tag SAML application.

Configuring AWS session tags for PingOne OIDC connections

You can configure AWS session tag support for OpenID Connect (OIDC) connections in PingOne by changing the attribute mappings in your OIDC application.

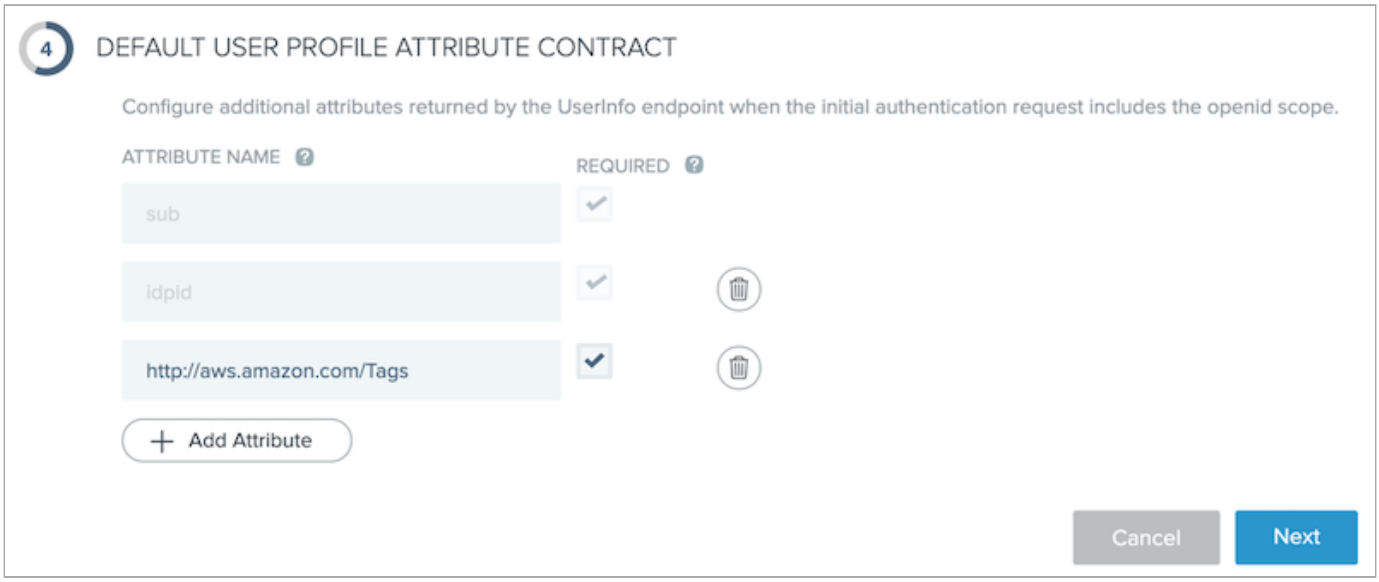
Before you begin

- Create an Amazon Web Services (AWS) console account and policy that uses session tags. For help, see AWS prerequisites (page 241) in the PingAccess documentation.
- Sign on to your PingOne account as an administrator.
- Configure an external identity provider, such as PingFederate, with a `Tags` attribute. The value must be in JSON format as specified by the AWS session tag feature.

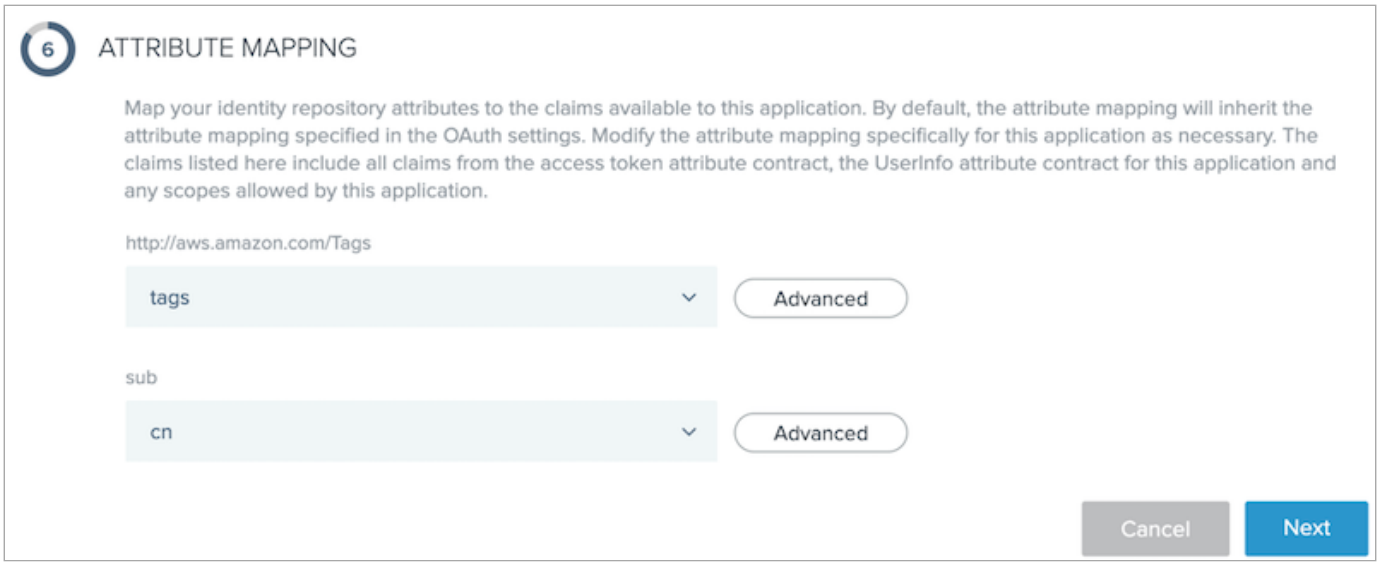
Steps

1. Go to **Applications > OIDC.**
2. Select the application you want to edit and click **Edit**.
3. In the **Default User Profile Attribute Contract** section, click **Add Attribute**.
4. Enter `http://aws.amazon.com/tags` as shown, then click **Next**.

Copyright © 2025 Ping Identity Corporation



5. In the **Attribute Mapping** section, map the `http://aws.amazon.come/tags` attribute to the external identity provider attribute that contains the JSON formatted session tags data as shown.



Copyright © 2025 Ping Identity Corporation

Apache Linux Integration Kit

The Apache Linux Integration Kit allows PingFederate to coordinate user authentication and single sign-on (SSO) between an Apache web application and an identity provider (IdP).

> **Tip**
>
> For new integrations, try the Agentless Integration Kit. It can integrate with a variety of platforms using a modern RESTful approach, and it doesn't require you to integrate agent software into your application.

Components

OpenToken Adapter

Installed in PingFederate, this adapter uses the secure OpenToken standard to pass user attributes and session information from PingFederate to the Apache agent on the Apache server.

Apache agent

Installed on the server running Apache, this program watches for protected resource requests and determines whether to grant access or redirect the user to PingFederate for authentication with an IdP.

Intended audience

This document is intended for PingFederate administrators.

Learn more about the setup process in the following sections of the PingFederate documentation:

- Managing SP adapters
- Configuring an OpenToken SP Adapter instance

System requirements

- PingFederate 11.3 or later
- OpenSSL 1.1.1 or later

The Apache agent supports the following platforms:

- Apache HTTP Server 2.4:
  - Red Hat Enterprise Linux:
    - Red Hat Enterprise Linux 7
    - Red Hat Enterprise Linux 8
    - Red Hat Enterprise Linux 9
  - Canonical Ubuntu:
    - Canonical Ubuntu 18.04

Copyright © 2025 Ping Identity Corporation

- Canonical Ubuntu 20.04
- Canonical Ubuntu 22.04
- Canonical Ubuntu 24.04

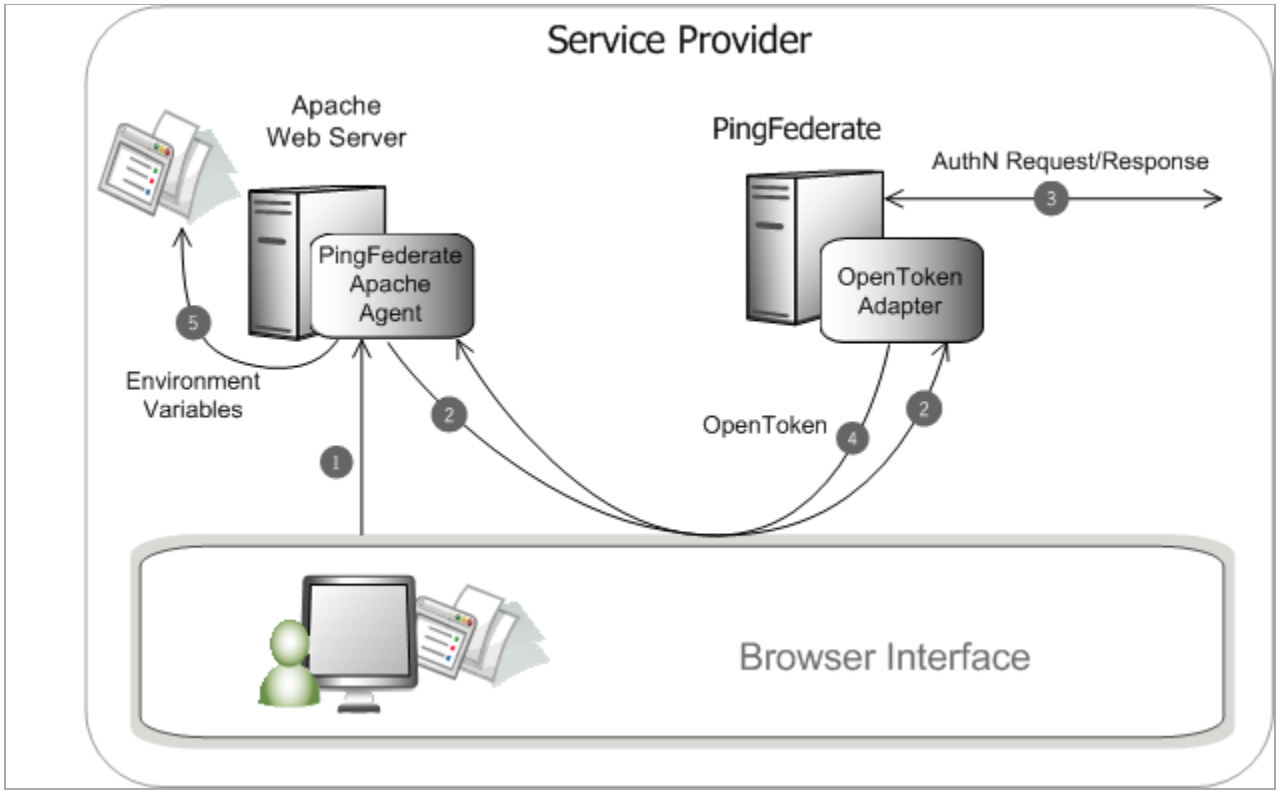
Overview of the SSO flow

The Apache agent acts as a filter in front of an external protected resource, such as an application.

For each request, the Apache agent does one of the following:

- If the request is for an unprotected resource, the Apache agent passes the request to the application.
- If the request is for a protected resource, the Apache agent checks to see if there is a PingFederate session available and if the session parameters meet session policy for that session.
- If a session exists and the session meets session policy for that request, the Apache agent passes the request through to the application.
- If a session doesn't exist or if the existing session doesn't meet the session policy for that request, the Apache agent redirects the browser through the PingFederate server to an identity provider (IdP) for authentication. After authentication, PingFederate redirects the user back to the protected resource with a valid session.

The following diagram illustrates a service provider (SP)-initiated single sign-on (SSO) scenario, showing the request flow and how the PingFederate OpenToken Adapter wraps attributes from an assertion into a secure token (OpenToken) and passes the token to Apache.



Copyright © 2025 Ping Identity Corporation

In this flow:

1. A user attempts to access a resource on the Apache server protected by the PingFederate Apache agent.
    1. The user is redirected to the PingFederate server for authentication.
    2. If an OpenToken session already exists, the user is granted immediate access.
2. The PingFederate server redirects the user's browser to an IdP for authentication using either the SAML or WS-Federation protocols. The IdP partner authenticates the user and returns a SAML assertion.
3. PingFederate validates the assertion and creates an OpenToken for the user including any configured attributes. PingFederate then redirects the browser, including the OpenToken, back to the Apache agent.
4. The Apache agent verifies the OpenToken and grants access to the protected resource. The User ID and any attributes from the OpenToken are exposed to the resource as HTTP request headers or Apache environment variables.

Apache agent information

The Apache agent lets your Apache server communicate with PingFederate. When a user tries to access a resource on your Apache server, the agent checks their session status. When a user starts a new session, the agent validates OpenToken information from PingFederate and lets the user access the resource.

SSL support

The Apache agent supports TLS/SSL, using standard Apache SSL support for connections to the server from browsers.

Session validation

PingFederate validates both an inactivity timeout and an overall session timeout:

Inactivity timeout

The amount of time that a session can be inactive when no new browser requests are received and before a user is required to reauthenticate.

Overall session timeout

The total amount of time that a session can be active, regardless of activity, before the user is required to re-authenticate.

If either of the timeout limits has expired, the Apache agent cancels the existing session and redirects the browser to the `PingFederateLoginPageUrl` address in your `<apache_home>/conf/mod_pf.conf` file. This starts a service provider (SP)-initiated single sign-on (SSO) request at the identity provider (IdP).

> **Note**
>
> Session cancellation enforces session cleanup in the PingFederate server and session cookie obsolescence.

Copyright © 2025 Ping Identity Corporation

Memory usage

The Apache agent uses the Apache Portable Runtime (APR) pools for allocation of most data: either the configuration-time pool for storing configuration variables or request-time pools for processing sessions.

Heap is used only for temporary data with a short usage time and sensitive size, such as:

- Dynamic reallocations on compressed-token decompression
- Parsing session information
- Operations with the OpenToken library

Session information and user attributes

The PingFederate Apache agent passes session information and user attributes from the adapter to the application.

The Apache agent includes the information in HTTP request headers or Apache environment variables. This information can then be used by the application for authorization decisions or for generation of content specific to the user making the request.

The following session and attribute information is exposed to the application:

Attributes from the OpenToken Adapter contract

The subject (`SUBJECT`) and any attributes that you add on the **Extended Contract** tab of the adapter configuration. Only the attributes fulfilled at runtime are exposed to the application. Attributes with a `NULL` value aren't included in the OpenToken.

`NOT-ON-OR-AFTER`

The time until inactivity timeout is reached.

`RENEW-UNTIL`

The time until overall session timeout is reached.

`AUTH_NOT-BEFORE`

The time when the session was created.

`AUTHNCONTEXT`

Information from the SAML assertion that describes how the user was authenticated at the IdP.

For security reasons, each HTTP request header or Apache environment variable is first pre-pended with a specific prefix. Learn more about configuring the prefix in Configuring the Apache agent. The Apache agent always removes and rewrites the prefixed request headers and environment variables for each request.

If you can't modify your applications to accept headers with this prefix, you can configure the Apache agent to add a prefix to the HTTP headers or environment variables. In this case, on the **Extended Contract** tab of the OpenToken Adapter configuration, include an attribute named `pf_attribute_list`. Map that attribute in your identity provider (IdP) connection as a text field containing a comma-separated list of all the attributes in the adapter contract. This attribute list is sent in the OpenToken and used by the Apache agent to overwrite headers in the request.

Copyright © 2025 Ping Identity Corporation

Learn more about Configuring target session fulfillment in the PingFederate documentation.

Error handling

When an error occurs, the Apache agent redirects the browser to the `PingFederateErrorPageURL` address in your `<apache_home>/conf/mod_pf.conf` file.

> **Note**
>
> This value isn't enabled by default, but examples are provided.

Session logout

The `PingFederateCancelURL` address in your `<apache_home>/conf/mod_pf.conf` file initiates a user-session logout. This URL specifies a resource that directs the Apache agent to:

1. Initiate a single logout (SLO), if configured
2. Expire the PingFederate session
3. Clean up any resources associated with the session
4. Pass control back to the application so that it can clean up its own session

Logging

The PingFederate Apache agent uses an Apache API logging scheme that writes into the standard `logs/error_log` file. This file is created automatically at startup with the verbosity level controlled by `LogLevel` in your `httpd.conf` file.

The PingFederate Apache agent has six internally distinguished verbosity levels, ranging from `0` to `5`. The first four correspond to Apache definitions in `error/warn/notice/info`. The last two levels are for logging HTTP requests, responses, and cURL-library debug output, if necessary. The default level is `0`, which logs only errors.

> **Note**
>
> The Apache agent logs all of its output at the `info` level. To access this output, set the Apache `LogLevel` to `info` in `httpd.conf`, then restart the Apache server.

Setup

Upgrading an existing deployment

If you're upgrading from a previous version of the Apache Linux Integration Kit, modify your existing configuration files.

Steps

1. Stop Apache.
2. Download the Apache Linux Integration Kit `.zip` archive from the **Add-ons** tab of the PingFederate downloads page.

Copyright © 2025 Ping Identity Corporation

3. From the `.zip` archive, copy `apache-agent/lib/<your platform>/libopentoken.so` to the Apache modules directory.
4. Edit your existing `mod_pf.conf` file:
    1. Add the following if it doesn't already exist.

```
# Enables or disables the "http only" attribute of the cookie. Http only cookies
# inform the browser that the cookie shouldn't be accessible by client-side scripts.
# The default is set to "yes"

PingFederateCookieHttpOnly              yes
```

    2. Set the value to `yes` or `no` to suit your environment. Save the file.
5. Edit your existing `mod_pf.so` file:
    1. Add the following if it doesn't already exist.

```
# (Required)
# The SameSite cookie attribute is set to this value. Set this to match the value of
# 'cookie-samesite-attribute' in the Agent configuration file (defined in
# PingFederateAgentConfigurationFile), if that is defined.
# The allowed values for this setting are: Strict, Lax, None, and Nothing
# The "Strict", "Lax", and "None" value changes the SameSite cookie attribute setting.
# The "Nothing" value leaves the SameSite cookie attribute unset in the OpenToken Session Cookie.
# For the "None" value, you must use secure attributes because cross-site cookies can only be
# accessed over HTTPS connections.
# If the cookie is not secure and the "None" value is selected, the SameSite cookie attribute
# will not be set.

PingFederateCookieSameSiteAttribute     Nothing
```

    2. Set the value to `Strict`, `Lax`, `None`, or `Nothing` to suit your environment. Save the file.
6. Add the following if it doesn't already exist.

```
# (Optional)
# Enables or disables fragment preservation in requests.
# When set to "yes", preserves request fragment and redirects user back
# to the URI with fragment.
# To prevent sensitive data leakage, ensure that no sensitive information
# is present in a preserved fragment.
# The default is "no".
PingFederateEnableFragmentPreservation   no

# (Optional)
# The HTML template used to generate client side (JavaScript-based) redirects for
# fragment preservation. If not specified, a prebuilt template is used.
# Path could be an absolute or relative to the httpd.conf ServerRoot
# definition.

#PingFederateFragmentPreservationPageTemplateFile conf/fragment_preservation_request_template.html
```


Copyright © 2025 Ping Identity Corporation

7. Start Apache.
8. Update the OpenToken Adapter in PingFederate as shown in Updating the OpenToken Adapter.
9. Reinstall the Apache agent as shown in Apache Agent setup.

Updating the OpenToken Adapter

The Apache Linux Integration Kit relies on the OpenToken Adapter that is distributed with the Java Integration Kit. Update the OpenToken Adapter to get the latest feature and security updates.

Steps

1. Download the Java Integration Kit `.zip` archive from the **Add-ons** tab of the PingFederate downloads page.
2. Stop PingFederate.
3. Delete the `opentoken-adapter-<version>.jar` file from your `<pf_install>/pingfederate/server/default/deploy` directory.
4. From the Java Integration Kit `.zip` archive, copy the contents of `dist/pingfederate` to your `<pf_install>/pingfederate` directory.

   > **Note**
   >
   > The `commons-collections`, `commons-beanutils`, and `commons-logging` libraries are provided as a convenience and should be installed only if they aren't already contained in the application `CLASSPATH`.

5. Start PingFederate.
6. If you operate PingFederate in a cluster, repeat steps 2-5 for each engine node.

Configuring an OpenToken SP Adapter instance

Configure the OpenToken Adapter to determine how PingFederate communicates with your service provider application.

Steps

1. In the PingFederate administrative console, go to **Authentication > Integration > IdP Adapters**. Click **Create new Instance**.
2. On the **Type** tab, set the basic adapter instance attributes.
   1. In the **Instance Name** field, enter a name for the adapter instance.
   2. In the **Instance ID** field, enter a unique identifier for the adapter instance.
   3. From the **Type** list, select **OpenToken Adapter**. Click **Next**.
3. On the **Instance Configuration** tab, configure the adapter instance by referring to Configuring an OpenToken SP Adapter instance in the PingFederate documentation. Click **Next**.

Copyright © 2025 Ping Identity Corporation

> **Note**
>
> If you use Apache to protect multiple sites on the same domain, in the OpenToken Adapter instance configuration, select **None** for SameSite Cookie, and select the **Secure Cookie** check box.

4. Export the configuration file:
    1. On the **Actions** tab, click **Download** and then click **Export**.
    2. Save `agent-config.txt`. Click **Next**.
5. On the **Extended Contract** tab, add any attributes that you expect to retrieve other than the SAML subject. Click **Next**.
6. On the **Target App Info** tab, enter the basic information about your SP application. Click **Next**.
7. On the **Summary** tab, check and save your configuration. Click **Save**.
8. Create or update an identity provider (IdP) connection to use the OpenToken Adapter instance as shown in Service provider SSO configuration in the PingFederate documentation.

Apache Agent setup

To set up the Apache agent, you must copy a configuration file and a library from the extracted integration-kit distribution to your Apache installation and then modify the configuration file.

> **Note**
>
> The Apache agent setup is required for new installations and for upgrades.

Deploying the Apache agent

To use the Apache agent, copy the files to your Apache directory and modify your Apache configuration.

About this task

The PingFederate Apache agent is represented by the `<apache_home>/conf/mod_pf.conf` Apache module (dynamic library) and an auxiliary OpenToken library. The behavior of the Apache agent is controlled by properties contained in the `mod_pf.conf` file.

Steps

1. Download the Apache Linux Integration Kit `.zip` archive from the Ping Identity Integration Directory.
2. In the Apache Linux Integration Kit `.zip` archive, copy the contents of the `apache-agent/lib` directory that corresponds to your Linux version into your Apache `/modules` directory. If the files already exist, overwrite them.

> **Note**
>
> For an Apache HTTP Server running on Canonical Ubuntu, use the modules found in the corresponding version's subdirectory: `Apache_2.4/Ubuntu<version>_64`.

Copyright © 2025 Ping Identity Corporation

3. For new installations, in the integration-kit `apache-agent/config` directory, copy the `mod_pf.conf`, `start_page_template.html`, `fragment_preservation_request_template.html`, and the `error_page_template.html` files into the `/conf` directory of your Apache installation (or the `/etc/apache2/conf-available/` directory in Ubuntu).
4. Copy the `agent-config.txt` file that you downloaded in Configuring an OpenToken SP Adapter instance to the Apache `/conf` folder.
5. If you're using Security Enhanced Linux, run the following commands as the root user:

   ```
   chcon --reference /usr/sbin/httpd /etc/httpd/modules/mod_pf.so
   chcon --reference /usr/sbin/httpd /etc/httpd/modules/libopentoken.so
   ```

   This allows the agent to run in the `httpd` context.

   > **Note**
   >
   > The preceding paths assume the default Linux installation.

6. In the Apache `httpd.conf` file, add the following statement above any other `LoadModule` statements:

   ```
   LoadModule access_compat_module modules/mod_access_compat.so
   LoadFile modules/libopentoken.so
   LoadModule pf_module modules/mod_pf.so
   PingFederateConfigurationFile conf/mod_pf.conf
   ```

   > **Note**
   >
   > Depending on the Ubuntu version, the module installation with default Apache could require additional steps. If so, perform steps 6a - 6c. Otherwise, skip to step 7.

   a. Place the `.so` files in the `/usr/lib/apache2/modules/` directory.

      Result:

      A configuration file (`/etc/apache2/mods-available/mod_pf.load`) is created to load the module. It contains the following statements:

      ```
      LoadModule access_compat_module /usr/lib/apache2/modules/mod_access_compat.so
      LoadFile  /usr/lib/apache2/modules/libopentoken.so
      LoadModule pf_module  /usr/lib/apache2/modules/mod_pf.so
      ```

   b. Link this file as enabled modules in the `/etc/apache2/mods-enabled/` directory:

      ```
      sudo ln -s ../mods-available/mod_pf.load mod_pf.load
      ```

   c. Link the `/etc/apache2/conf-available/mod_pf.conf` file as enabled conf in the `/etc/apache2/conf-enabled` file:

Copyright © 2025 Ping Identity Corporation

```
sudo ln -s ../conf-available/mod_pf.conf mod_pf.conf
```

7. Add the following statement within all `Directory` contexts that the agent should handle:

```
AuthType PFApacheAgent
```

> **Tip**
>
> Use a "deny by default" configuration for all directories that you want the Apache agent to protect:
>
> ```
> Order Deny,Allow
> Deny from all
> ```
>
> Learn more about `AuthType` examples in Apache Integration Kit AuthType examples in the Ping Identity Knowledge Base.

8. Restart Apache.

Configuring the Apache agent

Configure the Apache agent by modifying the `mod_pf.conf` file that you copied in Deploying the Apache agent.

Steps

1. Edit the `<apache_home>/conf/mod_pf.conf` file by modifying the example configuration and any optional properties based on your environment and needs.

   Comments above each property describe the function and possible values for that property.

2. Save the file and restart Apache to load the new configuration.

Setting up virtual hosts

Each virtual host can optionally have its own respective Apache agent configuration.

About this task

If no custom configuration is provided, a virtual host uses the agent configuration from the base server.

Steps

- If you want to use this feature, add the following to the Apache `httpd.conf` file in the virtual host context:

```
PingFederateConfigurationFile conf/mod_pf_vhost.conf
```


Copyright © 2025 Ping Identity Corporation

In this example, `mod_pf_vhost.conf` is an agent configuration file that contains settings unique to the virtual host. PingFederate attributes from the base server configuration aren't merged with the virtual host when the `PingFederateConfigurationFile` attribute is specified for the virtual host.

Testing the Apache agent

The Apache agent includes a protected start page to help you verify your configuration. This feature is for testing purposes only and is disabled by default.

About this task

The start page initiates a single sign-on transaction with the identity provider partner. If it succeeds, it displays the HTTP headers that the Apache agent will expose to your application. These headers correspond to attributes from the SAML assertion.

Steps

1. Turn on the test page:
    1. Edit the `<apache_home>/conf/mod_pf.conf` by adding `#` to comment out the `PingFederateStartPageURL` property.
    2. Save the file and restart Apache.
2. Modify the following URL for your environment and then open it in a browser.

```
http://apache-server/protected-path/?cmd=PingStartPage
```

3. When you finish testing, revert your changes to `mod_pf.conf`.

> **Important**
>
> For security reasons, you should disable this feature in production environments.

4. Restart Apache.

Release notes

Changelog

The following is the change history for the Apache Linux Integration Kit.

Apache Linux Integration Kit 4.2 – January 2025

- Added support for:
    - Red Hat Enterprise Linux 9
    - Canonical Ubuntu 22.04

Copyright © 2025 Ping Identity Corporation

- Canonical Ubuntu 24.04

Apache Linux Integration Kit 4.1.1 – August 2022

- Fixed an issue that could cause errors with specific string lengths for the cookie name.

Apache Linux Integration Kit 4.1 – June 2022

- Reimplemented fragment preservation with a new approach.

Apache Linux Integration Kit 4.0 – January 2022

- Added support for:
  - Red Hat Enterprise Linux 8
  - Canonical Ubuntu 18.04
  - Canonical Ubuntu 20.04
- Ended support for:
  - Red Hat Enterprise Linux 6
  - Canonical Ubuntu 14.04
  - Canonical Ubuntu 16.10
- Updated OpenSSL version to 1.1.1 for Red Hat Enterprise Linux 7.
- Improved redirect handling. When users are redirected back from PingFederate after a flow, the Apache Linux Integration Kit agent can now preserve fragments in the URI, such as `#MyHomePage`. You can enable this behavior with the **PingFederateEnableFragmentPreservation** setting in the `mod_pf.conf` file.

Apache Linux Integration Kit 3.3.5 – November 2021

- Improved the way the adapter handles browser cookie attributes and expired cookies.
- Removed the OpenToken Adapter from the `.zip` archive. The latest version is available in the Java Integration Kit. Learn more in Updating the OpenToken Adapter.
- Changed to a standardized `.zip` file structure to make automated deployments easier.

Apache Linux Integration Kit 3.3.4 – September 2020

- Fixed an issue that could insert bad characters into legacy cookie names.

Apache Linux Integration Kit 3.3.3 – January 2020

- Added support for the `SameSite` cookie flag in web browsers.

Apache Linux Integration Kit 3.3.2 – July 2018

- Added the ability to read cookie HTTP request header case insensitively.
- Removed module info from the Server HTTP response header.

Copyright © 2025 Ping Identity Corporation

Apache Linux Integration Kit 3.3.1 – February 2018

- Added support for the Amazon Web Services Application Load Balancer.
- Fixed a URL encoding issue with special characters.

Apache Linux Integration Kit 3.3 – July 2017

- Removed support for Red Hat Enterprise Linux 5.
- Improved query parameter encoding.
- Improved cookie name parsing.

Apache Linux Integration Kit 3.2.1 – February 2017

- Fixed an issue with retrieving the session token.

Apache Linux Integration Kit 3.2 – January 2017

- Added the ability to encode query parameters.
- Added the ability to disable virtual hosts by port value.
- Added support for Ubuntu 16.10.
- Added support for the `HttpOnly` flag on cookies. `HttpOnly` prevents the cookie from being intercepted or manipulated. This mitigates a significant portion of vulnerabilities, such as cross-site scripting (XSS), to which the cookie would otherwise be susceptible. This ensures the Apache agent's behavior and security considerations are consistent with behavior for other adapters and agents. You can control this behavior with the `PingFederateCookieHttpOnly` setting in the `mod_pf.conf` file.
- Bug fixes.

Apache Linux Integration Kit 3.1 – June 2016

- Added support for Oracle HTTP Server.

Apache Linux Integration Kit 3.1 – April 2015

- Added support for Apache 2.2 on Red Hat Enterprise Linux 5.0 (64-bit).
- Optimized OpenToken lookup.
- Fixed a security vulnerability.

Apache Linux Integration Kit 3.0.1 – December 2014

- Added Apache 2.4 support for Red Hat Enterprise Linux 7.0 and Ubuntu 14.04.
- Fixed an issue that caused the Apache agent to handle directories containing spaces improperly.

Apache Linux Integration Kit 3.0 – May 2014

- Added support for Red Hat Enterprise Linux 6.5.

Copyright © 2025 Ping Identity Corporation

- Added the ability to specify custom virtual host configurations.
- Added the ability to expose OpenToken attributes for unprotected resources in the `mod_pf.conf` file.
- Added the ability to use Allow/Deny directives in combination with `Satisfy Any` to achieve IP allow or deny directives.

Apache Linux Integration Kit 2.4.1 – June 2013

- Added support for Apache Server 2.4.

> **Note**
>
> Skipped versions 2.3.7 through 2.4 for internal configuration management.

Apache Linux Integration Kit 2.3.6 – March 2013

- Added support for:
  - Ubuntu 12.04 LTS.
  - Apache Server 2.2.

Apache Linux Integration Kit 2.3.1 – December 2012

- Fixed a security vulnerability.
- Added support for the OpenToken 2.5.1 adapter and agent.

Apache Linux Integration Kit 2.3 – December 2011

- Added the ability to disable the Apache agent for specific virtual hosts in the `mod_pf.conf` file.
- Added the ability to specify the Cache-Control HTTP header value in the `mod_pf.conf` file.
- Added support for passing multivalued attributes through the HTTP headers.
- Fixed an issue that caused automatic escaping of single and double quotes in attribute values. These are exposed to the application through the HTTP headers and server variables.

Apache Linux Integration Kit 2.2.1 – February 2010

- Fixed an issue that caused the Apache agent to truncate `POST` data.

Apache Linux Integration Kit 2.2 – September 2009

- Added a start page and error page for the Apache agent. Learn more in Download manifest.
- Added support for the default RHEL OpenSSL installation.
- Fixed an issue that caused Application Schemes, Hosts, and Port options to fail to work with the OpenToken when sent as a query parameter.
- Fixed an issue that caused the Apache agent to fail to check the `POST` and query values, then the cookie for a valid OpenToken.

Copyright © 2025 Ping Identity Corporation

Apache Linux Integration Kit 2.1 – March 2009

- Added support for:
  - Red Hat Enterprise Linux 4 and 5 (32-bit and 64-bit).
  - Apache 2.0 and 2.2.
  - Prefork and worker multiprocessing modules.
  - Reverse proxies
- Added richer support for single logout.
- Added the ability to use an alternative method to set session attributes as HTTP headers or environment variables without a prefix.

Apache Linux Integration Kit 2.0 – November 2008

- Removed deprecated functionality that wasn't specific to PingFederate to simplify the Apache module.
- Added the ability to expose attributes as HTTP request headers.
- Added the ability to log the SAML subject in the Apache `access_log`.
- Updated the OpenToken library to support password obfuscation.
- Updated the OpenToken library to support the `POST` transport method.
- Fixed an issue that required Cancel URLs to be contained in a protected resource.
- Fixed an issue that caused the cookie domain to be validated against the agent configuration file if `query` or `POST` is used as the initial transport method.

Apache Linux Integration Kit 1.1 – October 2008

- Added support for `dynamic TargetResource`.
- Removed items that aren't necessary for the PingFederate implementation to simplify the configuration.
- Added the ability for filters to use the full request URL, including query parameters, to determine whether to protect a resource.
- Fixed an issue that caused the module to fail to start up if the transport mode was set to **Query Parameter** in the OpenToken adapter setup. The OpenToken session now uses the `cookie-domain` property out of the `mod_plaa.conf` file instead of the `mod_pf.conf` file.
- The Apache Linux Integration Kit now ships with OpenToken 2.2.2. OpenToken 2.2.2 fixed an issue that appended a question mark (`?`) to the target resource URL, which Apache couldn't process.

Apache Linux Integration Kit 1.0 – September 2007

- Initial release.

Copyright © 2025 Ping Identity Corporation

Known issues and limitations

The following are known issues or limitations for the Apache Linux Integration Kit.

Known issues

- URL encoding in the `PingFederateFilter` and `PingFederateFilterAnonymous` isn't supported.
- In some configurations, Apache fails to load the module with the following being the last log entry: `Deobfuscating password for the password-based keys generation`.

  To overcome this issue, ensure the agent module is loaded immediately after the authz-related modules.

Known limitations

- `Satisfy All` isn't supported when used in conjunction with the `Allow` directive for access control.
- `RequireAll` isn't supported.
- The `PingFederateFilter` doesn't support:
  - URL encoded characters or non-ASCII characters.
  - Query strings
- The PingFederate Apache agent relies on path comparison to protect resources. Other modules in Apache such as `mod_speling` and `mod_rewrite` can affect the way the PingFederate Apache agent evaluates resources.
- To support fragment preservation, the # symbol in URL fragment must be URL encoded.

Download manifest

The following files are included in the Apache Linux Integration Kit `.zip` archive.

- `Legal.pdf` : Copyright and license information.
- `apache-agent` : Contains the PingFederate Apache agent files.
  - `config` : Contains configuration files.
    - `mod_pf.conf` : Apache agent module configuration file.
    - `start_page_template.html` : Apache agent start-page template.
    - `error_page_template.html` : Apache agent error-page template.
  - `lib/Apache_2.4`
    - `RHEL7_64` : Contains the RHEL 7.x 64-bit.
      - `libopentoken.so` : OpenToken library.
      - `mod_pf.so` : Apache agent module.

Copyright © 2025 Ping Identity Corporation

- `RHEL8_64`: Contains the RHEL 8.x 64-bit.
  - `libopentoken.so`: OpenToken library.
  - `mod_pf.so`: Apache agent module.
- `RHEL9_64`: Contains the RHEL 9.x 64-bit.
  - `libopentoken.so`: OpenToken library.
  - `mod_pf.so`: Apache agent module.
- `Ubuntu18_64`: Contains the Canonical Ubuntu 18.04 64-bit.
  - `libopentoken.so`: OpenToken library.
  - `mod_pf.so`: Apache agent module.
- `Ubuntu20_64`: Contains the Canonical Ubuntu 20.04 64-bit.
  - `libopentoken.so`: OpenToken library.
  - `mod_pf.so`: Apache agent module.
- `Ubuntu22_64`: Contains the Canonical Ubuntu 22.04 64-bit.
  - `libopentoken.so`: OpenToken library.
  - `mod_pf.so`: Apache agent module.
- `Ubuntu24_64`: Contains the Canonical Ubuntu 24.04 64-bit.
  - `libopentoken.so`: OpenToken library.
  - `mod_pf.so`: Apache agent module.

Copyright © 2025 Ping Identity Corporation

Apple Login Integration Kit

The Apple Login Integration Kit allows PingFederate to use Apple as an identity provider (IdP). This allows users to access service provider (SP) applications by signing in with their Apple ID.

Features

- Supports the PingFederate PingFederate authentication API.
- Supports the JavaScript Widget for the PingFederate Authentication API.

Components

Apple IdP Adapter

- Allows PingFederate to communicate with the Apple API to process sign-on requests and get user information.

Templates

- Allows the adapter to prompt the user to sign on. The template can be presented with a browser redirect or as a pop-up window.
- Allows you to modify the appearance of the sign-on prompt.

Language packs

- Allows you to customize or localize the messages returned by the PingFederate authentication API and shown on the templates during authentication. For help, see Localizing messages for end users in the PingFederate documentation.

Intended audience

This document is intended for PingFederate administrators.

If you need help during the setup process, learn more in the following resources:

- The following sections of the PingFederate documentation:
  - Identity provider SSO configuration
  - Managing IdP adapters
  - Authentication policies
  - SP connection management
- The following sections of the Apple developer documentation:
  - About Sign in with Apple
  - Sign in with Apple Overview
  - Sign in with Apple REST API

Copyright © 2025 Ping Identity Corporation

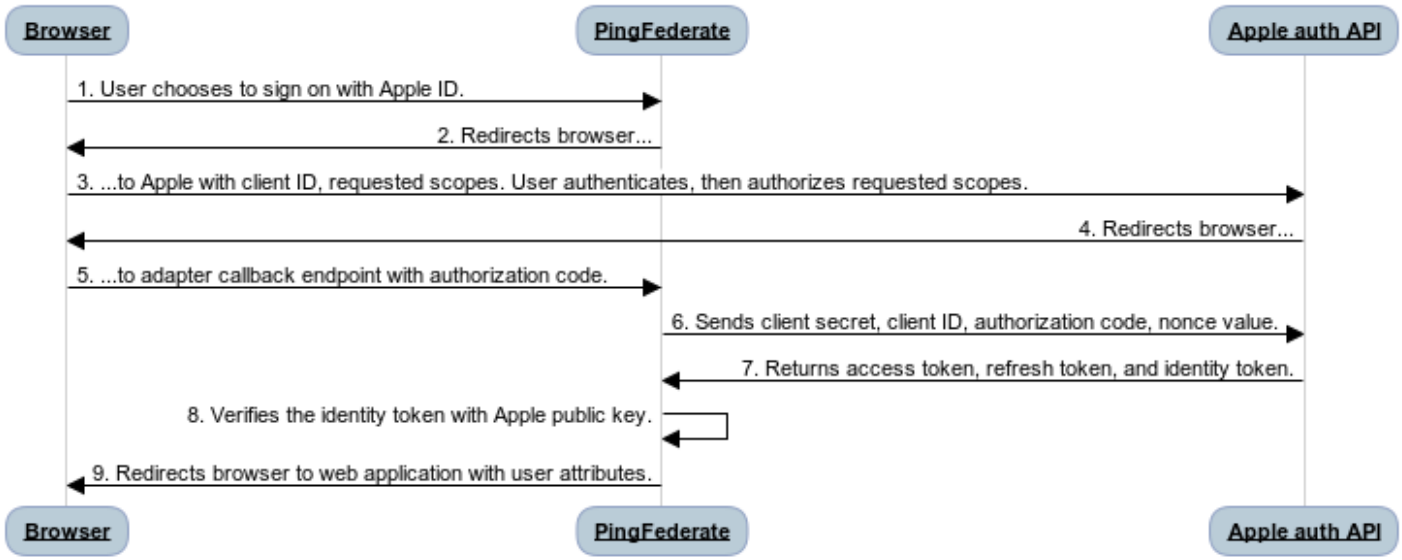
System requirements

- PingFederate 10.3 or later.
- An Apple Developer account. To sign up, visit What You Need To Enroll on the Apple Developer Program site.
- End users must have two-factor authentication set up to use Sign in with Apple.

Overview of the SSO flow

With the Apple Cloud Identity Connector, PingFederate includes an Apple authentication API in the sign-on flow.

The following figure illustrates a service provider (SP)-initiated single sign-on (SSO) scenario in which PingFederate authenticates users to an SP application using the Apple IdP Adapter.



Description

1. The user opens a web application and chooses the **Sign in with Apple** option.
2. The sign-on link points to the PingFederate Apple IdP Adapter, which redirects the browser...
3. ...to Apple with the client ID and a list of requested scopes. On the Apple site, the user authenticates their identity and then authorizes the requested scopes.
4. Apple redirects the browser...
5. ...to the PingFederate Apple IdP Adapter authorization callback endpoint with an authorization code.

   If the user fails to authenticate or does not authorize the request, the response includes an error code instead.
6. The Apple IdP Adapter generates a client secret JSON object. PingFederate sends the client secret, client ID, and nonce value to Apple.

Copyright © 2025 Ping Identity Corporation

> **Note**
>
> For more about the client secret object, see Creating the Client Secret in the Apple Developer documentation.

7. Apple returns an access token, refresh token, and an identity token.

> **Note**
>
> For more about the identity token object, see Retrieve the User's Information from Apple ID Servers in the Apple Developer documentation.

8. The Apple IdP Adapter uses the Apple public key to verify the identity token.
9. PingFederate redirects the user to the web application with the user attributes from the identity token.

Authentication API Support

You can use the PingFederate authentication API to integrate the Apple IdP Adapter into your application.

The PingFederate Authentication API provides access to the current state of the authentication flow as a user steps through the PingFederate authentication policy. For more information, see PingFederate authentication API in the PingFederate documentation. To integrate the Apple IdP Adapter into your authentication flow, configure your application based on the states, actions, and models available in the PingFederate Authentication API Explorer. For help, see Exploring the Authentication API in the PingFederate documentation.

Setup

Deploying the integration files

To get started with the integration, deploy the Apple Login Integration Kit files to your PingFederate directory.

About this task

> **Note**
>
> If you operate PingFederate in a cluster, the following steps refer to the console node.

Steps

1. Download the Apple Login Integration Kit `.zip` archive from the Ping Identity Integration Directory.
2. Stop PingFederate.
3. If you're upgrading an existing deployment, back up your customizations and delete earlier versions of the integration files:
    1. Back up any Apple Login Integration Kit files that you customized in the `<pf_install>/pingfederate/server/default/conf/` directory.

Copyright © 2025 Ping Identity Corporation

2. Delete the `pf-apple-idp-adapter-<version>.jar` file from your `<pf_install>/pingfederate/server/default/deploy` directory.

4. Extract the `.zip` archive and merge the contents of the `dist` directory with your `<pf_install>/pingfederate` directory.
5. If there is more than one version of the `pf-authn-api-sdk-<version>.jar` file in your `<pf_install>/pingfederate/server/default/lib` directory, delete all but the latest version of the file.
6. If you backed up any customized files, modify the new files with your customizations.
7. Start PingFederate.
8. If you operate PingFederate in a cluster, repeat steps 2-8 for each engine node.

Registering an Apple app

To allow PingFederate to process social sign-on requests with Apple, add PingFederate as an OpenID Connect application on the Apple Developer site.

Steps

1. Sign on to the Apple Certificates, Identifiers & Profiles page with an Apple Developer administrator account.
2. On the **Identifiers** screen, next to the **Identifiers** heading, click **Add (+)**.
3. On the **Register a new identifier** screen, select **App IDs**. Click **Continue**.
4. Select **App**. Click **Continue**.
5. On the **Register an App ID** screen, in the **Description** field, enter a description for your application.
6. In the **Bundle ID** field, enter a name for your application, such as `com.example.mywebapp`.
7. In the **Capabilities** section, select **Sign in with Apple**. Click **Continue**.
8. On the **Confirm your App ID** screen, under **App ID Prefix**, note your **Team ID**, such as "3R6RFC464N". You will use this in Configuring an adapter instance. Click **Register**.

Registering an Apple service

To allow PingFederate to process social sign-on requests with Apple, add PingFederate as a Sign in with Apple service on the Apple Developer site.

About this task

You can find official Apple documentation and troubleshooting suggestions in Configure Sign in with Apple for the web in the Apple Developer documentation.

Steps

1. Sign on to the Apple Developer site with an Apple Developer administrator account.
2. On the **Certificates, Identifiers & Profiles > Identifiers** page, next to the **Identifiers** heading, click **+**.

Copyright © 2025 Ping Identity Corporation

3. On the **Register a new identifier** page, select **Services IDs**. Click **Continue**.
4. On the **Register a Services ID** page, in the **Description** field, enter a description for your application.
5. In the **Identifier** field, enter a name for your application, such as `com.example.mywebservice`. You will use this as the **Services ID** in Configuring an adapter instance. Click **Continue**. Click **Register**.

> **Note**
>
> This is different from the bundle ID that you entered in Registering an Apple app.

6. On the **Identifiers** page, select the service that you created.
7. On the **Edit your Services ID Configuration** page, select **Sign in with Apple**. Click **Configure**.
8. On the **Web Authentication Configuration** modal, from the **Primary App ID** list, select the app that you registered in Registering an Apple app.
9. In the **Register Website URLs** section, in the **Domains and subdomains** field, enter the domain or domains for your web application, such as `example.com`.
10. In the **Return URLs** field, enter the Apple IdP Adapter callback endpoint on your PingFederate server, based on the following: `https://pf_host:pf_port/ext/apple-authn`. Click **Next**. Click **Done**.
11. Click **Continue**. Click **Save**.

Creating an Apple private key

To allow PingFederate to process social sign-on requests with Apple, add PingFederate as an OpenID Connect application in the Apple administrative console. PingFederate uses the resulting private key to sign the client secret JWT that it sends to Apple.

Steps

1. Sign on to the Apple Certificates, Identifiers & Profiles page using an Apple Developer administrator account.
2. On the **Keys** page, next to the **Keys** heading, click **Add (+)**.
3. On the **Register a New Key** page, in the **Key Name** field, enter a name, such as `My PingFederate Client Secret Signing Key`.
4. Select **Sign in with Apple**. Click **Configure**.
5. On the **Configure Key** page, from the **Choose a Primary App ID** list, select the app that you registered in Registering an Apple app. Click **Save**.
6. On the **Register a New Key** page, click **Continue**.
7. Verify your configuration, and then click **Register**.
8. On the **Download Your Key** page, note the **Key ID**. You will use this in Configuring an adapter instance.
9. Click **Download**, and then save the `.p8` file. You will use this in Configuring an adapter instance. Click **Done**.

Copyright © 2025 Ping Identity Corporation

> **Note**
>
> You can only download your private key once. Keep it in a secure place.

Configuring an adapter instance

Configure the Apple IdP Adapter to determine how PingFederate communicates with Apple Apple.

Steps

1. In the PingFederate administrative console, go to **Identity Provider > Adapters**. Click **Create New Instance**.
2. On the **Type** tab, set the basic adapter instance attributes.
    1. In the **Instance Name** field, enter a name for the adapter instance.
    2. In the **Instance ID** field, enter a unique identifier for the adapter instance.
    3. From the **Type** list, select **Apple IdP Adapter**. Click **Next**.
3. On the **IdP Adapter** tab, configure the adapter instance by referring to Apple IdP Adapter settings reference. Click **Next**.
4. On the **Extended Contract** tab, add any **Local Attributes** that you added in the **Apple Response Mappings** table. Click **Next**.
5. On the **Adapter Attributes** tab, set pseudonym and masking options as shown in Set pseudonym and masking options in the PingFederate documentation. Click **Next**.
6. On the **Adapter Contract Mapping** tab, configure the contract fulfillment details for the adapter as shown in Define the IdP adapter contract in the PingFederate documentation. Click **Next**.
7. On the **Summary** tab, click **Done**. On the **Manage IdP Adapter Instances** tab, click **Save**.

Apple IdP Adapter settings reference

Field descriptions for the Apple IdP Adapter configuration screen.

Standard fields

Field	Description
Team ID	The **Team ID** that you noted in Registering an Apple app. This is used as `iss` in the client secret.
Services ID	The **Identifier** that you registered in Registering an Apple service. This is used as the `client_id` when the adapter requests access and refresh tokens. It is also used as `sub` when the adapter creates the client secret JWT.
Key ID	The **Key ID** that you noted in Creating an Apple private key. This is used as `kid` in the client secret.

Copyright © 2025 Ping Identity Corporation

Field	Description
Client Secret Signing Key	The private key `.p8` file that you downloaded in Creating an Apple private key. PingFederate encrypts this file when you add it to your configuration.
Error Redirect URL	When an error occurs in the adapter, PingFederate redirects the browser to this URL instead of the default error page. This field is blank by default.
Authorization Callback Endpoint	The PingFederate endpoint that Apple uses to respond to authorization requests. If you set a custom endpoint in the **Authorization callback URL** field in Registering an Apple service, change this field to match. This default value is `/apple-authn`.

Advanced fields

Field	Description
Apple Authorization URL	The URL that PingFederate uses to send authorization requests to Apple. If Apple changes this endpoint, enter the new URL. The default value is `https://appleid.apple.com/auth/authorize`.
Apple Access Token URL	The URL that PingFederate uses to retrieve access tokens from Apple. If Apple changes this endpoint, enter the new URL. The default value is `https://appleid.apple.com/auth/token`.
Apple ID URL	The URL that PingFederate uses to communicate with Apple ID. The default value is `https://appleid.apple.com`.
Apple Public Key URL	The URL that PingFederate uses to retrieve the Apple public key. The default value is `https://appleid.apple.com/auth/keys`.
Client Secret Expiration	The amount of time in minutes that the client secret is valid. Enter a value between 1 and 262950 (6 months). The default value is `5`.
Scopes	The scopes that you want to request from Apple. Separate scopes with a space. See the note about scopes in Known issues and limitations. Valid scopes include `openid`, `email`, and `name`. The default value is `openid email name`.

Copyright © 2025 Ping Identity Corporation

Field	Description
Apple Sign-On Presentation	Determines how the adapter presents the Apple sign-on form. ***Redirect (default)*** The adapter redirects the browser to the Apple sign-on form. ***Pop-up window*** The adapter opens a new window with the Apple sign-on form on a PingFederate template. Use this option if automatic redirects are blocked by your users' browsers. This setting has no effect when using the adapter through the PingFederate authentication API.
Apple Pop-Up Template	The template file that presents the Apple sign-on form. Applies only when **Apple Login Presentation** is set to **Pop-up window**. The default value is `apple-pop-up-template.html`.
Apple Post-Auth Template	The template file that the adapter presents after the user signs on. Applies only when **Apple Login Presentation** is set to **Pop-up window**. The default value is `apple-post-auth-template.html`.
Apple Messages File	The language-pack file associated with the Apple pop-up template. The default value is `pingfederate-apple-adapter-messages`.
Retry Request	Determines whether PingFederate will retry requests after it receives a response with a failure code. This check box is cleared by default.
Maximum Retries Limit	Determines how many times PingFederate retries a request. The default value is `5`.
Retry Error Codes	A list of response codes that you want to trigger a retry. Separate response codes with a comma. The default value is `400`.
API Request Timeout	The amount of time in milliseconds that PingFederate waits for Apple to respond to requests. A value of `0` disables the timeout. The default value is `2000`.
Connection Timeout	The amount of time in milliseconds that PingFederate allows to establish a connection with Apple. A value of 0 disables the timeout. The default value is `2000`.
Proxy Settings	Defines proxy settings for outbound HTTP requests. The default value is **System Defaults**.
Custom Proxy Host	The proxy server host name to use when **Proxy Settings** is set to **Custom**. This field is blank by default.

Copyright © 2025 Ping Identity Corporation

Field	Description
Custom Proxy Port	The proxy server port to use when **Proxy Settings** is set to **Custom**. This field is blank by default.

Available user attributes

Lists the attributes supported by Sign in with Apple.

For more details about the following attributes, see Retrieving the User's Information from Apple ID Servers in the Apple Developer documentation.

For the `email`, `first_name`, and `lastname` attributes, see the note about scopes in Known issues and limitations.

Attribute	Description
`iss`	The issuer-registered claim key, which has the value https://appleid.apple.com.
`user_id`	The unique identifier for the user. This is Apple's primary identifier for the user.
`aud`	The **client_id** for your Apple Developer account.
`email`	The user's email address. The user may choose to share their real email address or an anonymous one that uses the private email relay service. Available with the `email` scope.
`email_verified`	A Boolean value that indicates whether the service has verified the email. The value of this claim is always true because the servers only return verified email addresses.
`access_token`	The access token provided by Apple.
`first_name`	The user's first name. Available with the `name` scope.
`last_name`	The user's last name. Available with the `name` scope.

Integrating Sign in with Apple into your application

To complete your Sign in with Apple integration, create an SP connection or adapter mapping and then add a sign-on hyperlink to your application.

Copyright © 2025 Ping Identity Corporation

Steps

1. If your application is outside the PingFederate domain, configure a service provider (SP) connection.
    1. Create an SP connection that uses your IdP adapter instance as shown in SP connection management and Mapping an adapter instance in the PingFederate documentation.
    2. In your web application, create a hyperlink to allow users to sign on to the SP application. Use the following URL and replace the variables based on the descriptions in the following table.

        https://*pf_host*:*pf_port*/idp/startSSO.ping?PartnerSpId=*ConnectionId*

Variable	Description
<pf_host>	The host name or IP address of the PingFederate server.
<pf_port>	The port number for PingFederate.
<ConnectionId>	The federation identifier of the SP for the connection that uses the Apple IdP Adapter instance.

    3. Brand your link with a **Sign in with Apple** button. For instructions, see Sign in with Apple Buttons in the Apple Developer documentation.
2. If your application is inside the PingFederate domain, configure an adapter-to-adapter mapping.
    1. On the **System > Protocol Settings > Roles & Protocols** tab, select the **Enable Identity Provider (IdP) role and support for the following** and **Enable Service Provider (SP) role and support for the following** check boxes.
    2. In both the **Enable Identity Provider** and **Enable Service Provider** sections, select any protocol check box, such as **SAML 2.0**. Click **Save**.

        > **Note**
        >
        > PingFederate requires a protocol selection to activate the roles. The protocol that you select is not used for this integration.

    3. On the **Service Provider > Adapters** tab, create or select an adapter instance that is integrated with the application as shown in SP application integration settings in the PingFederate documentation.
    4. On the **Identity Provider > Adapter-to-Adapter Mappings** tab, configure the IdP-to-SP adapter mapping as shown in Adapter-to-adapter mappings in the PingFederate documentation.
    5. In your web application, create a hyperlink to allow users to sign on to the SP application. Use the following URL and replace the variables based on the descriptions in the table below:

        https://*pf_host*:*pf_port*/pf/adapter2adapter.ping?IdpAdapterId=*IdpAdapterId*&SpSessionAuthnAdapterId=*SpAdapterId*

Copyright © 2025 Ping Identity Corporation

Variable	Description
<pf_host>	The host name or IP address of the PingFederate server.
<pf_port>	The port number for PingFederate.
<IdpAdapterId>	The instance ID of the Apple IdP Adapter instance.
<SpAdapterId>	The instance ID of the SP adapter instance that has been integrated with the application.

6. Brand your link with a **Sign in with Apple** button. For instructions, see Sign in with Apple Buttons in the Apple Developer documentation.

Troubleshooting

Enabling debug logging

To help with troubleshooting or monitoring, you can turn on activity logging for the adapter.

About this task

These steps are optional. You can find general information about logging in Enabling debug messages and console logging in the PingFederate documentation.

Steps

1. Open the `<pf_install>/pingfederate/server/default/conf/log4j2.xml` file for editing.
2. If you want to log activity for PingFederate and all adapters:
    1. Find the following section.

```
<AsyncRoot level="INFO" includeLocation="false">
        <!-- <AppenderRef ref="CONSOLE" /> -->
        <AppenderRef ref="FILE" />
</AsyncRoot>
```

    2. Change `INFO` to `DEBUG`.

```
<AsyncRoot level="DEBUG" includeLocation="false">
        <!-- <AppenderRef ref="CONSOLE" /> -->
        <AppenderRef ref="FILE" />
</AsyncRoot>
```

    3. If you want to see the adapter activity in the console, remove the comment tags.

Copyright © 2025 Ping Identity Corporation

```
<AsyncRoot level="INFO" includeLocation="false">
        <AppenderRef ref="CONSOLE" />
        <AppenderRef ref="FILE" />
</AsyncRoot>
```

3. If you want to log activity just for the Apple IdP Adapter, add the following line.

```
<Logger name="com.pingidentity.adapter.idp.apple" level="DEBUG"/>
```

4. Save the file.

Release notes

Changelog

The following is the change history for the Apple Login Integration Kit.

Apple Cloud Identity Connector 1.1.2 – February 2024

- Fixed an issue that caused `email_verified` data to be handled incorrectly.

Apple Cloud Identity Connector 1.1.1 – September 2021

- Fixed an issue that could cause the incorrect first and last name to be returned when a user signed on with Apple.

Apple Cloud Identity Connector 1.1 – November 2020

- Added a setting to use browser redirect or pop-up window for the sign-on presentation.
- Added customizable sign-on templates for the pop-up window presentation.
- Added customizable user-facing language-pack messages.
- Added support for the PingFederate authentication API.
- Added support for the JavaScript Widget for the PingFederate Authentication API.

Apple Cloud Identity Connector 1.0.1 – December 2019

- Added a workflow to serve the Apple domain verification file directly through PingFederate. (Note: Domain verification is no longer needed as of July 2020, and has been removed from the documentation.)

Apple Cloud Identity Connector 1.0 – November 2019

- Initial release.
- Added support for social login using Apple credentials.
- Added support for retrieving Apple user information, email, and name.

Copyright © 2025 Ping Identity Corporation

- Added the ability to override Apple API endpoints.
- Added the ability to customize the error redirect URL.
- Added the ability to configure when to retry a failed request.
- Added the ability to configure the maximum amount of retries to perform.
- Added the ability to override the system-default proxy settings.
- Added the ability to configure API connection timeout settings.

Known issues and limitations

The following are known issues or limitations for the Apple Cloud Identity Connector.

Known issues

There are no known issues.

Known limitations

- Sign in with Apple ID does not create a session in the browser, so sign in is required every time and there is no single logout function.
- Apple handles scope requests in a unique way:
    - The `email` scope behaves as follows:
        - If you request this scope the first time a user signs on, Apple will return the email address. If you then request this scope at a later time, Apple will return the email address.
        - If you do not request this scope the first time a user signs on, Apple will not return the email address. If you then request this scope at a later time, Apple will not return the email address.
    - The `name` scope behaves as follows:
        - If you request this scope the first time a user signs on, Apple will return the name. If you then request this scope at a later time, Apple will not return the name.
        - If you do not request this scope the first time a user signs on, Apple will not return the name. If you then request this scope at a later time, Apple will not return the name.
    - If you need these attributes, have your application capture and store the values.
    - To reset these scenarios, the user has to manually revoke permissions from your application and then sign in again.

Download manifest

The following files are included in the Apple Login Integration Kit `.zip` archive:

- `legal/Legal.pdf` : Copyright and license information.

Copyright © 2025 Ping Identity Corporation

- `dist/pingfederate/server/default` : Contains the integration files
  - `deploy` : Contains the Java libraries
    - `pf-apple-idp-adapter-<version>.jar` : A `.jar` file that contains the Apple IdP Adapter.
  - `conf` : Contains the HTML template that presents the Apple sign-on form.
    - `language-packs/apple-messages.properties` : A variable file that provides adapter-specific template messages
      - `pingfederate-apple-adapter-messages.properties` : A variable file that customizes the messages that appear on the template files.
    - `template` : Contains user-facing HTML template files
      - `apple-pop-up-template.html` : A template that opens a pop-up window to prompt the user to sign on.
      - `apple-post-auth-template.html` : A template that returns the user to the PingFederate sign-on flow after they sign on with Apple.
      - `assets` – contains functional scripts and files used by the template
        - `css/apple.css` : A CSS file that customizes the appearance of the template files.
        - `fonts/end-user` : Contains template fonts and icons
          - `icons` : Contains template icons
        - `images` : Contains template image files
          - `ping-logo.svg` : An image file with company branding
  - `lib/pf-authn-api-sdk-<version>.jar` : A `.jar` file that contains the PingFederate Authentication API SDK

Copyright © 2025 Ping Identity Corporation

Aquera Provisioner

The Aquera Provisioner allows PingFederate to integrate with any application supported by Aquera for user provisioning and single sign-on (SSO).

Features

- Manages users in the target service based on changes in an external data store that is attached to PingFederate.
  - Creates, updates, disables, and deletes users.
  - Allows you to enable the create, update, disable, and delete capabilities independently.
  - Allows you to choose whether to disable or delete users when deprovisioning.
  - Allows you to provision disabled users.
- Manages groups in the target service based on changes in an external data store that is attached to PingFederate.
  - Creates, updates, and deletes groups.
  - Updates group memberships.
- Enables browser-based SSO initiated by the service provider (SP) or identity provider (IdP).

The Aquera Provisioner implements the official SCIM specifications provided from simplecloud.info. The following table provides a brief summary.

Feature	Outbound provisioning
SCIM specification	Step 2.0
Data format	JSON
User and group CRUD operations	Yes
Custom schema support	Yes. The connector retrieves the target service schema from Aquera.
Filtering support	Users: Yes Groups: Yes, if the target service supports it. Otherwise, the connector attempts to get all groups and find a match.
PATCH	Users: No Groups: Yes
Authentication method	Basic authentication and bearer token authentication
Source data stores	Active Directory and other LDAPv3-compliant directory servers

Attribute support varies by target service. The Aquera Provisioner dynamically gets attributes from Aquera for each target service.

Copyright © 2025 Ping Identity Corporation

Intended audience

This document is intended for PingFederate administrators.

Learn more about the setup process from the following resources:

- PingFederate documentation:
  - SP connection management
  - Identity provider SSO configuration
  - Managing digital signing certificates and decryption keys
  - Datastores
  - Configuring outbound provisioning
  - Configuring outbound provisioning settings
- Aquera documentation:
  - Aquera User Guide
  - Connectors documentation for the target service

System requirements

- PingFederate 9.0 or later.
- To allow PingFederate to make outbound connections, you might need to allow the following Aquera endpoint (use your tenant ID):
  - http://api.aquera.com/Tenants/:*tenantid*

User and group management

The Aquera Provisioner synchronizes users and groups from your datastore to the target service. The behavior of each provisioning capability is described below.

Learn more about configuring these capabilities in the Creating a provisioning connection step of the setup process.

Synchronizing existing users

PingFederate synchronizes users based on the `userName` attribute in the target service. If a user already exists in your datastore and the target service, mapping this attribute correctly links the two records together.

For example:

- In the target service, Janet's `userName` is `jsmith`.
- In your datastore, Janet's `sAMAccountName` is `jsmith`.

Copyright © 2025 Ping Identity Corporation

- On the **Attribute Mapping** tab of your provisioning connection configuration, you map the `userName` attribute to `sAMAccountName`.
- When the provisioning connector runs, the datastore user is provisioned with a `userName` of `jsmith`. That matches Janet's existing `userName` in the target service, so her information in the datastore is synchronized to her the target service account.

User provisioning

PingFederate provisions users when one of the following happens:

- A user is added to the datastore group or filter that is targeted by the provisioning connector.
- A user with "disabled" status is added to the datastore group or filter that is targeted by the provisioning connector, and the **Provision disabled users** provisioning option is enabled.

The **Source Location** tab of your provisioning connection configuration defines which users PingFederate targets for provisioning.

User updates

PingFederate updates users when a user attribute changes in your datastore.

The **Attribute Mapping** tab of your provisioning connection configuration defines which attributes PingFederate monitors for changes.

User deprovisioning

PingFederate deprovisions users when one of the following happens:

- A user is deleted from the user store.
- A user is disabled in the user store.
- A user is removed from the datastore group or filter that is targeted by the provisioning connector.

The **Remove User Action** setting in your provisioning connection configuration defines whether PingFederate disables or deletes the user.

Some Aquera apps also include a **deleteOnDeactivation** setting on the Aquera console.

Synchronizing existing groups

PingFederate synchronizes groups from the datastore to the target service based on the group name.

For example:

- In the target service, there is a group is named `Accounting`.
- In your datastore, there is a group with a `CN` of `Accounting`.
- When the provisioning connector runs, the two groups are synchronized.

Copyright © 2025 Ping Identity Corporation

Group provisioning

PingFederate provisions groups when a group is added to the datastore filter that is targeted by the provisioning connector.

The **Source Location** tab in your provisioning connection configuration defines which groups PingFederate targets for provisioning and monitors for changes.

Some Aquera apps do not support group creation. In this case, create groups manually in the target service. The connector will then be able to update the groups.

Group name updates

PingFederate renames groups when they are renamed in the datastore.

Group membership updates

PingFederate updates group memberships when memberships change in the datastore, whether the change is in the group's properties or a user's properties.

Group memberships in the datastore overwrite the group memberships in the target service.

Group deletion

PingFederate deletes groups when any of the following happen:

- The group is deleted in the datastore.
- The group is removed from the datastore group or filter that is targeted by the provisioning connector.

Group deletions are permanent and cannot be undone.

Some Aquera apps do not support group deletion. In this case, delete groups manually in the target service.

Setup

Upgrading an existing deployment

If you're upgrading from a previous version of the Aquera Provisioner, note your existing service provider (SP) connection configuration and create a new connection.

Upgrading an existing deployment in PingFederate 10.1 or later

Steps

1. Back up your current PingFederate configuration as shown in Configuration archive in the PingFederate documentation.
2. In the PingFederate administrative console, go to **Applications > Integration > SP Connections** and select your connection.
3. Note the attribute mappings for your existing SP connection.

Copyright © 2025 Ping Identity Corporation

Learn more in Mapping attributes in the PingFederate documentation.

4. Delete your existing SP connection.
    1. Go to **Applications > Integration > SP Connections**.
    2. For your existing connection, click **Select action**, and then click **Delete**. Click **Confirm.**
5. Complete the steps in Deploying the integration files.
6. Complete the steps in Creating a provisioning connection.
    - From **Outbound Provisioning > Manage Channels > Channel**, on the **Attribute Mapping** tab, configure the attribute mappings based on your notes.
7. (Optional) Complete the steps in Adding SSO to a connection.

Upgrading an existing deployment in PingFederate 10.0 or earlier

Steps

1. Back up your current PingFederate configuration as shown in Configuration archive in the PingFederate documentation.
2. In the PingFederate administrative console, go to **Identity Provider > SP Connections** and select your connection.
3. Note the attribute mappings for your existing SP connection.

   Learn more in Mapping attributes in the PingFederate documentation.
4. Delete your existing SP connection.
    1. Go to **Identity Provider > SP Connections > Manage All**.
    2. For your existing connection, click **Select action**, and then click **Delete**. Click **Save.**
5. Complete the steps in Deploying the integration files.
6. Complete the steps in Creating a provisioning connection.
    - From **Outbound Provisioning > Manage Channels > Channel**, on the **Attribute Mapping** tab, configure the attribute mappings based on your notes.
7. (Optional) Complete the steps in Adding SSO to a connection.

Deploying the integration files

To get started with the integration, deploy the Aquera Provisioner files to your PingFederate directory.

Steps

1. Download the Aquera Provisioner `.zip` archive from the Ping Identity Integration Directory.
2. Stop PingFederate.
3. If you're upgrading an existing deployment, delete the `pf-aquera-quickconnection-<version>.jar` file from your `<pf_install>/pingfederate/server/default/deploy` directory.

Copyright © 2025 Ping Identity Corporation

4. Extract the `.zip` archive and merge the contents of the `dist` directory with your `<pf_install>/pingfederate/server/default/deploy` directory.
5. Enable the PingFederate provisioning engine:
    1. Open your `<pf_install>/pingfederate/bin/run.properties` file.
    2. Change `pf.provisioner.mode` to `STANDALONE`. Save the file.

    > **Note**
    >
    > Learn more about configuring the `FAILOVER` mode instead in Deploying provisioning failover in the PingFederate Server Clustering Guide.

6. Start PingFederate.
7. If you operate PingFederate in a cluster, repeat steps 2 - 4 and step 6 for each engine node.

Enabling single sign-on in PingFederate

To use PingFederate for provisioning and single sign-on, configure an external datastore and set a SAML entity ID.

About this task

Your external datastore acts as the source of data for provisioning. PingFederate also uses an internal datastore to store the state of synchronization between the source datastore and the target datastore.

Learn more in Datastores and Configuring outbound provisioning settings in the PingFederate documentation.

Enabling provisioning and single sign-on in PingFederate 10.1 or later

Steps

1. In the PingFederate administrative console, configure the datastore for PingFederate to use as the source of user data.

    Learn more in Datastores in the PingFederate documentation.
2. Go to **System > Server > Protocol Settings > Federation Info**.
3. In the **SAML 2.0 Entity ID** field, enter a name for PingFederate to use when identifying itself to SAML partners.
4. On the **Outbound Provisioning** tab, in the **Provisioner Data Store** list, select the internal database that will store the synchronization state. Click **Save**.

    Learn more in Configuring outbound provisioning settings in the PingFederate documentation.

Enabling provisioning and single sign-on in PingFederate 10.0 or earlier

Steps

1. In the PingFederate administrative console, configure the datastore for PingFederate to use as the source of user data.

    For help, see Datastores in the PingFederate documentation.

Copyright © 2025 Ping Identity Corporation

2. Go to **System > Protocol Settings > Roles & Protocols**.
3. Select the **Enable Identity Provider IdP Role and Support the Following** checkbox.
4. Select the **SAML 2.0** and **Outbound Provisioning** checkboxes. Click **Next**.
5. Click the **Federation Info** tab.
6. In the **SAML 2.0 Entity ID** field, enter a name for PingFederate to use to identify itself to SAML partners.
7. On the **Outbound Provisioning** tab, in the **Provisioner Data Store** list, select the internal database that will store the synchronization state. Click **Save**.

   Learn more in Configuring outbound provisioning settings in the PingFederate documentation.

Add an application in Aquera

Configure the target service as an application in Aquera and note the connection credentials.

Steps

1. Sign on to the Aquera Applications page.
2. Click **Add Application** and select the target service.
3. Configure the application. Learn more by searching for the the target service configuration steps in Connectors in the Aquera documentation.
4. On the **Applications** page, select the application that you added.
5. On the **Configure Application** window, from the **Copy this URL** field, note your SCIM URL.
6. Note your application credentials.

   Credentials can be a bearer token or a username and password pair. If both types of credentials are provided, note the bearer token.

   On the application details page, the credentials can appear in the following locations:

   - Right sidebar, in the **Authorization** section:
     - **Bearer**
   - Main window, in the **Credential To Access <*application name*>** section:
     - **Token**
     - **Username**, **Password**

     > **Tip**
     >
     > Aquera shows tokens and passwords as dots, but you can still copy the values.

Copyright © 2025 Ping Identity Corporation

Creating a provisioning connection

To allow PingFederate to manage users in the target service, create a service provider (SP) connection.

Steps

1. In the PingFederate administrator console, create a new SP connection:

   Choose from:

   - For PingFederate 10.1 or later: go to **Applications > Integration > SP Connections**. Click **Create Connection**.
   - For PingFederate 10.0 or earlier: go to **Identity Provider > SP Connections**. Click **Create Connection**.

2. Configure the basic connection details with the the target service quick connection template.
   1. On the **Connection Template** tab, select **Use a template for this connection**.
   2. From the **Connection Template** list, select **Aquera Provisioner**. Click **Next**.
   3. On the **Connection Type** tab select **Outbound Provisioning**. Click **Next**.
   4. On the **General Info** tab, in the **Connection Name** field, enter a name of your choosing. Click **Next**.
3. On the **Outbound Provisioning** tab, configure provisioning with the following details.

   Learn more in Configuring outbound provisioning in the PingFederate documentation.
   1. On the **Target** page, enter the URL and authentication details that you noted in Add an application in Aquera.

      > **Note**
      >
      > PingFederate verifies the authentication details when you activate the channel and SP connection.

   2. Under **Provisioning Options**, customize the provisioning connector behavior by referring to Aquera Connector settings reference. Click **Next**.
   3. On the **Manage Channels > Attribute Mapping** tab, at the end of the page, click **Refresh Fields**. Complete the required (`*`) attribute mappings and any optional mappings that you want.

      The **Refresh Fields** button causes PingFederate to get the attribute schema from the target service. The schema can include custom attributes (marked with `IANUser`) and standard SCIM attributes. Learn more about standard SCIM attributes in Supported attributes reference.

      Learn about mapping attributes in Managing channels in the PingFederate documentation.
4. On the **Activation and Summary** tab, above the **Summary** section, click the toggle to turn on the connection. Click **Save**.

Aquera Connector settings reference

Lists the configuration settings and provisioning options for the Aquera Provisioner.

Copyright © 2025 Ping Identity Corporation

Field Name	Description
SCIM URL	The SCIM URL that you noted in Add an application in Aquera.
Bearer Token Authentication	
Access Token	The bearer token for the target service. If you noted a bearer token in Add an application in Aquera, enter it here.
Basic Authentication	
Basic Authentication Username	The username of the administrator account on the target service. If you noted a username in Add an application in Aquera, enter it here.
Basic Authentication Password	The password of the administrator account on the target service. If you noted password in Add an application in Aquera, enter it here.
Provisioning Options **Note** Some Aquera apps do not support all of the following provisioning options. Check the documentation for the target service.	
User Create	***Selected (default)*** PingFederate creates users in the target service. ***Cleared*** PingFederate does not create users in the target service.
User Update	***Selected (default)*** PingFederate updates existing users in the target service. ***Cleared*** PingFederate does not update existing users in the target service.

Copyright © 2025 Ping Identity Corporation

Field Name	Description
User Disable / Delete	***Selected (default)*** PingFederate removes users from the target service according to the **Remove User Action** setting. **Note** You might need to enable **User Update** for this to work with some services. ***Cleared*** PingFederate does not remove users from the target service.
Provision Disabled Users	***Selected*** PingFederate creates users in the target service with a "disabled" status. ***Cleared (default)*** If a user has a "disabled" status, PingFederate does not create the user in the target service.

Note
If any of the above provisioning options are cleared, PingFederate logs a warning in the user workflow section of `provisioner.log` when the related action fails.

Copyright © 2025 Ping Identity Corporation

Field Name	Description
Remove User Action	This option applies when **User Disable / Delete** is selected, and either: • a previously-provisioned user no longer meets the condition set on the **Source Location** tab, or • a user has been disabled or deleted from the data store. ***Disable (default)*** PingFederate disables the user in the target service. ***Delete*** PingFederate deletes the user from the target service. **Note** Some target applications do not support hard deleting users through external interfaces. For those services, users are disabled. **Note** Some Aquera apps also include a **deleteOnDeactivation** setting on the Aquera console.

Supported attributes reference

The following list describes standard SCIM attributes. Attribute support varies by target service. The Aquera Provisioner dynamically gets attributes from Aquera for each target service.

Learn more about SCIM attributes in the SCIM specification and the target service documentation.

Attribute	Description
`userName`	A service provider's unique identifier for the user. This attribute is required.
`givenName`	The given name of the user, or first name in most Western languages (for example, 'Barbara' given the full name 'Ms. Barbara Jane Jensen, III').
`familyName`	The family name of the user, or last name in most Western languages (for example, 'Jensen' given the full name 'Ms. Barbara Jane Jensen, III').
`middleName`	The middle name(s) of the user (for example, "Jane" given the full name "Ms. Barbara Jane Jensen, III").

Copyright © 2025 Ping Identity Corporation

Attribute	Description
`honorificPrefix`	The honorific prefix(es) of the user, or title in most Western languages (for example, "Ms." given the full name "Ms. Barbara Jane Jensen, III").
`honorificSuffix`	The honorific suffix(es) of the user, or suffix in most Western languages (for example, "III" given the full name "Ms. Barbara Jane Jensen, III").
`formattedName`	The full name, including all middle names, titles, and suffixes as appropriate, formatted for display (for example, "Ms. Barbara Jane Jensen, III").
`workEmail`	Work email for the user (for example, "bjensen@example.com").
`displayName`	The name of the user, suitable for display to end-users.
`title`	The user's title, such as "Vice President".
`externalId`	A string that is an identifier for the resource as defined by the provisioning client.
`password`	This attribute is intended to be used as a means to set, replace, or compare (for example, filter for equality) a password.
`preferredLanguage`	Indicates the user's preferred written or spoken languages and is generally used for selecting a localized user interface.
`userType`	Used to identify the relationship between the organization and the user. Typical values used might be "Contractor", "Employee", "Intern", "Temp", "External", and "Unknown", but any value may be used.
`locale`	Used to indicate the user's default location for purposes of localizing such items as currency, date time format, or numerical representations.
`nickName`	The casual way to address the user in real life. For example,"Bob" or "Bobby" instead of "Robert".
`profileUrl`	A URI that is a uniform resource locator that points to a location representing the user's online profile (for example, a web page).

Copyright © 2025 Ping Identity Corporation

Attribute	Description
`profilePhotoUrl`	A URI that is a uniform resource locator that points to the user's profile photo. The resource MUST be a file (for example, a GIF, JPEG, or PNG image file) rather than a web page containing an image.
`profileThumbnailUrl`	A URI that is a uniform resource locator that points to the user's profile thumbnail. The resource MUST be a file (for example, a GIF, JPEG, or PNG image file) rather than a web page containing an image.
`timezone`	The user's time zone, in IANA Time Zone database format (for example, "America/Los_Angeles").
`workPhone`	The work phone number for the user (for example, "+1-201-555-0123").
`mobilePhone`	The mobile phone number for the user (for example, "+1-201-555-0123").
`pagerPhone`	The pager number for the user (for example, "+1-201-555-0123").
`faxPhone`	The fax number for the user (for example, "+1-201-555-0123").
`homePhone`	The home phone number for the user (for example, "+1-201-555-0123").
`otherPhone`	Another phone number that can be used to reach the user (for example, "+1-201-555-0123").
`workStreetAddress`	The work street address for the user, which may include house number, street name, P.O. box, and multi-line extended street address information.
`workCity`	The work city or locality component for the user's mailing address.
`workState`	The work state or region component for the user's mailing address.
`workPostalCode`	The work ZIP or postal code component for the user's mailing address.
`workCountry`	The work country component for the user's mailing address. When specified, the value MUST be in ISO 3166-1 "alpha-2" code format ISO3166; for example, the United States and Sweden are "US" and "SE", respectively.

Copyright © 2025 Ping Identity Corporation

Attribute	Description
`workFormattedAddress`	The user's full work address, formatted for display.
`homeStreetAddress`	The home street address for the user, which may include house number, street name, P.O. box, and multi-line extended street address information.
`homeCity`	The home city or locality component for the user's mailing address.
`homeState`	The home state or region component for the user's mailing address.
`homePostalCode`	The home ZIP or postal code component for the user's mailing address.
`homeCountry`	The home country component for the user's mailing address. When specified, the value MUST be in ISO 3166-1 "alpha-2" code format ISO3166; for example, the United States and Sweden are "US" and "SE", respectively.
`homeFormattedAddress`	The user's full home address, formatted for display.
`otherStreetAddress`	An alternate street address for the user, which may include house number, street name, P.O. box, and multi-line extended street address information.
`otherCity`	The alternate city or locality component for the user's mailing address.
`otherState`	The alternate state or region component for the user's mailing address.
`otherPostalCode`	The alternate ZIP or postal code component for the user's mailing address.
`otherCountry`	The alternate country component for the user's mailing address. When specified, the value MUST be in ISO 3166-1 "alpha-2" code format ISO3166; for example, the United States and Sweden are "US" and "SE", respectively.
`otherFormattedAddress`	The alternate address for the user, formatted for display.
`qqIm`	The QQ instant messaging address for the user.
`skypeIm`	The Skype instant messaging address for the user.
`gtalkIm`	The Google Talk instant messaging address for the user.

Copyright © 2025 Ping Identity Corporation

Attribute	Description
`aimIm`	The AOL Instant Messenger instant messaging address for the user.
`icqIm`	The ICQ instant messaging address for the user.
`yahooIm`	The Yahoo Messenger instant messaging address for the user.
`msnIm`	The MSN Messenger instant messaging address for the user.
`xmppIm`	The XMPP instant messaging address for the user.
`entitlements`	A list of entitlements for the user that represent a thing the user has. An entitlement may be an additional right to a thing, object, or service.
`roles`	A list of roles for the user that collectively represent who the user is. For example, "Student", "Faculty".
`certificates`	A list of certificates associated with the resource (for example, a user). Each value contains exactly one DER-encoded X.509 certificate (see Section 4 of RFC5280), which MUST be base64 encoded per Section 4 of [RFC4648.
`employeeNumber`	A string identifier, typically numeric or alphanumeric, assigned to a person, often based on order of hire or association with an organization.
`costCenter`	The cost center for the user.
`organization`	The organization for the user.
`division`	The division for the user.
`department`	The department for the user.

Configuring single sign-on

To allow PingFederate to act as an identity provider for the target service, modify your connection to include single sign-on (SSO).

About this task

> **Note**
>
> SSO is not required for provisioning. If you only want to use the Aquera Provisioner for provisioning, skip these steps.

Copyright © 2025 Ping Identity Corporation

Steps

1. Complete the steps in Adding SSO to a connection.
2. On the target service, configure single sign-on by following the documentation provided by the service. When prompted for information about PingFederate, see PingFederate SSO details for the service provider.

Adding SSO to a connection

To enable single sign-on (SSO), modify the provisioning connection that you created in Creating a provisioning connection.

Steps

1. On the PingFederate administrator console, open your existing SP connection.

   Choose from:

   - For PingFederate 10.1 or later: go to **Applications > Integration > SP Connections**. Select your connection.
   - For PingFederate 10.0 or earlier: go to **Identity Provider > SP Connections**. Select your connection.

2. On the **Connection Type** tab select **Browser SSO Profiles**. Click **Next**.
3. On the **Browser SSO** tab, configure your SSO settings.

   For specific connection details, consult the documentation and admin console provided by the target service.

   Learn more in Configuring IdP Browser SSO in the PingFederate documentation.
4. On the **Activation and Summary** tab, above the **Summary** section, click the toggle to turn on the connection. Click **Save**.

PingFederate SSO details for the service provider

When enabling single sign-on (SSO) in the target service, you will require some or all of the following information from PingFederate.

Metadata file

Some target services allow you to import a SAML metadata file that contains some of the information below. Learn more about exporting your metadata file in Metadata export in the PingFederate documentation.

SAML endpoint

The PingFederate SAML endpoint is:

```
https://<pf_hostname>:<pf_port>/idp/SSO.saml2
```

Copyright © 2025 Ping Identity Corporation

Identity provider issuer

This is SAML 2.0 Entity ID from PingFederate, which can be found under the **Server Settings** page. Learn more in Specifying federation information.

> **Tip**
>
> In order to override SAML 2.0 Entity ID on the **Server Settings** page for your SP Connection, navigate to the General Info screen to add a Virtual Server ID. This value will be sent as the SAML Issuer URL.

Signing certificate

This is the public signing certificate that PingFederate uses to sign the SAML assertion. Learn more about exporting your certificate in Managing digital signing certificates and decryption keys.

Release notes

Changelog

The following is the change history for the Aquera Provisioner.

Aquera Provisioner 1.0 – December 2020

- Initial release.
- Included support for user provisioning.
- Included support for group provisioning.
- Included support for adding users to groups.
- Included support for SCIM core and enterprise attributes.
- Included support for dynamic attribute retrieval of custom attributes.
- Included support for bearer token and HTTP basic authentication.
- Included configuration options for CRUD capabilities.
- Included configuration options for provisioning disabled users.
- Included configuration options for deprovisioning actions.

Known issues and limitations

The following are known issues or limitations with the Aquera Provisioner.

Known issues

There are no known issues.

Copyright © 2025 Ping Identity Corporation

Known limitations

SP connections

- The Unique User Identifier cannot be changed in an SP connection configuration. To change to a different Unique User Identifier, delete the existing connection, restart PingFederate, and then create a connection with the new Unique User Identitier.
- All SP connections with the same target must use the same Unique User Identifier. If multiple SP connections are created for the same target, every subsequent connection will use the Unique User Identifier configured in the first connection that was created.

Attributes

- When mapping attributes, Aquera can return invalid schemas for some services. This can cause unexpected behavior. If you experience issues, confirm the correct schema in the documentation for the target service.
- If the target service does not specify type or primary information on multi-value attributes (email, phone, address), unexpected behavior can occur. During an update, existing attributes on the SaaS may not be removed, and the desired value may not be correctly set as primary.
- The connector cannot clear a user attribute once it has been set.

Other

- This connector does not support PATCH user updates.
- When an LDAP user is deleted in a targeted group distinguished name (DN), the provisioning connector does not propagate the deletion until a new user is added to the group. This limitation is compounded when the **User Create** provisioning option is disabled. Learn more in SaaS provisioner does not remove the user in the Knowledge Base.
- If the target service interprets or implements the SCIM standards differently, it can result in unexpected behavior.

Download manifest

The following files are included in the Aquera Provisioner `.zip` archive:

- `ReadMeFirst.pdf` : Contains links to this online documentation
- `Legal.pdf` : Copyright and license information
- `dist` : Contains the integration files
  - `deploy/pf-aquera-quickconnection-<version>.jar` : The Aquera Provisioner quick connection template.

Copyright © 2025 Ping Identity Corporation

Atlassian integrations

Atlassian Cloud SSO configuration guide

To integrate PingFederate with Atlassian Cloud for single sign-on (SSO), you don't need to download anything. For setup instructions, see Configuring SAML SSO with Atlassian Cloud and PingFederate.

Atlassian Cloud Connector

The Atlassian Connector allows PingFederate to integrate with Atlassian Cloud for user and group provisioning as well as SSO.

Atlassian Integration Kit

The Atlassian Integration Kit allows PingFederate to coordinate SSO for on-premises deployments of Jira and Confluence.

Atlassian Cloud Provisioner

The Atlassian Cloud Provisioner allows PingFederate to integrate with Atlassian Cloud for user and group provisioning and single sign-on (SSO).

Features

- Manages users in Atlassian Cloud based on changes in an external data store that is attached to PingFederate.
  - Creates, updates, disable, and delete users.
  - Allows you to enable the create, update, disable, and delete capabilities independently.
  - Allows you to choose to disable or delete users when deprovisioning.
  - Allows you to provision disabled users.
  - Creates and deletes groups.
  - Updates group memberships.
- Browser-based single sign-on (SSO) initiated by the service provider (SP) or identity provider (IdP).
- Pre-populates some connection settings with the included quick connection template and SAML metadata file.

Intended audience

This document is intended for PingFederate administrators. Before you start, familiarize yourself with:

- The following sections of the Atlassian Cloud documentation:
  - Atlassian organizations
  - User provisioning
  - SAML single sign-on

Copyright © 2025 Ping Identity Corporation

- The following sections of the PingFederate documentation:
  - Identity provider SSO configuration
  - Datastores
  - Managing IdP adapters
  - Managing digital signing certificates and decryption keys
  - SP connection management
  - Configuring outbound provisioning
  - Configuring outbound provisioning settings

System requirements

- PingFederate 11.3 or later.

> **Note**
>
> To allow PingFederate to make outbound connections to the Atlassian Cloud API, you might need to add the following domains to the allow list for your firewall:
> https://api.atlassian.com

- An Atlassian Cloud administrator account.
- An Atlassian Access subscription.

User and group management

The Atlassian Cloud Provisioner links users and groups from the datastore to Atlassian Cloud. The behavior of each provisioning capability is described below.

Synchronizing existing users

By default, the provisioning connector synchronizes based on the `userName` attribute in Atlassian Cloud. You can change the default mapping in Creating a provisioning connection. For example:

- In Atlassian Cloud, Janet's `userName` is `jsmith`.
- In your datastore, Janet's `sAMAccountName` is `jsmith`.
- On the **Attribute Mapping** tab of your PingFederate channel configuration, you map the `userName` attribute to `sAMAccountName`.
- When the provisioning connector runs, the datastore user is provisioned with a `userName` of `jsmith`. That matches Janet's existing `userName` in Atlassian Cloud, so her information in the datastore is synchronized to her Atlassian Cloud account.

Copyright © 2025 Ping Identity Corporation

User provisioning

Triggered by either of the following:

- A user is added to the datastore group or filter targeted by the provisioning connector.
- A user with "disabled" status is added to the datastore group or filter targeted by the provisioning connector, and the **Provision disabled users** provisioning option is enabled.

The target is determined by the **Source Location** tab in the provisioning connector configuration.

User updates

Triggered when a change occurs to a user attribute that is mapped in the provisioning connector configuration, such as an attribute change or the user being re-enabled from a disabled status.

User deprovisioning

Triggered by any of the following:

- A user is deleted from the user store.
- A user is disabled in the user store.
- A user is removed from the datastore group or filter targeted by the provisioning connector.

The **Remove User Action** setting in the connection configuration determines whether the deprovisioning action disables or deletes the user.

Synchronizing existing groups

The provisioning connector synchronizes groups from the datastore to Atlassian Cloud based on the group name. For example:

- In Atlassian Cloud, there is a group is named `Accounting`.
- In your data store, there is a group with a `CN` of `Accounting`.
- When the provisioning connector runs, the two groups are synchronized.

Group provisioning

Triggered when a group is added to the datastore filter that the provisioning connector targets.

The target is determined by the **Source Location** tab in the provisioning connector configuration.

Group name updates

Renaming the group in the datastore triggers PingFederate to rename the group in Atlassian Cloud.

Group membership updates

Changing group memberships through the group's properties or a user's properties triggers PingFederate to update the group membership in Atlassian Cloud.

Group memberships in the datastore overwrite the group memberships in Atlassian Cloud.

Copyright © 2025 Ping Identity Corporation

Group deletion

Triggered by any of the following:

- Deleting the group in the datastore will trigger PingFederate to delete the group in Atlassian Cloud. Group deletions are permanent and cannot be undone.
- The group is removed from the datastore group or filter targeted by the provisioning connector.

Setup

Upgrading an existing deployment

If you're upgrading from a previous version of the Atlassian Cloud Provisioner, note your existing service provider (SP) connection configuration and create a new connection.

Upgrading an existing deployment in PingFederate 10.1 or later

Steps

1. Back up your current PingFederate configuration as shown in Configuration archive in the PingFederate documentation.
2. In the PingFederate administrative console, go to **Applications > Integration > SP Connections** and select your connection.
3. Note the attribute mappings for your existing SP connection.

   Learn more in Mapping attributes in the PingFederate documentation.
4. Delete your existing SP connection.
    1. Go to **Applications > Integration > SP Connections**.
    2. For your existing connection, click **Select action**, and then click **Delete**. Click **Confirm.**
5. Complete the steps in Deploying the integration files.
6. Complete the steps in Creating a provisioning connection.
    - From **Outbound Provisioning > Manage Channels > Channel**, on the **Attribute Mapping** tab, configure the attribute mappings based on your notes.
7. (Optional) Complete the steps in Creating a single sign-on connection.

Upgrading an existing deployment in PingFederate 10.0 or earlier

Steps

1. Back up your current PingFederate configuration as shown in Configuration archive in the PingFederate documentation.
2. In the PingFederate administrative console, go to **Identity Provider > SP Connections** and select your connection.

Copyright © 2025 Ping Identity Corporation

3. Note the attribute mappings for your existing SP connection.

   Learn more in Mapping attributes in the PingFederate documentation.

4. Delete your existing SP connection.

   1. Go to **Identity Provider > SP Connections > Manage All**.
   2. For your existing connection, click **Select action**, and then click **Delete**. Click **Save.**

5. Complete the steps in Deploying the integration files.
6. Complete the steps in Creating a provisioning connection.

   - From **Outbound Provisioning > Manage Channels > Channel**, on the **Attribute Mapping** tab, configure the attribute mappings based on your notes.

7. (Optional) Complete the steps in Creating a single sign-on connection.

Deploying the integration files

To get started with the integration, deploy the Atlassian Cloud Provisioner files to your PingFederate directory.

Steps

1. Download the Atlassian Cloud Provisioner `.zip` archive from the Ping Identity Integration Directory.
2. Stop PingFederate.
3. If you're upgrading an existing deployment, delete the `pf-atlassian-quickconnection-<version>.jar` file from your `<pf_install>/pingfederate/server/default/deploy` directory.
4. Extract the `.zip` archive and merge the contents of the `dist` directory with your `<pf_install>/pingfederate/server/default/deploy` directory.
5. Enable the PingFederate provisioning engine:

   1. Open your `<pf_install>/pingfederate/bin/run.properties` file.
   2. Change `pf.provisioner.mode` to `STANDALONE`. Save the file.

   > **Note**
   >
   > To configure the `FAILOVER` mode instead, see Deploying provisioning failover in the PingFederate Server Clustering Guide.

6. Start PingFederate.
7. If you operate PingFederate in a cluster, repeat steps 2-4 and step 6 for each engine node.

Getting an Atlassian API key

Create a user directory in Atlassian to generate API key for PingFederate.

Copyright © 2025 Ping Identity Corporation

Steps

1. Sign on to Atlassian as an administrator.
2. On the **Directory > User provisioning** page, click **Create a directory**.
3. On the **Create a directory** page, enter a name. Click **Create**.
4. On the **API key** page, note the **Directory base URL** and **API key** values. You'll use these again in Creating a provisioning connection. Click **Done**.
5. On the **Directory > User provisioning** page, on the **Product access** tab, click **Add a site**.
    1. Select the Jira or Confluence site that you want provisioned users to be able to access.
    2. Complete the on-screen instructions.

Creating a provisioning connection

To allow PingFederate to manage users in Atlassian Cloud, create a service provider (SP) connection.

Steps

1. In the PingFederate administrator console, configure the datastore that PingFederate will use as the source of user data.

   You can find specific instructions in Datastores in the PingFederate documentation.

    - When targeting users and groups for provisioning, exclude the user account that you will use to administer users in your connection to Slack. This prevents the PingFederate provisioning engine from interfering with the account that provisions users and groups.
2. Enable provisioning:
    1. On the **System > Protocol Settings > Roles & Protocols** tab, select **Enable Identity Provider IdP Role and Support the Following**.
    2. Select **Outbound Provisioning**. Click **Save**.
3. On the **Identity Provider** tab, in the **SP Connections** area, open an existing connection or create a new one as follows:
    1. Click **Create new**.
    2. On the **Connection Template** tab, select **Use a template for this connection**.
    3. In the **Connection Template** list, select **Atlassian Provisioner**.
    4. Click **Choose File**, select the `atlassian-saml-metadata.xml` file from the Atlassian Cloud Provisioner `.zip` archive, and then click **Open**. Click **Next**.
4. On the **Connection Type** tab, select **Outbound Provisioning** and clear any unwanted types. Click **Next**.
5. On the **General Info** tab, in the **Base URL** field, enter the **Directory base URL** that you noted in Getting an Atlassian API key. The rest of the connection information is populated by the metadata XML file. Click **Next**.

Copyright © 2025 Ping Identity Corporation

6. On the **Outbound Provisioning** tab, configure the provisioning target and channel as shown in Configuring outbound provisioning in the PingFederate documentation.
    1. Click **Configure Provisioning**.
    2. On the **Target** tab, in the **Directory base URL** field, enter the value that you noted in Getting an Atlassian API key.
    3. In the **API key** field, enter the value that you noted in Getting an Atlassian API key.

        > **Note**
        >
        > PingFederate verifies the token when you activate the channel and SP connection.

    4. Under **Provisioning Options**, customize the provisioning connector actions as shown in Provisioning options reference. Click **Next**.
    5. On the **Manage Channels** tab, create a channel as shown in Manage channels in the PingFederate documentation. Click **Done**.

        > **Note**
        >
        > For more information about the attributes available in your channel configuration, see Supported attributes reference.

    6. On the **Outbound Provisioning** tab, click **Next**.
7. On the **Activation and Summary** tab, above the **Summary** section, turn on the connection. Click **Save**.

Provisioning options reference

The following are setting descriptions for the Atlassian Cloud provisioning capabilities.

Field Name	Description
User Create	***Selected (default)*** PingFederate creates users in Atlassian Cloud. ***Cleared*** PingFederate does not create users in Atlassian Cloud.
User Update	***Selected (default)*** PingFederate updates existing users in Atlassian Cloud. ***Cleared*** PingFederate does not update existing users in Atlassian Cloud.

Copyright © 2025 Ping Identity Corporation

Field Name	Description
User Disable	***Selected (default)*** PingFederate disables users in Atlassian Cloud. ***Cleared*** PingFederate does not disable users in Atlassian Cloud.

> **Note**
> If any of the above options are cleared, PingFederate logs a warning in the user workflow section of the `provisioner.log` file when the related action fails.

Provision Disabled Users	***Selected*** PingFederate creates users in Atlassian Cloud with a "disabled" status. ***Cleared (default)*** If a user has a "disabled" status, PingFederate does not create the user in Atlassian Cloud.

Supported attributes reference

Lists the attributes that can be mapped for user provisioning to Atlassian Cloud.

Attribute	Description
`userName`	The user's unique identifier in Atlassian Cloud. This attribute cannot be updated. This attribute is required.
`workEmail`	The user's work email address, formatted as `bjenson@example.com`. PingFederate sets this as the user's primary email address. You can override this choice in Atlassian. This attribute is required.
`homeEmail`	The user's personal email address, formatted as `bjenson@example.com`.
`otherEmail`	The user's other email address, formatted as `bjenson@example.com`.
`givenName`	The given name of the user, or first name in most Western languages. For example, `Barbara`, given the full name Ms. Barbara Jane Jensen, III. This attribute cannot be updated.

Copyright © 2025 Ping Identity Corporation

Attribute	Description
`familyName`	The family name of the user, or last name in most Western languages. For example, `Jensen`, given the full name Ms. Barbara Jane Jensen, III. This attribute cannot be updated.
`honorificPrefix`	The honorific prefix or prefixes of the user, or title in most Western languages. For example, `Ms.`, given the full name Ms. Barbara Jane Jensen, III. This attribute cannot be updated.
`honorificSuffix`	The honorific suffix or suffixes of the user, or title in most Western languages. For example, `III`, given the full name Ms. Barbara Jane Jensen, III. This attribute cannot be updated.
`displayName`	The name of the user, suitable for display to end users. If this is empty, Atlassian combines the user's given and family names for the display name.
`title`	The user's title, such as `Vice President`.
`preferredLanguage`	Indicates the user's preferred written or spoken languages and is generally used for selecting a localized user interface.
`timezone`	The user's time zone, in IANA Time Zone database format. For example, `America/Los_Angeles`.
`homePhone`	The user's phone number, formatted as `+1-201-555-0123`. This attribute cannot be updated.
`mobilePhone`	The user's mobile phone number, formatted as `+1-201-555-0123`. This attribute cannot be updated.
`otherPhone`	The user's other phone number, formatted as `+1-201-555-0123`. This attribute cannot be updated.
`pagerPhone`	The user's pager number, formatted as `+1-201-555-0123`. This attribute cannot be updated.
`faxPhone`	The user's fax number, formatted as `+1-201-555-0123`. This attribute cannot be updated.
`organization`	The user's organization.
`department`	The user's department.

Configuring single sign-on

To allow PingFederate to act as an identity provider for Atlassian Cloud, enable single sign-on (SSO) in PingFederate and Atlassian Cloud, and create a connection.

Copyright © 2025 Ping Identity Corporation

About this task

> **Note**
>
> SSO integration is an optional part of this integration. If you only want to use the Atlassian Cloud Provisioner for provisioning, skip the following steps.

Steps

1. Complete the steps in Enabling single sign-on in PingFederate.
2. Complete the steps in Enabling single sign-on in Atlassian.
3. Complete the steps in Creating a single sign-on connection.

Enabling single sign-on in PingFederate

Before you can configure single sign-on in Atlassian Cloud, you need to set a SAML entity ID in PingFederate .

Enabling single sign-on in PingFederate 10.1 or later

Steps

1. On the PingFederate administrative console, go to **System > Protocol Settings > Federation Info**.
2. In the **SAML 2.0 Entity ID** field, enter a name for PingFederate to use to identify itself to SAML partners.
3. Click **Save**.

Enabling single sign-on in PingFederate 10.0 or earlier

Steps

1. On the PingFederate administrative console, go to **System > Protocol Settings > Roles & Protocols**.
2. Select the **Enable Identity Provider IdP Role and Support the Following** check box.
3. Select the **SAML 2.0** check box. Click **Next**.
4. Go to the **Federation Info** tab.
5. In the **SAML 2.0 Entity ID** field, enter a name for PingFederate to use to identify itself to SAML partners.
6. Click **Save**.

Copyright © 2025 Ping Identity Corporation

Enabling single sign-on in Atlassian

To allow PingFederate to manage authentication, enable single sign-on (SSO) in Atlassian Cloud.

Steps

1. Sign on to Atlassian as an administrator.
2. On the **Security > SAML single sign-on** page, click **Add SAML configuration**.
3. On the **Add SAML configuration** page, complete the SAML 2.0 details. Click **Save configuration.**
    1. In the **Identity provider Entity ID** field, enter the **SAML 2.0 Entity ID** that you created in Enabling single sign-on in PingFederate.
    2. In the **Identity provider SSO URL**, enter your PingFederate SAML 2.0 endpoint based on the following:

        `https://<pf_host>:<pf_port>/idp/SSO.saml2`
    3. In the **Public x509 certificate** field, enter the contents of your PingFederate signing certificate.

        To get your certificate, refer to the Exporting a certificate section of Managing digital signing certificates and decryption keys in the PingFederate documentation.
4. On the **SAML single sign-on** page, note the **SP Entity ID** and **SP Assertion Consumer Service URL**. You'll use these in Creating a single sign-on connection.

Creating a single sign-on connection

To allow PingFederate to act as an identity provider for Atlassian Cloud, create a service provider (SP) connection.

About this task

If you only want to use the Atlassian Cloud Provisioner for provisioning, skip these steps.

> **Tip**
>
> You can follow these steps to create a new SP connection, or you can modify an existing connection.

Steps

1. In the PingFederate administrator console, create a new SP connection:

Copyright © 2025 Ping Identity Corporation

Choose from:

- For PingFederate 10.1 or later: go to **Applications > Integration > SP Connections**. Click **Create Connection**.
- For PingFederate 10.0 or earlier: go to **Identity Provider > SP Connections**. Click **Create Connection**.

2. Configure an SP connection with the Atlassian Cloud quick connection template:
    1. On the **Connection Template** tab, select **Use a template for this connection**.
    2. From the **Connection Template** list, select **Atlassian Provisioner**. Click **Next**.
    3. On the **Metadata File** row, upload the `atlassian-saml-metadata.xml` file. Click **Next**.
    4. On the **Connection Type** tab, click **Next**.
    5. On the **General Info** tab, click **Next**.
3. On the **Connection Type** tab, select **Browser SSO Profiles** and clear any unwanted types. Click **Next**.
4. On the **General Info** tab, complete the following fields. The rest of the connection information is populated by the metadata XML file. Click **Next**.
    1. In the **Partner's Entity ID** field, enter the **SP Entity ID** that you noted in Enabling single sign-on in Atlassian.
    2. In the **Base URL** field, enter the base URL that you noted in Getting an Atlassian API key.
5. On the **Browser SSO** tab, configure browser SSO:

    You can find a complete guide in Configure IdP Browser SSO in the PingFederate documentation.

    1. On the **Browser SSO > SAML Profiles** tab, select the **IdP-Initiated** and **SP-Initiated** checkboxes.
    2. On the **Browser SSO > Protocol Settings > Assertion Consumer Service** tab, in the **Binding** list, select **POST**. In the **Endpoint URL** field, enter the **SP Assertion Consumer Service URL** that you noted in Enabling single sign-on in Atlassian. Click **Add**.
    3. On the **Browser SSO > Protocol Settings > Assertion Creation > Attribute Contract** tab, for **SAML_SUBJECT**, in the **Subject Name Format** list, select **urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress**.
6. On the **Credentials** tab, configure the connection credentials. Click **Next**.

    You can find a complete guide in Configure credentials in the PingFederate documentation.
7. On the **Outbound Provisioning** tab, configure the provisioning target and channel as shown in Configuring outbound provisioning in the PingFederate documentation.
8. On the **Activation and Summary** tab, above the **Summary** section, click the toggle to turn on the connection. Click **Save**.

Release notes

Changelog

The following is the change history for the Atlassian Cloud Provisioner.

Copyright © 2025 Ping Identity Corporation

Atlassian Cloud Provisioner 2.0 - February 2025

- The Atlassian Cloud Provisioner now performs group updates with the `PATCH` operation instead of the `PUT` operation. This affects the group management capabilities described in User and group management.

Atlassian Cloud Provisioner 1.0 – May 2020

- Initial Release
- Included support for user and group provisioning.
- Included support for Atlassian Cloud attributes.
- Included support for API key authentication.
- Included support for provisioning disabled users.
- Included configuration options for create, update, and disable/delete capabilities.
- Included configuration options for deprovisioning actions.

Known issues and limitations

The following are known issues or limitations with the Atlassian Cloud Provisioner.

Known issues

There are no known issues.

Known limitations

- Attributes
  - Due to limitations in PingFederate, user attributes cannot be cleared after they are set.
- Deprovisioning
  - When an LDAP user is deleted in a targeted group distinguished name (DN), the provisioning connector does not propagate the deletion until a new user is added to the group. This limitation is compounded when the **User Create** provisioning option is disabled. For solutions, refer to SaaS provisioner does not remove the user in the Knowledge Base.
- Provisioning
  - Atlassian only allows users with a verified domain to be created. If a user seems to have synced successfully, but does not appear on the **Managed accounts** page in Atlassian, the user's domain is likely unverified.
  - Due to a limitation in Atlassian, group names cannot be updated.

Copyright © 2025 Ping Identity Corporation

Download manifest

The following files are included in the Atlassian Cloud Provisioner `.zip` archive:

- `legal/Legal.pdf` : Copyright and license information.
- `dist` : Contains the integration files
  - `pf-atlassian-quickconnection-<version>.jar` : The Atlassian Cloud Provisioner quick connection template.
- `atlassian-saml-metadata.xml` : A modifiable metadata file used to create the connection.

Atlassian Integration Kit

The Atlassian Integration Kit allows PingFederate to coordinate single sign-on (SSO) for on-premises deployments of Jira and Confluence.

> **Tip**
>
> To integrate with Atlassian Cloud, see Atlassian integrations.

This integration requires the use of the Reference ID adapter, which is part of the Agentless Integration Kit. The Reference ID adapter is used to pass user identity information from PingFederate to the specific Atlassian product.

Components

- Authenticator
  - This component uses the Atlassian Seraph API to enable SSO for Jira and Confluence.
- Atlassian plugin
  - This plugin allows you to configure the authenticator with the appropriate PingFederate settings.

Intended audience

This document is intended for PingFederate administrators.

Before you start, you should be familiar with the following parts of the PingFederate documentation:

- Identity provider SSO configuration
- Managing IdP adapters

System requirements

- PingFederate 8.0 or later
- Agentless Integration Kit 1.5 or later

Copyright © 2025 Ping Identity Corporation

For Atlassian Jira integrations:

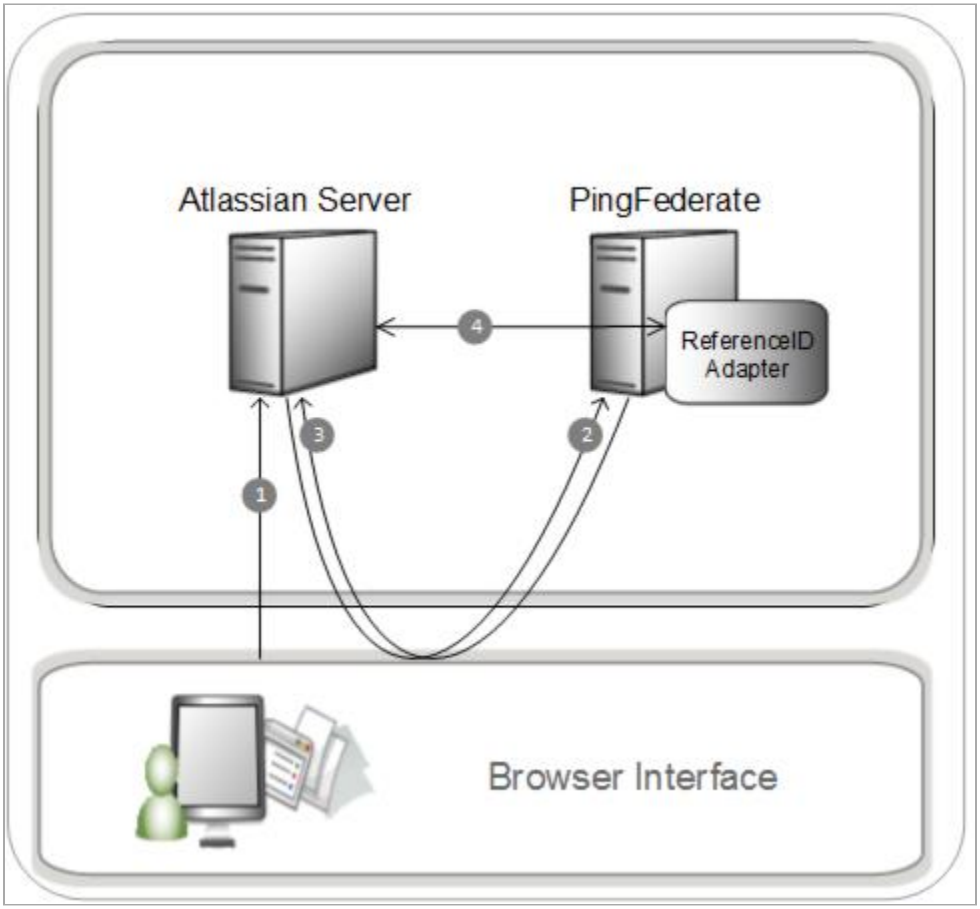
- Atlassian Jira 8.13
- Java 8 or later

For Atlassian Confluence integrations:

- Atlassian Confluence 7.4
- Java 8

Overview of the SSO flow

The following figure illustrates an example SSO process flow.



Processing Steps

1. The user navigates to an Atlassian application, such as Jira or Confluence.
2. The user is redirected to PingFederate server to perform SSO. After the user logs in PingFederate creates a unique identifier and associates it with the user's attributes.
3. PingFederate redirects the user back to Atlassian. The URL contains a reference ID which is used to request user attributes from PingFederate in the next step.

Copyright © 2025 Ping Identity Corporation

4. The authenticator issues a request (which contains the identifier obtained from the previous step) to PingFederate's Reference ID adapter. After PingFederate ensures the identifier is valid and hasn't timed out, the user's attributes are returned to the Authenticator and the session is created.

Setup

Deploying the Agentless Integration Kit

About this task

The Atlassian Integration Kit relies on the Reference ID adapter that is included in the Agentless Integration Kit.

Steps

1. In a browser, go to the Ping Identity Integration Directory.
2. Search for "Agentless".
3. Click Agentless Integration Kit.
4. On the sidebar, click **Download Integration**.
5. On the sidebar, click **Get Documentation**.
6. Follow the documentation to deploy the Agentless Integration Kit.

Upgrading instructions

Steps

1. From the distribution package, copy the file `dist/pf-authenticator-x.jar` to the Atlassian library directory:

   Choose from:

   - For Confluence: `<Atlassian Installation Directory>/Confluence/confluence/WEB-INF/lib`
   - For Jira: `<Atlassian Installation Directory>/Jira/atlassian-jira/WEB-INF/lib`

   If installing to Jira, follow the steps outlined in Configuring Jira.
2. Restart the Atlassian platform and attempt single sigh-on (SSO) as an administrator.
3. Within the Atlassian platform, remove the old plugin from **Manage add-ons** and install the new plugin from the distribution `.zip` `dist/pf-authenticator-plugin-x.jar`.
4. Go to the plugin configuration page, http[s]://*atlassian_hostname*:*atlassian_port*/plugins/servlet/ping/config, to ensure it displays the existing configuration correctly.
5. For Jira, if you customized your `login.jsp` file, back up your changes. Deploy the latest `login.jsp` file from the Atlassian Integration Kit, and then modify it with your customizations.

Copyright © 2025 Ping Identity Corporation

Defining the Authenticator configuration directory

Before using the plugin to configure the authenticator, you must define where the plugin will store the configuration settings. This information is stored in an environment variable.

Defining the Authenticator configuration directory (Unix)

Steps

1. Create a directory of your choosing on the Atlassian server. We suggest creating a subdirectory in your Atlassian instance: e.g. `<Atlassian Installation Directory>/pingfederate-settings`

   > **Note**
   >
   > The user that is running Jira must have read/write permissions for the created directory.

2. Within your Jira root directory, edit `./bin/setenv.sh` (`setenv.bat` on Windows)
3. At the top of this file insert the following (modify the actual path to suit your environment): `PINGFEDERATE_ATLASSIAN_DATA_PATH=<Atlassian Installation Directory>/pingfederate-settings; export PINGFEDERATE_ATLASSIAN_DATA_PATH`
4. Restart the Jira server.

Defining the Authenticator configuration directory (Windows)

Steps

1. Create a directory of your choosing on the Atlassian server.

   > **Note**
   >
   > You should create a subdirectory in your Atlassian instance, such as `<Atlassian Installation Directory>\pingfederate-settings` .

2. Go to your Jira root directory and edit `.\bin\setenv.bat` .
3. At the start of the `.\bin\setenv.bat` file, insert the following:

   Choose from:

   - If Jira or Confluence is running as a standalone service (started using `startup.bat`):

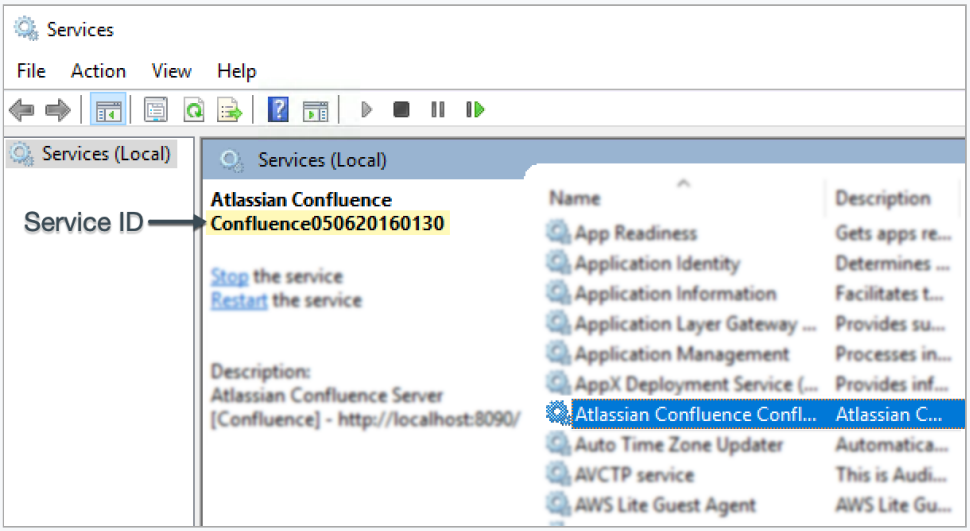
     ```
     set PINGFEDERATE_ATLASSIAN_DATA_PATH=<Atlassian Installation Directory>\pingfederate-settings
     ```

   - If Jira or Confluence is started using Windows service, go to Jira's `bin` directory with a command prompt running as Administrator, where `Tomcat9.exe` binary is located and enter the following.

     ```
     tomcat9 //US//<service id>  --Environment=PINGFEDERATE_ATLASSIAN_DATA_PATH=<Atlassian Installation Directory>\pingfederate-settings
     ```

Copyright © 2025 Ping Identity Corporation

> **Note**
>
> Depending on your version of Atlassian, you might need to change `tomcat9` to `tomcat8` in the command. To find out which version of Tomcat you have, check the documentation for your Atlassian product:
>
> - Bundled Tomcat and Java versions for Confluence
> - Bundled Tomcat and Java versions for Jira

> **Tip**
>
> To determine your service ID, open `services.msc` and select the Jira or Confluence service. The service ID appears in the details area. A service ID is formatted as `JIRA123456123456` .
>


4. Restart the Jira server.

Uploading the plugin

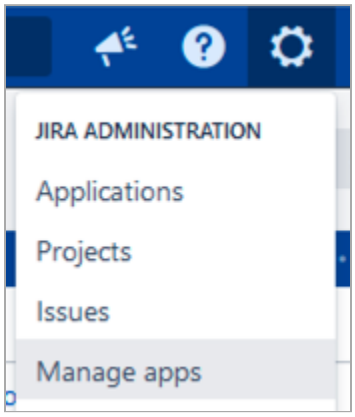
About this task

The process is similar for deploying the plugin in both Jira and Confluence.

Steps

1. Log in to Jira as an administrator.
2. From the **Settings** menu, click **Manage apps**.

Copyright © 2025 Ping Identity Corporation



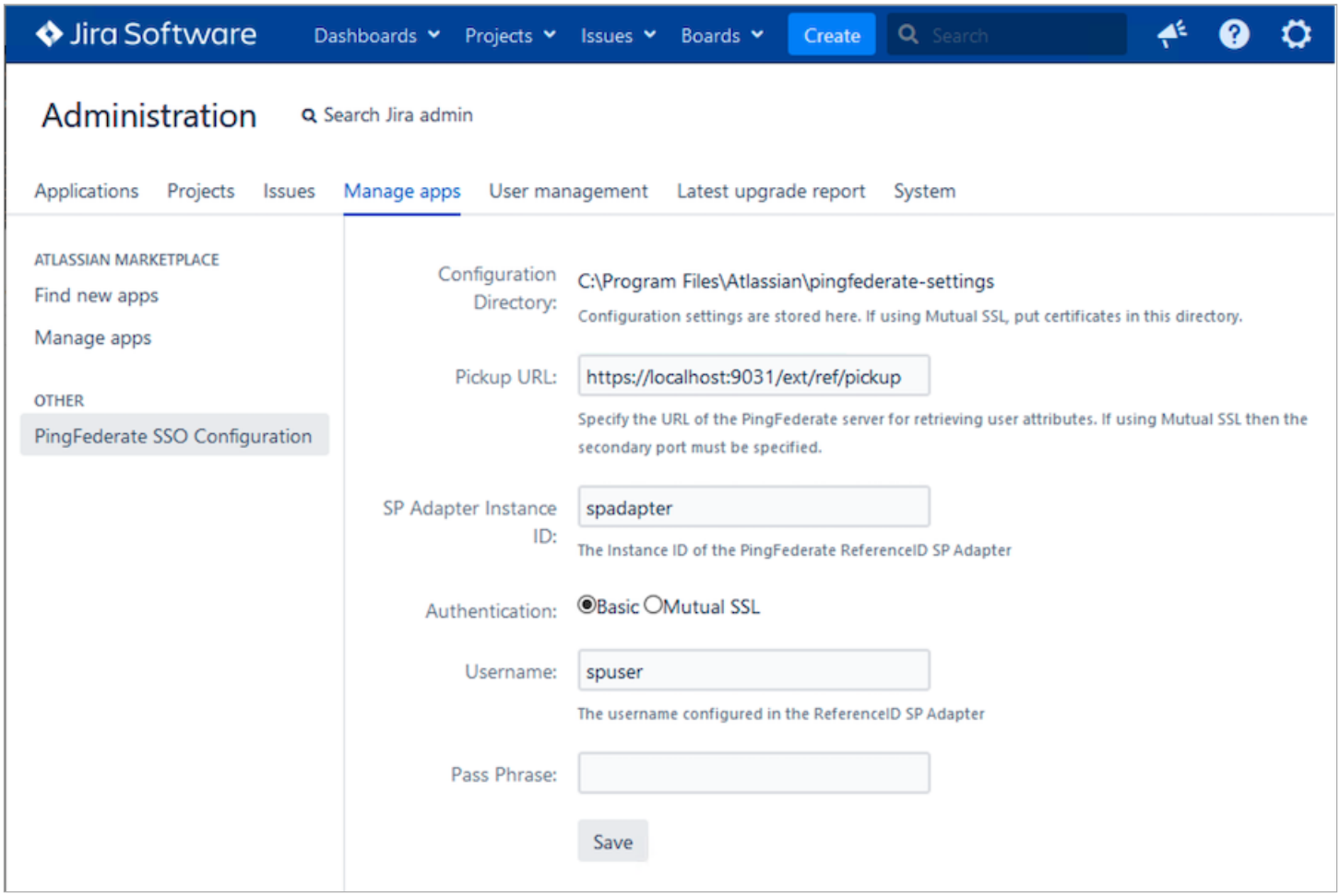
3. On the sidebar, click **Manage apps**.
4. Click **Upload app**.
5. Go to the integration kit contents, and select the `dist/pf-authenticator-plugin-x.jar` file. Click **Upload**.
6. Click **Configure**.

Configure the plugin

The URL to configure the Authenticator is `http[s]://atlassian_hostname:atlassian_port/plugins/servlet/ping/config`

The configuration page is also accessible at **Manage apps > PingFederate SSO Configuration > /**:

Copyright © 2025 Ping Identity Corporation



Configuration options

Field	Description
Configuration Directory	Configuration settings are stored here. If using Mutual SSL, put certificates in this directory. For more information, see Defining the Authenticator configuration directory.
Pickup URL	`https://pf_host:pf_port/ext/ref/pickup` When configuring a pickup URL: • Always use HTTPS over HTTP • Enable the Require SSL/TLS option within the PingFederate SP Reference ID adapter configuration • If using Mutual SSL Authentication, `pf_port` in the Pickup URL must match the port specified in the PingFederate `run.properties` for the `pf.secondary.https.port` (Step 1 of Configuring PingFederate for mutual SSL authentication)
SP Adapter Instance ID	The Instance ID of your PingFederate SP Reference ID adapter.

Copyright © 2025 Ping Identity Corporation

Field	Description
Authentication	Determines the type of authentication to be performed between the Authenticator and the Reference ID adapter. **Mutual SSL**: You should use this option as it provides the highest level of security. For more information, see Mutual SSL authentication **Basic**: Uses a username and passphrase to authenticate with PingFederate . For more information, see Basic authentication

Mutual SSL authentication

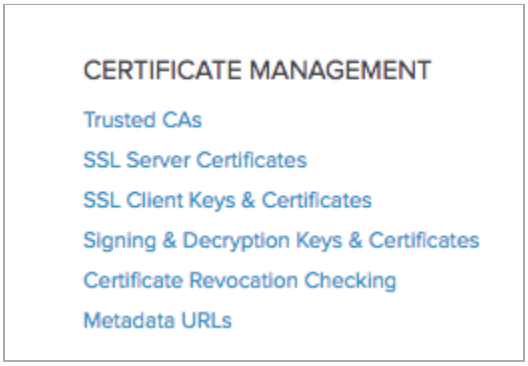
Mutual SSL Authentication configuration requires changes to both the Atlassian and PingFederate configurations.

- Configuring PingFederate for mutual SSL authentication
- Configuring Atlassian

Configuring PingFederate for mutual SSL authentication

Steps

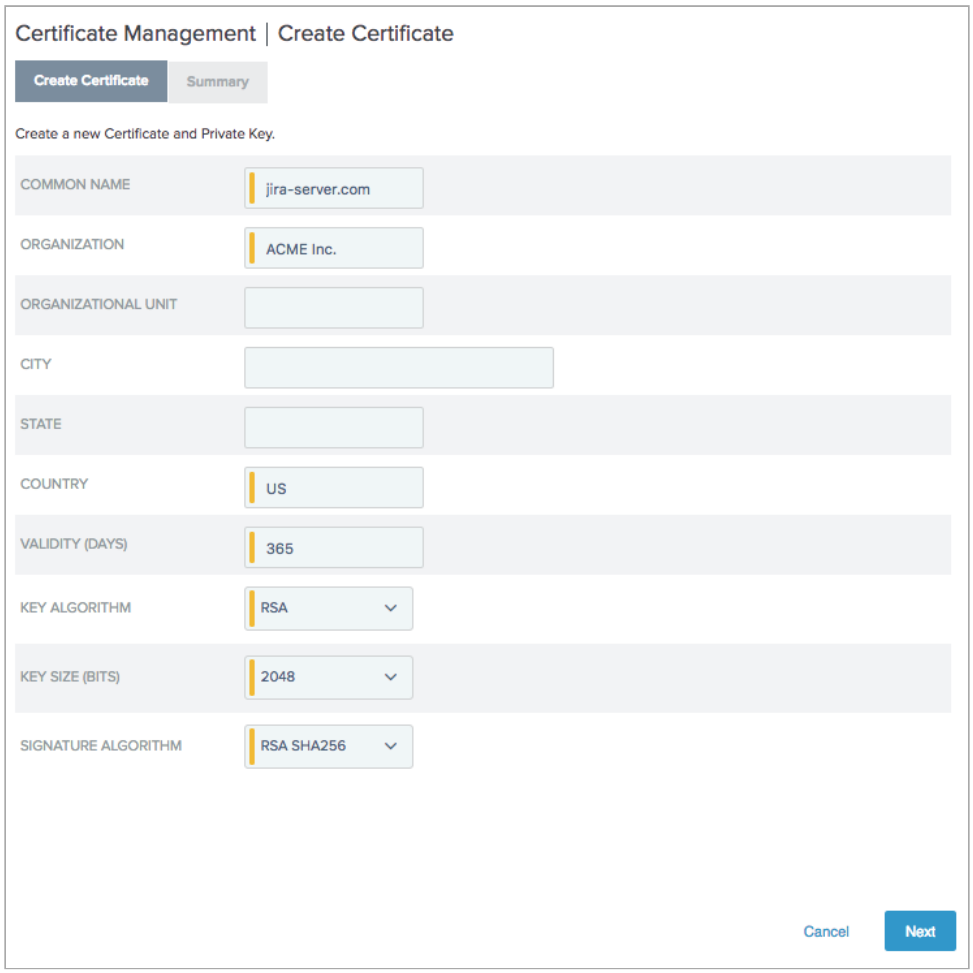
1. Configure a secondary SSL port. See the property `pf.secondary.https.port` in the table under Configuring PingFederate properties.
2. Import the SSL Certificate of your Atlassian server into PingFederate. See Manage trusted certificate authorities in the PingFederate documentation. If you do not have an SSL certificate for the Atlassian server you can use PingFederate's Client Key/Certificate Utility to create one:
    1. From PingFederate's Admin Page go to: `Server Configuration > Certificate Management > SSL Client Keys & Certificates`:



    2. Click `Create New`.
    3. For Common Name, enter the domain of your Atlassian server.
    4. Input a value for Organization.

Copyright © 2025 Ping Identity Corporation

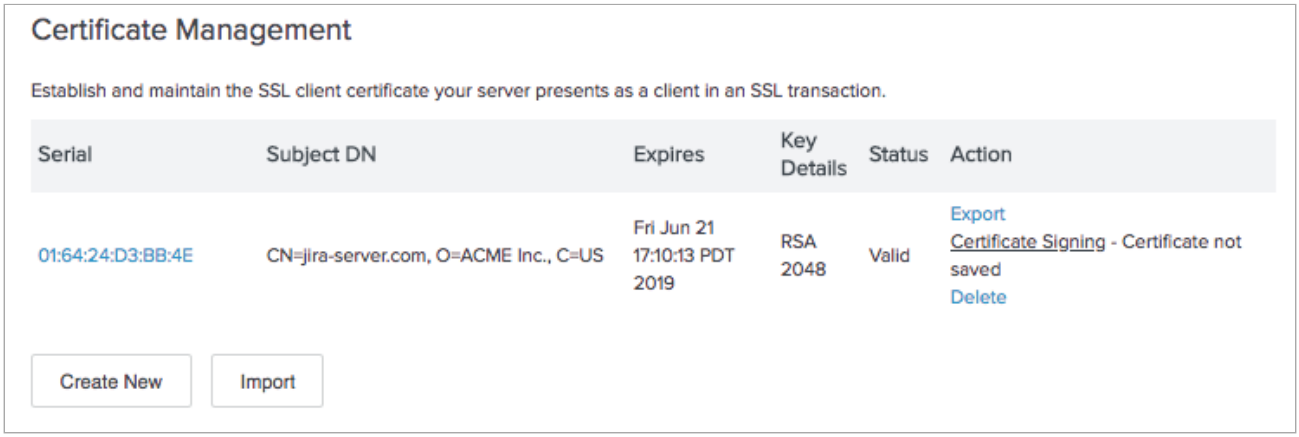
5. Optionally fill out the remaining fields.



6. Click `Next`.

7. Click `Done`.

8. From the list of certificates click `Export` on your new certificate:



9. Choose Certificate and private key.

   This creates a PKCS 12 certificate file (p12 extension), which will be used later in this guide.

Copyright © 2025 Ping Identity Corporation

10. Click Previous and `Export` once again but this time choose Certificate.

    This will export the certificate file (crt extension).

11. Import this certificate into PingFederate as a Trusted CA (`Server Configuration > Certificate Management > Trusted CAs`).

    See Manage trusted certificate authorities in the PingFederate documentation for more information.

3. Create an instance of Reference ID SP Adapter.

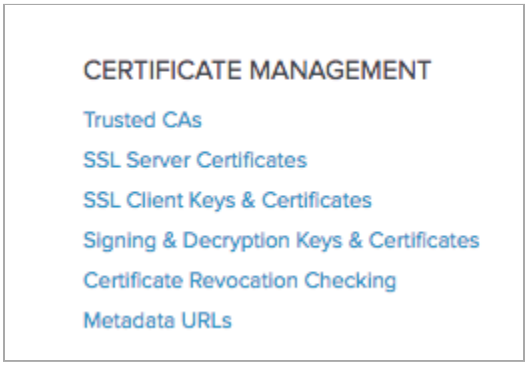
   For a complete guide, see Configuring a Reference ID SP Adapter instance in the Agentless Integration Kit documentation.

   1. Clear the **User Name** and **Pass Phrase** fields.
   2. Input a value for **Allowed Subject DN**. For example, for the certificate shown in section 2.h. above, the DN would be: `CN=jira-server.com, O=ACME Inc., C=US` .
   3. From the **Transport Mode** list, select **Query Parameter**.

Configuring Atlassian

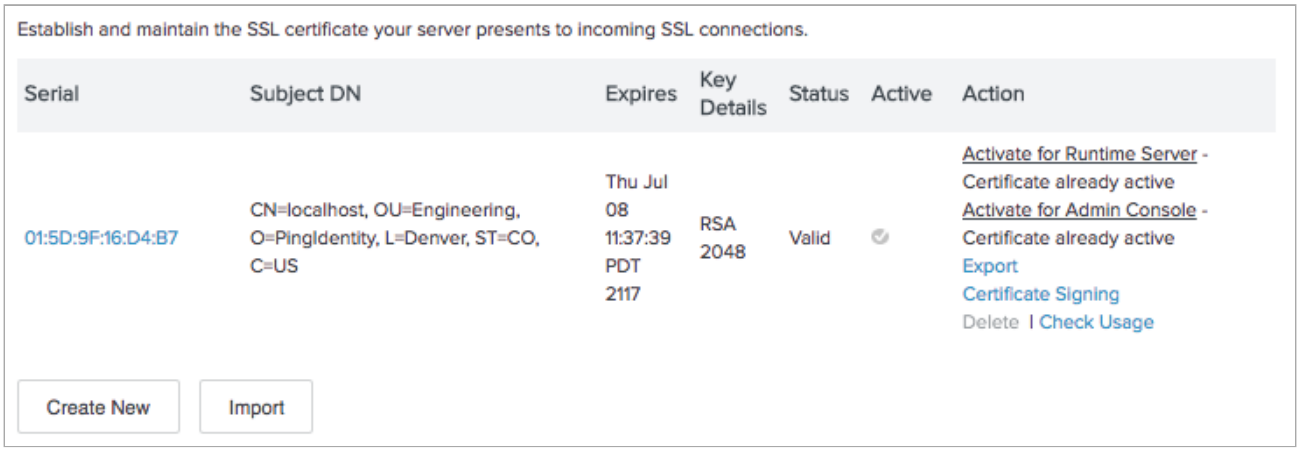
Steps

1. Copy the p12 file created in step 2.i from the previous section into the Configuration Directory of your Atlassian Server.
2. Export PingFederate's SSL Server Certificate:
   1. From the Administrator page choose **SSL Server Certificates**.



   2. Click **Export** next to your Server Certificate.

Copyright © 2025 Ping Identity Corporation



3. Choose **Certificate Only** and click **Next**.
4. Click **Export** to save the certificate.



3. Copy the Server Certificate you exported in the previous step into the Configuration Directory of your Atlassian Server.
4. Complete the configuration of the Authenticator Plugin:

Field	Description
Client Keystore File	The full name of the PKCS 12 file from Step 1 (do not include the full path, just the file name)
Client Keystore Password	If there was a password assigned to the client certificate enter that here
PingFederate Certificate	The full name of the certificate from Step 3 (do not include the full path, just the file name)

Copyright © 2025 Ping Identity Corporation

> **Note**
>
> When the configuration options are saved, the plugin will check to make sure the client and server certificates can be found.

> **Important**
>
> If you are using a self-signed certificate, you'll need to import it into the JVM keystore. See the Importing your self-signed certificate into Atlassian section for more information.

Basic authentication

- Configuring PingFederate for basic authentication
- Atlassian configuration

Configuring PingFederate for basic authentication

Create an instance of the Reference ID SP Adapter.

About this task

For a complete guide, see Configuring a Reference ID SP Adapter instance in the Agentless Integration Kit documentation.

Steps

1. In the **User Name** and **Pass Phrase** fields, enter your credentials.
2. Clear the **Allowed Subject DN** and **Allowed Issuer DN** fields.
3. From the **Transport Mode** list, select **Query Parameter**.

Atlassian configuration

Complete the configuration of the Authenticator Plugin:

Field	Description
Username	The User Name of the SP Reference ID Adapter
Pass Phrase	The Pass Phrase of the SP Reference ID Adapter

Copyright © 2025 Ping Identity Corporation

> **Important**
>
> If you are using a self-signed certificate you'll need to import it into the JVM keystore. See the Importing your self-signed certificate into Atlassian for more information.

Install the authenticator

From the distribution package, copy file: `dist/pf-authenticator-x.jar` to the Atlassian library directory:

- For Confluence, `<Atlassian Installation Directory>/Confluence/confluence/WEB-INF/lib`
- For Jira, `<Atlassian Installation Directory>/Jira/atlassian-jira/WEB-INF/lib`

Configuring Jira

About this task

When accessing a direct link to a subpage within Jira with no user session, the login link does not work correctly. This is the workaround.

Steps

1. Copy the following files from the distribution package to the Atlassian jsp directory: `<Atlassian Installation Directory>/Jira/atlassian-jira/`
    - `dist/jira/login.jsp`
    - `dist/jira/default.jsp`
2. Edit the two jsp files copied in the previous step and update the values for the four variables at the top of the file. For information on what these should be, refer to step 3 of Configuring Seraph.
3. Make a backup copy of `<Installation Directory>/Jira/atlassian-jira/WEB-INF/web.xml`.
4. Open this file in a text editor: `<Installation Directory>/Jira/atlassian-jira/WEB-INF/web.xml`
5. Disable the default and login pages by adding comment blocks around the <servlet> and <servlet-mapping> nodes.

Copyright © 2025 Ping Identity Corporation

```
<!--
    <servlet>
        <servlet-name>jsp.default_jsp</servlet-name>
        <servlet-class>jsp.default_jsp</servlet-class>
    </servlet>
-->

<!--
    <servlet>
        <servlet-name>jsp.secure.default_jsp</servlet-name>
        <servlet-class>jsp.secure.default_jsp</servlet-class>
    </servlet>
-->

<!--
    <servlet>
        <servlet-name>jsp.secure.project.default_jsp</servlet-name>
        <servlet-class>jsp.secure.project.default_jsp</servlet-class>
    </servlet>
-->

<!--
    <servlet>
        <servlet-name>jsp.secure.admin.default_jsp</servlet-name>
        <servlet-class>jsp.secure.admin.default_jsp</servlet-class>
    </servlet>
-->

<!--
    <servlet-mapping>
        <servlet-name>jsp.default_jsp</servlet-name>
        <url-pattern>/default.jsp</url-pattern>
    </servlet-mapping>
-->

<!--
    <servlet-mapping>
        <servlet-name>jsp.secure.default_jsp</servlet-name>
        <url-pattern>/secure/default.jsp</url-pattern>
    </servlet-mapping>
-->

<!--
    <servlet-mapping>
        <servlet-name>jsp.secure.project.default_jsp</servlet-name>
        <url-pattern>/secure/project/default.jsp</url-pattern>
    </servlet-mapping>
-->

<!--
    <servlet-mapping>
        <servlet-name>jsp.secure.admin.default_jsp</servlet-name>
        <url-pattern>/secure/admin/default.jsp</url-pattern>
    </servlet-mapping>
```

Copyright © 2025 Ping Identity Corporation

```
-->

<!--
        <servlet>
            <servlet-name>jsp.login_jsp</servlet-name>
            <servlet-class>jsp.login_jsp</servlet-class>
        </servlet>
-->

<!--
        <servlet-mapping>
            <servlet-name>jsp.login_jsp</servlet-name>
            <url-pattern>/login.jsp</url-pattern>
        </servlet-mapping>
-->
```

Configuring Seraph

About this task

The final step to linking the two systems is to configure the Seraph file within the Atlassian product.

Steps

1. Backup the existing seraph-config.xml. It's crucial to perform this step, because if there is an issue with your configuration you may need to restore this file to get back into your Atlassian server. This file can be located at:

   Choose from:

   - For Confluence:

     `<Atlassian Installation Directory>/Confluence/confluence/WEB-INF/classes`

   - For Jira:

     `<Atlassian Installation Directory>/Jira/atlassian-jira/WEB-INF/classes`

2. Open `seraph-config.xml` in your favorite editor.
3. Change the param-value of `login.url` and `link.login.url` to:

   `https://<pf_host>:<pf_port>/sp/startSSO.ping?PartnerIdpId=`

   `<idp_connection_entity_id>&SpSessionAuthnAdapterId=`

   `<sp_refid_adapter_instance_id>&TARGET=${originalurl}`

   `pf_host` : The PingFederate host

   `pf_port` : The PingFederate port. This should be the same value that was specified for the secondary SSL port in the Mutual SSL authentication

   `idp_connection_entity_id` : The Partner Entity ID for the IdP connection (found under General Info section of the IdP Connection).

Copyright © 2025 Ping Identity Corporation

`sp_refid_adapter_instance_id`: The Instance ID for the SP Reference ID adapter.

> **Note**
>
> This value determines where the user is redirected to in step 2 of Overview of the SSO flow.

> **Important**
>
> The values for `login.url` and `link.login.url` must be URL-encoded and HTML-escaped.

4. Change the authenticator class:

   Choose from:

   - For Confluence, remove this line:

     ```
     <authenticator class="com.atlassian.confluence.user.ConfluenceAuthenticator"/>
     ```

     Add this line:

     ```
     <authenticator
     class="com.pingidentity.adapters.atlassian.confluence.PFConfluenceAuthenticator"/>
     ```

   - For Jira, remove this line:

     ```
     <authenticator class="com.atlassian.jira.security.login.JiraSeraphAuthenticator"/>
     ```

     Add this line:

     ```
     <authenticator class="com.pingidentity.adapters.atlassian.jira.PFJiraAuthenticator"/>
     ```

5. Save the `seraph-config.xml`.
6. For Jira, follow these additional steps:

   1. Open `<Atlassian Installation Directory>/Jira/atlassian-jira/WEB-INF/classes/jira-application.properties`
   2. Disable the login gadget.

      `jira.disable.login.gadget=true`

   3. Save the file.

Importing your self-signed certificate into Atlassian

If you are using a self-signed certificate for PingFederate, import it into the Java virtual machine (JVM) key store of the Atlassian server. This is required when the authentication type is mutual SSL or Basic authentication.

Copyright © 2025 Ping Identity Corporation

About this task

> **Tip**
>
> The default key store password on most systems is `changeit`. On MacOS 10.7 or earlier, the password is `changeme`.

Steps

1. Run the following command as an administrative user (or sudo):

```
keytool -importcert -alias {pingfed} -file /<full_path_to_cert>/<certificate_name>.crt -keystore /
<full_path_to_jre>/lib/security/cacerts
```

2. Replace the variables in the command as follows:

Variable	Description
`full_path_to_cert`	Based on previous configuration steps, this value is: `$PINGFEDERATE_ATLASSIAN_DATA_PATH`
`certificate_name`	The file name of the certificate.
`full_path_to_jre`	The Java Runtime Environment (JRE) being used by Atlassian is usually listed when you start Jira or Confluence.

Troubleshooting

The following table lists potential problems administrators might encounter during the setup or deployment of the Atlassian Integration Kit, along with possible solutions:

Problem	Possible Solution
(SSO Failure) An SSO attempt results in a continuous loop between PingFederate and the Atlassian product.	1. Ensure that the IdP User exists in the Atlassian Product. 2. Ensure that the PingFederate Reference ID Adapter settings match your Authenticator Plugin configuration in the Atlassian Product. 3. If using a self-signed certificate, refer to the following instructions (Importing your self-signed certificate into Atlassian). 4. If using Mutual SSL Authentication, ensure the following: 1. Import the SSL Certificate of your Atlassian server into PingFederate (Import Certificate into PingFederate). 2. Ensure the Atlassian and PingFederate server certificates are in the Configuration Directory. 3. Ensure the Reference ID Adapter contains the correct Issuer or Subject DN values.

Copyright © 2025 Ping Identity Corporation

Users are logged out at inappropriate times	This can be due to clashing sessions caused by Jira and Confluence running on the same host. This may be prevented by using different hostname aliases and changing the base URLs of Jira and Confluence to use different hostnames. For example you could set the following base URL for Jira: `http://jira:8080` and the following base URL for Confluence: `http://confluence:8090`

Enabling logging

To help with troubleshooting or monitoring, you can turn on activity logging for the authenticator.

Steps

1. Open the log4j properties file by doing one of the following:

   Choose from:

   - For Jira: `<jira-installation>/Jira/atlassian-jira/WEB-INF/classes/log4j.properties`
   - For Confluence: `<confluence-installation>/Confluence/confluence/WEB-INF/classes/log4j.properties`

2. Add the following lines:

```
log4j.appender.PingFederateLog=org.apache.log4j.RollingFileAppender
log4j.appender.PingFederateLog.File=pingfederate.log
log4j.appender.PingFederateLog.MaxFileSize=20480KB
log4j.appender.PingFederateLog.MaxBackupIndex=5
log4j.appender.PingFederateLog.layout=com.atlassian.logging.log4j.NewLineIndentingFilteringPatternLayout
log4j.appender.PingFederateLog.layout.ConversionPattern=%m%n
log4j.logger.com.pingidentity.adapters.atlassian = INFO, console, {pingfed} Log
log4j.additivity.com.pingidentity.adapters.atlassian=false
```

3. **Optional:** Replace `INFO` with `DEBUG` to turn on debug logging.
4. Save the file.

Release Notes

Changelog

Atlassian Integration Kit 2.2 – April 2021

- Fixed a cross-site scripting (XSS) vulnerability by improving parameter sanitation.
- Added support for the Atlassian Application Server REST API.

Copyright © 2025 Ping Identity Corporation

Atlassian Integration Kit 2.1 – June 2019

- Added support for Atlassian Confluence 6.15.
- Added support for Atlassian Jira 8.x.
- Fixed a redirect loop that occurred when a user existed in the identity provider but not in Confluence.
- Improved error-handling for redirect situations.

Atlassian Integration Kit 2.0.1 – June 2018

- Fixed a defect in AutoAddGroups functionality.

Atlassian Integration Kit 2.0 – March 2018

- Compatibility updates for the latest Atlassian platforms.

Atlassian Integration Kit 1.1.1 – July 2016

- Fixed an issue so Basic and Mutual SSL options are selectable in the plugin configuration.

Atlassian Integration Kit 1.1 – April 2016

- Added support for Default Group Memberships (autoAddGroups)
- Added support for Jira 7
- Bug fixes

Atlassian Integration Kit 1.0 – March 2015

- Atlassian Integration Kit 1.0 – March 2015

Known issues and limitations

Known issues

There are no known issues.

Known limitations

- When a user logs into Jira from the Logout screen, using the Login link in the top right navigation, the user is successfully logged in, but the user is prompted with a Logout confirmation dialog upon login.
- The prerequisite before performing SSO is that a valid user must exist in both the IdP and the Atlassian product. In a scenario where a user exists for an IdP but does not exist in the Confluence server, an SSO attempt will result in an error message for Jira and Confluence.

Copyright © 2025 Ping Identity Corporation

- The PingFederate server will perform a CRL check of the certificate received by the Authenticator class. If PingFederate cannot connect to the CRL server to verify the certificate this can cause a delay during SSO. To remove this delay make sure that the PingFederate server can resolve the CA that issued the certificate by using DNS or a adding entries to the hosts file that resolve the CA server.
- The kit can't be used with the built in SAML SSO that comes with the Atlassian applications. The user will be asked to log in twice in some cases, or the Atlassian integration kit will take over the login method for SAML SSO.

Download manifest

The distribution `.zip` archive for the kit contains the following:

- `ReadMeFirst.pdf` – contains links to this online documentation
- `Legal.pdf` – copyright and license information
- `/dist` – contains libraries needed to run the kit:
  - `pf-authenticator-<version>.jar`
  - `pf-authenticator-plugin-<version>.jar`
  - `/jira`
    - `default.jsp`
    - `login.jsp`

Copyright © 2025 Ping Identity Corporation

Azure AD integrations

Azure AD Password Credential Validator

Password credential validators (PCV) allow PingFederate administrators to define a centralized location for username and password validation, allowing validator instances to be referenced by various PingFederate configurations.

PingFederate Integration with Azure Active Directory and Office 365

This guide provides detailed steps for integrating PingFederate with Microsoft Azure Active Directory and Microsoft Office 365.

Azure AD Password Credential Validator

Password credential validators (PCV) allow PingFederate administrators to define a centralized location for username or password validation, allowing validator instances to be referenced by various PingFederate configurations. The PingFederate Azure AD PCV uses Microsoft Graph API for credentials validation.

Features

- Sign-on with full usernames, such as john.smith@mydomain.com.
- Support for Azure AD Custom Properties (Directory Schema Extensions).
- All user group memberships are included in the response.
- Failed sign-on feedback, such as "invalid credentials", "account is disabled", "forced password change".
- Supports non-federate single and multi-tenant Azure AD user accounts.

Learn more in Known issues and limitations.

Intended audience

This document is intended for PingFederate administrators and application developers.

Learn more about the setup process with the following resources:

- Managing SP adapters
- Configuring an OpenToken SP Adapter instance

System requirements

- PingFederate 9.0 or later
- A Microsoft Azure account with Active Directory or Active Directory B2C configured

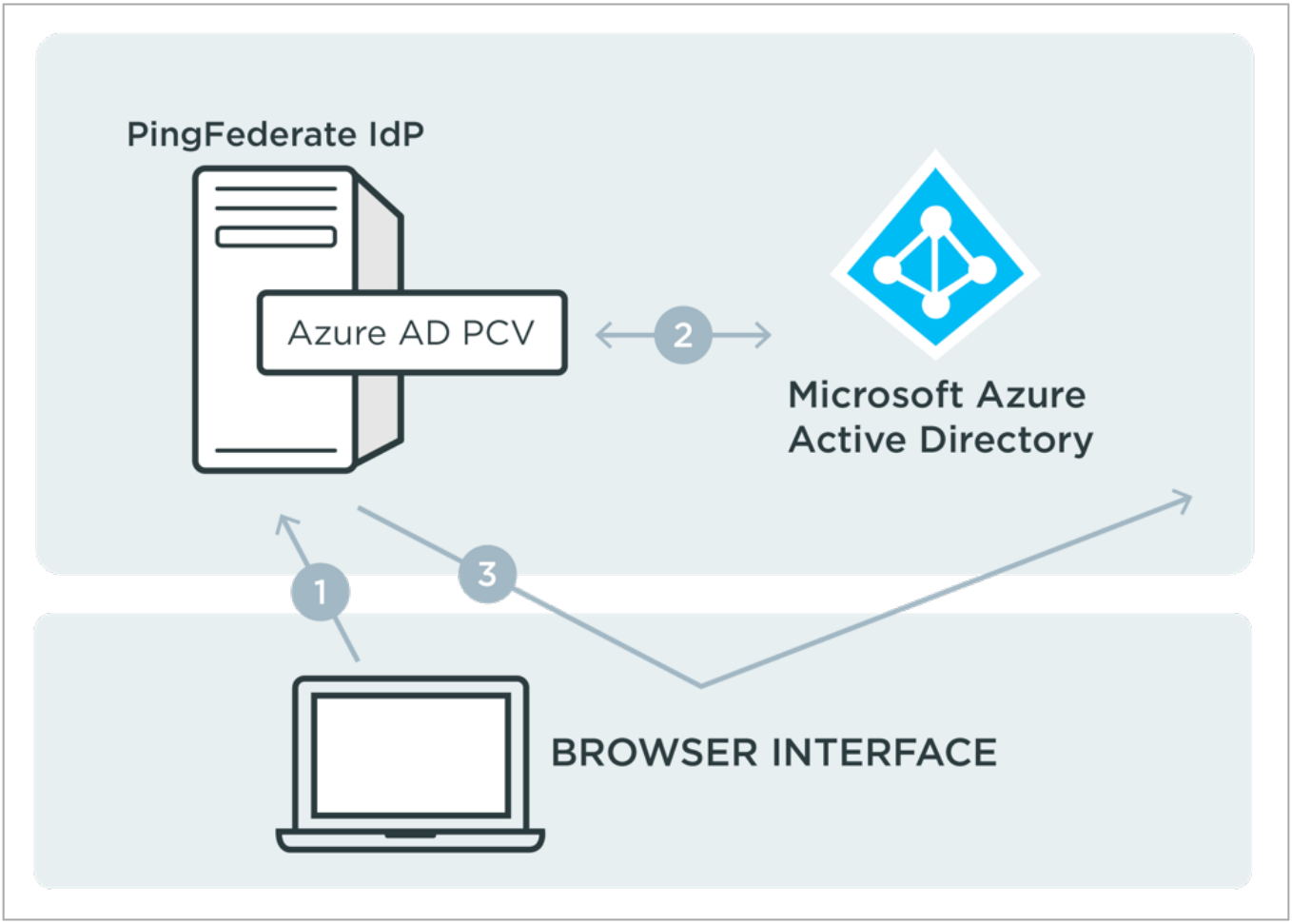
> **Note**
>
> Learn more about supported user account types in Known issues and limitations.

Copyright © 2025 Ping Identity Corporation

- To allow PingFederate to make outbound connections to the Microsoft API, you might need to allow the following endpoints in your firewall:
    - Token endpoint: https://login.microsoftonline.com/<*tenant*>/oauth2/v2.0/token
    - User attributes endpoint: https://graph.microsoft.com/v1.0/me/
    - Group membership endpoint: https://graph.microsoft.com/v1.0/me/memberOf

Overview of the SSO flow

The following figure illustrates an example user login flow using the Azure AD PCV:



Processing Steps

1. The user initiates SSO.
2. User credentials are validated by Microsoft Graph. Upon successful authentication, user attributes and group memberships are retrieved from Microsoft Graph.
3. The PingFederate IdP server generates an assertion, which contains the user's attributes and passes it to the SP application through the browser.

Copyright © 2025 Ping Identity Corporation

Setup

Deploying the integration files

To get started with the integration, deploy the Azure AD Password Credential Validator files to your PingFederate directory.

About this task

> **Note**
>
> If you operate PingFederate in a cluster, the following steps refer to the console node.

Steps

1. Download the Azure AD Password Credential Validator `.zip` archive from the Ping Identity Integration Directory.
2. Stop PingFederate.
3. If you're upgrading an existing deployment, delete the `pf-azure-ad-pcv-<version>.jar` file from your `<pf_install>/pingfederate/server/default/deploy` directory.
4. Extract the `.zip` archive and merge the contents of the `dist` directory with your `<pf_install>/pingfederate` directory.
5. Start PingFederate.
6. If you operate PingFederate in a cluster, repeat steps 2-5 for each engine node.

Required Azure AD permissions

The Azure AD application needs to have the following permissions enabled:

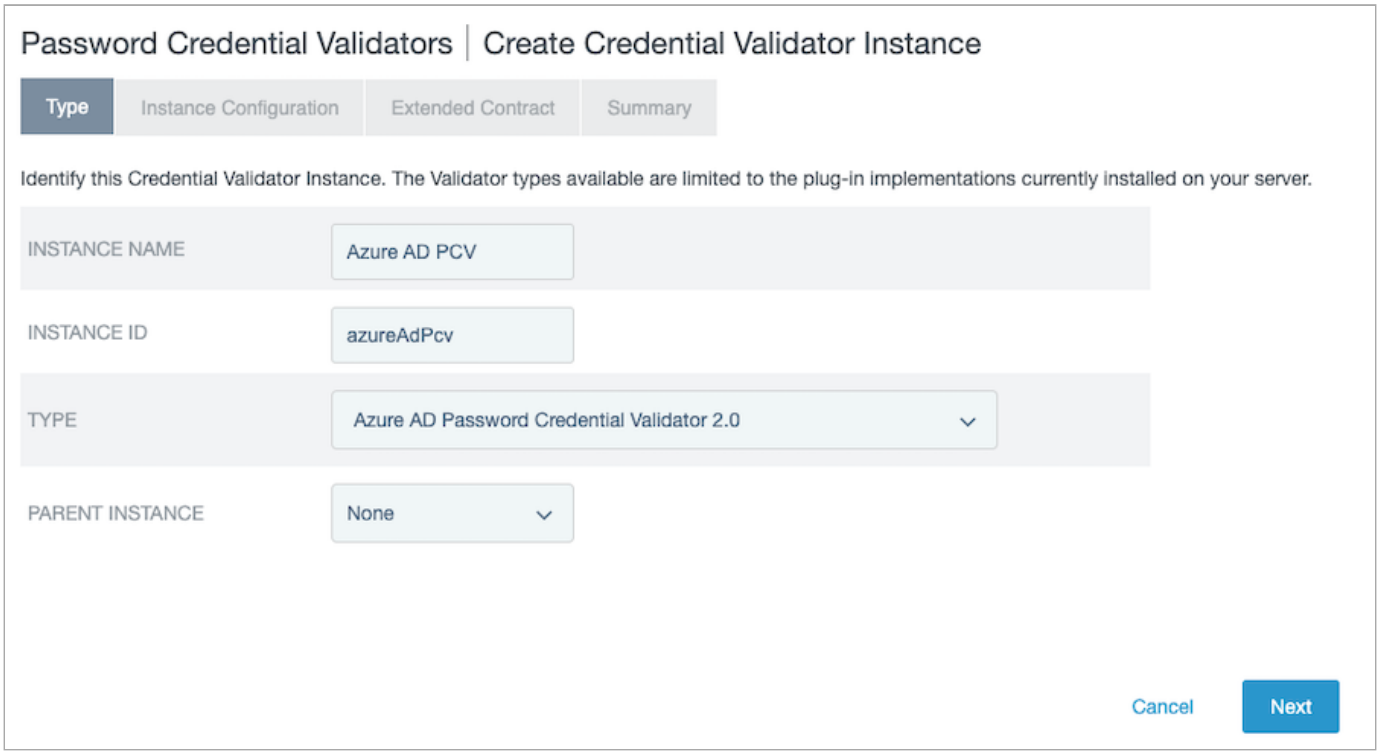
- Microsoft Graph > Delegated Permission
  - Sign in and read user profile
  - Read directory data

Configuring a password credential validator instance

Steps

1. Sign on to the PingFederate administrative console and go to **Server Configuration** > **Password Credential Validators** > **Create New Instance**.
2. On the **Type** tab, populate the **Instance Name** and **Instance ID** fields, choose **Azure AD Password Credential Validator 2.0** in the **Type** field, and click **Next**.

Copyright © 2025 Ping Identity Corporation



3. On the **Instance Configuration** tab, populate the **Tenant ID**, **Client ID** and **Client Secret** fields using the information specific to your Azure AD application, and click **Next**.

> **Note**
>
> Learn more about obtaining the **Client ID** and **Client Secret** in Register a Microsoft Entra app and create a service principal.

Configuration fields

Field	Type	Description
Tenant ID	String. Required.	The tenant ID generated by Microsoft when you register an application in Azure.
Client ID	String. Required.	The client ID generated by Microsoft when you register an application in Azure.
Client Secret	String. Required.	The client secret generated by Microsoft when you register an application in Azure.
Disable User Group Retrieval	Checkbox	Disable the PCV from retrieving the `memberOf` attribute for users.

Copyright © 2025 Ping Identity Corporation

Field	Type	Description
Microsoft Login Base URL	String. Required.	The base URL used by Microsoft for any authentication calls. The default value is `https://login.microsoftonline.com/`.
Microsoft Token Endpoint	String. Required.	The endpoint used by Microsoft to retrieve an access token. The default value is `/oauth2v2.0/token`.
User Attributes Endpoint	String. Required.	The endpoint used to retrieve user attributes. The default value is `https://graph.microsoft.com/v1.0/me`.
Group Membership Endpoint	String. Required.	The endpoint used to retrieve group membership info. The default value is `https://graph.microsoft.com/v1.0/me/memberOf`.
API Request Timeout		The amount of time in milliseconds that PingFederate waits for Microsoft APIs to respond to requests. A value of `0` disables the timeout. The default value is `5000`.
Proxy Settings	◦ No Proxy ◦ System Defaults ◦ Custom	Defines proxy settings for outbound HTTP requests. The default value is **System Defaults**.
Custom Proxy Host	String. Optional.	The proxy server hostname to use when **Proxy Settings** is set to **Custom**.
Custom Proxy Port	String. Optional.	The proxy server port to use when **Proxy Settings** is set to **Custom**.

> **Note**
>
> If the user's group memberships aren't required, select the option to **Disable User Group Retrieval**.

4. The attribute contract can be extended with any additional Azure AD attributes, including Azure AD custom properties. Learn more in Known issues and limitations.

Copyright © 2025 Ping Identity Corporation

> **Important**
>
> If you're upgrading from Azure AD Password Credential Validator 1.2 or earlier, and used the `objectID` attribute in your extended contract, change the attribute to `ID`.

The core contract contains the following attributes:

1. `displayName`
2. `givenName`
3. `mail`
4. `memberOf`
5. `surname`
6. `username`
7. `userPrincipalName`

5. Click **Next**, review your settings, then click **Save**.

Configuring an IdP adapter instance

The new PCV instance, created in the previous section, needs to be configured with an HTTP Basic or HTML Form IdP Adapter. The following steps associate your password credential validator (PCV) instance with an identity provider (IdP) adapter.

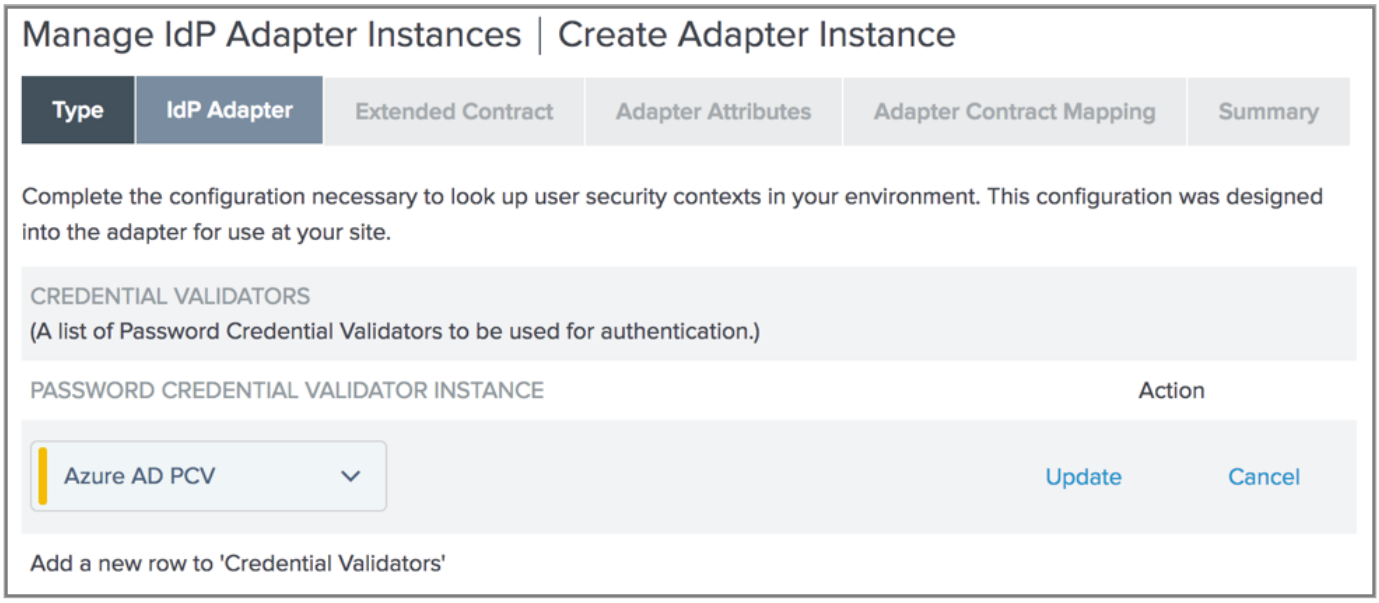
About this task

> **Note**
>
> Learn more about getting help with IdP adapters in HTTP Basic Adapter and HTML Form Adapter in the PingFederate documentation.

Steps

1. In the PingFederate administrative console, navigate to **IdP Configuration** > **Adapters** and select the configured IdP adapter.
2. Select **IdP Adapter**.
3. Select **Add a new row to 'Credential Validators'** to associate the new Azure AD PCV with this adapter:

Copyright © 2025 Ping Identity Corporation



4. **Optional:** Extend the contract.

> **Note**
>
> When extending the contract, ensure you add the attributes in the PCV configuration and in the SP Connection. If using PingFederate as the SP, the attributes also have to be extended in the SP Adapter and in the IdP Connection.

5. Save the configuration.

Testing

To test the configuration, navigate to the **SP Connection** in the PingFederate console under **IdP Configuration**. Open the URL listed next to **SSO Application Endpoint** to initiate SSO.

Release notes

Changelog

The following is the change history for the Azure AD Password Credential Validator.

Azure AD Password Credential Validator 2.1 – January 2023

- Added the **User Attributes Endpoint** and **Group Membership Endpoint** fields in the password credential validator (PCV) to accommodate customization of the user attributes endpoint and group membership endpoint URLs.

Azure AD Password Credential Validator 2.0 – November 2021

- Added the **Microsoft Login Base URL** and **Microsoft Token Endpoint** fields in the PCV to accommodate any changes Microsoft makes to the API endpoint.

Copyright © 2025 Ping Identity Corporation

- Replaced **Read Timeout** and **Connection Timeout** fields with the **API Request Timeout** field in the PCV.
- Changed to a standardized `.zip` file structure to make automated deployments easier.

> **Important**
>
> If you're upgrading from Azure AD PCV 1.2 or earlier and used the `objectID` attribute in your extended contract, change the attribute to `ID`.

Azure AD Password Credential Validator 1.2 – July 2019

- Improved support for passwords with special characters.
- Added proxy connection override settings.
- Added connection read and timeout settings.

Azure AD Password Credential Validator 1.1 – July 2017

- Retrieving user memberships can now be disabled. This will reduce network traffic and API calls in configurations where this data is not required.
- Bug fix: The PCV will properly failover to the next PCV in the chain of an HTML form adapter.

Azure AD Password Credential Validator 1.0 – Dec 2016

- Initial release.

Known issues and limitations

The following are known issues or limitations for the Azure AD Password Credential Validator.

Known issues

There are no known issues.

Known limitations

- User account support:
  - Supported: single-tenant and multi-tenant Azure AD user accounts.
  - Not supported by Azure AD or Azure AD B2C: Federated accounts, personal accounts, and personal accounts that have been invited to an Azure AD tenant even as a guest.
- Usernames
  - Short usernames, such as john.smith, aren't supported.

Copyright © 2025 Ping Identity Corporation

- Azure AD custom properties have the following naming convention: extension_applicationId_attributename. When extending the contract with custom properties, use the full attribute name. For example: extension_1e194ef156124ddcb1cedf75d43f1794_st. When these attributes are returned from the adapter they will be in the same format.
- The custom properties would have to be explicitly requested in API. Update the PCV instance fields **User Attributes Endpoint** and **Group Membership Endpoint** accordingly. Learn more in the Get user Microsoft documentation.

Download manifest

The following files are included in the Azure AD Password Credential Validator `.zip` archive.

- `Legal.pdf` : Copyright and license information.
- `dist/pingfederate/server/default` : Contains the integration files.
  - `deploy` : Contains the Java libraries.
    - `pf-azure-ad-pcv-<version>.jar` : The Azure AD PCV.

Azure AD and Office 365 Integration Guide

Integrating PingFederate with Microsoft Azure Active Directory (AD) and Microsoft Office 365 requires careful preparation and planning, which varies depending on the applications and use cases you're targeting.

After configuration, PingFederate provides single sign-on (SSO) into Azure AD, enabling access to Office 365 and other applications connected to this service.

Customers already running Azure AD Connect

If you reached this page by following the configuration export from Azure AD Connect, your Azure Active Directory domain will be configured for single sign-on for you and you can skip to Install and configure PingFederate.

Otherwise, complete the following prerequisite procedures:

- Getting ready to integrate Azure Active Directory
- Preparing Active Directory for federation

Prerequisites

Getting ready to integrate Azure Active Directory

Perform the following steps to enable use of PingFederate and Azure Active Directory (AD).

Copyright © 2025 Ping Identity Corporation

About this task

By performing these steps, you can use PingFederate and Azure AD together to:

- Access Azure online services including Office 365 without having to remember an additional username and password.
- Authenticate email access on mobile devices with usernames and passwords that are stored centrally in AD.
- Revoke access to email and secure content by disabling an account in AD.

> **Tip**
>
> To simplify the configuration experience, you can download Azure AD Connect from Microsoft and run it locally on a Windows server. Azure AD Connect guides you through the setup procedures for PingFederate and provides additional sync features to improve integration with existing AD infrastructure.

Steps

1. Sign up for an Azure Active Directory account.

   Microsoft offers various Azure AD and Office 365 plans for different types of organizational needs. Not all of them support Web SSO, but all enterprise plans support federation. Learn more about signing up for Office 365 at Office 365.

2. Set up AD and enable directory synchronization.
3. Create a federated domain and prove ownership of it.
4. Install and configure PingFederate 8.4 or later.

   > **Note**
   >
   > If you need to support active clients, such as native desktop applications, for use with Office 365, ensure that PingFederate is installed with a license that enables the WS-Trust Security Token Service (STS).

5. Replace the default self-signed SSL server certificate included with PingFederate with one that is signed by a public certificate authority (CA).

   This enables Azure AD to establish a trusted SSL session with PingFederate. Learn more about Manage SSL server certificates in the PingFederate documentation.

Preparing Active Directory for federation

When you configure federation, your local Active Directory (AD) users should be provisioned to Azure AD. You can do this by using synchronization tools like Azure AD Connect.

> **Important**
>
> The following installation steps are provided for a configuration where the `objectGUID` attribute is selected for `ImmutableID`. If you're using different attribute for this purpose, such as `msDS-ConsistencyGuid`, ensure to align it accordingly.

Copyright © 2025 Ping Identity Corporation

Before configuring synchronization, you might need to prepare your local AD, including:

- Configuring UPN suffix for non-routable domain names
- Cleaning up AD objects
- Planning filters for AD users
- Planning for multiple forests

> **Tip**
>
> To simplify your configuration, Microsoft provides Azure AD Connect, which will automate many of the required steps. You can download Azure AD Connect at Microsoft Azure Active Directory Connect.
> Azure AD Connect automatically configures your Azure AD domain and exports a configuration file that provides the settings needed to complete the federated single sign-on connection from PingFederate. If you're using Azure AD Connect and have the configuration file, proceed to install and configure PingFederate to complete your setup.

Configuring a federated domain

Add a new federated domain for your account.

About this task

After signing up for Office 365, the only domain associated with your account is the onmicrosoft.com subdomain chosen during registration, such as contoso.onmicrosoft.com. To enable single sign-on (SSO) to Azure AD and Office 365, you should have another domain added to the environment.

If you've already added and verified such a domain, skip to Step 2.

> **Tip**
>
> Running the Azure AD Connect tool and following its prompts makes these required configuration changes automatically. The steps outlined here can be run manually if required.

Steps

1. Add a federated domain to your account: Authenticate to Office 365 using the `Connect-MsolService` PowerShell `cmdlet` and enter the same credentials used when authenticating to the Microsoft Online Services portal.

   Choose from:

   - Add a new domain using Azure AD or Office 365 Admin Portals. Learn more in the following sections of the Microsoft documentation:
     - Add a domain to Microsoft 365
     - Add your custom domain name using the Azure Active Directory portal
   - Add a new domain manually with PowerShell.

Copyright © 2025 Ping Identity Corporation

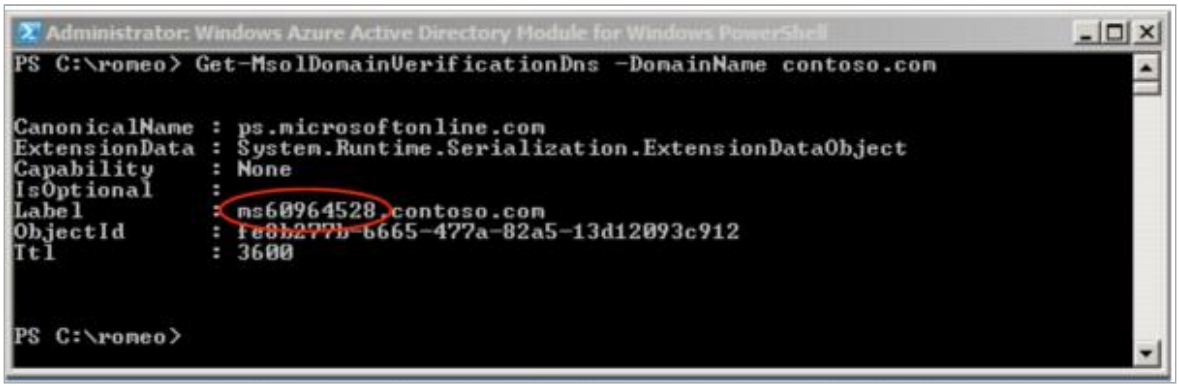
> **Tip**
>
> You can load this and the other `cmdlets` described here by launching PowerShell from the **Microsoft Azure Active Directory Module for Windows PowerShell** desktop and **Start** menu shortcuts.

1. To add a new domain, run the `New-MsolDomain -Name <name> -Authentication Managed` command.
2. To get DNS verification records for the new domain, run the `Get-MsolDomainVerificationDns -DomainName <name>` command.
3. To prove that you control the domain, use the output of the `Get-MsolDomainVerificationDNS` command to create a `.txt` record on the DNS server of the domain used in the previous step.

   > **Note**
   >
   > This server must be accessible over the Internet so that Microsoft servers can resolve and access them.

   The DNS record name should match the Domain Name and the DNS record value should be `MS=<ms portion of the Label>`.

   The following is an example from the `Get-MsolDomainVerificationDNS` command.

   > **Note**
   >
   > Creating a DNS record value can vary between different DNS host providers. Learn more about how to Add a domain to Microsoft 365 in the Microsoft documentation.

   Example Values for Creating a Text Record

Record Type	Alias or hostname	Destination or Points to Address	TTL
`.txt`	`@ or jkdoctest.com`	`MS=ms60016396`	`1 Hour`
`MX`	`@ or jkdoctest.com`	`Ms60016396.msv1.invalid.outlook.com`	`1 Hour`

4. To prove your control of the domain, run the `Confirm-MsolDomain -DomainName <name>` command.

2. Complete the steps in Enabling federated authentication.

Copyright © 2025 Ping Identity Corporation

3. Complete the steps in Configuring multiple domains.
4. To verify that the domain settings are up to date and in effect, run the `Get-MsolDomainFederationSettings -DomainName <name>` command.
5. To change domain settings after the domain is created and verified, run the `Set-MsolDomainFederationSettings -DomainName` command with extra arguments for the settings that you want to change.

Enabling federated authentication

The default authentication type in the Azure AD or Office 365 domain is **managed**, which means that access is provided only for Azure AD cloud user identities. To allow usage of the on-prem user accounts, change the authentication type to **federated**.

About this task

> **Tip**
>
> Running the Azure AD Connect tool and following its prompts makes these required configuration changes automatically. The steps outlined here can be run manually if required.

You can complete the configuration manually using the `Set-MsolDomainAuthentication` PowerShell cmdlet. When you run it, you must provide the URLs for PingFederate, the public portion of its signing certificate, and some other inputs.

> **Note**
>
> The `IssuerURI` parameter should be unique so that Office 365 can identify your identity provider (IdP).

Steps

1. Export the signing certificate from PingFederate:
    1. Go to **Server Configuration > Signing & Decryption Keys & Certificates**.
    2. Export the applicable signing certificate to the local file system, such as `C:\temp\pf-signing.crt`.
    3. When prompted, select **Certificate Only**.
2. Use PowerShell to remove unnecessary lines from the certificate file and configure federation for your Azure AD domain.

    Example:

Copyright © 2025 Ping Identity Corporation

```
$certFile = "C:\temp\pf-signing.crt"
$cert = [IO.File]::ReadAllText($certFile)
$cert = $cert.replace("-----BEGIN CERTIFICATE-----","")
$cert = $cert.replace("-----END CERTIFICATE-----","")
$cert = $cert.replace("`r","")
$cert = $cert.replace("`n","")
$domainName = "<Federated Domain Name>"
$hostName = "<Hostname>.$domainName"
$port = 9031
${pingfed} = "https://${hostName}:$port"
$brandName = "<Federated Domain Alias>"
$issuer = "<WS-Federation Realm or Virtual Server ID>"
$spId = "urn:federation:MicrosoftOnline"
$activeLogOn = "$pingfederate/idp/sts.wst"
$logOff = "$pingfederate/idp/prp.wsf"
$metaData = "$pingfederate/pf/sts_mex.ping?PartnerSpId=$spId"
$passiveLogOnPF="$pingfederate/idp/prp.wsf"
Set-MsolDomainAuthentication -Authentication Federated -DomainName
"$domainName" -ActiveLogOnUri  "$activeLogOn" -FederationBrandName
"$brandName" -IssuerUri "$issuer"  -LogOffUri "$logOff" -MetadataExchangeUri
"$metaData" -PassiveLogOnUri  "$passiveLogOnPF" -SigningCertificate "$cert"
```

Learn more about the **`Set-MsolDomainAuthentication`** command in Set-MsolDomainAuthentication in the Microsoft documentation.

3. To verify that the domain settings are up-to-date and in effect, run the **`Get-MsolDomainFederationSettings -DomainName <name>`** command.

> **Tip**
>
> If you have multiple subdomain accounts in Office 365, you can connect to them in one service provider (SP) connection using multiple virtual server IDs in PingFederate 7.2 or later.

Configuring multiple domains

About this task

A single PingFederate server connection can support multiple Azure AD and Office 365 domains. This functionality relies on using Virtual Server ID (VSID) feature.

The VSID value overrides the WS-Federation Realm value configured in the server settings. It should be created for each domain you're going to federate with and added to the connection configuration.

The same VSID value should be used for the `IssuerURI` parameter while configuring federation settings on the Azure AD side. The encoded VSID value should be added to all endpoint URIs, so the incoming requests can be processed correctly by the PingFederate server.

> **Note**
>
> If you use the Azure AD Connect wizard to federate with PingFederate server, it generates the VSID value for each specified domain. This parameter is listed in the configuration file, which should be exported from the wizard to obtain required PingFederate server settings. The Azure AD Connect wizard adds the encoded VSID value to all corresponding URIs while completing the Azure AD side configuration.

Copyright © 2025 Ping Identity Corporation

In a situation where you decide to use manual configuration, if both the engineering and marketing departments of contoso.com have their own departmental subdomains, engineering.contoso.com and marketing.contoso.com, they are both registered in Azure AD or Office 365 under the parent domain, contoso.com. As an example, their `IssuerUri` values are `Engineering` and `Marketing`, respectively.

> **Note**
>
> When you run the **`Set-MsolDomainAuthentication`** PowerShell command, you must include the base64-encoded value of the VSID presenting that subdomain in the paths for the `ActiveLogOnUri`, `LogoffUri`, `PassiveLogOnUri`, and `MetadataExchangeUri` parameters.

Steps

1. Construct a JSON object containing a key-value pair of the virtual server ID in the format of `{"vsid":"<VirtualServerIdValue>"}`.

   Example:

   `{"vsid":"Engineering"}`

2. Base64url-encode the JSON object.

   Example:

   `eyJ2c2lkIjoiRW5naW5lZXJpbmcifQ`

3. Insert the base64url-encoded value between `/idp` or `/pf` and the rest of the respective endpoint for `ActiveLogOnUri`, `LogoffUri`, `PassiveLogOnUri`, and `MetadataExchangeUri`.

   Example:

   ```
   $activeLogOn = “$pingfederate/idp/eyJ2c2lkIjoiRW5naW5lZXJpbmcifQ/sts.wst”
   $logOff = “$pingfederate/idp/eyJ2c2lkIjoiRW5naW5lZXJpbmcifQ/prp.wsf”
   $metaData = "$pingfederate/pf/eyJ2c2lkIjoiRW5naW5lZXJpbmcifQ/sts_mex.ping?PartnerSpId=$spId"
   $passiveLogOnPF = “$pingfederate/idp/eyJ2c2lkIjoiRW5naW5lZXJpbmcifQ/prp.wsf”
   ```

4. Repeat step 1-3 for each subdomain.

   Learn more about base64url in RFC 4648 on the Internet Engineering Task Force (IETF) website.

Setup

Install and configure PingFederate

Using PingFederate for SSO to Azure Active Directory (AD) makes it a key component of daily operations.

When using PingFederate in this way, it’s important that the deployment be highly available. Learn more about installation and high availability in the PingFederate Server Clustering Guide. After installing PingFederate, configure it for use with Azure AD by creating an identity provider (IdP) adapter and a WS-Federation connection at minimum.

Copyright © 2025 Ping Identity Corporation

Configuring PingFederate to omit line breaks in digital signatures

Configure PingFederate to omit line breaks in digital signatures by performing the following steps.

Steps

1. Open `<pf_install>/pingfederate/bin/run.properties` in a text editor.
2. If not already present, add a new line to the file as follows: `org.apache.xml.security.ignoreLineBreaks=true`
3. Save the file.
4. Stop PingFederate and restart it to pick up the change.

Next steps

If running a cluster, you must apply these steps on all nodes.

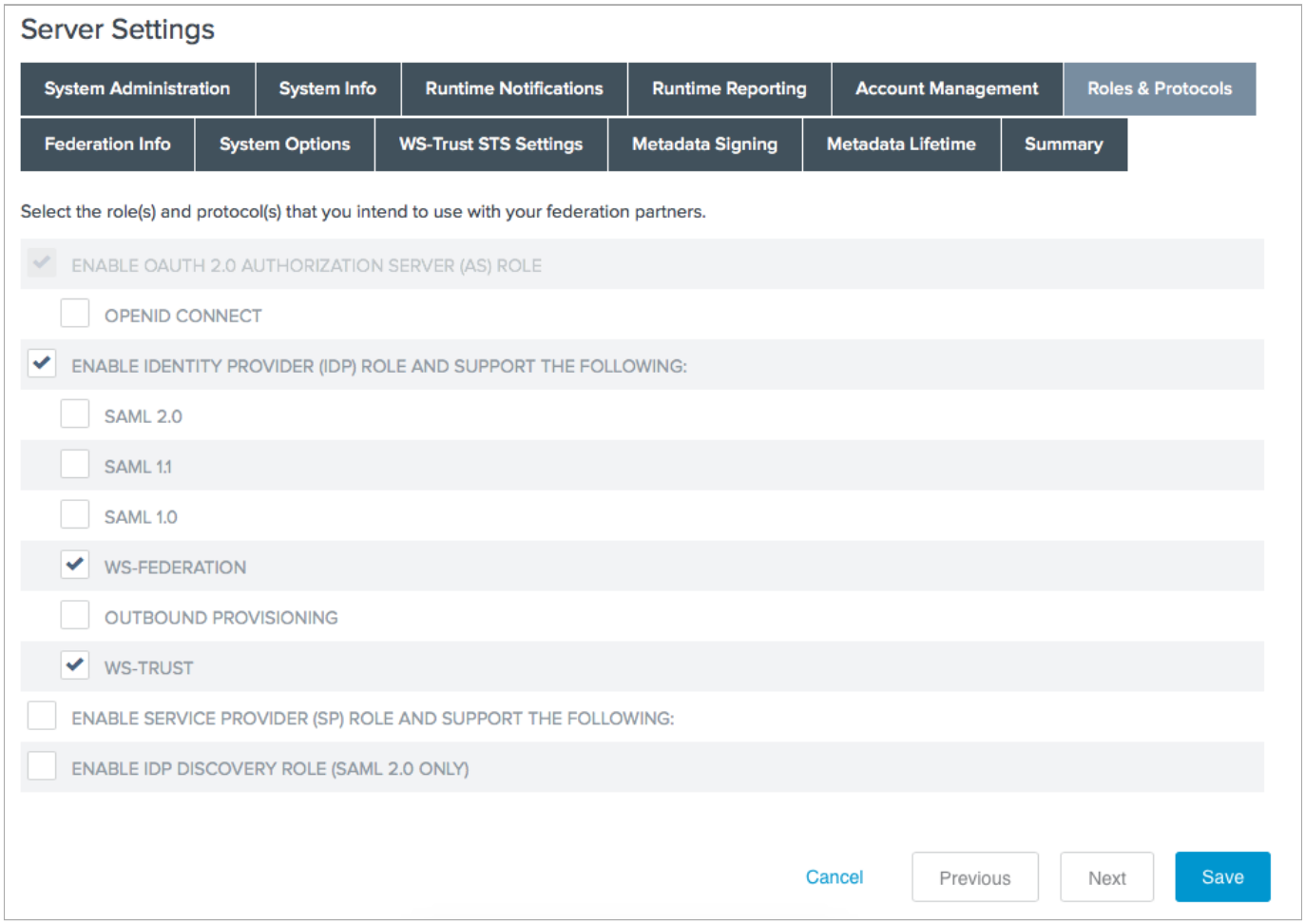
Enabling server protocols

Steps

1. In the PingFederate administrative console, go to **Server Settings > Roles and Protocols**.
2. Ensure that **WS-Federation** is selected under the IdP role.
3. Select **WS-Trust** if you need to support active clients.

   This protocol selection is available only if an STS-enabled license is installed.

Copyright © 2025 Ping Identity Corporation



Replacing the default SSL certificates

When PingFederate is installed, the included SSL certificate is self-signed. If left as is, email and other Office 365 applications won't work because Microsoft's WS-Trust clients aren't able to establish an SSL connection with PingFederate.

Steps

- To ensure proper functionality, ensure you replace the default SSL certificate with one that is signed by a public certificate authority (CA).

  Learn more about instructions for Manage SSL server certificates in the PingFederate documentation.

Configuring an LDAP connection

If you're planning to provide SSO to users whose accounts reside in a directory server, ensure you have an LDAP data store defined for it in PingFederate.

Copyright © 2025 Ping Identity Corporation

About this task

Learn more about Datastores in the PingFederate documentation.

> **Note**
>
> Ensure the `objectGUID` attribute is set to binary. It must be a binary attribute to create a connection to Office 365. Learn more in Creating a connection to Azure Active Directory.

> **Note**
>
> If you need to support multiple Office 365 subdomain accounts through one SP connection in PingFederate 7.2 or later, create additional LDAP data store connections to LDAP servers of the subdomains.

Steps

1. In the PingFederate administrative console go to **Server Configuration > System Settings > Data Stores**.
2. Click **Add New Data Store**.
3. Select LDAP in the **Data Store Type** tab and click **Next**.
4. Populate the fields in the **LDAP Configuration** tab.
    1. In the **Hostname(s)** field enter the DNS name or IP address of the data store, which might include a port number such as `181.20.42.130:389`. For failover, you can enter multiple LDAP servers, each separated by a space.
    2. In the **LDAP Type** field, select **Active Directory**.
    3. Enter the **User DN** and password of a user account with read permission in Active Directory.
5. Click **Advanced** and then click the **LDAP Binary Attributes** tab.
6. In the **Binary Attribute Name** field, enter `objectGUID` and click **Add**.
7. Click **Done** and then click **Next**.
8. Review the summary and click **Save**.

Creating a password credential validator

After configuring an LDAP connection, the next step is to create an LDAP password credential validator (PCV).

Copyright © 2025 Ping Identity Corporation

Steps

- Complete the steps in Configuring the LDAP Username Password Credential Validator in the PingFederate documentation.

> **Tip**
>
> If you need to support multiple Office 365 subdomain accounts using one SP connection in PingFederate 7.2 or later, create additional LDAP PCVs against your LDAP data store connections. Learn more in Configuring an LDAP connection.

Configuring an IdP adapter

PingFederate supports a wide selection of integration kits that plug into the PingFederate server enabling it to interface with various identity management systems. After authentication, PingFederate can look up more attributes in various data stores to collect additional information that is placed in the SAML token passed to Office 365.

About this task

Regardless of which integration kit is used or the source of the attributes, two things need to be provided to Office 365:

User Principal Name (UPN)

Format as an email address and the domain name must match the domain name registered with Office 365. For example, if the domain contoso.com is created using the `New-MsolDomain` PowerShell command, then the UPN attribute value in the SAML assertion for all users must be their username followed by @contoso.com.

> **Note**
>
> The UPN of the user in AD can be different from what is placed in the SAML assertion created by PingFederate.

ImmutableID

The Azure AD Connect copies this ID to the cloud when it creates Azure AD accounts. The ImmutableID, which uniquely represents the user in AD, is an immutable identifier used to associate local and remote identities. The AD attribute is a binary value and must be base-64 encoded to be transmitted in a SAML token.

> **Note**
>
> The expected value can be determined by examining the **ImmutableID** attribute output by the `Get-MsolUser` PowerShell command after synchronization is set up.

PingFederate packages an HTML Form adapter that renders a simple HTML form where users can enter their username and password. This credential can be checked against AD using the previously configured password credential validator. Follow these steps to set up this adapter in PingFederate.

Copyright © 2025 Ping Identity Corporation

> **Note**
>
> These installation steps are provided for a configuration where `objectGUID` attribute is selected for **ImmutableID**. If you are using different attribute for this purpose (such as `msDS-ConsistencyGuid`), be sure to align it accordingly.

Steps

1. In the PingFederate administrative console got to **My IdP Configuration > Adapters**.
2. On the **Manage IdP Adapter Instances** tab, click **Create New Instance**.
3. On the **Type** tab, enter an **Instance Name** and **Instance ID** and select **HTML Form IdP Adapter** as the **Type**.
4. On the **IdP Adapter** tab, click the **Add a new row to 'Credential Validators'**.
5. In the **Password Credential Validator Instance** list, select the validator ID you previously configured and click **Update**.
6. Click **Next**.
7. On the **Adapter Attributes** tab, select the **username** checkbox under **Pseudonym** and click **Next**.
8. On the **Summary** tab, click **Done**.
9. Click **Save** to complete the HTML Form IdP Adapter configuration.

Configuring a username token processor instance

To allow email clients, mobile phones, and other active clients that use Office 365 to authenticate, users must provide the username and password of their AD domain account.

About this task

> **Note**
>
> This configuration isn't required for browser-only implementations, such as passive WS-Federation.

For this credential to be verified, Office 365 relays them to PingFederate using the WS-Trust protocol. For the username and password to be validated, a username token processor is set up to bind to the domain controller. Whenever requests are sent to PingFederate, they include a UsernameToken element that PingFederate passes along for authentication.

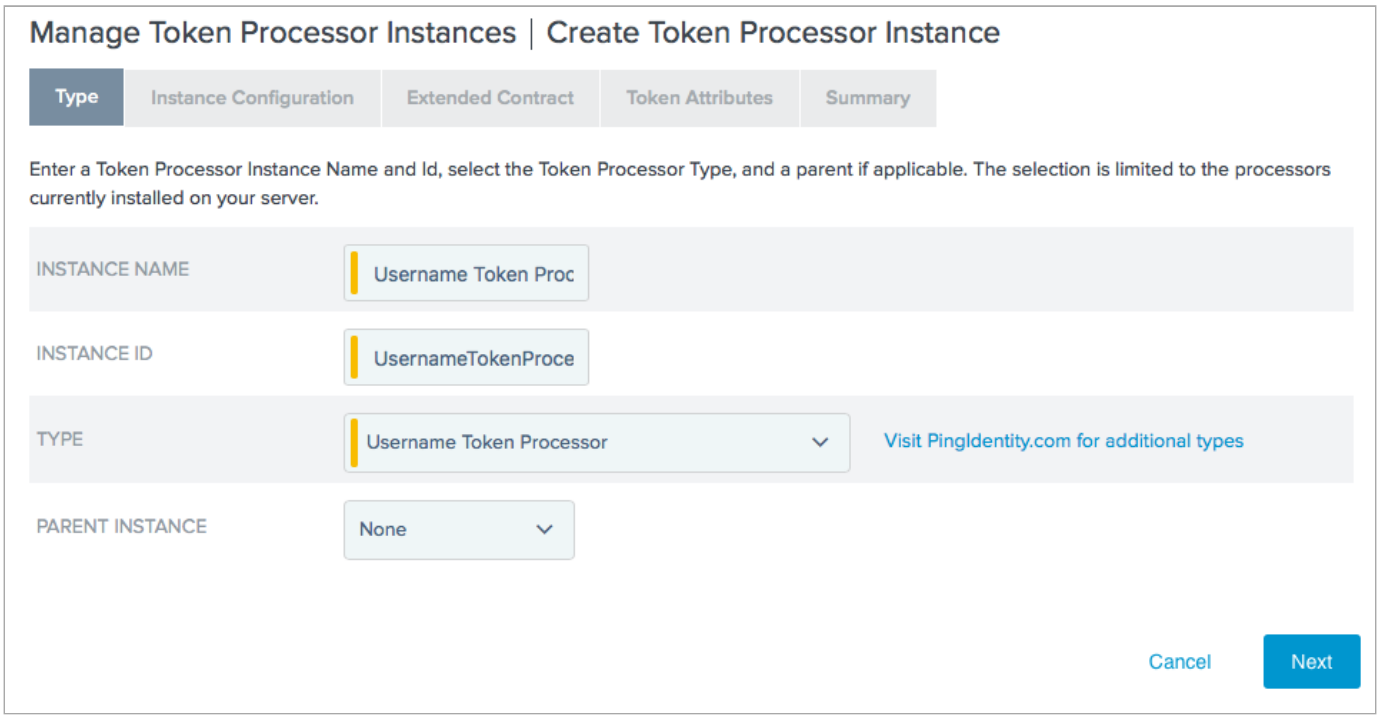
> **Tip**
>
> With PingFederate 6.11 or later, you can also configure the Kerberos token processor to allow the STS to accept and validate Kerberos tokens and to enable SSO for clients that support Kerberos authentication.

Copyright © 2025 Ping Identity Corporation

Steps

1. In the PingFederate administrative console, go to **Authentication > Token Exchange > Token Processors**.
2. Click **Create New Instance**.
3. On the **Type** tab, in the **Instance Name** field, enter a name for the token processor.
4. In the **Instance ID** field, enter an ID.
5. In the **Type** list, select **Username Token Processor**.

> **Note**
>
> For PingFederate 7.2 or later, select **Username Token Processor** from in the **Type** list and follow the steps in the Configuring a Username Token Processor instance section of the PingFederate documentation. When finished, skip to step 8.



6. Click **Next**.
7. On the **Instance Configuration** tab, select the LDAP Password Credential Validation instance that was previously configured.
8. Click **Next** on both the **Instance Configuration** and **Token Attributes** tabs.
9. Click **Done** on the **Summary** tab.
10. Click **Save** on the **Manage Token Processors** tab.

Copyright © 2025 Ping Identity Corporation

> **Tip**
>
> If you need to support multiple Office 365 subdomain accounts using one SP connection in PingFederate 7.2 or later, repeat steps 1-6 to create additional token processors against your LDAP password credential validators. Learn more in Creating a password credential validator.

Enable Kerberos authentication

Creating the Kerberos Token Processor is an optional step, but it's necessary if you're going to use certain features for users located inside your corporate network.

Here are some examples:

- Automatic Office 365 ProPlus registration
- Automatic Azure AD device registration for Windows 10 devices
- Seamless legacy authentication for desktop clients like Skype for Business
- Integrated Active Directory authentication for Azure SQL server

Creating a connection to Azure Active Directory

Setting up single sign-on (SSO) in PingFederate is done on a connection-by-connection basis.

Steps

1. In the PingFederate administrator console, create a new SP connection:
    - For PingFederate 10.1 or later: Go to **Applications > Integration > SP Connections**. Click **Create Connection**.
    - For PingFederate 10.0 or earlier: Go to **Identity Provider > SP Connections**. Click **Create Connection**.
2. Configure the connection protocols.
    1. On the **Connection Template** tab, select **Do not use a template for this connection**. Click **Next**.
    2. On the **Connection Type** tab select **Browser SSO Profiles**.
    3. In the **Protocol** list, select **WS-Federation**.
    4. In the **WS-Federation Token Type** list, select **SAML 1.1**.
    5. If you want to support active federation, select the **WS-Trust STS** check box.

> **Note**
>
> Passive federation is done with browser redirects and a sign-on form, whereas active federation allows an application to take responsibility for supplying credentials without further user interaction.

Copyright © 2025 Ping Identity Corporation

6. Click **Next**.
7. On the **Connection Options** tab, click **Next**.

3. On the **General Info** tab, configure the basic connection information.
    1. In the **Partner's Realm (Connection ID)** field, enter `urn:federation:MicrosoftOnline`.
    2. In the **Connection Name** field, enter a name of your choosing.
    3. In the **Virtual Server IDs** field, enter the domain name to be used with Office 365, such as `contoso.com`, or leave this field blank to use the system default.

        This domain name must match the issuer entity ID from the PowerShell script described in Configuring a federated domain. When this field is blank, the connection uses the PingFederate system default. You must set either the **Virtual Server ID** or the system default.

        Learn more about setting the system default in Specifying federation information in the PingFederate documentation.

        > **Note**
        >
        > To support multiple Office 365 subdomain accounts with one service provider (SP) connection, add a virtual server ID for each subdomain. Each virtual server ID value should match the issuer entity ID (`IssuerUri`) of the respective subdomain account in Office 365.

    4. Click **Next**.
4. On the **Browser SSO** tab, complete the steps in Configuring browser SSO and then click **Next**.
5. If you see the **WS-Trust STS** tab, complete the steps in Configuring WS-Trust STS and then click **Next**.
6. On the **Credentials** tab, configure the connection credentials.
    1. Click **Configure Credentials**.
    2. On the **Digital Signature Settings** tab, in the **Signing Certificate** list, select the certificate that you want to use for the Office 365 connection.

        > **Note**
        >
        > If you need to create a new certificate, click **Manage Certificates**. Learn more about Managing digital signing certificates and decryption keys in the PingFederate documentation.

    3. Click **Next**.
    4. On the **Summary** tab, click **Done**.
    5. On the **Credentials** tab, click **Next**.
7. On the **Activation and Summary** tab, above the **Summary** section, click the toggle to turn on the connection. Click **Save**.

Copyright © 2025 Ping Identity Corporation

Configuring browser SSO

Configure the single sign-on (SSO) details in your connection to Azure Active Directory.

About this task

Learn more about Configuring IdP Browser SSO in the PingFederate documentation.

Steps

1. On the **Browser SSO** tab, click **Configure Browser SSO**.
2. On the **Assertion Lifetime** tab, click **Next**.
3. On the **Assertion Creation** tab, complete the steps in Configuring assertion creation.
4. On the **Protocol Settings** tab, configure the protocol settings and then click **Next**.
    1. Click **Configure Protocol Settings**.
    2. On the **WS-Trust Version** tab, click **Next**.
    3. On the **Service URL** tab, in the **Endpoint URL** field, enter `https://login.microsoftonline.com/login.srf`.
    4. In the **Valid Domain Name** field, enter `*.microsoftonline.com`.
    5. Select the **Require HTTPS** and **Allow Any Query/Fragment** checkboxes. Click **Add**.
    6. Click **Next**.
    7. On the **Summary** tab, click **Done**.
5. On the **Browser SSO > Summary** tab, click **Done**.

Configuring assertion creation

Configure the assertion by mapping attributes from the necessary sources.

Steps

1. On the **Assertion Creation** tab, click **Configure Assertion Creation**.
2. On the **Identity Mapping** tab, select **User Principal Name**. Click **Next**.
3. On the **Attribute Contract** tab, add the following attributes and then click **Next**.

Attribute	Attribute Name Format
`ImmutableID`	http://schemas.microsoft.com/LiveID/Federation/2008/05
`UPN`	http://schemas.xmlsoap.org/claims
`SAML_NAME_FORMAT`	http://schemas.xmlsoap.org/claims

Copyright © 2025 Ping Identity Corporation

> **Note**
>
> The `SAML_NAME_FORMAT` is a pseudo-attribute that is used to define a specific attribute name format for `SAML_SUBJECT`. The current setting is ignored and a text value specified in the attribute contract fulfillment later on is used as the attribute name format for `SAML_SUBJECT`.

4. On the **Authentication Source Mapping** tab, map your HTML Form adapter instance.

   Learn more about how to map your HTML form adapter instance in Mapping an HTML form adapter instance.

5. On the **Assertion Creation > Summary** tab, click **Done**.

Configuring attribute source and user lookup for HTML Form Adapter instances

Complete the following to configure attribute source and user lookups for the HTML Form Adapter instances.

Steps

1. On the **Attribute Sources & User Lookup** tab, click **Add Attribute Source**.
2. On the **Data Store** tab, enter an attribute source ID and description of your choosing.
3. In the **Active Data Store** list, select the datastore connection that you created in Configuring an LDAP connection.
4. On the **LDAP Directory Search** tab, in the **Base DN** field, enter the base DN that contains the users whose attributes you want to retrieve.

   Example:

   ```
   CN=Users,DC=contoso,DC=com
   ```

5. In the **Root Object Class** list, select **<Show All Attributes>**.
6. In the **Attribute** list, select **objectGUID**. Click **Add Attribute**.
7. In the **Attribute** list, select **userPrincipalName**. Click **Add Attribute**. Click **Next**.
8. On the **LDAP Binary Attribute Encoding Types** tab, click **Next**.
9. On the **LDAP Filter** tab, enter a filter to limit the search and then click **Next**.

   Example:

   ```
   (|(sAMAccountName=${username})(userPrincipalName=${username}))
   ```

   > **Note**
   >
   > In this context, `${username}` contains the username from the HTML Form Adapter that PingFederate presents during browser-based SSO.

10. Click **Done**.

Mapping an HTML form adapter instance

Copyright © 2025 Ping Identity Corporation

Complete the following to map an HTML form adapter instance.

Steps

1. On the **Authentication Source Mapping** tab, click **Map New Adapter Instance**.
2. On the **Adapter instance** tab, in the **Adapter Instance** list, select the HTML Form Adapter instance that you created in Configuring an IdP adapter. Click **Next**.
3. On the **Virtual Server IDs** tab, if you have a single domain, click **Next**.
4. If you configured virtual server IDs on the **General Info** tab of the connection and have a separate HTML Form Adapter instances for each subdomain:
    1. Select the **Restrict Virtual Server IDs** checkbox.
    2. Select the checkbox for the virtual server ID that represents the subdomain associated with this HTML Form Adapter instance.
    3. Click **Next**.
5. If you configured virtual server IDs on the **General Info** tab of the connection and use a single HTML Form Adapter instances for all subdomains:
    1. Leave the **Restrict Virtual Server IDs** checkbox unselected. Click **Next**.
    2. (Recommended) When you reach the **Issuance Criteria** tab later in the setup process, create an OGNL expression to protect against unauthorized access.
6. On the **Mapping Method** tab, select **Retrieve additional attributes from multiple data stores using one mapping**. Click **Next**.
7. On the **Attribute Sources & User Lookup** tab, complete the steps in Configuring attribute source and user lookup for HTML Form Adapter instances, and then click **Next**.
8. On the **Attribute Contract Fulfillment** tab, create the following mappings and then click **Next**.

Attribute Contract	Source	Value
`ImmutableID`	**LDAP (*<Your datastore>*)**	**objectGUID**
`SAML_SUBJECT`	**LDAP (*<Your datastore>*)**	**objectGUID**
`UPN`	**LDAP (*<Your datastore>*)**	**userPrincipalName**
`SAML_NAME_FORMAT`	**Text**	`urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified`

9. If you use a single HTML Form Adapter instance for users in multiple subdomains, on the **Issuance Criteria** tab, create an OGNL expression verify the virtual server ID and other conditions, such as group membership.

Copyright © 2025 Ping Identity Corporation

> **Note**
>
> Learn more about OGNL expressions in Defining issuance criteria for IdP Browser SSO, Enabling and disabling expressions, and Constructing OGNL expressions in the PingFederate documentation.

10. Click **Next**
11. On the **Summary** tab, click **Done**.

Configuring WS-Trust STS

If you enabled WS-Trust STS to support active federation, configure the WS-Trust protocol and token settings.

Steps

1. On the **WS-Trust STS** tab, click **Configure WS-Trust STS**.
2. On the **Protocol Settings** tab,
    1. In the **Partner Service Identifier** field, enter `urn:federation:MicrosoftOnline`. Click **Add**.
    2. In the same field, enter the active logon URI.

       To get this value, take the URL that was passed as the `ActiveLogOnUri` parameter to the `Set-MsolDomainFederationSettings` PowerShell `cmdlet`, then remove the URI scheme.

       Example:

       ```
       pf01.contoso.com:9031/idp/sts.wst
       ```

    3. From the **Default Token Type** list, select **SAML 1.1 for Office 365**.
    4. Click **Next**.
3. On the **Token Lifetime** tab, click **Next**.
4. On the **Token Creation** tab, configure the token.
    1. Click **Configure Token Creation**.
    2. On the **Attribute Contract** tab, map the following attributes and then click **Next**.

Attribute	Attribute Name Format
`ImmutableID`	**http://schemas.microsoft.com/LiveID/Federation/2008/05**
`UPN`	**http://schemas.xmlsoap.org/claims**

Copyright © 2025 Ping Identity Corporation

Attribute	Attribute Name Format
`SAML_NAME_FORMAT`	**http://schemas.xmlsoap.org/claims**

3. On the **Request Contract** tab, click **Next**.
4. On the **IdP Token Processor Mapping** tab, click **Map New Token Processor Instance**.
5. On the **Token Processor Instance** tab, from the **Token Processor Instance** list, select **UsernameTokenProcessor**. Click **Next**.
6. If you have virtual server IDs for multiple subdomains: On the **Virtual Server IDs** tab, select the **Restrict Virtual Server IDs** checkbox.
7. Select the checkbox for the virtual server ID that represents the subdomain associated with this token processor. Click **Next**.

   Learn more about Restricting a token processor to certain virtual server IDs in the PingFederate documentation.

   > **Tip**
   >
   > If you have a single token processor for users across multiple subdomains, create an OGNL expression to verify the virtual server ID and other conditions, such as group membership.
   > Learn more about OGNL expressions in Defining issuance criteria for IdP Browser SSO, Enabling and disabling expressions, and Constructing OGNL expressions in the PingFederate documentation.

8. On the **Attribute Retrieval** tab, select **Retrieve additional attributes from data stores to fulfill the attribute contract**. Click **Next**.
9. On the **Attribute Sources & User Lookup** tab, complete the steps in Configuring attribute source and user lookup for token processors and then click **Next**.
10. On the **Attribute Contract Fulfillment** tab, create the following mappings and then click **Next**.

Attribute Contract	Source	Value
`ImmutableID`	**LDAP (<*Your datastore*>)**	**objectGUID**
`SAML_NAME_FORMAT`	**Text**	`urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified`
`TOKEN_SUBJECT`	**LDAP (<*Your datastore*>)**	**objectGUID**
`UPN`	**LDAP (<*Your datastore*>)**	**userPrincipalName**

11. If you use a single token translator instances for users in multiple subdomain, on the **Issuance Criteria** tab, create an OGNL expression to verify the virtual server ID and other conditions, such as group membership.

Copyright © 2025 Ping Identity Corporation

> **Note**
>
> Learn more about OGNL expressions in Defining issuance criteria for IdP Browser SSO, Enabling and disabling expressions, and Constructing OGNL expressions in the PingFederate documentation.

12. Click **Next**.
13. On the **Summary** tab, click **Done**.
14. If you have more token processor instances (username or Kerberos token processors), repeat steps d–l for each one.

> **Note**
>
> When completing step f, each username token processor must have a different virtual server ID.

15. On the **Token Creation > IdP Token Processor Mapping** tab, click **Next**.
16. On the **Summary** tab, click **Done**.

5. On the **WS-Trust STS > Token Creation** tab, click **Next**.
6. On the **Summary** tab, click **Done**.
7. On the **SP Connection > WS-Trust STS** tab, click **Next**.

Configuring attribute source and user lookup for token processors

Complete the following to configure attribute source and user lookups for the HTML Form Adapter instance and token processor parts of the setup process.

Steps

1. On the **Attribute Sources & User Lookup** tab, click **Add Attribute Source**.
2. On the **Data Store** tab, enter an attribute source ID and description of your choosing.
3. From the **Active Data Store** list, select the datastore connection that you created in Configuring an LDAP connection.
4. On the **LDAP Directory Search** tab, in the **Base DN** field, enter the base DN that contains the users whose attributes you want to retrieve. For example, `CN=Users,DC=contoso,DC=com`.
5. From the **Root Object Class** list, select **<Show All Attributes>**.
6. From the **Attribute** list, select **objectGUID**. Click **Add Attribute**.
7. From the **Attribute** list, select **userPrincipalName**. Click **Add Attribute**. Click **Next**.
8. On the **LDAP Binary Attribute Encoding Types** tab, click **Next**.
9. On the **LDAP Filter** tab, enter a filter to limit the search, and then click **Next**.

   Example filter for a username token processor:

```
(|(sAMAccountName=${username})(userPrincipalName=${username}))
```

Copyright © 2025 Ping Identity Corporation

Example filter for a Kerberos token processor:

```
userPrincipalName=${principal}
```

> **Note**
>
> In this context, the username is always sent as a UPN.

10. Click **Done**.

Troubleshooting

Testing your configuration

Perform the following steps to test browser-based access to Office 365 applications, such as the team portal.

Steps

1. Open a browser window and navigate to Office 365.
2. In the sign on window, in the **User ID** field, specify the address of a user that has been synchronized.
3. Click the **Sign in at <your federated domain brand name>** link.
4. When presented with a sign on form, enter the credentials of the user and click **Submit**.

Result

Your browser should be redirected to the Office 365 team portal and you should be logged in. If you receive an error stating that the user doesn't have a license to use Office 365, check that the user has been allocated a license. You might also need to grant the user rights if you're testing SharePoint Online. After making these changes, retest.

Next steps

To test access to the federated instance of Office 365 using Microsoft Outlook or some other client that supports POP3 or IMAP4, follow the instructions specific to the application you're using. Learn more about the server settings you need to configure the client in Office 365 in POP, IMAP, and SMTP settings for Outlook.com in the Microsoft Outlook documentation.

After you find the correct POP/IMAP server settings and configure your client, try to send and receive email. This should trigger a prompt for a password. This password relays back to PingFederate through WS-Trust, so monitor its log files for activity. If you don't see any activity and receive an error in your email program about the password not being valid, the SSL certificate that PingFederate is using is probably self-signed. Check that this SSL certificate is signed by a CA that Microsoft trusts and then retest.

Performance considerations

Testing performed in-house has shown that an Outlook client using the default configuration, on average, makes between 9 and 12 requests to the PingFederate STS endpoint per hour.

Copyright © 2025 Ping Identity Corporation

Additionally, the first time an editor is opened, a request is made to the PingFederate STS endpoint.

To help size your deployment, performance results for the username and Kerberos token processors are documented in the PingFederate Capacity Planning Guide and PingFederate Performance Tuning Guide.

Troubleshooting

The following links provide help for troubleshooting sign-in issues with Microsoft Lync.

- Troubleshooting Azure AD connection configuration errors: Ping Identity support article for troubleshooting Azure AD connection configuration errors.
- Troubleshooting Skype for Business Online sign-in errors for administrators: Describes how to troubleshoot sign-in issues with Microsoft Skype for Business Online. Details how to clear the credential cache that can occur on some operating systems.
- Troubleshoot sign-in issues for Lync for Mac 2011 in Skype for Business Online: Describes how to troubleshoot common issues between Lync for Mac and Office 365.

Azure AD Identity Protection Integration Kit

The Azure AD Identity Protection Integration Kit allows PingFederate to communicate with Azure AD Identity Protection for risk-based authentication.

By sending a Microsoft user ID to Azure AD Identity Protection when a user signs on, PingFederate can get a security Result based on the user's history. You can use this to dynamically adjust the authentication requirements. For example, you could configure your PingFederate authentication policy to require multifactor authentication (MFA) when a user with a high-risk level signs on.

Features

- Uses the Azure AD Identity Protection "riskyUsers" resource
- Supports the PingFederate PingFederate authentication API
- Supports the JavaScript Widget for the PingFederate Authentication API

Components

- Azure AD Identity Protection IdP Adapter:
  - When a user signs on through PingFederate, the adapter sends the user ID to Azure AD Identity Protection.
  - The adapter receives the user's Result and makes it available in the PingFederate authentication policy.

Intended audience

This document is intended for PingFederate administrators.

Copyright © 2025 Ping Identity Corporation

Learn more about the setup process with the following resources:

- The following sections of the PingFederate documentation:
  - Managing IdP adapters
  - Authentication policies
- The following sections of the Azure AD Identity Protection documentation:
  - What is Identity Protection?
  - What is risk
  - How To: Investigate risk

System requirements

- PingFederate 9.3 or later.
- A valid Azure AD Identity Protection license. Learn more about License requirements in the Azure AD Identity Protection documentation.
- This integration uses the Microsoft Login Integration Kit to get Microsoft user IDs. Setup details are provided in Setting up the Microsoft Login Integration Kit.

Setup

Setting up the Microsoft Login Integration Kit

The Azure AD Identity Protection IdP Adapter needs a user's Microsoft user ID to request their risk information. To make the user ID available, prompt your users authenticate with Microsoft as part of the sign-on flow.

About this task

Follow the Microsoft Login Integration Kit documentation to download and configure the Microsoft IdP Adapter, with the following details:

Steps

1. Complete the steps in Deploying the integration files.
2. Complete the steps in Registering PingFederate as an application in Azure with the following details:
   - For **Supported Account Types**, select **Single tenant** or one of the **Multitenant** options.

     You can use Microsoft sign on for users with personal accounts or from other domains, but Azure AD Identity Protection risk levels will be available only for users from the "single tenant" domain.
3. In Azure, grant risk permissions to your application as shown in the **Configure API permissions** section of Azure Active Directory Identity Protection and the Microsoft Graph PowerShell SDK in the Azure AD Identity Protection documentation.

Copyright © 2025 Ping Identity Corporation

4. Complete the steps in Configuring an adapter instance and note your **Tenant ID**, **Client ID**, and **Client Secret**.

   You'll use these same credentials when configuring your Azure AD Identity Protection IdP Adapter in Configuring an adapter instance.

Next steps

Set up the Azure AD Identity Protection Integration Kit by continuing to Deploying the integration files.

Deploying the integration files

To get started with the integration, deploy the Azure AD Identity Protection Integration Kit files to your PingFederate directory.

About this task

> **Note**
>
> If you operate PingFederate in a cluster, the following steps refer to the console node.

Steps

1. Download the Azure AD Identity Protection Integration Kit `.zip` archive from the Ping Identity Integration Directory.
2. Stop PingFederate.
3. If you're upgrading an existing deployment, delete the `pf-azure-ad-identity-protection-adapter-<version>.jar` file from your `<pf_install>/pingfederate/server/default/deploy` directory.
4. Extract the `.zip` archive and merge the contents of the `dist` directory with your `<pf_install>/pingfederate` directory.
5. Start PingFederate.
6. If you operate PingFederate in a cluster, repeat steps 2-5 for each engine node.

Configuring an adapter instance

Configure the Azure AD Identity Protection IdP Adapter to determine how PingFederate communicates with Azure AD Identity Protection.

Steps

1. In the PingFederate administrative console, create a new IdP adapter instance.

   Choose from:

   - For PingFederate 10.1 or later: go to **Authentication > Integration > IdP Adapters**. Click **Create New Instance**.
   - For PingFederate 10.0 or earlier: go to **Identity Provider > Adapters**. Click **Create New Instance**.

2. On the **Type** tab, set the basic adapter instance attributes:

   1. In the **Instance Name** field, enter a name for the adapter instance.

Copyright © 2025 Ping Identity Corporation

2. In the **Instance ID** field, enter a unique identifier for the adapter instance.
3. From the **Type** list, select **Azure AD Identity Protection IdP Adapter**. Click **Next**.

3. On the **IdP Adapter** tab, in the **Azure AD Identity Protection API Response Mappings** section, map user data from the Azure AD Identity Protection response to the attribute contract:

   These attributes become available in your PingFederate authentication policy.

   1. Click **Add a new row to 'Azure AD Identity Protection API Response Mappings'**.
   2. In the **Local Attribute** field, enter a name of your choosing for an attribute.
   3. In the **Azure Attribute Mapping** field, enter the JSON Pointer syntax for the value of the matching Azure AD Identity Protection attributes as shown in JSON Pointer syntax reference. Alternately, leave the field blank to include the entire response as the value.
   4. In the **Action** column, click **Update**.
   5. To add more attributes, repeat steps a-d.

4. On the **IdP Adapter** tab, configure the adapter instance by referring to Azure AD Identity Protection IdP Adapter settings reference. Click **Next**.
5. On the **Extended Contract** tab, add any attributes that you included in the **Azure Response Mappings** section of the **IdP Adapter** tab. Click **Next**.
6. On the **Adapter Attributes** tab, set pseudonym and masking options as shown in Set pseudonym and masking options in the PingFederate documentation. Click **Next**.
7. On the **Adapter Contract Mapping** tab, configure the contract fulfillment details for the adapter as shown in Define the IdP adapter contract in the PingFederate documentation. Click **Next**.
8. On the **Summary** tab, check and save your configuration.

   Choose from:

   - For PingFederate 10.1 or later: click **Save**.
   - For PingFederate 10.0 or earlier: click **Done**. On the **Manage IdP Adapter Instances** tab, click **Save**.

JSON Pointer syntax reference

JavaScript Object Notation (JSON) Pointer defines a syntax for identifying a specific value within a JSON payload. Using the sample payload and JSON Pointer examples below, identify the attributes that you want to use to populate your attribute contract.

Learn more about a complete technical description of JSON Pointer syntax in JavaScript Object Notation (JSON) Pointer on ietf.org.

Learn more about the list of all possible values for each attribute in Risky user in the Azure AD documentation.

Copyright © 2025 Ping Identity Corporation

Example JSON payload

```
{
   "@odata.type": "#microsoft.graph.riskyUser",
   "id": "41a31b00-3b3b-42d9-8f1c-6d4f14e74c69",
   "isDeleted": false,
   "isProcessing": false,
   "riskLastUpdatedDateTime": "2019-05-03T03:50:34.9565578Z",
   "riskLevel": "low",
   "riskState": "remediated",
   "riskDetail": "userPerformedSecuredPasswordReset",
   "userDisplayName": "Allan Deyoung",
   "userPrincipalName": "AllanD@contoso.OnMicrosoft.com"
}
```

JSON Pointer syntax

Description	JSON Pointer	Example value
Risk level	`/riskLevel`	`low`
Risk state	`/riskState`	`remediated`
User name	`/userDisplayName`	`Allan Deyoung`

To populate an attribute with the entire JSON response, leave the **Azure Response Mappings** field blank.

Azure AD Identity Protection IdP Adapter settings reference

The following are setting descriptions for the Azure AD Identity Protection IdP Adapter.

Standard fields

Field	Description
Tenant ID	The tenant ID that you noted in Setting up the Microsoft Login Integration Kit. This field is blank by default.
Client ID	The client ID that you noted in Setting up the Microsoft Login Integration Kit. This field is blank by default.
Client Secret	The client secret that you noted in Setting up the Microsoft Login Integration Kit. This field is blank by default.

Copyright © 2025 Ping Identity Corporation

Advanced fields

Field	Description
Microsoft Token Base URL	The Microsoft authentication base URL. If Microsoft changes this URL, enter the new URL. The default value is `https://login.microsoftonline.com`.
Microsoft Graph API Base URL	The Microsoft Graph API base URL. If Microsoft changes this URL, enter the new URL. The default value is `https://graph.microsoft.com/v1.0`.
Microsoft Token Endpoint	The Microsoft endpoint used to get an access token. If Microsoft changes this URL, enter the new URL. The default value is `/oauth2/v2.0/token`.
Scopes	The scopes to request from Microsoft. If Microsoft changes this URL, enter the new URL. Separate multiple scopes with a comma. The default value is `https://graph.microsoft.com/.default`.
API Request Timeout	The amount of time in milliseconds that PingFederate allows when establishing a connection with Azure AD Identity Protection or waiting for a response to a request. A value of 0 disables the timeout. The default value is `5000`.
Proxy Settings	Defines proxy settings for outbound HTTP requests. The default value is **System Defaults**.
Custom Proxy Host	The proxy server host name to use when **Proxy Settings** is set to **Custom**. This field is blank by default.
Custom Proxy Port	The proxy server port to use when **Proxy Settings** is set to **Custom**. This field is blank by default.

Adding review statuses to your authentication policy

By modifying your PingFederate authentication policy to include the Result from Azure AD Identity Protection, you can dynamically change authentication requirements based on security risk level.

About this task

These steps are designed to help you add to an existing authentication policy. Learn more general information about configuring authentication policies in PingFederate authentication API in the PingFederate documentation.

Copyright © 2025 Ping Identity Corporation

Steps

1. In the PingFederate administrative console, go to the **Policies** tab.

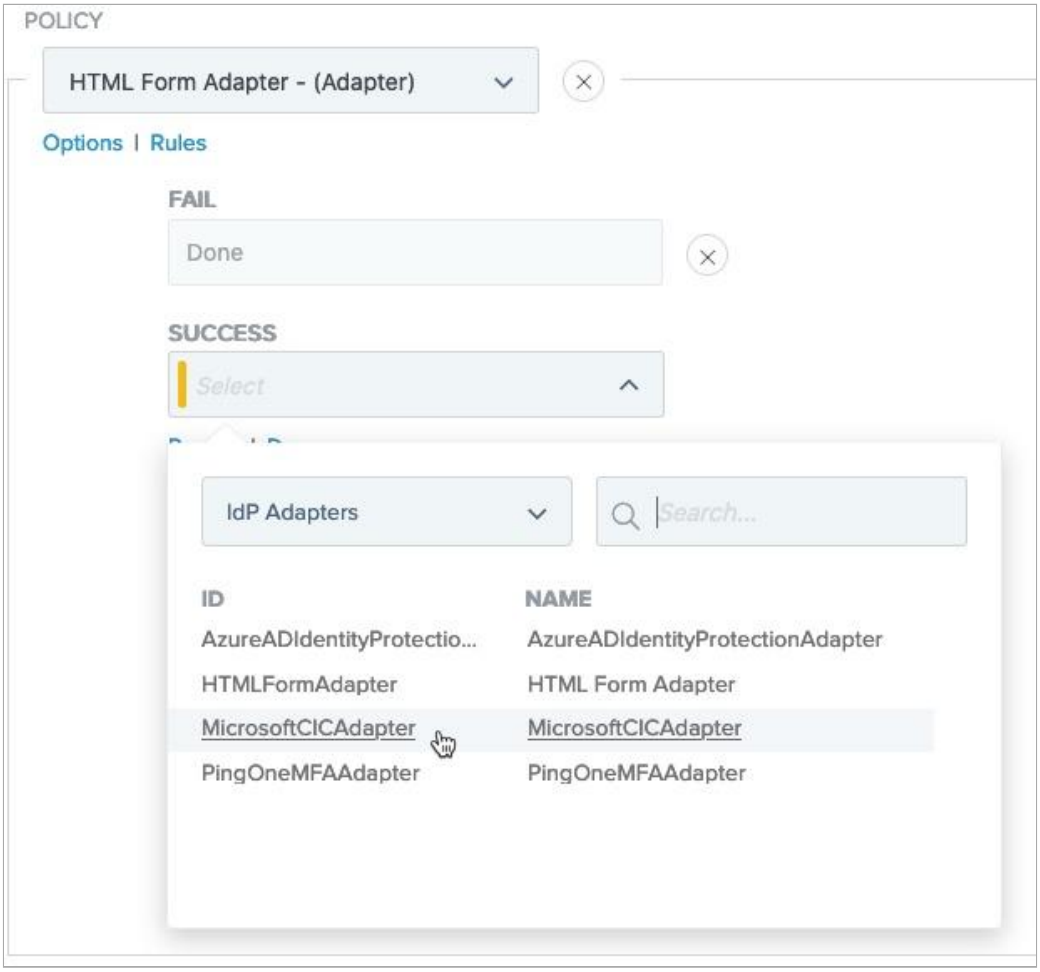
   Choose from:

   - For PingFederate 10.1 or later: go to **Authentication > Policies > Policies**.
   - For PingFederate 10.0 or earlier: go to **Identity Provider > Authentication Policies > Policies**.

2. Select the **IdP Authentication Policies** check box.
3. Open an existing authentication policy, or click **Add Policy**.

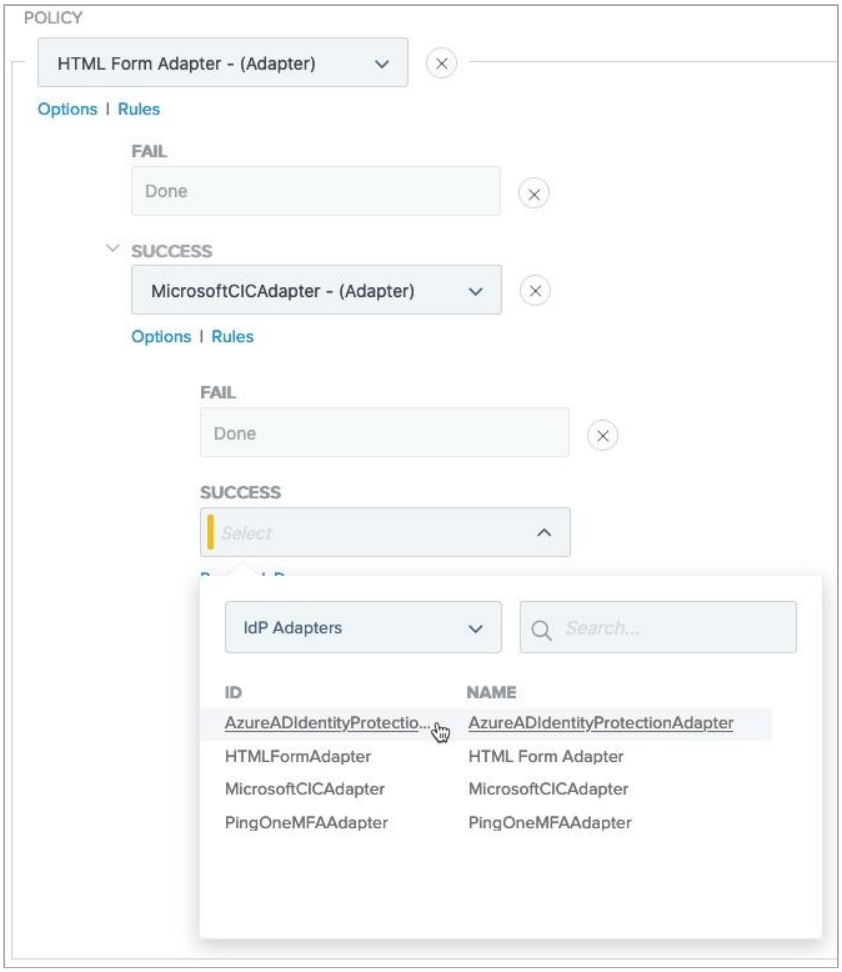
   Learn more about Defining authentication policies in the PingFederate documentation.

4. In the **Policy** area, select the Microsoft IdP Adapter instance that you created in Setting up the Microsoft Login Integration Kit.

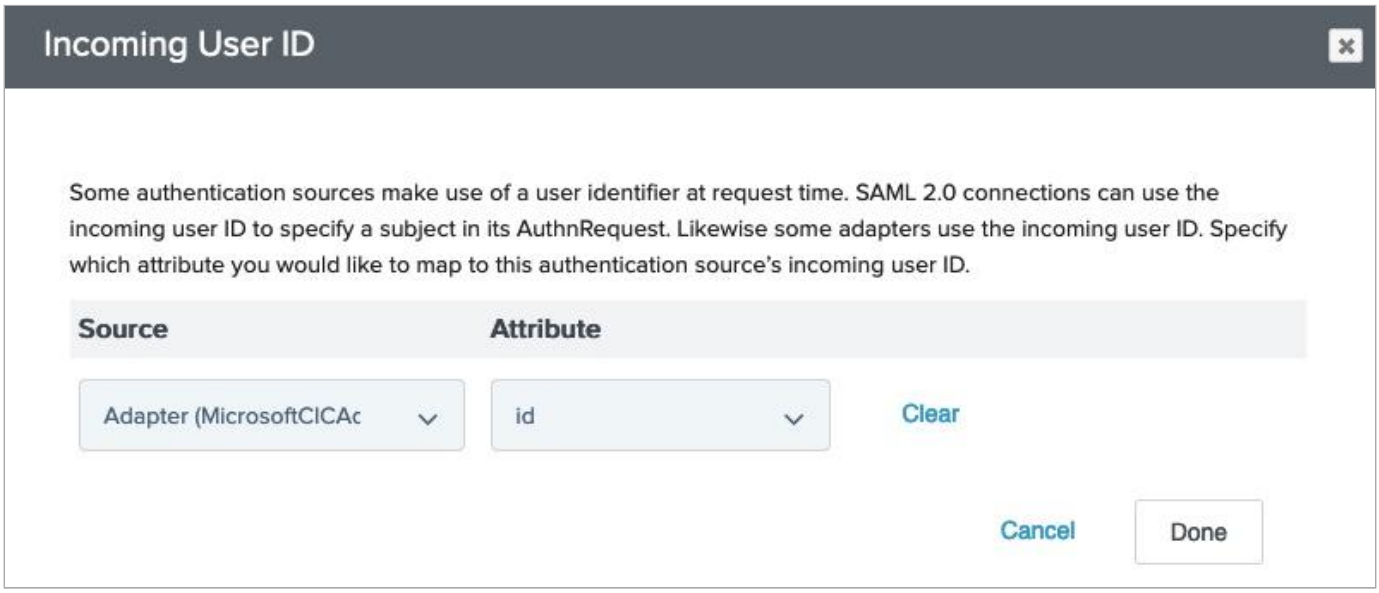


5. In the **Success** branch following the Microsoft IdP Adapter instance, select your Azure AD Identity Protection IdP Adapter instance.

Copyright © 2025 Ping Identity Corporation



6. Map the Microsoft user ID into the Azure AD Identity Protection IdP Adapter instance.

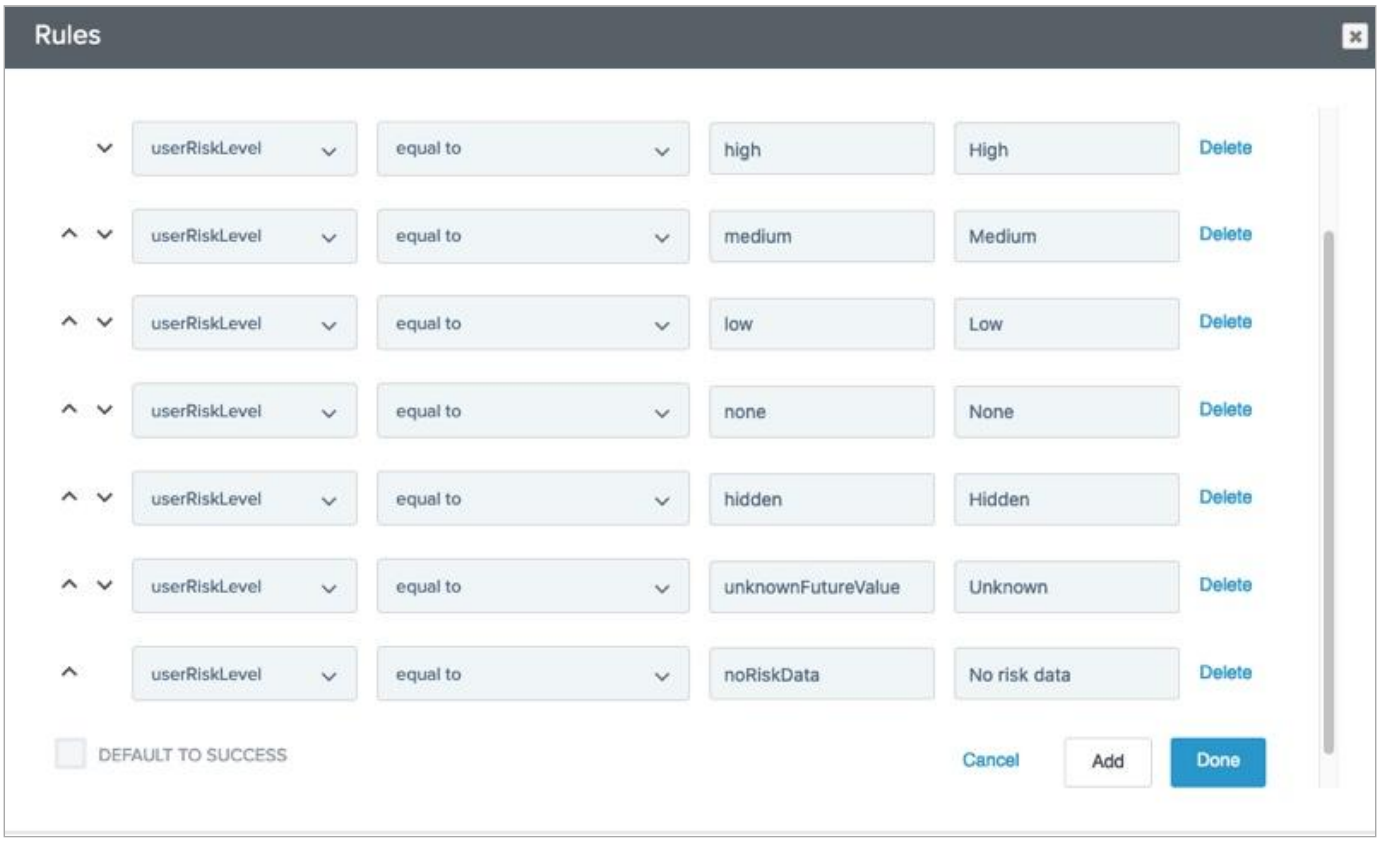


1. Under the Azure AD Identity Protection IdP Adapter instance, click **Options**.
2. On the **Options** dialog, from the **Source** list, select your Microsoft IdP Adapter instance.

Copyright © 2025 Ping Identity Corporation

3. From the **Attribute** list, select the **id**. Click **Done**.

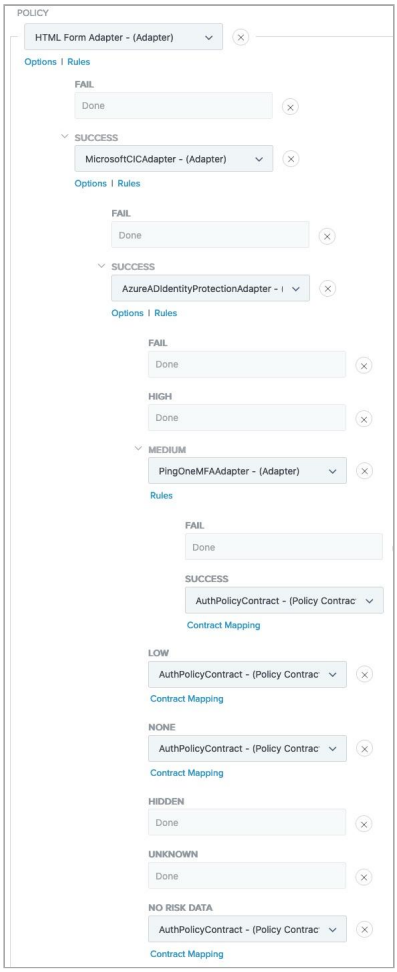
7. Define policy paths based on the information provided by Azure AD Identity Protection:



1. Under the Azure AD Identity Protection IdP Adapter instance, click **Rules**.
2. On the **Rules** dialog, from the Attribute Name list, select **userRiskLevel**.
3. From the **Condition** list, select **equal to**.
4. In the **Value** field, enter "low", "medium", or "high", or one of the utility values "none", "hidden", "unknownFutureValue", or "noRiskData". The "noRiskData" value is set by the adapter when it does not find risk data for the user.
5. In the **Result** field, enter a name. This appears as a new policy path that branches from the Azure AD Identity Protection IdP Adapter.
6. If you want to add more authentication paths, click **Add** and repeat steps b-e.
7. **Optional:** Clear the **Default to success** checkbox.
8. Click **Done**.

8. Configure each of the authentication paths.

Copyright © 2025 Ping Identity Corporation



9. Click **Done**. In the **Policies** window, click **Save**.

Authentication API support

You can use the PingFederate authentication API to integrate the Azure AD Identity Protection IdP Adapter into your application.

The PingFederate Authentication API provides access to the current state of the authentication flow as a user steps through the PingFederate authentication policy. Learn more about PingFederate authentication API in the PingFederate documentation.

There are no states or actions for the Azure AD Identity Protection IdP Adapter in the authentication API, so you don't need to modify your application to use it. By including it in your authentication policy, it will be triggered automatically.

Troubleshooting

Enabling debug logging

To help with troubleshooting or monitoring, you can turn on activity logging for PingFederate, the Azure AD Identity Protection IdP Adapter, or both.

Copyright © 2025 Ping Identity Corporation

About this task

This task is optional. You can use logging for troubleshooting or analytics.

Learn more about logging in Enabling debug messages and console logging in the PingFederate documentation.

Steps

1. Open the `<pf_install>/pingfederate/server/default/conf/log4j2.xml` file for editing.
2. To log activity for PingFederate and all adapters:
    1. Find the following section in the file.

```
<AsyncRoot level="INFO" includeLocation="false">
        <!-- <AppenderRef ref="CONSOLE" /> -->
        <AppenderRef ref="FILE" />
</AsyncRoot>
```

    2. Change `INFO` to `DEBUG`.

       The following code snippet shows `DEBUG` in bold for visibility.

```
<AsyncRoot level="DEBUG" includeLocation="false">
        <!-- <AppenderRef ref="CONSOLE" /> -->
        <AppenderRef ref="FILE" />
</AsyncRoot>
```

    3. **Optional:** To see the adapter activity in the console and the log file, remove the comment tags (`<!--` and `-->`) that surround the `CONSOLE` line.

```
<AsyncRoot level="INFO" includeLocation="false">
        <AppenderRef ref="CONSOLE" />
        <AppenderRef ref="FILE" />
</AsyncRoot>
```

3. If you want to log activity relating to the Azure AD Identity Protection IdP Adapter, do one of the following:

> **Tip**
>
> You can use this information with a third-party log analysis tool to monitor for important events, such as when a sign-on event has a high-risk Result.

Choose from:

- To log activity for the Azure AD Identity Protection IdP Adapter and for its HTTPS and component activity, add the following line.

Copyright © 2025 Ping Identity Corporation

```
<Logger name="com.pingidentity.adapters.azure.ad.identity.protection" level="DEBUG"/>
```

- To log activity for the adapter's HTTPS activity and other components, but not for the adapter itself, add the following line.

```
<Logger name="com.pingidentity.adapters.azure.ad.identity.protection.shade" level="DEBUG"/>
```

- To log activity for the Azure AD Identity Protection IdP Adapter, but not for its HTTPS or component activity, add the following lines.

```
<Logger name="com.pingidentity.adapters.azure.ad.identity.protection" level="DEBUG"/>
<Logger name="com.pingidentity.adapters.azure.ad.identity.protection.shade" level="INFO"/>
```

4. Save the file.

Release notes

Changelog

The following is the change history for the Azure AD Identity Protection Integration Kit.

Azure AD Identity Protection Integration Kit 1.0 - October 2021

- Initial release.
- Added support for getting Result from the Azure AD Identity Protection "riskyUsers" resource.
- Added the ability to map any Azure AD Identity Protection API response attribute to an attribute in the PingFederate authentication policy.
- Added support for the PingFederate authentication API.
- Added support for the JavaScript Widget for the PingFederate Authentication API.
- Added settings for API connection and request timeouts.
- Added settings to override the PingFederate system-default proxy settings.

Known issues and limitations

The following are known issues or limitations for the Azure AD Identity Protection Integration Kit.

Known issues

There are no known issues.

Copyright © 2025 Ping Identity Corporation

Known limitations

There are no known limitations.

Download manifest

The following files are included in the Azure AD Identity Protection Integration Kit `.zip` archive.

- `Legal.pdf` : Copyright and license information.
- `dist/pingfederate/server/default` : Contains the integration files.
  - `deploy` : Contains the Java libraries.
    - `pf-azure-ad-identity-protection-adapter-<version>.jar` : The Azure AD Identity Protection IdP Adapter.

Copyright © 2025 Ping Identity Corporation

Box Provisioner

The Box Provisioner allows PingFederate to integrate with Box for provisioning and single sign-on (SSO).

Features

- Manages users in Box based on changes in an external data store that is attached to PingFederate.
  - Creates, updates, disables, and deletes users.
  - Allows you to enable the create, update, disable, and delete capabilities independently.
  - Allows you to provision disabled users.
  - Allows you to choose whether to disable or delete users when deprovisioning.
- Manages groups in Box based on changes in an external data store that is attached to PingFederate.
  - Creates and deletes groups.
  - Updates group memberships.
- Enables browser-based SSO initiated by the service provider (SP) or identity provider (IdP).
- Pre-populates some connection settings with the included quick connection template.

Intended audience

This document is intended for PingFederate administrators.

If you need help during the setup process, see the following sections of the PingFederate documentation:

- SP connection management
- Identity provider SSO configuration
- Datastores
- Configuring outbound provisioning
- Configuring outbound provisioning settings

System requirements

- PingFederate 7.3 or later.
- A Box administrator account.
- To allow PingFederate to make outbound connections to the Box API, you might need to allow the following domain in your firewall:
  - https://account.box.com
  - https://api.box.com

Copyright © 2025 Ping Identity Corporation

User and group management

The Box Provisioner synchronizes users and groups from your datastore to Box. The following describes the behavior of each provisioning capability.

> **Tip**
>
> You can configure the following capabilities and specify which users to provision when you get to the Configure provisioning part of the setup process.

Synchronizing existing users

PingFederate synchronizes users based on the `mail` attribute in Box. If a user already exists in your datastore and Box, mapping this attribute correctly links the two records together.

For example:

- In Box, Janet's `mail` is `jsmith@example.com`.
- In your datastore, Janet's `mail` is `jsmith@example.com`.
- On the **Attribute Mapping** tab of your provisioning connection configuration, map the `mail` attribute to `mail`.
- When the provisioning connector runs, the datastore user is provisioned with a `mail` of `jsmith@example.com`. That matches Janet's existing `mail` in Box, so her information in the datastore is synchronized to her Box account.

User provisioning

PingFederate provisions users when any of the following happens:

- A user is added to the datastore group or filter that is targeted by the provisioning connector.
- A user with `disabled` status is added to the datastore group or filter that is targeted by the provisioning connector, and the **Provision disabled users** provisioning option is enabled. This feature is not available in all provisioning connector versions.

You can define which users PingFederate targets for provisioning on the **Source Location** tab of your provisioning connection configuration.

User updates

PingFederate updates users when a user attribute changes in your datastore.

You can define which attributes PingFederate monitors for changes on the **Attribute Mapping** tab of your provisioning connection configuration.

User deprovisioning

PingFederate deprovisions users when any of the following happens:

- A user is deleted from the user store.

Copyright © 2025 Ping Identity Corporation

- A user is disabled in the user store.
- A user is removed from the datastore group or filter that is targeted by the provisioning connector.

The **Remove User Action** setting in the connection configuration determines whether the deprovisioning action disables or deletes the user.

Synchronizing existing groups

PingFederate synchronizes groups from the datastore to the target service based on the group name.

For example:

- In Box, there is a group is named `Accounting`.
- In your datastore, there is a group with a `CN` of `Accounting`.
- When the provisioning connector runs, the two groups are synchronized.

Group provisioning

PingFederate provisions groups when a group is added to the datastore filter that is targeted by the provisioning connector.

You can define which groups PingFederate targets for provisioning and monitors for changes on the **Source Location** tab in your provisioning connection configuration.

Group name updates

PingFederate renames groups when they are renamed in the datastore.

Group membership updates

PingFederate updates group memberships when memberships change in the datastore, whether the change is in the group's properties or a user's properties.

Group memberships in the datastore overwrite the group memberships in Box.

Group deletion

PingFederate deletes groups when any of the following happens:

- The group is deleted in the datastore.
- The group is removed from the datastore group or filter that is targeted by the provisioning connector.

> **Important**
>
> Group deletions are permanent and cannot be undone.

Setup

Copyright © 2025 Ping Identity Corporation

Obtain client ID and secret from Box

About this task

The Box Connector's outbound provisioning functionality is built using Box's REST API, which requires an OAuth 2.0 access token for authentication. To obtain the access token, you will need to first obtain an app key and secret from Box.

Steps

1. Sign on to your Box account as an administrative user for your organization.

   https://app.box.com/developers/console

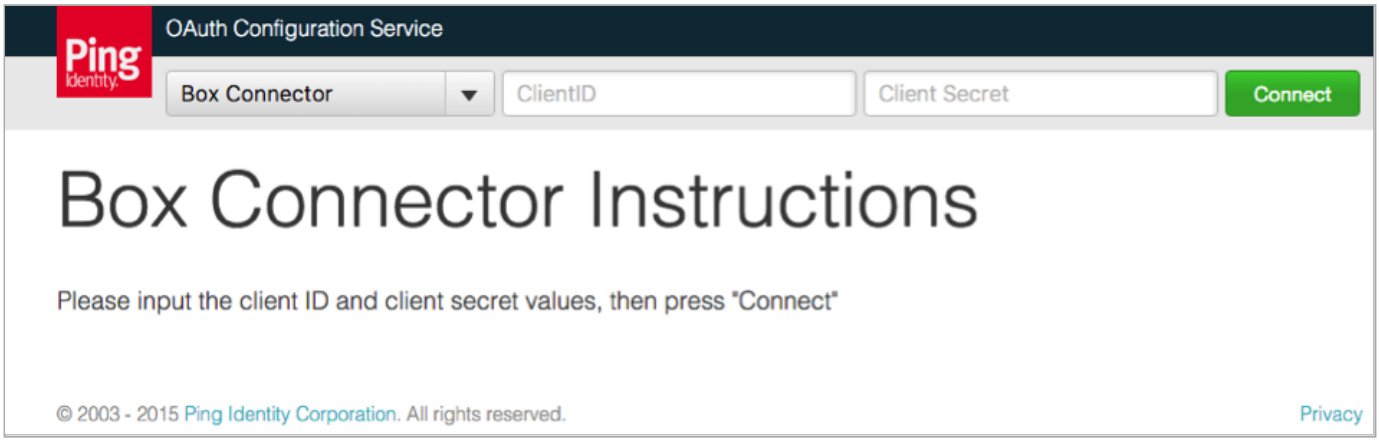
2. Go to **Dev Console**.
3. Go to **My Apps**.
4. Select **Create New App**.
5. Select **Enterprise Integration**.
6. On the **Authentication Method** screen, select **Standard OAuth 2.0 (User Authentication)**.
7. Give your application a name, such as 'PingFederate Provisioning'.
8. In the **OAuth 2.0 Credentials** section, copy the **Client ID** and **Client Secret** values for later use.
9. Update the **Redirect URI** field with the following URL:
   - https://oauth.pingone.com/ocs/ppm/rest/v1/oauth/oastempcredresponse/
10. Click **Save Changes**.

Generate OAuth access and refresh tokens

Steps

1. Go to the Ping OAuth Configuration Service at the following URL: https://oauth.pingone.com/ocs/ppm/rest/v1/oauth/oasrequestform
2. In the list, select **Box Connector**.

Copyright © 2025 Ping Identity Corporation

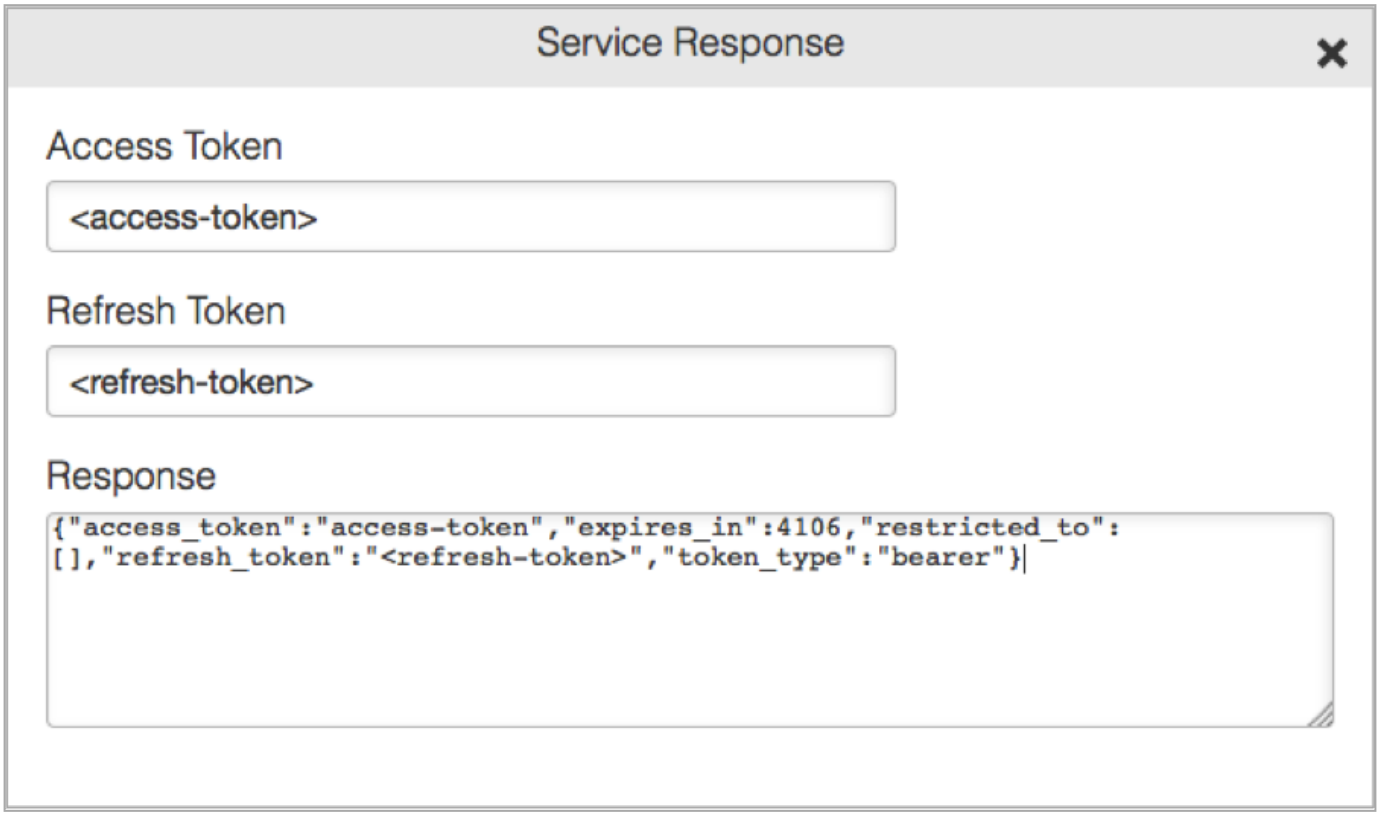


3. Enter the **Client ID** value you previously obtained into the **ClientID** field.
4. Enter the **Client Secret** value you previously obtained into the **Client Secret** field.
5. Click the **Connect** button.
6. Sign on to Box as an administrative user for your organization.

> **Note**
>
> If you are already signed on to Box you will not be asked to sign on again. Make sure that the account you are signed in under is an administrative account.

7. Click **Grant access to Box** to generate your access and refresh tokens.
8. You should be redirected to the OAuth Configuration Service and presented with an access token and refresh token.

Copyright © 2025 Ping Identity Corporation



9. Copy the **Access Token** and **Refresh Token** to use when configuring the Box Connector.

Download Box SAML 2.0 metadata file

About this task

This connector's quick-connection template uses a metadata `.xml` file from Box to assist in configuring many settings in the SP connection. When asked during the SP connection configuration steps, import the `boxmetadata.xml` that you downloaded from Box.

Steps

1. Download the SAML 2.0 metadata from Box.
2. Save the `.xml` file to a desired location.

Upgrade an existing connector

Steps

1. Before stopping the PingFederate server to upgrade the Box Connector, access the **Attribute Mapping** screen for existing channel configurations and note the current configuration.

Copyright © 2025 Ping Identity Corporation

> **Warning**
>
> The upgrade process removes existing mappings and defaults on the **Attribute Mapping** screen. These must be configured again before activating the channel configuration.

2. Disable the existing service provider (SP) connection where the Box Connector is configured.
3. Delete the existing Box Connector SP connection and save.
4. Stop the PingFederate server if it is running.
5. Unzip the new Box Connector distribution `.zip` archive into a holding directory.
6. Remove any older versions of `pf-box-quickconnection-[version].jar` from: `<pf_install>/pingfederate/server/default/deploy`
7. From the `dist` directory of the new version of the connector, copy the `pf-box-quickconnection-[version].jar` file into the `<pf_install>/pingfederate/server/default/deploy` directory.
8. Start the PingFederate server.
9. Create a new SP connection, using **Box Connector** as the connection template.
10. Follow the instructions in Configure provisioning and Updating Box OAuth tokens to configure the connection.
11. Access the **Attribute Mapping** screen for existing channel configurations and click **Refresh Fields**.
12. Ensure all new required fields (if any), are mapped appropriately or have a default value.
13. When you have completed the attribute configuration, click **Done**, **Done**, and **Save**.
14. Activate the SP connection to resume outbound provisioning.

Install the connector

About this task

This section describes the common steps required to install the PingFederate Box Connector.

If you experience issues with deployment, installation or configuration, visit Ping Identity Support & Community.

Steps

1. Stop the PingFederate server if it is running.
2. Extract the Box Connector distribution `.zip` archive.
3. Copy the contents of the `dist` directory into the `<pf_install>/pingfederate/server/default/deploy` directory.
4. (Optional) If you plan to use the connector for outbound provisioning, edit the `run.properties` file located in `<pf_install>/pingfederate/bin`, changing the property `pf.provisioner.mode` to `STANDALONE`. For example:

   ```
   pf.provisioner.mode=STANDALONE
   ```

Copyright © 2025 Ping Identity Corporation

> **Note**
>
> Learn more about using the FAILOVER setting for runtime deployment in Deploying provisioning failover.

5. Start the PingFederate server.

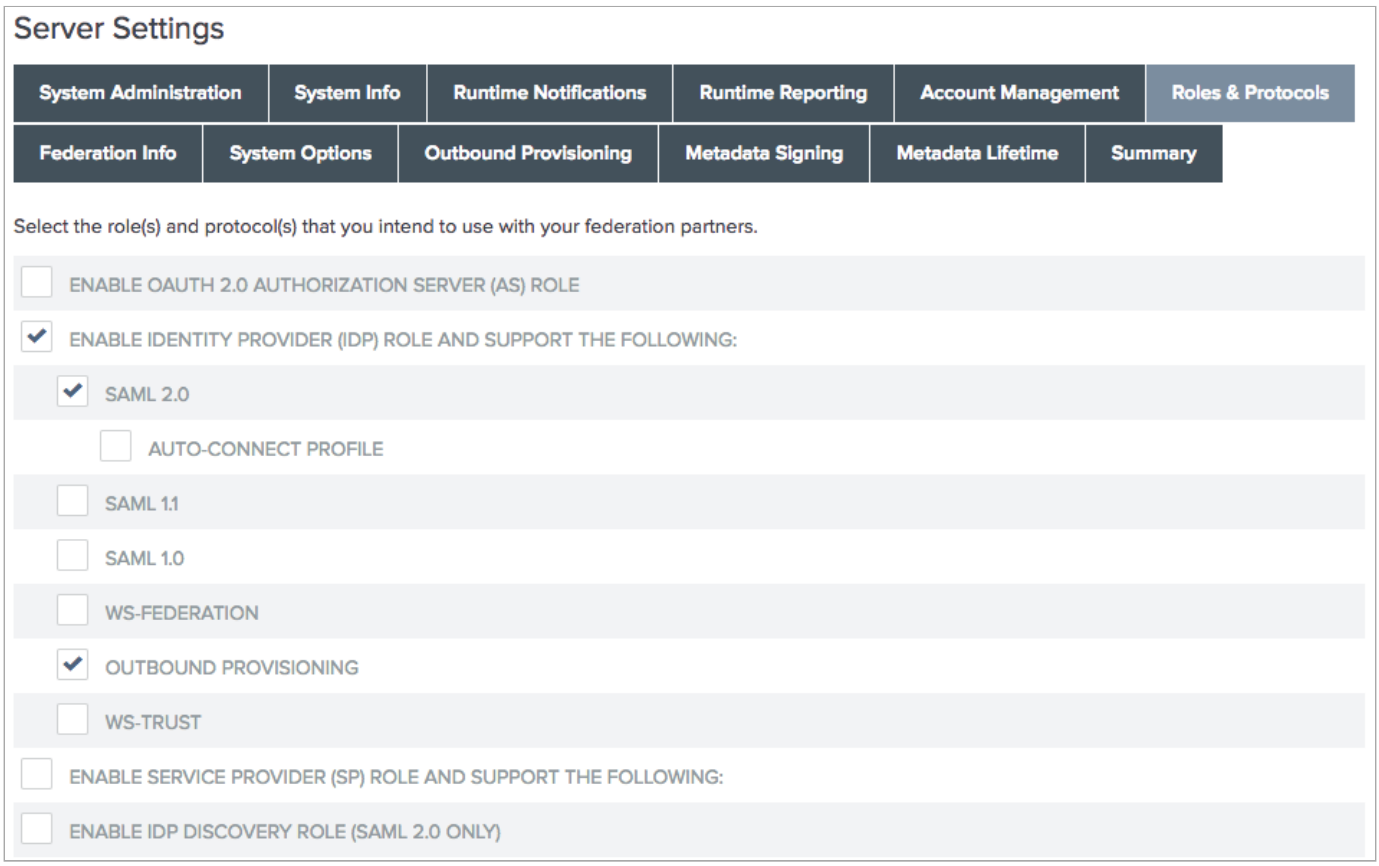
Enable outbound provisioning

About this task

After enabling outbound provisioning in the `<pf_install>/pingfederate/bin/run.properties` file, you must also activate the outbound provisioning role in the administrative console.

Steps

1. Go to the **Server Configuration** > **Server Settings** > **Roles & Protocols** screen.
2. Select the **Outbound Provisioning** check box.



> **Note**
>
> Enabling outbound provisioning adds the outbound provisioning screen, requiring the selection of a database to facilitate provisioning. Learn more in Configuring outbound provisioning settings or click **Help** from the configuration screen.

Copyright © 2025 Ping Identity Corporation

Configure provisioning

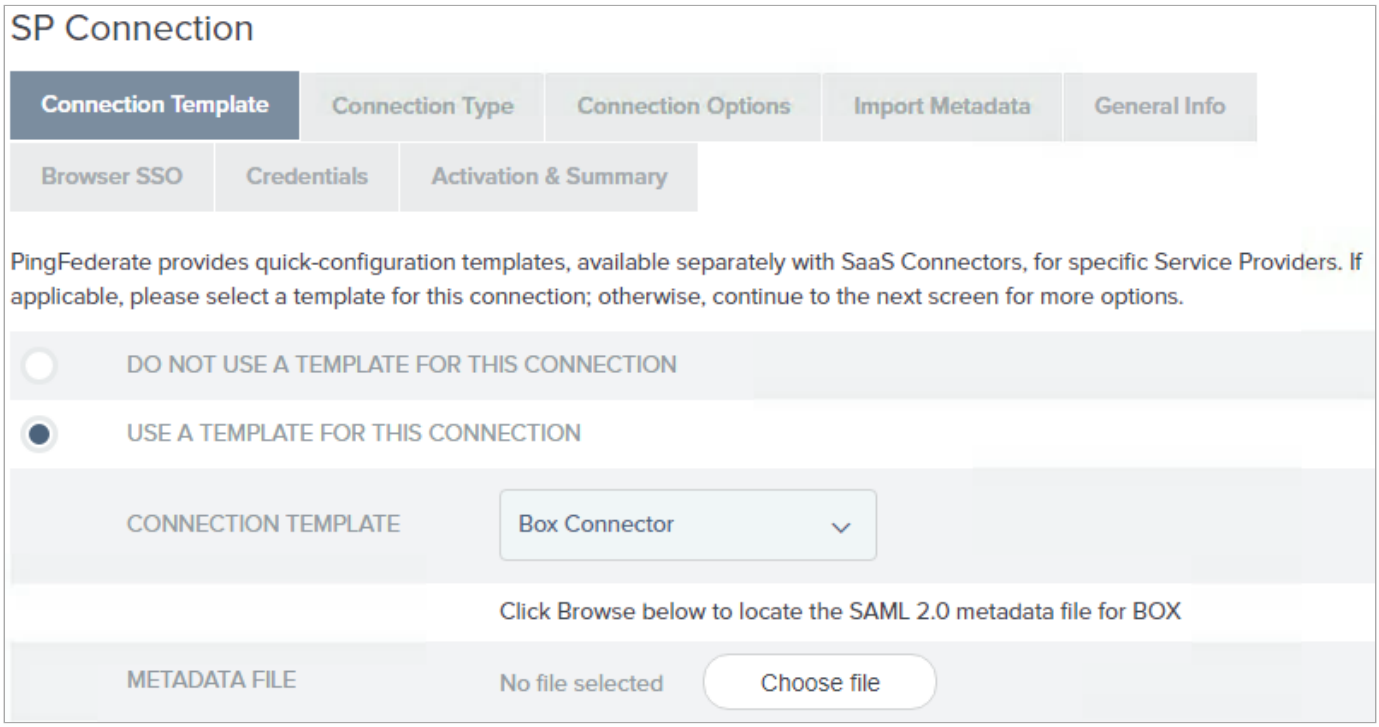
About this task

To configure a connection for outbound provisioning to Box, follow the instructions in this section.

Outbound provisioning details are managed within a service provider (SP) connection. You can configure outbound provisioning with or without Browser SSO, WS-Trust STS, or both when you create a new SP connection. You can also add outbound provisioning to an existing SP connection.

Steps

1. In the PingFederate administrator console, configure the data store that PingFederate will use as the source of user data. Learn more in Datastores in the PingFederate documentation.
    - When targeting users and groups for provisioning, exclude the user account that you will use to administer users in your connection to Box. This prevents the PingFederate provisioning engine from interfering with the account that provisions users and groups.
2. Create a new SP connection or select an existing SP connection from the **SP Configuration** menu.
3. On the **Connection Template** screen, select the **Use a template for this connection** option and choose **Box Connector** from the **Connection Template** drop-down list. You will be asked to provide the `boxmetadata.xml` file you obtained earlier in Download Box SAML 2.0 metadata file.

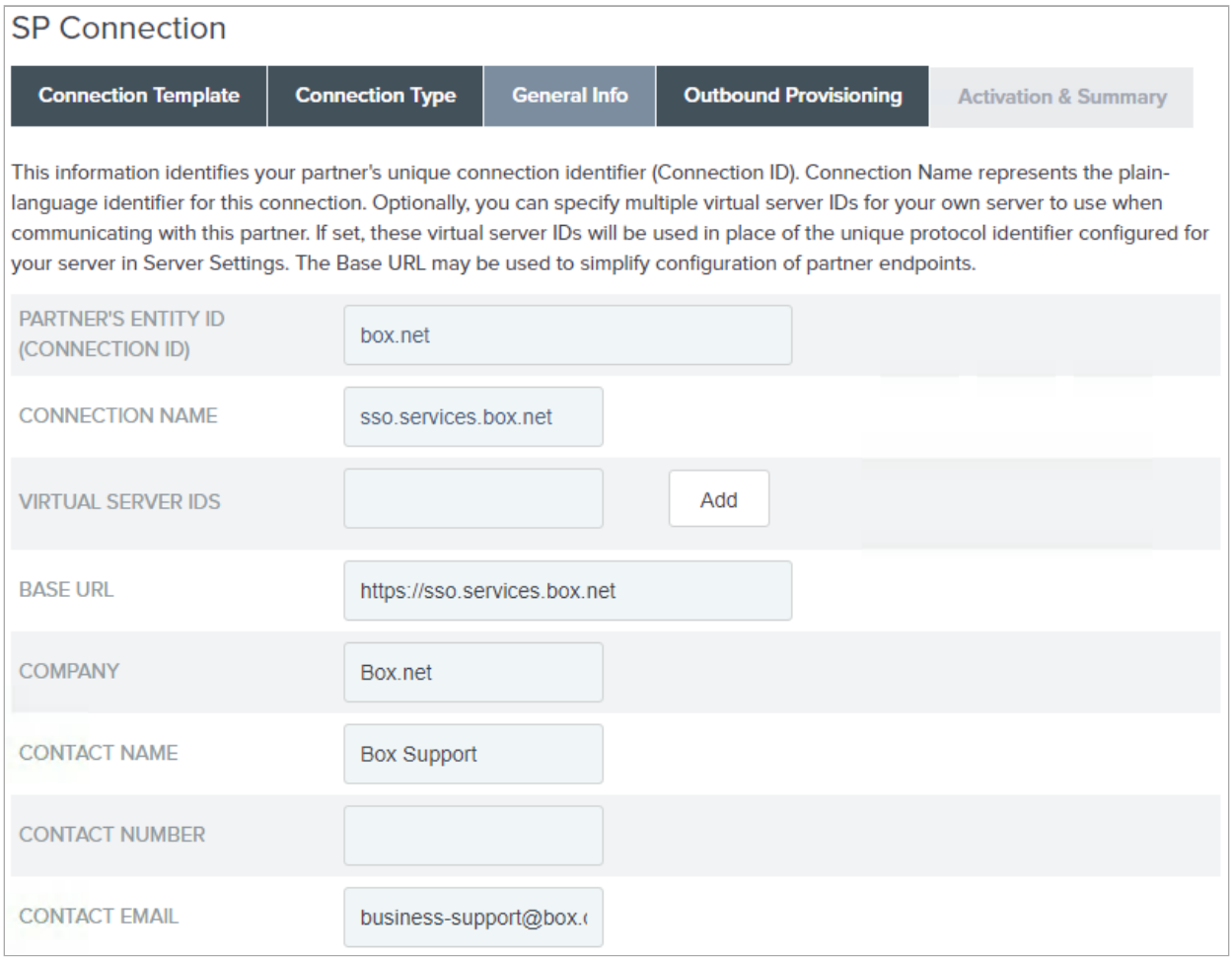


> **Tip**
>
> If this selection is not available, verify the connector installation and restart PingFederate.

4. On the **Connection Type** screen, ensure the **Outbound Provisioning** check box is selected, and the **Browser SSO Profiles** check box is cleared (if appropriate).

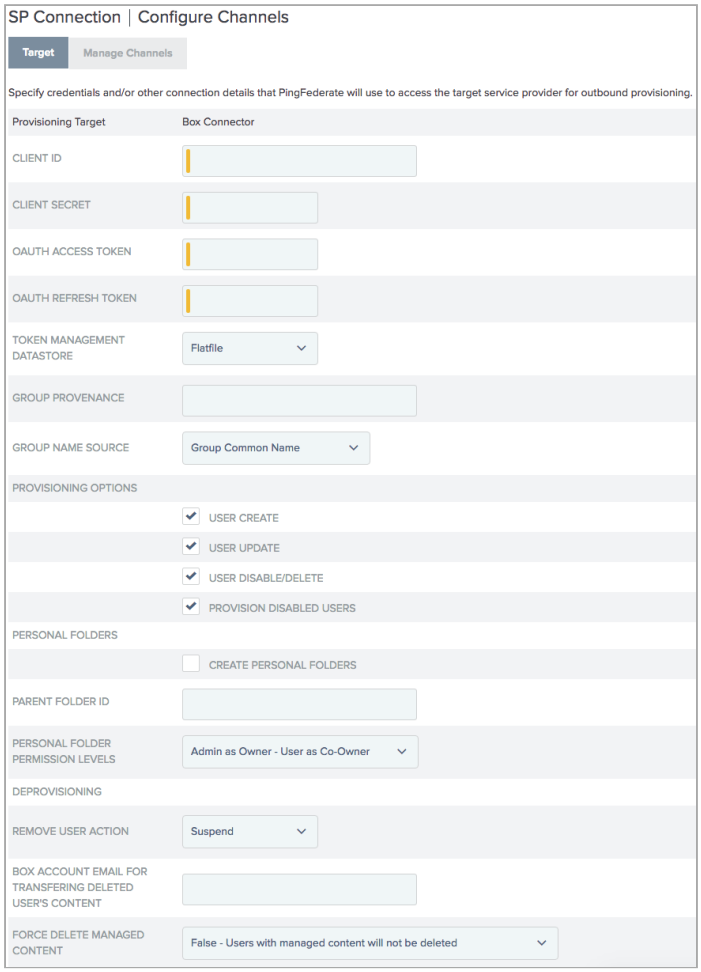
Copyright © 2025 Ping Identity Corporation

5. On the **General Info** screen, the default values are taken from the metadata file you selected in step 2. We recommend using the metadata default values.



6. Follow the connection wizard to configure the connection.
7. On the **Outbound Provisioning** screen, click **Configure Provisioning**.
8. On the **Target** screen, enter the values for each field as required by the Box Connector.

Copyright © 2025 Ping Identity Corporation



Field Name	Description
Client ID	The client ID for the application created in Box. Learn more about obtaining a client ID in Obtain client ID and secret from Box.
Client Secret	The client secret generated during application creation for Box. Learn more about obtaining a client secret in Obtain client ID and secret from Box.
OAuth Access Token	The OAuth access token generated by the OAuth Configuration Service. Learn more about obtaining authorized OAuth tokens in Generate OAuth access and refresh tokens.
OAuth Refresh Token	The OAuth refresh token generated by the OAuth Configuration Service. Learn more about obtaining authorized OAuth tokens in Generate OAuth access and refresh tokens.

Copyright © 2025 Ping Identity Corporation

Field Name	Description
Token Management Datastore	Select where tokens will be stored: ***Flatfile (default)*** Tokens will be stored in a flatfile named `boxoauthtoken.conf` located at `<pf_install>/server/default/data/adapter-config`. **Note** The flatfile option is only available when running in Standalone mode, since the flatfile cannot be accessed by multiple nodes. The flatfile option cannot be used in a clustered environment even if only one node is used for provisioning. ***Database*** Tokens can also be stored in a database. If you have set up databases in PingFederate, they will appear in the list as options. If you have a multinode PingFederate configuration for clustered or failover use cases, tokens must be stored in a database. The `saas_connection_fields` table must be first created in the database using the appropriate supplied script. The same database used for PingFederate provisioning can be used to create the `saas_connection_fields` table. This must be completed before configuring the Box Connector. During configuration, choose the database that contains the `saas_connection_fields` table.
Group Provenance (Optional)	Optional and for group provisioning only. This allows you to keep track of which external source this group is coming from (for example, "Active Directory", "Google Groups", "Facebook Groups"). This field should be a human-readable identifier up to 255 characters long. Setting this will also prevent Box users from editing this group directly through Box. This is desirable for one-way syncing of groups.

Copyright © 2025 Ping Identity Corporation

Field Name	Description
Group Name Source	Possible values: ***Group Common Name (default)*** When selected, groups will be provisioned in Box with the name equal to the common name of the group in Active Directory. ***Group Distinguished Name*** When selected, groups will be provisioned in Box with the name equal to the distinguished name of the group in Active Directory.
Provisioning Options	
User Create	***True (default)*** Users will be created in Box. ***False*** Users will not be created in Box. **Note** The `provisioner.log` will display a warning within the create user workflow that the user was not created in Box.
User Update	***True (default)*** Users will be updated in Box. ***False*** Users will not be updated in Box. **Note** The `provisioner.log` will display a warning within the update user workflow that the user was not updated in Box.

Copyright © 2025 Ping Identity Corporation

Field Name	Description
User Disable/Delete	***True (default)*** Users will be disabled or deleted in Box. **Note** A disabled user can only be re-enabled if **User Update** is true. ***False*** Users will not be disabled or deleted in Box. **Note** The `provisioner.log` will display a warning indicating that the user was not disabled or deleted in Box.
Provision Disabled Users	This option is only relevant if **User Create** is true. ***True (default)*** If a disabled user in the user store is targeted for provisioning, it will be created in a disabled state in Box. ***False*** Box users will not be created in a disabled state. This is desirable for scenarios where there are disabled users in the data store, not intended for creation in Box during initial synchronization. **Note** The `provisioner.log` will display a warning within the create user workflow indicating that the user was not created in Box.
Personal Folders	

Copyright © 2025 Ping Identity Corporation

Field Name	Description
Create Personal Folders	***False (default)*** Personal folders will not be created. ***True*** On the creation of a new user, a personal Box folder will be created for the user. The new folder will be created in a parent folder you must specify on this page (see **Parent Folder ID**), and its name will be the value of the user's **Personal Folder Name** attribute. You can find details in Supported attributes reference. **Note** If an issue occurs during the creation of the personal folder, the `server.log` will display a warning indicating that the user's personal folder was not created in Box. If you have applied Additional logging changes, this will be shown in the `provisioner.log`.
Parent Folder ID	This option is required if **Create Personal Folders** is true. The ID of the parent folder in which new personal folders will be created. The administrator account used when obtaining the client ID and client secret must be the owner of this folder. You can easily find the ID of a folder by navigating to the desired parent folder in the Box web portal and copying the string of numbers at the end of the url. For example, the ID of the folder located at: ◦ https://myconnector.app.box.com/folder/12345678910 would be `12345678910`.
Personal Folder Permission Levels	This option is only relevant if **Create Personal Folders** is true. ***Admin as Owner - User as Co-Owner (default)*** When a new personal folder is created, the admin will be the owner of the folder, and the user will be a co-owner. ***Admin as Owner - User as Editor*** When a new personal folder is created, the admin will be the owner of the folder, and the user will be an editor. ***User as Owner*** When a new personal folder is created, the user will be the owner, and the admin will have no permissions on or visibility into that folder.

Copyright © 2025 Ping Identity Corporation

Field Name	Description
Deprovisioning	
Remove User Action	Select a deprovision method (**Suspend** or **Delete**). Deprovisioning is triggered when previously provisioned users no longer meet the condition set in the **Source Location** screen, or when a user has been suspended or deleted from the data store. This option is only applicable if **User Disable/Delete** is set to true. ***Suspend (default)*** When selected, if you delete a user from Active Directory, the user will be suspended in Box (also known as a soft delete). ***Delete*** When selected, if you delete a user from Active Directory, the user will be deleted in Box (also known as a hard delete).
Box Account Email For Transferring Deleted User's Content	Optional and only relevant when **User Disable/Delete** is true and **Remove User Action** is **Delete**. The email associated with a valid and active Box account. When this is specified, if a user is deleted, their content (for example, files) would be transferred to the specified Box account instead of being deleted with the user.

Copyright © 2025 Ping Identity Corporation

Field Name	Description
Force Delete Managed Content	This option is only relevant if **User Disable/Delete** is true, **Remove User Action** is **Delete**, and the **Box Account Email For Transferring Deleted User's Content** is valid. ***False - Users with managed content will not be deleted (default)*** Attempts to delete users who own files will fail unless the files can be transferred to another account. If a value has been provided for **Box Account Email For Transferring Deleted User's Content** field then the user will be deleted after the data is transferred. The `provisioner.log` will show an error and retry on the next provisioning cycle. ***True - CAUTION: Users and their managed content will be deleted*** Attempts to delete users will proceed regardless of whether they own files. Any content owned by the user, including folders and files, will be permanently deleted from Box. If a value has been provided for the **Box Account Email For Transferring Deleted User's Content** field, then the transfer of files will be attempted prior to the user and content being deleted. Only select this option if revoking user access to content is of a higher priority than preserving the content.

Important

Once PingFederate is restarted, these and subsequent authorized OAuth tokens are stored in either the flatfile or the database, depending on the **Token Management Datastore** choice.
If these values in your SP connection require updating at any time in the future, follow the steps in Updating Box OAuth tokens.

9. Click **Next** to continue the provisioning configuration. Learn more in the following sections under Outbound provisioning for IdPs in the PingFederate documentation:
    - Managing channels
    - Specifying channel information
    - Identifying the source datastore
    - Modifying source settings
    - Specifying a source location
    - Mapping attributes

Copyright © 2025 Ping Identity Corporation

- Reviewing channel settings

> **Note**
>
> Credentials will be verified when the channel and SP connection is set to Active and provisioning is initiated.

> **Tip**
>
> If you are not ready to complete the provisioning configuration, you can click **Save** and return to the configuration page later. To return to the configuration page, select the connection from **Identity Provider > SP Connections > Manage All**.

Supported attributes reference

The following attributes can be mapped for user provisioning to Box.

Attribute	Description
Name	The name of the user. This attribute is required.
Login	The email address the user uses to sign on. This attribute is required. **Note** This value must be in the format of an email. Box Connector version 2.2 and later enables updating a user's login. However, the user has to have signed on previously for the update to succeed.
Language	The user's language. Learn more in Language Codes in the Box documentation.
Timezone	The user's timezone. Input format follows tz database timezones.
Space Amount	The user's total available space amount in bytes. A value of -1 grants unlimited storage.
Inactive Status Default	The user's default inactive status. The three inactive defaults include: • inactive • cannot_delete_edit • cannot_delete_edit_upload **Note** When a user is disabled, the user status will be the value you have specified here. If no value, the default is inactive. Deleting the user in LDAP will always default to inactive regardless of the attribute's value or the **Target** screen configuration for **Remove User Action**.
Job Title	The user's job title.

Copyright © 2025 Ping Identity Corporation

Attribute	Description
Phone	The user's phone number.
Address	The user's address.
Role	The user's enterprise role. Valid options include: • coadmin • user
Can See Managed Users	Whether the user can see other enterprise users in their contact list. Valid options include: • true • false
Is Sync Enabled	Whether or not the user can use Box Sync. Valid options include: • true • false
Is Exempt from Device Limits	Whether to exempt the user from Enterprise device limits. Valid options include: • true • false
Is Exempt from Login Verification	Whether or not this user must use two-factor authentication. Valid options include: • true • false
Is External Collab Restricted	Whether this user is allowed to collaborate with users outside her enterprise. Valid options include: • true • false
Personal Folder Name	The name to be used for the personal folder which can optionally be created along with a new user. The personal folder name format can be configured using OGNL expressions. Learn more in Sample OGNL expressions. Learn more about personal folder fields in Configure provisioning.

Copyright © 2025 Ping Identity Corporation

Sample OGNL expressions

PingFederate provides an advanced option allowing administrators to map user attributes by way of an expression language. The following are sample OGNL expressions for common use case requirements with the PingFederate Box Connector. Learn more about enabling expressions in PingFederate in Enabling and disabling expressions in the PingFederate documentation.

The below sample expressions are intended for use with the **Personal Folder Name** attribute. This can be configured under **SP Connection > Configure Channels > Channel > Attribute Mapping** when **Create Personal Folders** is enabled. To configure the **Personal Folder Name** attribute, select **Edit**, add all necessary attributes for the evaluation of the expression, and configure the expression.

Example

Lastname, Firstname

The following expression would produce the format `Collins, Audrey` for the user Audrey Collins.

```
#lastname = #this.get("sn").toString(),
#firstname = #this.get("givenName").toString(),
#foldername = #lastname + ", " + #firstname
```

Lastname, Firstname (Department)

The following expression would produce the format `Collins, Audrey (Sales)`, where the user Audrey Collins works in the Sales department.

```
#lastname = #this.get("sn").toString(),
#firstname = #this.get("givenName").toString(),
#department = #this.get("department").toString(),
#foldername = #lastname + ", " + #firstname + " (" + #department + ")"
```

Remove Domain from Email Address

The following expression would produce the format `acollins`, where the user Audrey Collins has an email address `acollins@example.com`.

```
#this.get("mail").toString().split("@")[0]
```


Copyright © 2025 Ping Identity Corporation

Configure SAML SSO

The following section describes the steps for configuring single sign-on (SSO) to Box.

> **Note**
>
> Configuring SSO is optional for outbound provisioning.

- Configure PingFederate for SSO
- Configure Box for SSO

Configure PingFederate for SSO

About this task

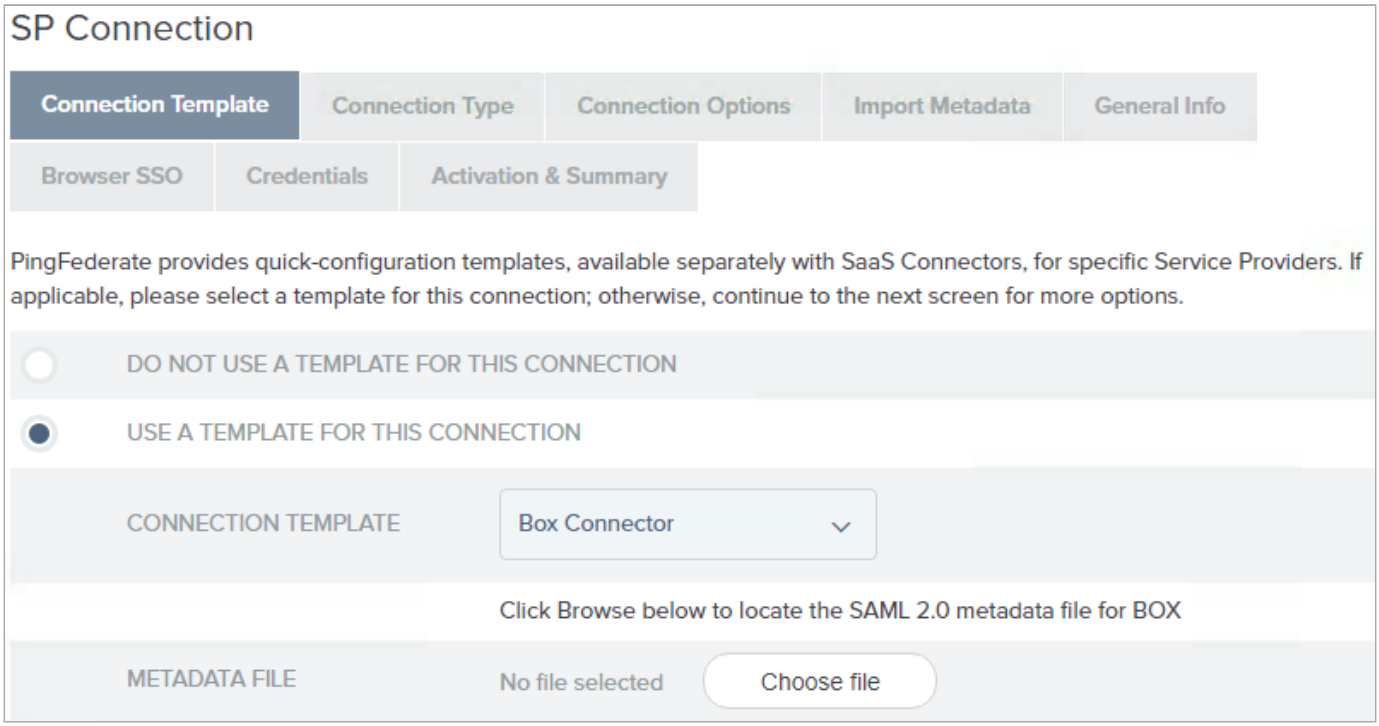
The following section describes the steps for configuring single sign-on (SSO) to Box. Configuring SAML SSO involves configuring both the PingFederate SP connection and Box.

> **Note**
>
> Configuring SSO is optional for outbound provisioning.

Steps

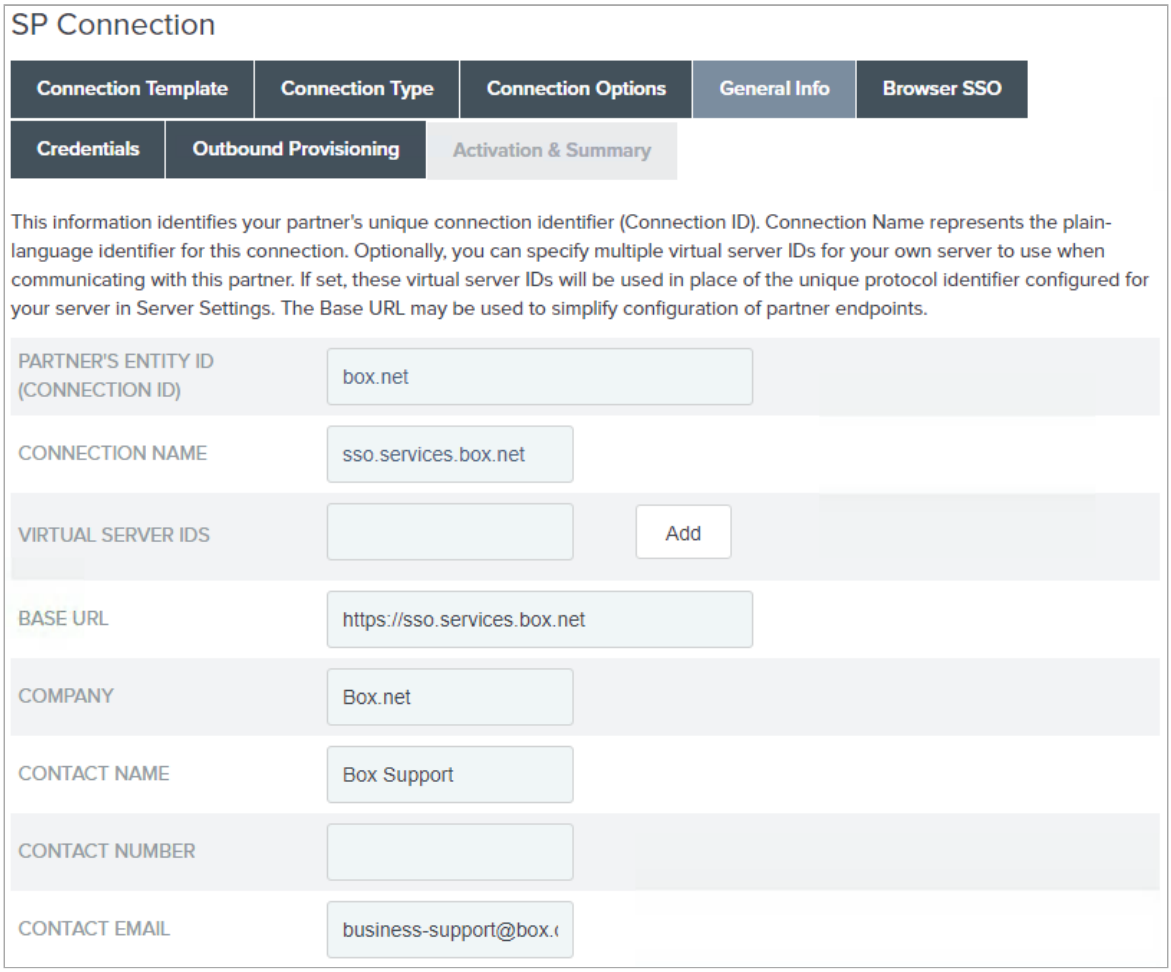
1. Create a new SP connection or select an existing SP connection from the **SP Configuration** menu.
2. On the **Connection Template** screen, select the **Use a template for this connection** option and choose **Box Connector** in the **Connection Template** list. You will be asked to provide the `boxmetadata.xml` file you obtained earlier in Download Box SAML 2.0 metadata file.

Copyright © 2025 Ping Identity Corporation



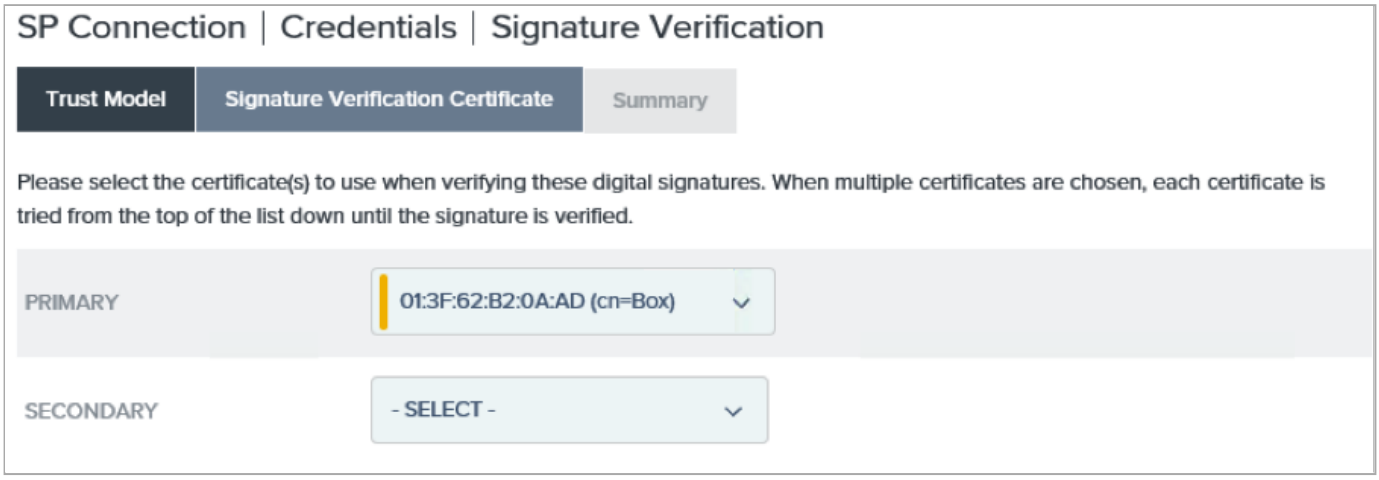
3. On the **Connection Type** screen, ensure that the **Browser SSO Profiles** check box is selected.
4. On the **General Info** screen, the default values are taken from the metadata file you selected in step 2. We recommend using the metadata default values.

Copyright © 2025 Ping Identity Corporation



5. Click **Next** to continue the Browser SSO configuration. Learn more in the following sections under Identity provider SSO configuration:
    - Managing IdP adapters
    - Configure IdP Browser SSO
    - Configuring credentials
6. On the authentication adapter's **Attribute Contract Fulfillment** screen, map SAML_SUBJECT to email address.
7. On the **Protocol Settings > Allowable SAML Bindings** screen, ensure that both **POST** and **SOAP** are selected.
8. On the **Credentials** screen, click **Configure Credentials**.
9. On the **Back-Channel Authentication** screen, click **Configure**.
10. On the **Inbound Authentication Type** screen, select **Digital Signature (Browser SSO profile only)** and click **Done**.
11. On the **Credentials > Digital Signature Settings** screen, select the signing certificate.
12. On the **Signature Verification Settings** screen, click **Manage Signature Verification Settings**.
13. On the **Trust Model** screen, ensure **Unanchored** is selected and click **Next**.
14. On the **Signature Verification Certificate** screen, select the Box certificate as the primary certificate and click **Next**.

Copyright © 2025 Ping Identity Corporation



15. On the **Activation & Summary** screen, set **Connection Status** to Active, then click **Save**.

Configure Box for SSO

About this task

To configure Box for single sign-on (SSO), you will require metadata from your PingFederate Identity Provider (IdP) setup. Learn more in Exporting connection-specific SAML metadata

To export your IdP metadata:

1. Go to **Server Configuration > Metadata Export**.
2. On the **Metadata Mode** screen, choose **Use a Connection for Metadata Generation** and click **Next**.
3. On the **Connection Metadata** screen, select the Box connection created in Configure PingFederate for SSO.
4. On the **Metadata Signing** screen, click **Next**.
5. On the **Export & Summary** screen, click **Export** to retrieve your metadata `.xml` file.

To configure Box for SSO:

Steps

1. Submit a ticket through the Box SSO Setup Support Form and upload your PingFederate metadata.
2. Click **Submit**.

Troubleshooting

The following table lists potential problems administrators might encounter during the setup or deployment of the Box Connector, along with possible solutions. Learn more in Common Errors in the Box documentation.

Copyright © 2025 Ping Identity Corporation

Problem	Possible Solution
The exception {"error":"invalid_grant", "error_description":"Refresh token has expired"} appears in `server.log`.	The error indicates that the refresh token has expired. A refresh token from Box has a lifespan of 60 days. Additionally, if a new refresh token has been requested, the old one will become expired. Learn more about configuring PingFederate with a new refresh token in Updating Box OAuth tokens.
The exception {"type":"error","status":403, "code":"access_denied_insufficient_permissions", "help_url":"http:\/\/developers.box.com\/docs\/#errors", "message":"Access denied - insufficient permission"} appears in `server.log`.	Revert the email to the original and have the user sign on and set their password, then update the user's email.
The exception {"errors": [{"reason":"invalid_parameter", "name":"email","message":"Invalid value '<email value>. This email address already has a Box account. You can only add emails that do not have an account already."}]} appears in `server.log`.	Revert the email to the original and have the user log in and set their password, then update the user's email.
The exception {"type":"error","status": 500,"code":"internal_server_error", "help_url":"http:\/\/developers.box.com\/docs\/#errors","message":"Internal Server Error","request_id":"xxxxxxxx"} appears in `provisioner.log`.	If there is a value in the **Target** field **Box Account Email for Transferring Deleted User's Content**, ensure that it is a valid email belonging to a Box user in your instance, and that the user is active.

Additional logging

The following logging options can be updated in `log4j2.xml` after the section `AsyncLogger name="com.pingidentity.provisioner"`. This will provide additional logging details in the `provisioner.log` for troubleshooting purposes.

Learn more about log4j2 in Log4j 2 logging service and configuration. You can also learn more from Ping Identity Support & Community.

Copyright © 2025 Ping Identity Corporation

```
<AsyncLogger name="com.pingidentity.integrations"
                    level="INFO" additivity="false" includeLocation="false">
            <appender-ref ref="ProvisionerLog" />
            <!--
                <appender-ref ref="CONSOLE-PROVISIONER" />
                <appender-ref ref="ProvisionerLogToOracleDB-FAILOVER"/>
                <appender-ref ref="ProvisionerLogToSQLServerDB-FAILOVER"/>
                <appender-ref ref="ProvisionerLogToMySQLDB-FAILOVER"/>
            -->
        </AsyncLogger>

<AsyncLogger name="com.pingidentity.saas"
                    level="INFO" additivity="false" includeLocation="false">
            <appender-ref ref="ProvisionerLog" />
            <!--
                <appender-ref ref="CONSOLE-PROVISIONER" />
                <appender-ref ref="ProvisionerLogToOracleDB-FAILOVER"/>
                <appender-ref ref="ProvisionerLogToSQLServerDB-FAILOVER"/>
                <appender-ref ref="ProvisionerLogToMySQLDB-FAILOVER"/>
            -->
</AsyncLogger>
```

Updating Box OAuth tokens

About this task

Use the following procedure to manually update the Box OAuth tokens.

Steps

1. Obtain new Box OAuth tokens. Learn more in Generate OAuth access and refresh tokens.
2. If using the flatfile method for token storage, delete the `boxoauthtoken.conf` file located at `<pf_install>/server/default/data/adapter-config`. If using the database method for token storage, delete the row that contains the Box tokens from the `saas_connection_fields` table.
3. On the **Target** screen of the SP connection, update the following fields with the new OAuth token values:
    - `OAUTH_ACCESS_TOKEN`
    - `OAUTH_REFRESH_TOKEN`
4. Restart PingFederate. This will regenerate the flatfile or repopulate the database with the new credentials.

Release notes

Copyright © 2025 Ping Identity Corporation

Changelog

Box Provisioner 2.5 – April 2018 (current release)

- Added option to create personal folders on user creates
- Added option to force delete users with managed content

Box Provisioner 2.4.1 – October 2017

- Resolved issue when saving new tokens to the database (token renewal workflow)
- Improved logging for token renewal workflow

Box Provisioner 2.4 – July 2017

- Added option to provision groups with common name or distinguished name

Box Provisioner 2.3 – July 2017

- Added option to store Box credentials in a flatfile or in a database
    - Flatfile credential storage is limited to a single instance of PingFederate
    - Database credential storage has the ability to support multiple PingFederate instances (for example, failover use case)
- Added support for disabling or deleting users without updating any user attributes
- Added configuration control of provisioning disabled users

Box Provisioner 2.2.2 – March 2017

- Fixed synchronization on update of users, that were previously created with "User Create Enabled" set to false in configurable options

Box Provisioner 2.2.1 – January 2017

- Fixed OAuth token issues
- Fixed the provisioner handling channels with multiple threads
- Improved handling of different letter case logins and aliases

Box Provisioner 2.2 – November 2016

- Added support for updating user emails

Box Provisioner 2.1 – May 2016

- Added support for transferring box content to an admin account during deprovisioning (hard delete only)

Copyright © 2025 Ping Identity Corporation

Box Provisioner 2.0 – March 2016

- Added support for additional user attributes
- Added support for group provisioning
- Added support for adding users to groups
- Added configuration options for CRUD capabilities

Box Provisioner 1.0 – June 2014

- Initial release
- Added support for user provisioning

Known issues and limitations

- Due to a limitation with PingFederate 8.1 and earlier versions, when configuring two SP connections with the same provisioner, the second connection built may be pre-populated with the channel from the first connection. To avoid conflicts, delete this pre-populated channel and create a unique channel for each connection.
- User attributes cannot be cleared once set.
- Version 2.2 and later enables updating a user's login. However, the user has to have logged in previously for the update to succeed.
- The provenance attribute is the only supported attribute for group provisioning.
- The provenance attribute cannot be cleared once set for a group.
- The Inactive Status Default user attribute will have no effect if the Box connector is configured to delete (hard delete) instead of disable (soft delete) for user deprovisioning. Additionally, deleting a user in LDAP will always set that user as inactive in Box.
- When an LDAP user is deleted in a targeted group distinguished name (DN), the provisioning connector does not propagate the deletion until a new user is added to the group. This limitation is compounded when the **User Create** provisioning option is disabled. For solutions, see SaaS provisioner does not remove the user in the Knowledge Base.
- A Box API limitation prevents logins of different letter case (but otherwise the same), from being updated by the provisioner. In scenarios where the letter case differs, the login will be omitted from the API operation. For example, `USER@TEST.COM` in Box, cannot be updated to `user@test.com`. In an update operation, the login would be omitted, but any other attributes that may have changed would be provisioned and updated.
- Due to Box API requirements, only primary, validated email addresses can be used to sync users.

Box token failover support limitations

- Tokens entered during SP connection configuration will be invalidated and replaced with new tokens upon first use. Afterwards, they will be updated with new tokens once they expire.
- If a SP connection is re-configured to change the connection datasource from a database to a flatfile, or vice versa, any pre-existing entries in the new datasource for the given client ID must be deleted before the SP connection is updated.

Copyright © 2025 Ping Identity Corporation

Performance limitations

- Enabling personal folder functionality will diminish initial synchronization provisioning performance.

Download manifest

The distribution `.zip` archive for the connector contains the following:

- `ReadMeFirst.pdf` : Contains links to this online documentation.
- `/legal`
  - `Legal.pdf` : Copyright and license information.
- `/dist` : Contains libraries needed for the connector:
  - `pf-box-quickconnection-[version].jar` : PingFederate Box Connector
- `/scripts`
  - saas_connection_fields-postgresql.sql
  - saas_connection_fields-oracle.sql
  - saas_connection_fields-mysql.sql
  - saas_connection_fields-mssql.sql

Copyright © 2025 Ping Identity Corporation

Citrix ADC Integration Guide

This guide describes how to configure PingFederate and Citrix ADC to enable single sign-on (SSO).

This allows you to use PingFederate as an identity provider (IdP) for Citrix ADC and Gateway.

The integration uses a standard SAML configuration and does not require any additional software.

Features

- Browser-based SSO initiated by the IdP or service provider (SP).

Intended audience

This document is intended for PingFederate administrators. Before you start, you should be familiar with the following:

- Managing IdP adapters in the PingFederate documentation
- The following sections of the Citrix documentation:
  - SAML authentication
  - Authentication, authorization, and auditing Application Traffic

System requirements

- Citrix ADC 13.0 or later
- A DNS configuration that routes authentication requests to a Citrix ADC virtual IP address
- An SSL certificate installed on the Citrix ADC appliance

Enabling single sign-on in PingFederate

Before you can configure single sign-on in Citrix, you need to set a SAML entity ID in PingFederate.

Enabling single sign-on in PingFederate 10.1 or later

Steps

1. On the PingFederate administrative console, go to **System > Protocol Settings > Federation Info**.
2. In the **SAML 2.0 Entity ID** field, enter a name for PingFederate to use to identify itself to SAML partners.
3. Click **Save**.

Enabling single sign-on in PingFederate 10.0 or earlier

Steps

1. On the PingFederate administrative console, go to **System > Protocol Settings > Roles & Protocols**.

Copyright © 2025 Ping Identity Corporation

2. Select the **Enable Identity Provider IdP Role and Support the Following** checkbox.
3. Select the **SAML 2.0** checkbox. Click **Next**.
4. Go to the **Federation Info** tab.
5. In the **SAML 2.0 Entity ID** field, enter a name for PingFederate to use to identify itself to SAML partners.
6. Click **Save**.

Exporting your PingFederate signing certificate

To allow Citrix to communicate with PingFederate, export your PingFederate signing certificate.

Steps

1. On the PingFederate administrative console, go to **Security > Signing & Decryption Keys & Certificates**.
2. If you do not have a certificate, create one.

   Learn more in Managing digital signing certificates and decryption keys in the PingFederate documentation.
3. For the certificate that you want to use, in the **Action** column, click **Export**.
4. On the **Export Certificate** tab, click **Next**.
5. On the **Export & Summary** tab, click **Export**.

Creating an authentication server in Citrix

To allow PingFederate to handle authentication requests, add your PingFederate signing certificate and sign-on URL.

Steps

1. In Citrix ADC, complete the steps in Enabling authentication, authorization, and auditing in the Citrix documentation.
2. Go to **Configuration > Authentication > Dashboard**. Click **Add**.
3. In the **Create Authentication Server** window, in the **Choose Server Type** list, select **SAML**.
4. In the **Name** field, enter a name to represent your PingFederate server, such as `PF_Auth_Server`.
5. Clear the **Import Metadata** checkbox.
6. In the **Redirect URL** field, enter the following PingFederate sign-on URL. Substitute your hostname and port, and choose a connection ID.

   ```
   https://pf_host:pf_port/idp/startSSO.ping?PartnerSpId=connectionID
   ```

   You will use the connection ID again in Creating a single sign-on connection.
7. In the **SAML Binding** and **Logout Binding** lists, select **POST**.

Copyright © 2025 Ping Identity Corporation

8. Add your PingFederate signing certificate.
    1. Under **IDP Certificate Name**, click **Add**.
    2. In the **Certificate-Key Pair Name** field, enter a name, such as `PF_Certificate`.
    3. In the **Certificate File Name** section, upload the certificate that you saved in Exporting your PingFederate signing certificate. Click **Install**.
9. In the **Issuer Name** field, enter a name to represent Citrix, such as `CitrixSAML`.
10. In the **Reject Unsigned Assertion** list, select **OFF**. Click **More**.
11. In the **Signature Algorithm** section, click **RSA-SHA256**.
12. In the **Digest Method** section, click **SHA256**. Click **Create**.

Creating a SAML policy in Citrix

Create a SAML authentication policy and associate it with the authentication server that you created.

About this task

Learn more in NetScaler as a SAML SP in the Citrix documentation.

Steps

1. In Citrix ADC, go to **Configuration > Security > AAA - Application Traffic > Policies > Authentication > Basic > Policies > SAML**. Click **Add**.
2. In the **Create Authentication SAML Policy** window, in the **Name** field, enter a name, such as `PF_Policy`.
3. In the **Server** list, select the server that you created in Creating an authentication server in Citrix.
4. In the **Expression** section, in the **Saved Policy Expressions** list, select **ns_true**. Click **Create**.

Creating an authentication virtual server in Citrix

Configure a virtual server to direct authentication requests to PingFederate.

About this task

Learn more in Setting up an authentication virtual server in the Citrix documentation.

Steps

1. In Citrix ADC, go to **Configuration > Security > AAA - Application Traffic > Virtual Servers**. Click **Add**.

Copyright © 2025 Ping Identity Corporation

2. In the **Authentication Virtual Server** window, in the **Basic Settings** section, enter the basic server information.
    1. In the **Name** field, enter a name, such as `PF_Auth_VServer`.
    2. In the **IP Address** field, enter the IP address of your PingFederate server. Click **OK**.
3. In the **Certificates** section, select your PingFederate certificate.
    1. Click **No Server Certificate**.
    2. In the **Server Certificate Binding** pane, in the **Select Server Certificate** section, click **Click to select**.
    3. In the **Server Certificates** pane, select the certificate that you added in Creating an authentication server in Citrix. Click **Select**.
    4. In the **Server Certificate Binding** pane, click **Bind**. Click **Continue**.
4. In the **Advanced Authentication Policies** section, click **Continue**.
5. In the **Basic Authentication Policies** section, add a policy.
    1. Click **Add (+)**.
    2. In the **Choose Type** pane, in the **Policies** section, in the **Choose Policy** list, select **SAML**.
    3. In the **Choose Type** list, select **Primary**. Click **Continue**.
    4. In the **Policy Binding** section, in the **Select Policy** list, select the policy that you created in Creating a SAML policy in Citrix.
    5. Click **Bind**. Click **Continue**.
6. Click **Done**.

Enabling SAML policy on the Gateway virtual server

Set the SAML policy you create as the default for the Gateway virtual server.

Steps

1. In Citrix ADC, go to **Configuration > Citrix Gateway > Virtual Servers**. Select your virtual server.
2. In the **Basic Authentication Policies** section, select the policy that you created in Creating a SAML policy in Citrix.
    1. Click **Add (+)**.
    2. In the **Choose Type** pane, in the **Policies** section, in the **Choose Policy** list, select **SAML**.
    3. In the **Choose Type** list, select **Primary**. Click **Continue**.
    4. In the **Policy Binding** section, in the **Select Policy** list, select the policy that you created in Creating a SAML policy in Citrix.
    5. Click **Bind**.

Copyright © 2025 Ping Identity Corporation

3. Click **Done**.

Creating a single sign-on connection

To allow PingFederate to handle single sign-on (SSO) to Citrix, create a service provider (SP) connection.

Steps

1. In the PingFederate administrator console, create a new SP connection:

   Choose from:

   - For PingFederate 10.1 or later: go to **Applications > Integration > SP Connections**. Click **Create Connection**.
   - For PingFederate 10.0 or earlier: go to **Identity Provider > SP Connections**. Click **Create Connection**.

2. If you see the **Connection Template** tab, select **Do not use a template for this connection**. Click **Next**.
3. On the **Connection Type** tab, select only **Browser SSO Profiles**. Click **Next**.
4. On the **Connection Options** tab, select only **Browser SSO**. Click **Next**.
5. On the **Import Metadata** tab, select **None**. Click **Next**.
6. On the **General Info** tab, enter the basic connection information. Click **Next**.
   1. In the **Partner's Entity ID** field, enter the **Issuer Name** that you chose in Creating an authentication server in Citrix.
   2. In the **Connection Name** field, enter the connection ID portion of the **Redirect URL** that you entered in Creating an authentication server in Citrix.
   3. In the **Base URL** field, enter the base URL for your Citrix Gateway server.
7. On the **Browser SSO** tab, configure browser SSO. Click **Next**.

   Learn more in Configure IdP Browser SSO in the PingFederate documentation.

   1. On the **Browser SSO > SAML Profiles** tab, select **IdP-Initiated SSO** and **SP-Initiated SSO**. Click **Next**.
   2. On the **Browser SSO > Assertion Creation > Authentication Source Mapping** tab, complete the steps in Configuring an adapter instance. Click **Next**.
   3. On the **Browser SSO > Protocol Settings > Assertion Consumer Service** tab, in the **Binding** list, select **POST**.
   4. In the **Endpoint URL** field, enter `/cgi/samlauth`, and then click **Add**. Click **Next**.
   5. On the **Browser SSO > Protocol Settings > Allowable SAML Bindings** tab, select only **POST**. Click **Next**.
8. On the **Credentials** tab, configure the connection credentials as shown in Configuring credentials in the PingFederate documentation. Click **Next**.
9. On the **Activation and Summary** tab, above the **Summary** section, click the toggle to turn on the connection. Click **Save**.
10. Note the **SSO Application Endpoint** URL. Click **Save**.

Copyright © 2025 Ping Identity Corporation

The **SSO Application Endpoint** URL should match the **Redirect URL** that you entered in Creating an authentication server in Citrix. If it doesn't, update the URL in Citrix ADC.

11. To test the integration, make sure your test credentials exist in both the PingFederate data store and Citrix, then go to your Citrix ADC URL in a browser and sign on.

Configuring an adapter instance

Create an adapter instance to make the username attribute available in your service provider connection to Citrix Gateway.

Steps

1. Click **Map New Adapter Instance**.
2. On the **Adapter Instance** tab, in the **Adapter Instance** list, select an adapter instance that can access the username in your user store, such as **HTML Form Adapter**. Click **Next**.
3. On the **Mapping Method** tab, click **Next**.
4. On the **Attribute Contract Fulfillment** tab, on the **SAML_SUBJECT** line, in the **Source** list, select **Adapter**.
5. On the **SAML_SUBJECT** line, in the **Value** list, select **username**. Click **Next**.
6. On the **Issuance Critera** tab, click **Next**.
7. On the **Summary** tab, check that the configuration is correct, and then click **Done**.

Copyright © 2025 Ping Identity Corporation

Code42 Integration Guide for PingFederate

PingIdentity.

You can integrate PingFederate with Code42 for user provisioning and single sign-on (SSO). This integration uses the downloadable SCIM Connector.

Features

- Manages users in Code42 based on changes in a datastore that is attached to PingFederate.
  - Creates, updates, and disables users.
  - Allows you to enable the create, update, and disable capabilities independently.
- Enables browser-based single sign-on initiated by the service provider (SP).

Intended audience

This document is intended for PingFederate administrators.

If you need help during the setup process, see the following resources:

- The following sections of the PingFederate documentation:
  - Identity provider SSO configuration
  - Datastores
  - Managing IdP adapters
  - Managing digital signing certificates and decryption keys
  - SP connection management
  - Configuring outbound provisioning
  - Configuring outbound provisioning settings
- The following sections of the Code42 documentation:
  - Identity management
  - Identity management reference
  - Introduction to SCIM provisioning
  - How to configure SCIM provisioning
  - Introduction to single sign-on
- The following sections of the SCIM Connector documentation:
  - SCIM Provisioner
  - Known issues and limitations

Copyright © 2025 Ping Identity Corporation

System requirements

- PingFederate 9.0 or later.
- The Ping Identity SCIM Connector integration. See Deploying the integration files.
- To allow PingFederate to make outbound connections to the Code42 API, you might need to allow the following domain in your firewall:
    - https://*.code42.com

Updating an existing deployment

If you're upgrading from a previous version of the SCIM Connector, note your existing service provider (SP) connection configuration and create a new connection.

Updating an existing deployment in PingFederate 10.1 or later

Steps

1. Back up your current PingFederate configuration as shown in Configuration archive in the PingFederate documentation.
2. In the PingFederate administrative console, go to **Applications > Integration > SP Connections** and select your connection.
3. Note the attribute mappings for your existing SP connection.

   For help, see Mapping attributes in the PingFederate documentation.
4. Delete your existing SP connection.
    1. Go to **Applications > Integration > SP Connections**.
    2. For your existing connection, click **Select action**, and then click **Delete**. Click **Confirm.**
5. Complete the steps in Deploying the integration files.
6. Complete the steps in Creating a provisioning connection.
    - From **Outbound Provisioning > Manage Channels > Channel**, on the **Attribute Mapping** tab, configure the attribute mappings based on your notes.
7. Complete the steps in Creating a single sign-on connection.

Updating an existing deployment in PingFederate 10.0 or earlier

Steps

1. Back up your current PingFederate configuration as shown in Configuration archive in the PingFederate documentation.
2. In the PingFederate administrative console, go to **Identity Provider > SP Connections** and select your connection.
3. Note the attribute mappings for your existing SP connection.

Copyright © 2025 Ping Identity Corporation

For help, see Mapping attributes in the PingFederate documentation.

4. Delete your existing SP connection.
    1. Go to **Identity Provider > SP Connections > Manage All**.
    2. For your existing connection, click **Select action**, and then click **Delete**. Click **Save.**
5. Complete the steps in Deploying the integration files.
6. Complete the steps in Creating a provisioning connection.
    - From **Outbound Provisioning > Manage Channels > Channel**, on the **Attribute Mapping** tab, configure the attribute mappings based on your notes.
7. Complete the steps in Creating a single sign-on connection.

Deploying the integration files

To get started with the integration, deploy the SCIM Connector files to your PingFederate directory.

Steps

1. Download the SCIM Connector `.zip` archive from the Ping Identity Integration Directory.
2. Stop PingFederate.
3. If you're upgrading an existing deployment, delete the `pf-scim-quickconnection-<version>.jar` file from your `<pf_install>/pingfederate/server/default/deploy` directory.
4. Extract the `.zip` archive and merge the contents of the `dist` directory with your `<pf_install>/pingfederate/server/default/deploy` directory.
5. Enable the PingFederate provisioning engine:
    1. Open your `<pf_install>/pingfederate/bin/run.properties` file.Change `pf.provisioner.mode` to `STANDALONE`.
    2. Save the file.

    > **Note**
    >
    > To configure the `FAILOVER` mode instead, see Deploying provisioning failover in the PingFederate Server Clustering Guide.
6. Start PingFederate.
7. If you operate PingFederate in a cluster, repeat steps 2-4 and step 6 for each engine node.

Enabling provisioning and single sign-on in PingFederate

To use PingFederate for provisioning and single sign-on, configure an external datastore and set a SAML entity ID.

Copyright © 2025 Ping Identity Corporation

About this task

Your external datastore acts as the source of data for provisioning. PingFederate also uses an internal datastore to store the state of synchronization between the source datastore and the target datastore.

For more information, see Datastores and Configuring outbound provisioning settings in the PingFederate documentation.

Enabling provisioning and single sign-on in PingFederate 10.1 or later

Steps

1. In the PingFederate administrative console, configure the datastore for PingFederate to use as the source of user data.

   For help, see Datastores in the PingFederate documentation.
2. Go to **System > Server > Protocol Settings > Federation Info**.
3. In the **SAML 2.0 Entity ID** field, enter a name for PingFederate to use when identifying itself to SAML partners.
4. On the **Outbound Provisioning** tab, in the **Provisioner Data Store** list, select the internal database that will store the synchronization state. Click **Save**.

   For more information, see Configuring outbound provisioning settings in the PingFederate documentation.

Enabling provisioning and single sign-on in PingFederate 10.0 or earlier

Steps

1. In the PingFederate administrative console, configure the datastore for PingFederate to use as the source of user data.

   For help, see Datastores in the PingFederate documentation.
2. Go to **System > Protocol Settings > Roles & Protocols**.
3. Select the **Enable Identity Provider IdP Role and Support the Following** check box.
4. Select the **SAML 2.0** and **Outbound Provisioning** check boxes. Click **Next**.
5. Click the **Federation Info** tab.
6. In the **SAML 2.0 Entity ID** field, enter a name for PingFederate to use to identify itself to SAML partners.
7. On the **Outbound Provisioning** tab, in the **Provisioner Data Store** list, select the internal database that will store the synchronization state. Click **Save**.

   For more information, see Configuring outbound provisioning settings in the PingFederate documentation.

Provisioning setup

Adding PingFederate as SCIM provider in Code42

To get your base URL and credentials to access Code42 API, add PingFederate as a SCIM provider.

Copyright © 2025 Ping Identity Corporation

Steps

1. Sign on to Code42 as an administrator.
2. Navigate to **Integrations > Identity Management**.
3. On the **Provisioning** tab, click **Add provisioning provider**, and then click **Add SCIM provider**.
4. On the **Add SCIM provisioning** dialog, enter a name and select the **OAuth token** credential type. Click **Next**.
5. On the **SCIM Provider Created** dialog, note the **Base URL** and **Token** values. You will use these in Creating a provisioning connection. Click **Done**.

Creating a provisioning connection

To allow PingFederate to manage users in Code42 using SCIM, create a service provider (SP) connection.

Steps

1. In the PingFederate administrator console, create a new SP connection:
   - For PingFederate 10.1 or later: go to **Applications > Integration > SP Connections**. Click **Create Connection**.
   - For PingFederate 10.0 or earlier: go to **Identity Provider > SP Connections**. Click **Create Connection**.
2. Configure the basic connection details.
   1. On the **Connection Template** tab, select **Do not use a template for this connection**. Click **Next**
   2. On the **Connection Type** tab, select only **Outbound Provisioning**. From the **Type** list, select **SCIM Connector**. Click **Next**.
   3. On the **General Info** tab, in the **Partner's Entity ID** and **Connection Name** fields, enter a name of your choosing. Complete any other optional fields. Click **Next**.
3. On the **Outbound Provisioning** tab, configure the provisioning target and channel. Click **Next**.

   For help, see Configuring outbound provisioning in the PingFederate documentation.
   1. Click **Configure Provisioning**.
   2. On the **Target** tab, in the **SCIM URL** field, enter the **Base URL** that you noted in Adding PingFederate as SCIM provider in Code42.
   3. From the **SCIM Version** list, check that **2.0** is selected.
   4. From the **Authentication Method** list, select **Basic Authentication** or **OAuth 2 Bearer Token** based on the authentication method you selected in Adding PingFederate as SCIM provider in Code42.
   5. In the **Username** and **Password** fields or **Access Token** field, enter the authentication details that you noted in Adding PingFederate as SCIM provider in Code42.

      > **Note**
      > PingFederate verifies the authentication details when you activate the channel and SP connection.

Copyright © 2025 Ping Identity Corporation

6. From the **Unique User Identifier** list, select **userName**.
7. From the **Remove User Action** list, select **Disable**.
8. Under **Provisioning Options**, customize the provisioning connector actions as shown in SCIM Connector settings reference for Code42. Click **Next**.
9. On the **Manage Channels** tab, follow the steps in Creating a channel. Click **Done**.
10. On the **Outbound Provisioning** tab, click **Next**.

4. On the **Activation and Summary** tab, above the **Summary** section, click the toggle to turn on the connection. Click **Save**.

SCIM Connector settings reference for Code42

Lists the configuration settings for the SCIM Connector when integrating with Code42.

Field Name	Description
SCIM URL	The Code42 base URL that you noted in Adding PingFederate as SCIM provider in Code42.
SCIM Version	Select **2.0**.
Authentication Methods	Select **OAuth 2 Bearer Token**.
OAuth 2 Bearer Token	
Access Token	The OAuth access token that you noted in Adding PingFederate as SCIM provider in Code42.
SCIM Overrides	
Unique User Identifier	The attribute that uniquely identifies a user when PingFederate does not have access to the unique user ID that the target application assigns to a user. Select **userName**.
Provisioning Options	
User Create	**Selected** (default) – PingFederate creates users in the target service. **Cleared** – PingFederate does not create users in the target service.
User Update	**Selected** (default) – PingFederate updates existing users in the target service. PingFederate can also re-enable disabled users. **Cleared** – PingFederate does not update existing users in the target service.
User Disable / Delete	**Selected** (default) – If **User Update** is also enabled, PingFederate removes users in the target service according to the **Remove User Action** setting in the connection configuration. **Cleared** – PingFederate does not remove users in the target service.

Copyright © 2025 Ping Identity Corporation

Field Name	Description
Provision Disabled Users	Code42 does not support provisioning disabled users. If a disabled user is targeted in your data store, PingFederate creates an active user in Code42.
	Note If any of the above provisioning options are cleared, PingFederate logs a warning in the user workflow section of `provisioner.log` when the related action fails.
Remove User Action	This option applies when: • **User Disable / Delete** is selected, and • a previously-provisioned user no longer meets the condition set on the **Source Location** screen, or • a user has been disabled or deleted from the data store. Select **Disable**. Code42 does not support the delete option.

Creating a channel

Integrating PingFederate with Code42 requires a specific attribute mapping configuration.

About this task

For general information about creating a channel, see Managing channels in the PingFederate documentation

For more information about the attributes available in your channel configuration, see Supported attributes reference.

Steps

1. On the **Manage Channels** tab, click **Create**.
2. On the **Channel Info** tab, in the **Channel Name** field, enter a name for the channel. Click **Next**.
3. On the **Source** tab, from the **Active Data Store** list, select the data store that you created in Creating a provisioning connection. Click **Next**.
4. On the **Source Settings** and **Source Location** tabs, complete the configuration based on your environment.
5. On the **Attribute Mapping** tab, configure the attributes. You can use the following LDAP mappings, or adapt them for your data store type. Click **Next**.

Code42 attribute	Data store attribute
`userName`	`mail`
`workEmail`	`mail`

Copyright © 2025 Ping Identity Corporation

Code42 attribute	Data store attribute
`department`	`department`
`division`	`division`
`externalId`	`objectGUID`
`familyName`	`sn`
`givenName`	`givenName`
`manager`	Populate this attribute with the Code42 ID that represents this user's manager. For example, Jake's manager Alice has a user ID of `123098456876789543` in Code42. In Active Directory, you set Jake's `msDS-cloudExtensionAttribute10` attribute to `123098456876789543`. Then, map this `manager` attribute to `msDS-cloudExtensionAttribute10`.
`title`	`title`
`userType`	`employeeType`
`workCity`	`l` (lowercase "L")
`workCountry`	`c`
`workState`	`st`
`workStreet`	`streetAddress`
`workPostalCode`	`postalCode`

Supported attributes reference

Lists the attributes that can be mapped for user provisioning to Code42.

For more information on SCIM attributes see the SCIM specification and the SCIM enabled service provider documentation.

Attribute	Description
userName	A service provider's unique identifier for the user. This attribute is required.
givenName	The given name of the user, or first name in most Western languages (for example, 'Barbara' given the full name 'Ms. Barbara Jane Jensen, III').
familyName	The family name of the user, or last name in most Western languages (for example, 'Jensen' given the full name 'Ms. Barbara Jane Jensen, III').

Copyright © 2025 Ping Identity Corporation

Attribute	Description
workEmail	Work email for the user (for example, "bjensen@example.com").
title	The user's title, such as "Vice President".
externalId	A string that is an identifier for the resource as defined by the provisioning client.
workCity	The work city or locality component for the user's mailing address.
workState	The work state or region component for the user's mailing address.
workCountry	The work country component for the user's mailing address. When specified, the value MUST be in ISO 3166-1 "alpha-2" code format ISO3166; for example, the United States and Sweden are "US" and "SE", respectively.
workStreet	The work street address for the user, which can include the street number, street name, P.O. box, and multi-line extended street address information.
workPostalCode	The work ZIP or postal code component for the user's mailing address.
division	The division for the user.
department	The department for the user.
manager	The user ID for the user's manager in Code42.

Single sign-on setup

Exporting SAML metadata from PingFederate

Export a metadata file that describes your PingFederate identity provider configuration.

About this task

You can find general information about these steps in Metadata export in the PingFederate documentation.

Steps

1. In the PingFederate administrative console, go to the **Metadata Export** window.

   Choose from:

   - For PingFederate 10.1 or later: go to **System > Protocol Metadata > Metadata Export**.
   - For PingFederate 10.0 or earlier: go to **System > Metadata Export**.

2. If you see the **Metadata Role** tab, select **I am the identity provider (IdP)**. Click **Next**.
3. On the **Metadata Mode** tab, select **Select information to include in metadata manually**. Click **Next**.

Copyright © 2025 Ping Identity Corporation

4. On the **Protocol** tab, click **Next**.
5. On the **Attribute Contract** tab, click **Next**.
6. On the **Signing Key** tab, select a signing certificate. Click **Next**.
7. (Optional) On the **Metadata Signing** tab, select a certificate to sign the metadata XML file. Click **Next**.
8. On the **XML Encryption Certificate** tab, select the certificate that you want to use to encrypt the XML content. Click **Next**.
9. On the **Export & Summary** tab, click **Export**.
10. Save `metadata.xml`.
11. Click **Done**.

Adding PingFederate as an authentication provider in Code42

To allow PingFederate to coordinate authentication with Code42, upload your SAML metadata.

Steps

1. Sign on to Code42 as an administrator.
2. Navigate to **Integrations > Identity Management**.
3. On the **Authentication** tab, click **Add authentication provider**.
4. On the **Add authentication provider** dialog, enter a name and select **Upload file**. Select the file that you exported in Exporting SAML metadata from PingFederate. Click **Create Provider**.
5. On the provider detail page, note the **Code42 Service Provider Metadata URL**.

Creating a single sign-on connection

To allow PingFederate to handle single sign-on (SSO) to Code42, create a service provider (SP) connection.

About this task

> **Note**
> You can follow these steps to create a new SP connection, or you can modify your provisioning connection.

Steps

1. In the PingFederate administrator console, create a new SP connection:

   Choose from:

   - For PingFederate 10.1 or later: go to **Applications > Integration > SP Connections**. Click **Create Connection**.
   - For PingFederate 10.0 or earlier: go to **Identity Provider > SP Connections**. Click **Create Connection**.

2. On the **Connection Template** tab, select **Do not use a template for this connection**. Click **Next**.

Copyright © 2025 Ping Identity Corporation

3. On the **Connection Type** tab, select **Browser SSO Profiles** and clear any unwanted types. Click **Next**.
4. On the **Connection Options** tab, select only **Browser SSO**. Click **Next**.
5. On the **Import Metadata** tab, select **URL**, and then configure the Code42 metadata information.
    1. Click **Manage Partner Metadata URLs**.
    2. On the **SP Connection > Partner Metadata URLs** tab, click **Add New URL**.
    3. On the **SP Connection > Partner Metadata URLs > Metadata URL** tab, in the **Name** field, enter a name, such as `Code42`.
    4. In the URL field, paste the **Code42 Service Provider Metadata URL** that you noted in Adding PingFederate as an authentication provider in Code42. Clear the **Validate Metadata Signature** check box. Click **Next**.
    5. On the **Summary** tab, click **Done**.
    6. On the **SP Connection > Partner Metadata URLs** tab, click **Save**.
    7. On the **SP Connection > Import Metadata** tab, from the **Metadata URL** list, select the Code42 URL. Click **Load Metadata**. Click **Next**.
6. On the **General Info** tab, the basic connection information is populated by the metadata XML file. Click **Next**.
7. On the **Browser SSO** tab, configure browser SSO. Click **Next**.

    You can find a complete guide in Configure IdP Browser SSO in the PingFederate documentation.
    1. On the **Browser SSO > SAML Profiles** tab, select only **SP-Initiated SSO**.
    2. On the **Browser SSO > Protocol Settings > Allowable SAML Bindings** tab, select only **POST**.
    3. On the **Browser SSO > Protocol Settings > Signature Policy** tab, select **Always sign assertion**.
8. On the **Credentials** tab, configure the connection credentials.

    You can find a complete guide in Configuring credentials in the PingFederate documentation.
9. On the **Activation and Summary** tab, above the **Summary** section, turn on the connection. Click **Save**.

Copyright © 2025 Ping Identity Corporation

Code42 Integration Guide for PingOne

You can integrate PingOne with Code42 for user provisioning and single sign-on (SSO).

Features

- Manages users in Code42 based on changes in the PingOne identity store.
  - Creates, updates, and disables users.
- Enables browser-based single sign-on initiated by the identity provider (IdP).

Intended audience

This document is intended for PingOne administrators.

If you need help during the setup process, see the following resources:

- The following sections of the PingOne documentation:
  - Provisioning
  - Creating a SCIM connection
  - Adding an application
  - Adding a user
- The following sections of the Code42 documentation:
  - Identity management
  - Identity management reference
  - Introduction to SCIM provisioning
  - How to configure SCIM provisioning
  - Add authentication provider

Provisioning setup

Adding PingOne as SCIM provider in Code42

To get your base URL and credentials to access Code42 API, add PingOne as a SCIM provider.

Steps

1. Sign on to Code42 as an administrator.
2. Navigate to **Integrations > Identity Management**.
3. On the **Provisioning** tab, click **Add provisioning provider**, and then click **Add SCIM provider**.

Copyright © 2025 Ping Identity Corporation

4. On the **Add SCIM provisioning** dialog, enter a name and select the **OAuth token** credential type. Click **Next**.
5. On the **SCIM Provider Created** dialog, note the **Base URL** and **Token** values. You will use these in Creating a provisioning connection in PingOne. Click **Done**.

Creating a provisioning connection in PingOne

To enable PingOne to manage users in Code42, create a provisioning connection.

Before you begin

If you don't want to provision users from an existing PingOne population, you can add a new one. For help, see Managing populations in the PingOne documentation.

Steps

1. In PingOne, go to **Integrations > Provisioning**.
2. Click **+**, then click **New Connection**.
3. In the **Choose a connection type** section, select **Identity Store**.
4. On the **SCIM Outbound** tile, click **Select**, and then click **Next**.
5. Enter a name and description for the provisioning connection, then click **Next**.

   The connection name will appear in your configured connections list after you complete and save the connection.
6. In the **Configure Authentication** section, enter the Code42 details:
   1. In the **SCIM base URL** field, enter the base URL that you noted in Adding PingOne as SCIM provider in Code42.
   2. For **Users Resource**, use the default value of **/Users**.
   3. For **SCIM Version**, use the default selection, **2.0**.
   4. In the **Authentication Method** list, select **OAuth 2 Bearer Token**.
   5. In the **OAuth Access Token** field, enter the **Token** that you noted in Adding PingOne as SCIM provider in Code42.
   6. For the **Auth Type Header**, use the default selection, **Bearer**.
   7. Click **Test Connection**.
   8. Resolve any issues that are reported, and then click **Next**.
7. In the **Configure Preferences** section, enter the provisioning options.
   1. Use the following default filter and deprovisioning actions:

Setting	Value
User Filter Expression	`username Eq "%s"`

Copyright © 2025 Ping Identity Corporation

Setting	Value
User Identifier	`userName`
Remove Action	**Disable** **Note** Code42 doesn't support user deletion.

2. **Optional:** Customize the provisioning options in the following table:

Setting	Description
Allow users to be created	Determines whether PingOne creates a user in Code42 when the user is created in PingOne.
Allow users to be updated	Determines whether PingOne updates a user's attributes in Code42 when the user attributes change in PingOne.
Allow users to be disabled	Determines whether PingOne disables a user in Code42 when the user is disabled in PingOne.

3. Click **Save**.

8. To enable your new provisioning connection, in its details pane, click the toggle.

Creating a provisioning rule in PingOne

Before you begin

Complete the Creating a provisioning connection in PingOne procedure.

About this task

Create a provisioning rule and assign the provisioning connection that you created in the previous procedure as the **Target** connection.

To create a provisioning rule:

Steps

1. Go to **Integrations > Provisioning**.
2. Click **+**, then **New Rule**.
3. Enter a name and description for the rule, and then click **Create Rule**.

Copyright © 2025 Ping Identity Corporation

> **Note**
>
> **Description** is an optional field.

4. On the **Configuration** tab, assign a target connection:
    1. Click **Target**.
    2. In the **Available Connections** section, in the row for the provisioning connection that you created in the previous procedure, click + to add it as the target connection.

       You can add a disabled connection to the target, but you must enable the connection before you can enable the associated rule.

       Result:

       PingOne is automatically selected as the **Source** connection.
5. Click **Save**.

   Result:

   The rule's name appears in the list on the **Rules** tab.
6. To enable the new rule, in its details pane, click the toggle.

Configuring a user filter on the provisioning rule

Before you begin

Complete the following procedures in order:

1. Creating a provisioning connection in PingOne
2. Creating a provisioning rule in PingOne

About this task

Configure a user filter on the provisioning rule that you created in the Creating a provisioning rule in PingOne procedure to specify which PingOne user populations to provision to Code42:

Steps

1. On the **Configuration** tab of the provisioning rule, click **User Filter**.
2. Click the **Pencil** icon to edit the filter.
3. Define the filter that determines which identities to provision to Code42.

   For more information, see Adding a user filter and Example user filters.
    1. Enter the first condition:

Copyright © 2025 Ping Identity Corporation

All

Select **All** or **Any** to determine how to evaluate the linked conditions. **All** functions as the boolean logical operator `AND`.

Any

Select **All** or **Any** to determine how to evaluate the linked conditions. **Any** functions as the boolean logical operator `OR`.

Attribute

The user attribute to filter by.

> **Note**
>
> Use the **Population Name** attribute.

Operator

The operator determines the context of the attribute and its value.

> **Note**
>
> Equals is the only operator supported at this time.

Value

Enter an appropriate value for the specified attribute.

> **Note**
>
> - For outbound provisioning rules, you must specify either a population or a group to define the users to be provisioned.
> - If you select a group in the filter, then updating or deleting the group can cause the provisioning rule to re-sync.
> - If you select a group in the filter, the filter includes all users with any kind of membership in the group, whether direct, dynamic membership based on a user filter, or inherited from parent groups. For more information, see Groups.

2. Click **+ Add** to add more conditions or condition sets until you've specified all the PingOne user populations that you want to provision to Code42.

4. Click **Save**.

Configuring attribute mapping on the provisioning rule

Before you begin

Complete the following procedures in order:

1. Creating a provisioning connection in PingOne
2. Creating a provisioning rule in PingOne

Copyright © 2025 Ping Identity Corporation

About this task

Configure attribute mapping on the provisioning rule that you created in the Creating a provisioning rule in PingOne procedure to map the PingOne user attributes from the SCIM attributes in the Code42 identity store. For outbound provisioning, the mapping is applied to the attribute coming from the PingOne directory before it is saved to the target identity store.

To configure attribute mapping:

Steps

1. On the **Configuration** tab of the provisioning rule, click **Attribute Mapping**.
2. Click the **Pencil** icon to edit the mapping and confirm that the following default attributes are within the mapping.

Code42 SCIM Attribute	PingOne PingOneUser Attribute
`userName`	`Username`
`givenName`	`Given Name`
`familyName`	`Family Name`
`workPhone`	`Primary Phone`
`workEmail`	`Email Address`

   1. If any of the default attributes are missing from the attribute mapping, add them. To add an attribute mapping, click **+ Add**, then select the source and target attribute.

      You must click **+ Add** for each attribute that you need to map.

      Learn more about configuring attribute mapping in Adding attribute mapping for outbound provisioning.

3. Map the following Code42 SCIM attributes.

Code42 SCIM Attribute	PingOne PingOneUser Attribute
`externalId`	`External ID`
`workCity`	`Locality`
`workStreetAddeess`	`Street Address`
`workCountry`	`Country Code`
`workPostalCode`	`Postal Code`
`userType`	`Type`
`title`	`Title`

Copyright © 2025 Ping Identity Corporation

Code42 SCIM Attribute	PingOne PingOneUser Attribute
`nickName`	`Nickname`
`formattedName`	`Formatted`

4. Click **Save**.

Testing user provisioning

Check your provisioning connection by adding a user.

Steps

1. In PingOne, create a user:
    1. Go to **Directory > Users** and click the **+** icon.
    2. In the **Username** field, enter a username in the form of an email address.
    3. In the **Population** list, select the population that you chose in Creating a provisioning connection in PingOne.
    4. Click **Save**.
2. Check for the user in Code42:
    1. In Code42, go to **Environment > Users**.
    2. Look for the user that you created in PingOne.

       If the user doesn't appear in Code42:
        1. In PingOne, go to **Directory > Users** and open the details pane for the use that you created.
        2. In the **Services** list, select **Sync Status**.
        3. Verify that the user is synced to your provisioning identity rule.

Single sign-on setup

Creating an SSO connection in PingOne

To allow PingOne to handle single sign-on (SSO) to Code42, add a Security Assertion Markup Language (SAML) application profile.

Steps

1. In PingOne, go to **Applications > Applications** and click **+**.
2. On the **Add Application** pane, enter a unique **Application Name**.
3. In the **Application Type** section, click **SAML Application**, then click **Configure**.

Copyright © 2025 Ping Identity Corporation

4. On the **SAML Configuration** page, enter your SAML details:
    1. Click **Manually Enter**.
    2. Enter the following connection details using your own Code42 console address.

    > **Tip**
    >
    > You can find your Code42 console address in the address bar of your browser when you sign on to Code42.
    >
    > 
    >

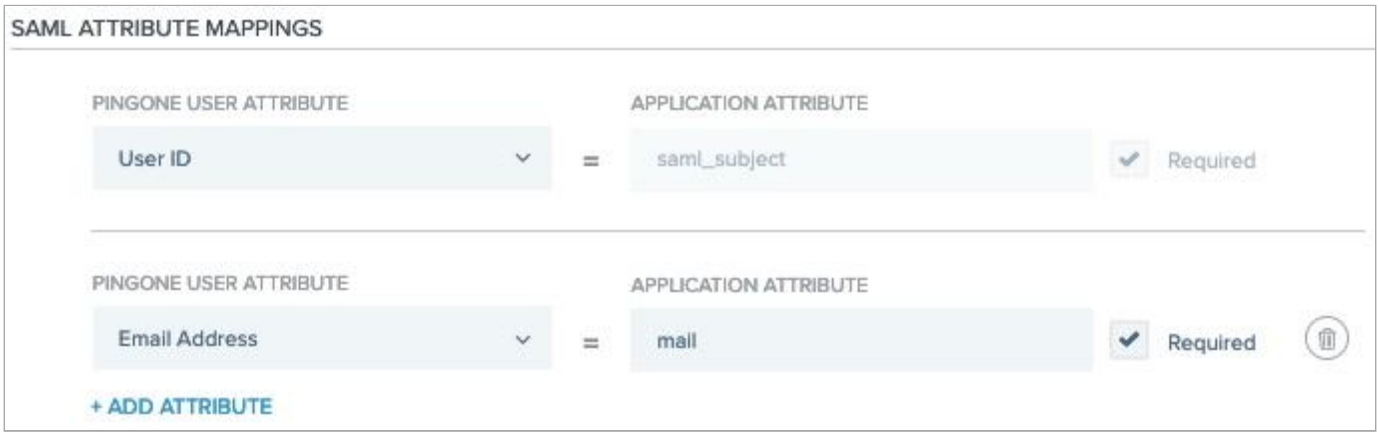
Setting	Value
ACS URLs	`https://Code42-console-address/api/SsoAuthLoginResponse`
Entity ID	`https://Code42-console-address`

    3. Click **Save**, then click the **Pencil** icon and enter the following configuration details.

Setting	Value
Assertion Validity Duration	`180` or your preferred duration.
Target Application URL	`https://Code42-console-address/console`

    Use the default settings for the rest of the configuration.

    4. Click **Save**.
5. On the **Attribute Mappings** tab, map the PingOne `Email Address` attribute to `mail`:



    1. Click the **Pencil** icon, then click **+ Add**.
    2. In the **Attributes** field, enter **mail**.
    3. In the **PingOne Mappings** list, select **Email Address**.

Copyright © 2025 Ping Identity Corporation

> **Tip**
>
> If the `Email Address` attribute is unavailable, go to **Directory > User Attributes**, select the **Show disabled attributes** checkbox, and click the toggle next to **email**. Return to the **Attribute Mapping** tab for your application.

4. Select the **Required** checkbox, then click **Save**.

6. To enable the application, in its details pane, click the toggle.

7. On the **Configuration** tab, click **Download Metadata** to save the `saml2-metadata-idp-***.xml` file.

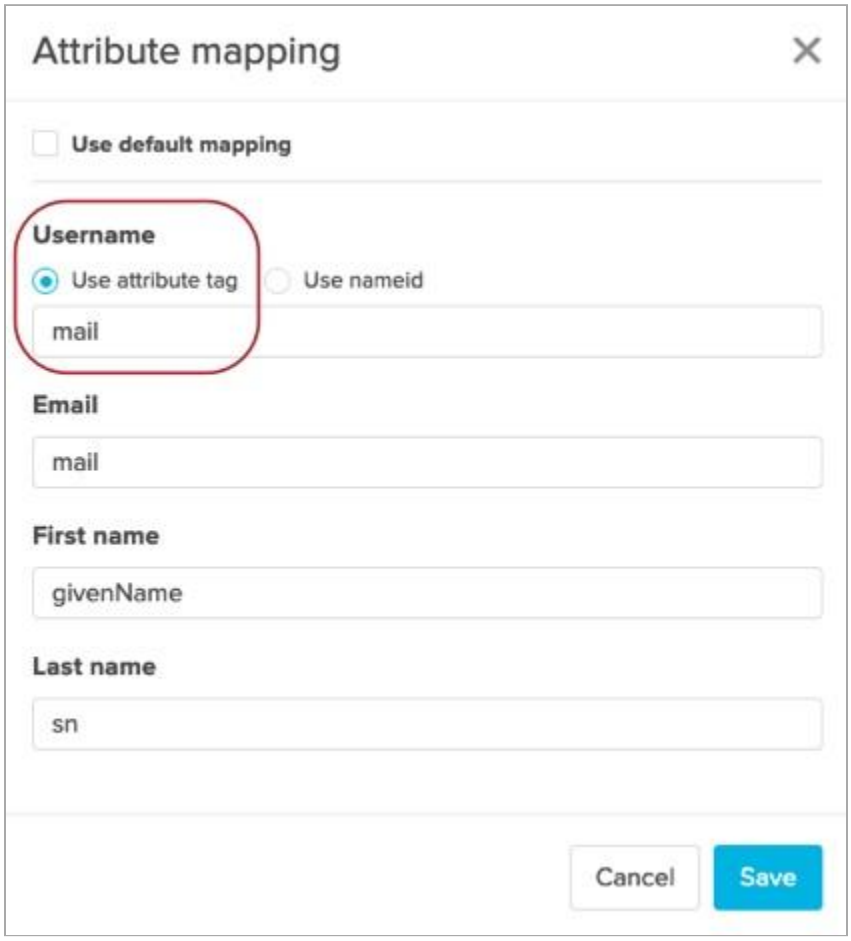
Adding PingOne as an authentication provider in Code42

To allow PingOne to coordinate authentication with Code42, upload your SAML metadata.

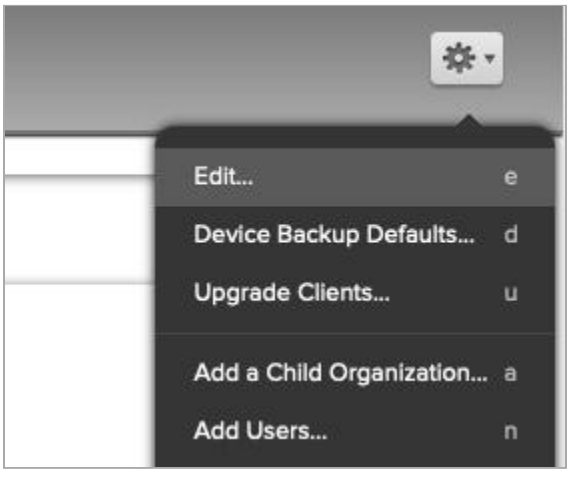
Steps

1. Sign on to Code42 as an administrator.
2. Navigate to **Integrations > Identity Management**.
3. On the **Authentication** tab, click **Add authentication provider**.
4. On the **Add authentication provider** dialog, enter a name and select **Upload file**. Select the `saml2-metadata-idp-***.xml` file that you downloaded in Creating an SSO connection in PingOne. Click **Create provider**.
5. In the **Attribute mapping** section, click the pencil icon.
6. Clear the **Use default mapping** checkbox.
7. In the **Username** field, enter `mail`.

Copyright © 2025 Ping Identity Corporation



8. Click **Save**.
9. Go to **Environment > Organizations**.
10. Add a new organization, or select an existing one.
11. In the organization details, on the **Settings** list, select **Edit**.



12. On the **Security** tab, in the **Servers** section, clear the **Inherit security settings from parent** checkbox.
13. On the **Select an authentication method** list, select **SSO**, then select the authentication provider you added.

Copyright © 2025 Ping Identity Corporation

14. On the **Select a directory service** list, select **Local** for testing purposes.
15. Click **Save**.
16. If you see a prompt to enable single sign-on, enter `ENABLE`, then click **Enable**.

Testing single sign-on

Check your single sign-on (SSO) connection by adding a user and signing on.

Steps

1. Create a user and provision it to Code42. For help, see the **Configure provisioning** section.
2. Set a password for the user in PingOne.
3. In Code42, make sure the user is part of the organization you configured in Adding PingOne as an authentication provider in Code42.
4. Go to https://*Code42-console-address*/login/#/login and enter the username.
5. When you are redirected to PingOne, enter the username and password to sign on.

Copyright © 2025 Ping Identity Corporation

Concur Provisioner

The PingFederate Concur Provisioner enables an enterprise to provision users to Concur. A quick connection template is also included to simplify the configuration of single sign-on (SSO).

> **Note**
>
> Updates and support have ended for this integration. The availability of this user guide and the related product download is intended only for those who have existing solutions that use this integration. Learn more about Concur in the Concur website.

Features

- Browser-based IdP-initiated SSO
- Includes support for user life cycle management (including creates, updates, and disables).

Intended audience

This document is intended for PingFederate administrators.

If you need help during the setup process, see the following sections of the PingFederate documentation:

- SP connection management
- Identity provider SSO configuration
- Datastores
- Configuring outbound provisioning
- Configuring outbound provisioning settings

System Requirements

- PingFederate 7.3 or later with Java 8.

User management

The Concur Provisioner synchronizes users from your datastore to Concur. The following describes the behavior of each provisioning capability.

> **Tip**
>
> You can configure the following capabilities and specify which users to provision when you get to the Configure provisioning part of the setup process.

Copyright © 2025 Ping Identity Corporation

Synchronizing existing users

PingFederate synchronizes users based on the `empId` attribute in Concur. If a user already exists in your datastore and Concur, mapping this attribute correctly links the two records together.

For example:

- In Concur, Janet's `empId` is `123abc`.
- In your datastore, Janet's `employeeID` is `123abc`.
- On the **Attribute Mapping** tab of your provisioning connection configuration, map the `empId` attribute to `employeeID`.
- When the provisioning connector runs, the datastore user is provisioned with a `empId` of `123abc`. That matches Janet's existing `empId` in Concur, so her information in the datastore is synchronized to her Concur account.

User provisioning

PingFederate provisions users when any of the following happens:

- A user is added to the datastore group or filter that is targeted by the provisioning connector.
- A user with `disabled` status is added to the datastore group or filter that is targeted by the provisioning connector, and the **Provision disabled users** provisioning option is enabled. This feature is not available in all provisioning connectors.

> **Tip**
>
> You can define which users PingFederate targets for provisioning on the **Source Location** tab of your provisioning connection configuration.

User updates

PingFederate updates users when a user attribute changes in your datastore.

> **Tip**
>
> You can define which attributes PingFederate monitors for changes on the **Attribute Mapping** tab of your provisioning connection configuration.

User deprovisioning

PingFederate deprovisions users when any of the following happens:

- A user is deleted from the user store.
- A user is disabled in the user store.
- A user is removed from the datastore group or filter that is targeted by the provisioning connector.

Setup

Copyright © 2025 Ping Identity Corporation

Obtain key and secret

About this task

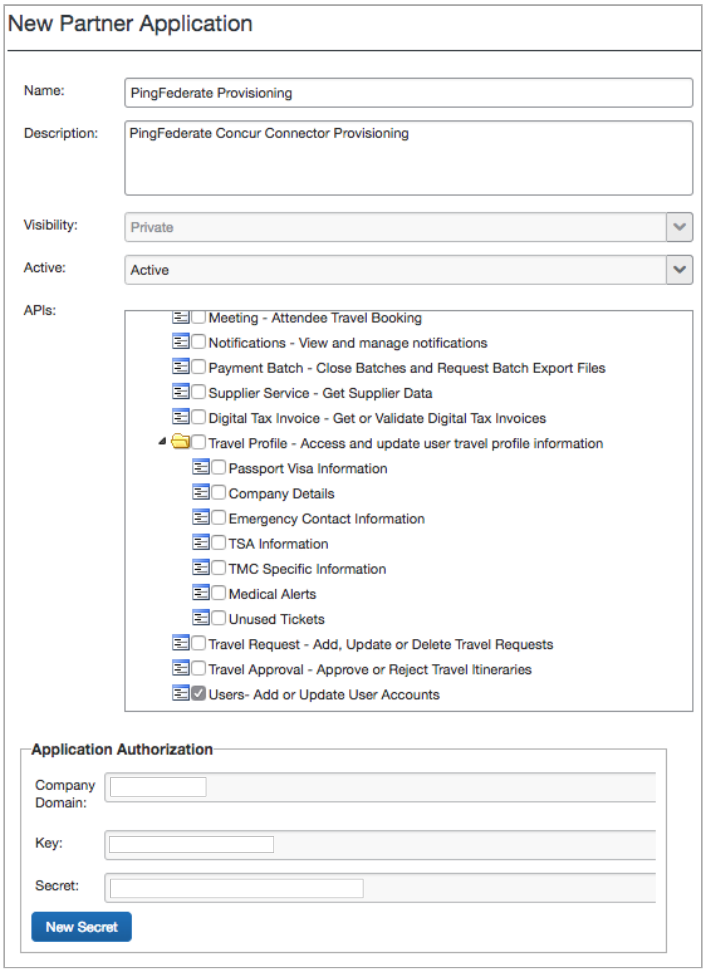
The Concur Connectors Outbound Provisioning functionality is built using Concur's REST API, which requires an OAuth 2.0 access token for authentication. To obtain the access token, you must first obtain your key and secret from Concur.

> **Note**
>
> Concur provides Web Services, such as provisioning, as an optional service to customers. Verify with Concur that you have Web Services available as part of your setup.

Steps

1. Sign on to your Concur account as an administrative user for your organization.
2. Go to **Administration > Company > Web Services**.
3. Select **Register Partner Application**.
4. Create a new application or modify an existing application with the following options:
    1. Enter a name for the application.
    2. Enter a description for the application.
    3. Ensure the **Active** field is set to **Active**.
    4. In the **APIs** list, ensure **Users- Add or Update User Accounts** is selected.
    5. Copy the Application Authorization's **Key** and **Secret** value for later use in Generate OAuth access token.

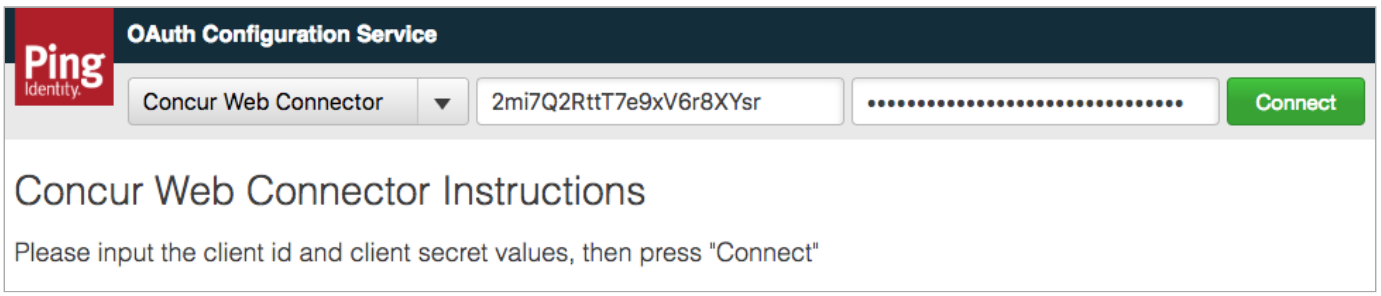
Copyright © 2025 Ping Identity Corporation



Generate OAuth access token

Steps

1. In a browser, go to the Ping Identity OAuth Configuration Service.
2. Select **Concur Web Connector** in the list.



3. Enter the **Key** value you previously obtained into the **ClientID** field.
4. Enter the **Secret** value you previously obtained into the **Client Secret** field.
5. Click **Connect**.

Copyright © 2025 Ping Identity Corporation

6. Sign on to Concur as an administrative user for your organization.

> **Note**
>
> If you are already signed into Concur you will not be asked to sign on again. Please be sure that the account you are signed in under is an administrative account.

7. Click the **Allow** button to generate your access token.
8. You should be redirected to the OAuth Configuration Service and presented with an access token.



9. Copy the **Access Token** to use when configuring the Concur Connector.

Install the connector

About this task

This section describes the common steps required to install the PingFederate Concur Connector.

If you experience issues with deployment, installation, or configuration, visit Ping Identity Support & Community.

Steps

1. Stop the PingFederate server if it is running.
2. Extract the Concur Connector distribution `.zip` archive.
3. Copy the contents of the `dist` directory into the directory:

   `<pf_install>/pingfederate/server/default/deploy`

Copyright © 2025 Ping Identity Corporation

4. (Optional) If you plan to use the connector for outbound provisioning, edit the `run.properties` file located in `<pf_install>/pingfederate/bin`, changing the property `pf.provisioner.mode` to `STANDALONE`. For example:

   ```
   pf.provisioner.mode=STANDALONE
   ```

   > **Note**
   >
   > To configure the `FAILOVER` mode instead, see Deploying provisioning failover in the PingFederate *Server Clustering Guide*.

5. Start the PingFederate server.

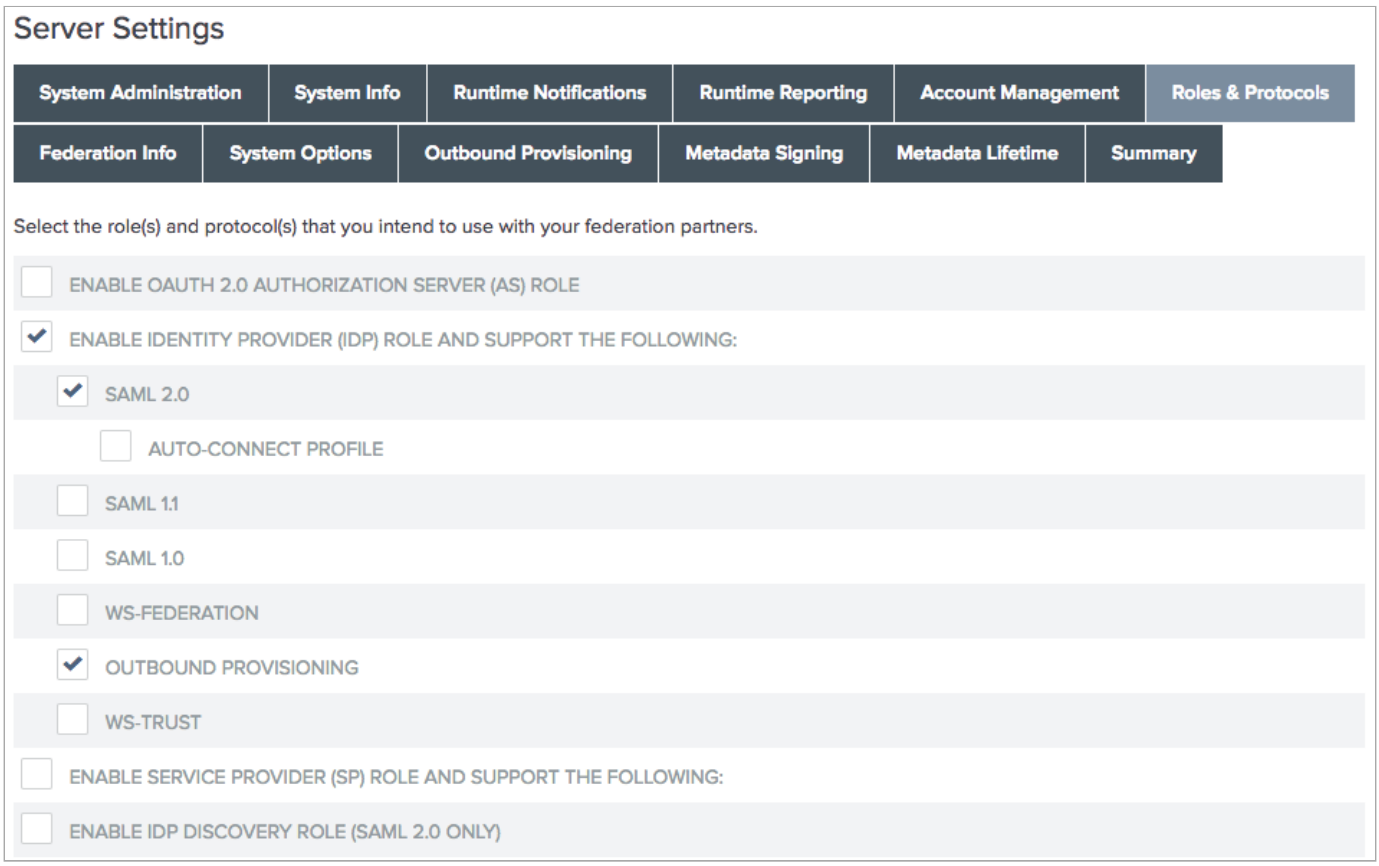
Enable outbound provisioning

About this task

After enabling outbound provisioning in the `<pf_install>/pingfederate/bin/run.properties` file, you must also activate the outbound provisioning role in the administrative console.

Steps

1. Go to the **Server Configuration > Server Settings > Roles & Protocols** page.
2. Select the **Outbound Provisioning** checkbox.



Copyright © 2025 Ping Identity Corporation

> **Note**
>
> Enabling outbound provisioning adds the outbound provisioning page, requiring the selection of a database to facilitate provisioning. Learn more in Configuring outbound provisioning settings or click **Help** from the configuration page.

Configure provisioning

About this task

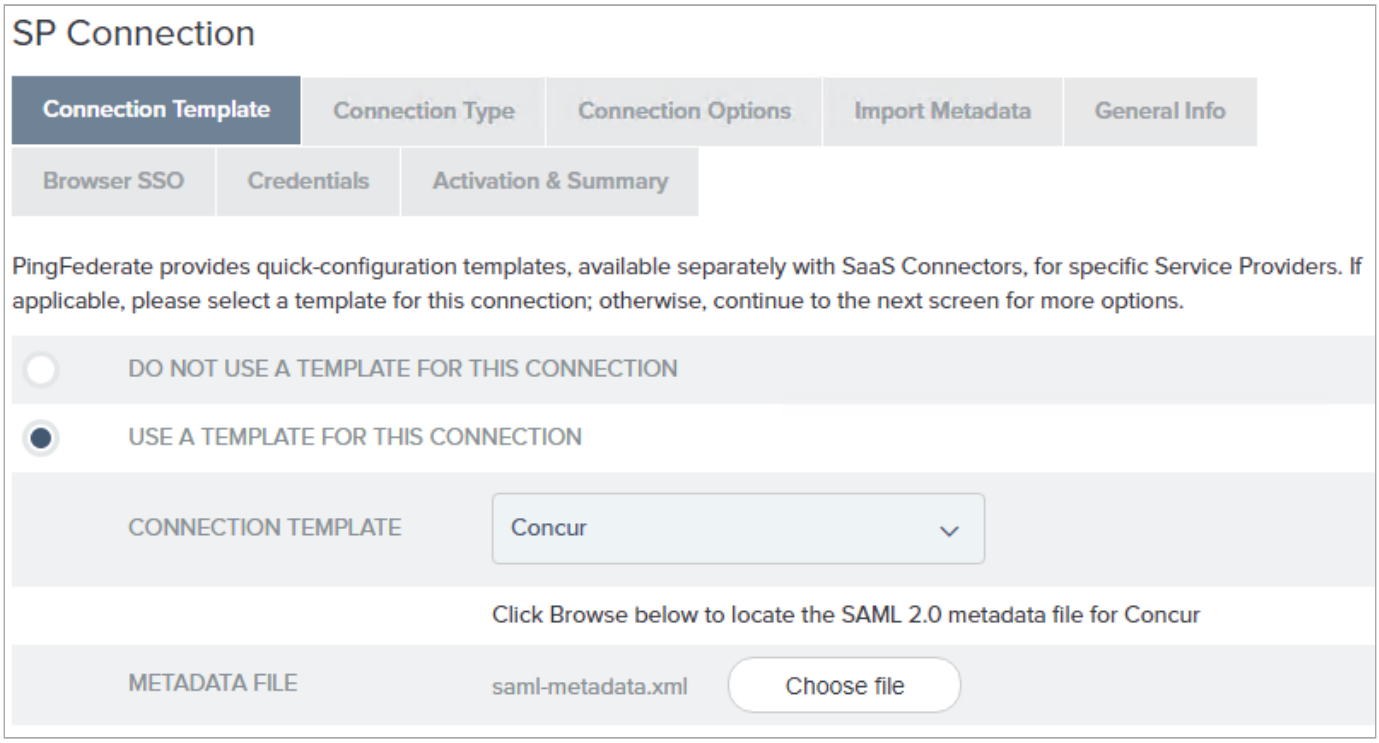
To configure a connection for outbound provisioning to Concur, follow the instructions in this section.

Outbound provisioning details are managed within a service provider (SP) connection. You can configure outbound provisioning with or without Browser SSO, WS-Trust STS, or both when you create a new SP connection. You can also add outbound provisioning to an existing SP connection.

Steps

1. In the PingFederate administrative console, configure the datastore that PingFederate will use as the source of user data. Learn more in Datastores in the PingFederate documentation.
    - When targeting users and groups for provisioning, exclude the user account that you will use to administer users in your connection to Concur. This prevents the PingFederate provisioning engine from interfering with the account that provisions users and groups.
2. Create a new SP connection or select an existing SP connection from the **SP Configuration** menu.
3. On the **Connection Template** page, select **Use a template for this connection** and select **Concur** in the **Connection Template** drop-down list. When asked during the connection configuration steps, import the `saml-metadata.xml` packaged with this connector.

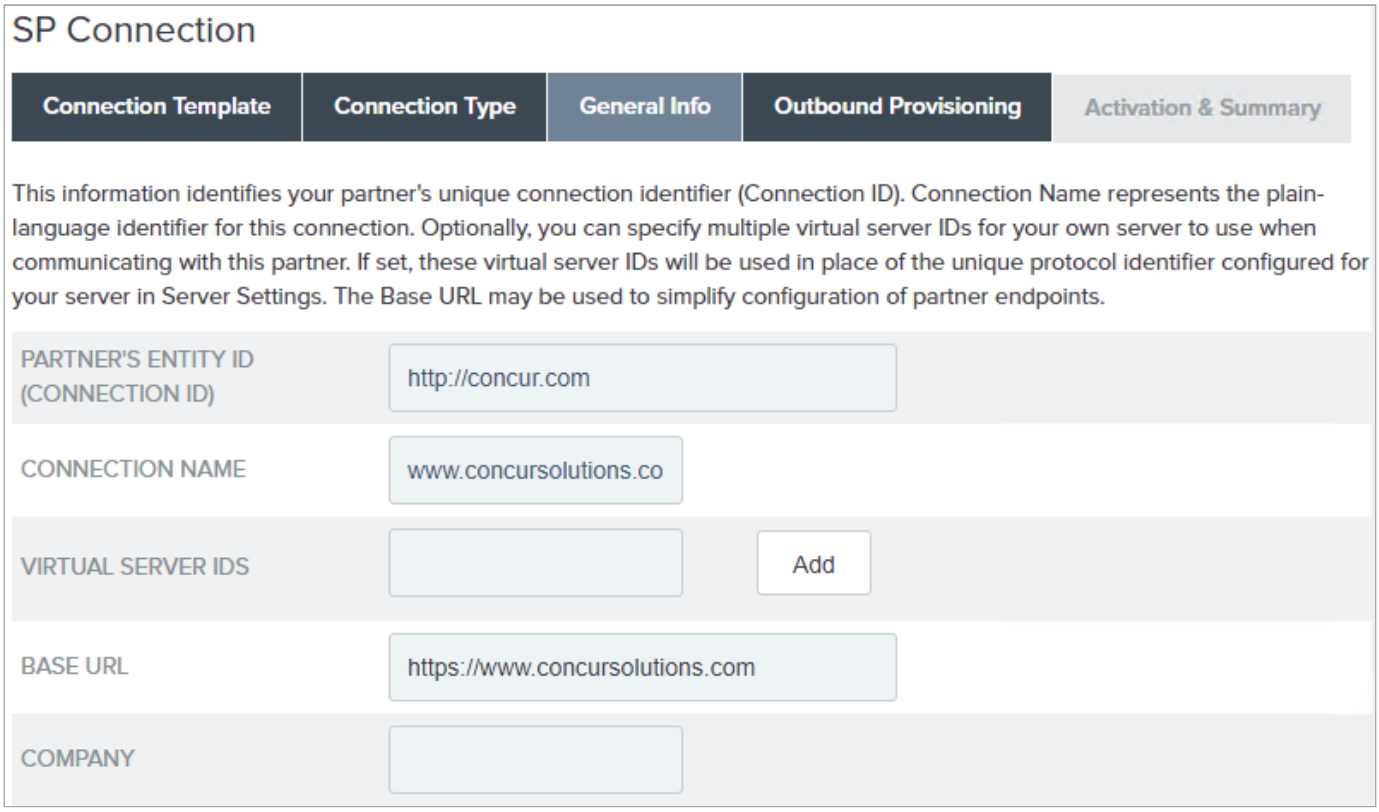
Copyright © 2025 Ping Identity Corporation



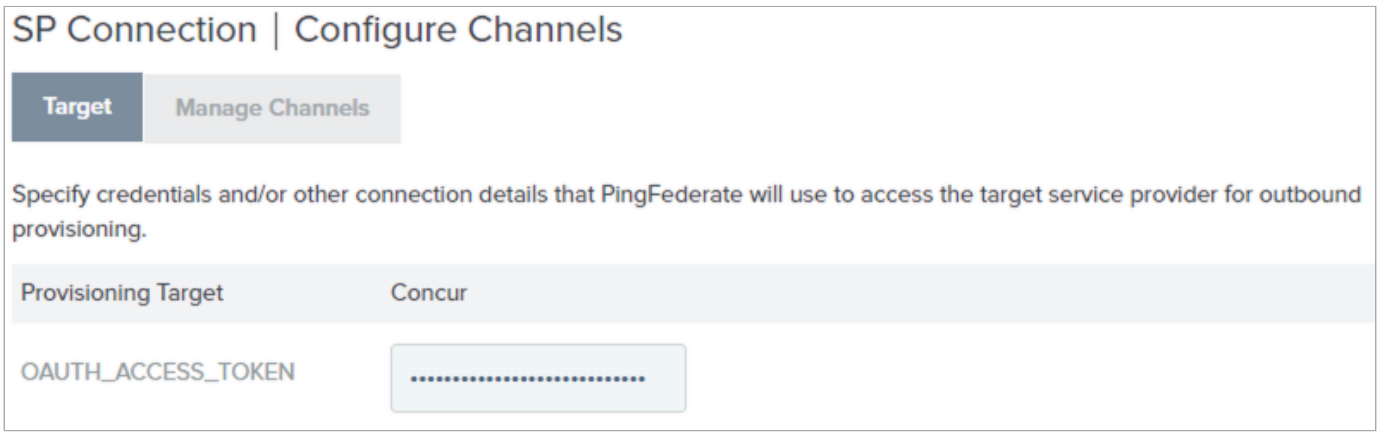
> **Tip**
>
> If this selection is not available, verify the connector installation and restart PingFederate.

4. On the **Connection Type** page, ensure the **Outbound Provisioning** checkbox is selected, and the **Browser SSO Profiles** checkbox is cleared (if appropriate).
5. On the **General Info** page, the default values are taken from the metadata file you selected in step 2. We recommend using the metadata default values.

Copyright © 2025 Ping Identity Corporation



6. Follow the connection wizard to configure the connection.
7. On the **Outbound Provisioning** page, click **Configure Provisioning**.
8. On the **Target** page, enter the values for each field as required by the Concur Connector.



Field Name	Description
OAUTH_ACCESS_TOKEN	The OAuth access token for the Concur account. Learn more in Obtain key and secret and Generate OAuth access token.

Copyright © 2025 Ping Identity Corporation

9. Click **Next** to continue the provisioning configuration. Learn more in the following sections under Outbound provisioning for IdPs in the PingFederate documentation:
    - Managing channels
    - Specifying channel information
    - Identifying the source datastore
    - Modifying source settings
    - Specifying a source location
    - Mapping attributes
    - Reviewing channel settings

    > **Important**
    >
    > Many fields are required based on your Concur account configuration. Ensure that you are sending data for all user fields that are required.

    > **Note**
    >
    > Credentials will be verified when the channel and SP connection is set to **Active** and provisioning is initiated.

    > **Tip**
    >
    > If you are not ready to complete the provisioning configuration, you can click **Save** and return to the configuration page later. To return to the configuration page, select the connection from **Identity Provider > SP Connections > Manage All**.

Supported attributes reference

The following table consists of the attributes that can be mapped on a user during provisioning.

Learn more in User in the Concur API reference documentation.

> **Important**
>
> Many fields are required based on your Concur account configuration. Ensure that you are sending data for all user fields that are required.

Attribute	Description
loginId	The URL-encoded Concur login of the user. This value must be unique. This attribute is required.
empId	The unique identifier for the user. This attribute is required.

Copyright © 2025 Ping Identity Corporation

Attribute	Description
emailAddress	The user's email address.
Password	The user's password. This element can be used to enter the password for a new user, but cannot be used to update the password for an existing user. This attribute is required.
crnKey	The 3-letter ISO 4217 currency code for the user's reimbursement currency. For example, United States Dollar is USD.
ctryCode	The ISO 3166-1 alpha-2 country code. For example, United States is US.
ctrySubCode	The user's two-character country code and two-character state or province code. For example, Washington State, United States is US-WA.
custom1 through custom21	The custom fields on the Employee form. Varies depending on configuration. There are two types of custom lists: simple lists and connected (multi-level) lists. We do not support connected lists.
expenseUserApprover	Whether the user is an Expense approver. Available options include: Y/N.
expenseUser	Whether the user has access to Expense. Available options include: Y/N.
firstName	The user's first name.
invoiceUserApprover	Whether the user is an Invoice approver. Available options include: Y/N.
invoiceUser	Whether the user has access to Invoice. Available options include: Y/N.
lastName	The user's last name.
ledgerKey	The user's assigned account code ledger. For example, Default. NOTE: This is required for new users.
localeName	The user's language locale code. Learn more in Locale Codes in the Concur API reference documentation. For example, United States English is en_US.
mI	The user's middle initial.
orgUnit1 through orgUnit6	The custom organizational unit fields on the Employee form. Varies depending on configuration.
tripUser	Whether the user has access to Travel. Available options include: Y/N.
isTestEmp	Whether the user is a test user. Available options include: Y/N.

Copyright © 2025 Ping Identity Corporation

Configure SAML SSO

The following section describes the steps for configuring single sign-on (SSO) to Concur.

> **Note**
>
> Configuring SSO is optional for outbound provisioning.

Configure PingFederate for SSO

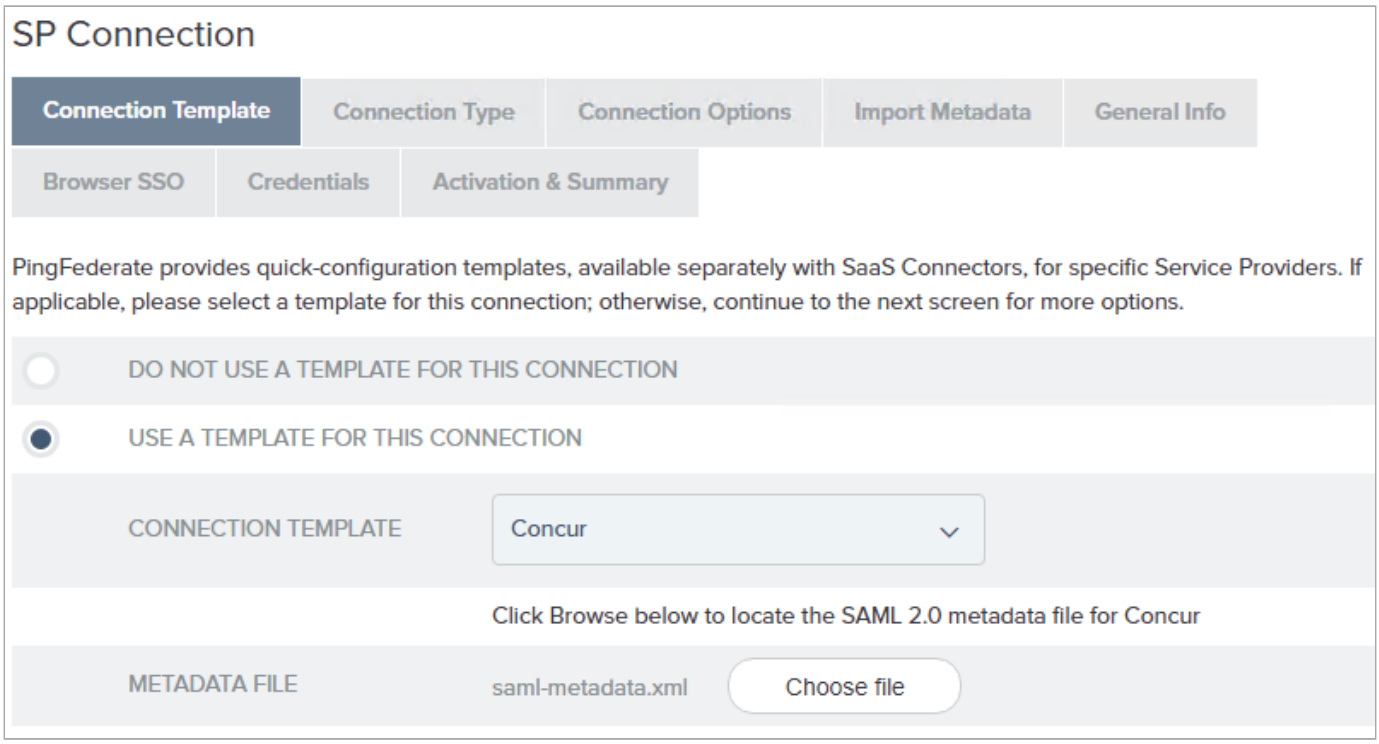
About this task

The following section describes the steps for configuring single sign-on (SSO) to Concur. Configuring SAML SSO involves configuring both the PingFederate SP connection and Concur.

> **Note**
>
> Configuring SSO is optional for outbound provisioning.

Steps

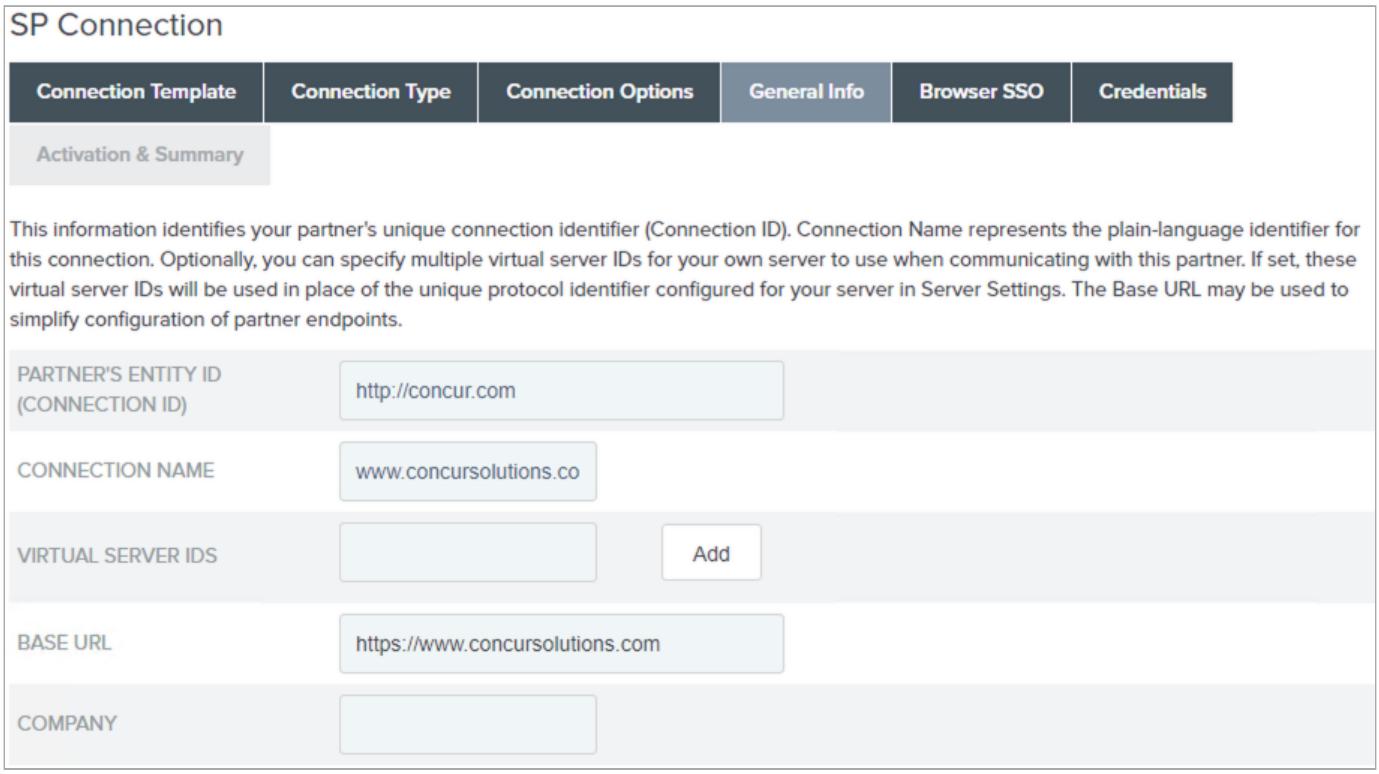
1. Create a new SP connection or select an existing SP connection from the **SP Configuration** menu.
2. On the **Connection Template** page, select the **Use a template for this connection** option and select **Concur** in the **Connection Template** drop-down list. When asked during the connection configuration steps, import the `saml-metadata.xml` packaged with this connector.



3. On the **Connection Type** page, ensure that the **Browser SSO Profiles** checkbox is selected.

Copyright © 2025 Ping Identity Corporation

4. On the **General Info** page, the default values are taken from the metadata file you selected in step 2. We recommend using the metadata default values.



5. Click **Next** to continue the Browser SSO configuration. Learn more in the following sections under Identity provider SSO configuration:
    - Managing IdP adapters
    - Configure IdP Browser SSO
    - Configuring credentials

> **Important**
>
> The SAML_SUBJECT configured on the **IdP Adapter Mapping > Attribute Contract Fulfillment** page must match the user's loginId configured in Concur.

6. On the **Credentials > Digital Signature Settings** page, select the signing certificate.
7. On the **Activation & Summary** page, set **Connection Status** to **Active**, then click **Save**.

> **Tip**
>
> If you are not ready to complete the SSO configuration, you can click **Save** and return to the configuration page later. To return to the configuration page, select the connection from **Identity Provider > SP Connections > Manage All**.

Copyright © 2025 Ping Identity Corporation

Configure Concur for SSO

About this task

To proceed with configuring Concur for single sign-on (SSO), you must have the following information from PingFederate.

- Identity provider certificate
  - This is the X.509 certificate from PingFederate used for SSO in your SP connection. Learn more in Managing digital signing certificates and decryption keys.

To configure Concur for SSO:

Steps

1. Obtain the Base64 X.509 certificate that will be used for SSO in your SP connection.
2. Contact your Concur account representative to obtain a work order which will enable the Concur technical team to assist you in setting up SSO for your organization. Ensure the X.509 certificate is included in your request.

Release notes

Changelog

Concur Connector 1.0 – May 2015 (current release)

- Initial release
- Added support for user provisioning
- Added support for browser-based single sign-on

Known issues and limitations

- Due to a limitation with PingFederate 8.1 and earlier versions, when configuring two SP connections with the same provisioner, the second connection built may be pre-populated with the channel from the first connection. To avoid conflicts, delete this pre-populated channel and create a unique channel for each connection.
- When an LDAP user is deleted in a targeted group distinguished name (DN), the provisioning connector does not propagate the deletion until a new user is added to the group. This limitation is compounded when the **User Create** provisioning option is disabled. Learn more about solutions in SaaS provisioner does not remove the user in the Knowledge Base.
- Due to PingFederate limitations, user attributes cannot be cleared once set.
- Custom attributes that are connected (multi-level) lists are not supported.
- Concur does not have support UTF-8 for all user fields.
- Due to API limitations, fields required in your Concur organization cannot be made create-only.

Copyright © 2025 Ping Identity Corporation

- Due to API limitations, users that are hard-deleted or no longer targeted by the provisioner in the data store are not disabled on Concur. If this occurs, an error will continue to appear in the provisioning logs stating what has happened. To resolve the issue in PingFederate, the user's record must be manually removed from the `channel_user` table in the database that monitors provisioning.
- Due to API limitations, updating a user's `loginId` and `employeeId` in the same request will result in a new user being created rather than the existing user being updated.

Download manifest

The distribution `.zip` : Archive for the connector contains the following:

- `ReadMeFirst.pdf` : Contains links to this online documentation.
- `saml-metadata.xml` : The metadata used for Browser SSO
- `/legal` :
  - `Legal.pdf` : Copyright and license information.
- `/dist` : Contains libraries needed for the connector:
  - `pf-concur-quickconnection-1.0.jar` : PingFederate Concur Connector
  - `prov-cpl-2.0.2.jar` : PingFederate Common Provisioning Layer

Copyright © 2025 Ping Identity Corporation

Contentful Integration Guide

You can integrate PingFederate with Contentful for provisioning and single sign-on (SSO). This integration uses the downloadable SCIM Connector.

Features

- Manages users in Contentful based on changes in a datastore that is attached to PingFederate.
  - Creates and deletes users.
  - Allows you to enable the create and delete capabilities independently.

    > **Note**
    >
    > Contentful doesn't allow attribute updates for existing users. When a user is "deleted", they are removed from the organization, but not removed from the Contentful platform.

- Manages groups in Contentful based on changes in your datastore.
  - Creates, updates, and deletes groups.
  - Updates group memberships.
- Enables browser-based SSO initiated by the identity provider (IdP) or service provider (SP).

Intended audience

This document is intended for PingFederate administrators.

If you need help during the setup process, refer to the following resources:

- The following sections of the PingFederate documentation:
  - General setup
    - Datastores
    - SP connection management
  - Provisioning setup
    - Configuring outbound provisioning
    - Configuring outbound provisioning settings
  - SSO setup
    - Password Credential Validators
    - Managing digital signing certificates and decryption keys
    - Identity provider SSO configuration

Copyright © 2025 Ping Identity Corporation

- The following section of the Contentful documentation:
  - SCIM FAQ
  - FAQ / Single sign-on (SSO)
- The following sections of the SCIM Connector documentation:
  - SCIM Provisioner
  - Known issues and limitations

System requirements

- PingFederate 9.0 or later.
- To allow PingFederate to make outbound HTTPS connections, you might need to allow the https://*.contentful.com. hostname in your firewall.
- Owner access to your Enterprise Contentful organization.

Upgrading an existing deployment

If you're upgrading from a previous version of the SCIM Connector, note your existing service provider (SP) connection configuration and create a new connection.

Upgrading an existing deployment in PingFederate 10.1 or later

Steps

1. Back up your current PingFederate configuration as shown in Configuration archive in the PingFederate documentation.
2. In the PingFederate administrative console, go to **Applications > Integration > SP Connections** and select your connection.
3. Note the attribute mappings for your existing SP connection.

   Learn more in Mapping attributes in the PingFederate documentation.
4. Delete your existing SP connection.
   1. Go to **Applications > Integration > SP Connections**.
   2. For your existing connection, click **Select action**, and then click **Delete**. Click **Confirm.**
5. Complete the steps in Deploying the integration files.
6. Complete the steps in Creating a provisioning connection.
   - From **Outbound Provisioning > Manage Channels > Channel**, on the **Attribute Mapping** tab, configure the attribute mappings based on your notes.
7. (Optional) Complete the steps in Creating a single sign-on connection.

Copyright © 2025 Ping Identity Corporation

Upgrading an existing deployment in PingFederate 10.0 or earlier

Steps

1. Back up your current PingFederate configuration as shown in Configuration archive in the PingFederate documentation.
2. In the PingFederate administrative console, go to **Identity Provider > SP Connections** and select your connection.
3. Note the attribute mappings for your existing SP connection.

   Learn more in Mapping attributes in the PingFederate documentation.
4. Delete your existing SP connection.
    1. Go to **Identity Provider > SP Connections > Manage All**.
    2. For your existing connection, click **Select action**, and then click **Delete**. Click **Save.**
5. Complete the steps in Deploying the integration files.
6. Complete the steps in Creating a provisioning connection.
    - From **Outbound Provisioning > Manage Channels > Channel**, on the **Attribute Mapping** tab, configure the attribute mappings based on your notes.
7. (Optional) Complete the steps in Creating a single sign-on connection.

Deploying the integration files

To get started with the integration, deploy the SCIM Connector files to your PingFederate directory.

Steps

1. Download the SCIM Connector `.zip` archive from the Ping Identity Integration Directory.
2. Stop PingFederate.
3. If you're upgrading an existing deployment, delete the `pf-scim-quickconnection-<version>.jar` file from your `<pf_install>/pingfederate/server/default/deploy` directory.
4. Extract the `.zip` archive and merge the contents of the `dist` directory with your `<pf_install>/pingfederate` directory.
5. Enable the PingFederate provisioning engine:
    1. Open your `<pf_install>/pingfederate/bin/run.properties` file.
    2. Change `pf.provisioner.mode` to `STANDALONE`. Save the file.

    > **Note**
    >
    > To configure the `FAILOVER` mode instead, refer to Deploying provisioning failover in the PingFederate Server Clustering Guide.

Copyright © 2025 Ping Identity Corporation

6. Start PingFederate.
7. If you operate PingFederate in a cluster, repeat steps 2 - 4 and step 6 for each engine node.

Enabling provisioning and single sign-on in PingFederate

To use PingFederate for provisioning and single sign-on, configure an external datastore and set a SAML entity ID.

About this task

Your external datastore acts as the source of data for provisioning. PingFederate also uses an internal datastore to store the state of synchronization between the source datastore and the target datastore.

Learn more in Datastores and Configuring outbound provisioning settings in the PingFederate documentation.

Enabling provisioning and single sign-on in PingFederate 10.1 or later

Steps

1. In the PingFederate administrative console, configure the datastore for PingFederate to use as the source of user data.

   Learn more in Datastores in the PingFederate documentation.
2. Go to **System > Server > Protocol Settings > Federation Info**.
3. In the **SAML 2.0 Entity ID** field, enter a name for PingFederate to use when identifying itself to SAML partners.
4. On the **Outbound Provisioning** tab, in the **Provisioner Data Store** list, select the internal database that will store the synchronization state. Click **Save**.

   Learn more in Configuring outbound provisioning settings in the PingFederate documentation.

Enabling provisioning and single sign-on in PingFederate 10.0 or earlier

Steps

1. In the PingFederate administrative console, configure the datastore for PingFederate to use as the source of user data.

   Learn more in Datastores in the PingFederate documentation.
2. Go to **System > Protocol Settings > Roles & Protocols**.
3. Select the **Enable Identity Provider IdP Role and Support the Following** check box.
4. Select the **SAML 2.0** and **Outbound Provisioning** check boxes. Click **Next**.
5. Click the **Federation Info** tab.
6. In the **SAML 2.0 Entity ID** field, enter a name for PingFederate to use to identify itself to SAML partners.
7. On the **Outbound Provisioning** tab, in the **Provisioner Data Store** list, select the internal database that will store the synchronization state. Click **Save**.

Copyright © 2025 Ping Identity Corporation

Learn more in Configuring outbound provisioning settings in the PingFederate documentation.

Configuring provisioning

You can configure PingFederate to manage users in your Contentful organization based on the users in your datastore.

Steps

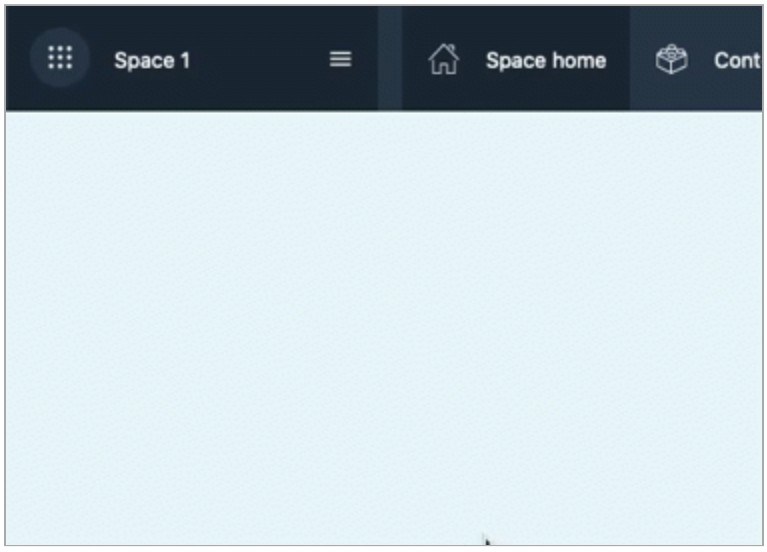
1. To give PingFederate access to Contentful, complete the steps in Creating an access token in Contentful.
2. To configure PingFederate for provisioning to Contentful, complete the steps in Creating a provisioning connection.

Creating an access token in Contentful

To allow PingFederate to access Contentful, generate an access token and get the SCIM URL.

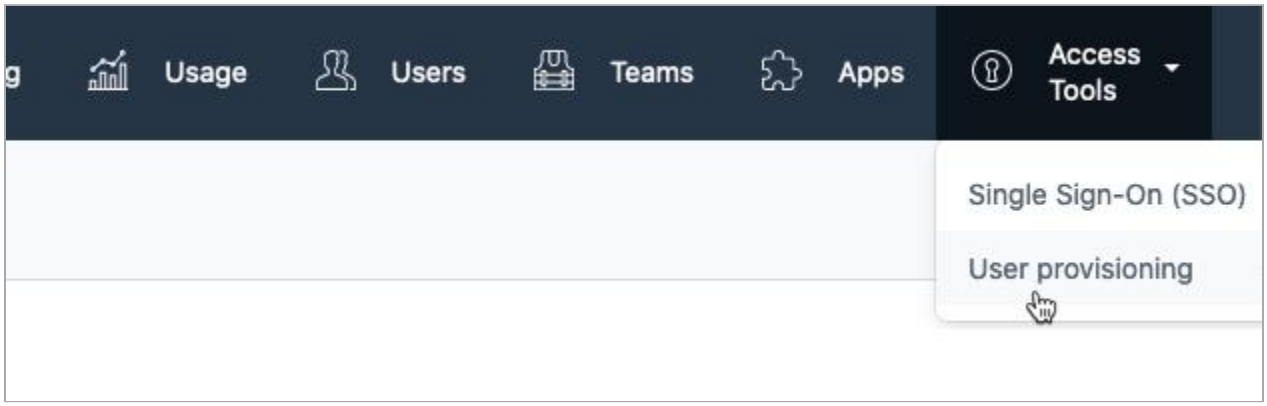
Steps

1. Sign on to Contentful as the organization owner.
2. Go to **Organization settings > User provisioning**.
    1. On the Contentful dashboard, from the **Space** menu, select **Organization settings & subscriptions**.



    2. From the **Access Tools** menu, select **User provisioning**.

Copyright © 2025 Ping Identity Corporation



3. On the **User Provisioning** window, click **Generate personal access token**. On the **Generate Personal Access Token** modal, enter a name of your choosing. Click **Generate**.
4. Note the access token and the **SCIM URL**.

   You'll use these in Creating a provisioning connection.

Creating a provisioning connection

To allow PingFederate to manage users in Contentful using SCIM, create a service provider (SP) connection.

Steps

1. In the PingFederate administrator console, create a new SP connection:

   Choose from:

   - For PingFederate 10.1 or later: go to **Applications > Integration > SP Connections**. Click **Create Connection**.
   - For PingFederate 10.0 or earlier: go to **Identity Provider > SP Connections**. Click **Create Connection**.

2. Configure the basic connection details.

   1. On the **Connection Template** tab, select **Do not use a template for this connection**. Click **Next**
   2. On the **Connection Type** tab, select only **Outbound Provisioning**. In the **Type** list, select **SCIM Connector**. Click **Next**.
   3. On the **General Info** tab, in the **Partner's Entity ID** and **Connection Name** fields, enter a name of your choosing. Complete any other optional fields. Click **Next**.

3. On the **Outbound Provisioning** tab, configure provisioning with the following details.

   Learn more in Configuring outbound provisioning in the PingFederate documentation.

   1. Click **Configure Provisioning**.
   2. On the **Target** tab, in the **SCIM URL** field, enter the **SCIM URL** that you noted in Creating an access token in Contentful.
   3. In the **SCIM Version** list, verify that **2.0** is selected.

Copyright © 2025 Ping Identity Corporation

4. In the **Authentication Method** list, select **OAuth 2 Bearer Token**.
5. In the **Access Token** field, enter the access token that you noted in Creating an access token in Contentful.

> **Note**
>
> PingFederate verifies the authentication details when you activate the channel and SP connection.

6. In the **Unique User Identifier** list, make sure **userName** is selected.
7. In the **Provisioning Options** section, select only **User Create** and **User Disable/Delete**.
8. In the **Remove User Action** list, select **Delete**. Click **Next**.

> **Note**
>
> This option deletes the user from your Contentful organization but not the Contentful platform. Contentful doesn't support the "disable" option.

9. On the **Manage Channels** tab, follow the steps in Creating a channel.
10. On the **Outbound Provisioning** tab, click **Next**.

4. On the **Activation and Summary** tab, above the **Summary** section, turn on the connection. Click **Save**.

Creating a channel

Integrating PingFederate with Contentful requires a specific attribute mapping configuration.

About this task

Learn more about creating a channel in Managing channels in the PingFederate documentation

Steps

1. On the **Configure Channels > Target** tab, click **Create**.
2. On the **Channel > Channel Info** tab, in the **Channel Name** field, enter a name for the channel. Click **Next**.
3. On the **Source** tab, from the **Active Data Store** list, select the data store that you created in Enabling provisioning and single sign-on in PingFederate. Click **Next**.
4. On the **Source Settings** and **Source Location** tabs, complete the configuration based on your environment.

   Learn more in Modifying source settings and Specifying a source location in the PingFederate documentation.
5. On the **Attribute Mapping** tab, configure the attributes as follows. You can use the following LDAP mappings, or adapt them for your datastore type. Click **Next**.

   Learn more about user attributes in the Users section of **SCIM API endpoints** in the Contentful documentation.

Copyright © 2025 Ping Identity Corporation

Contentful attribute	Datastore attribute
`userName`	`mail`
`email`	`mail`
`lastName`	`sn`
`firstName`	`givenName`

6. On the **Activation and Summary** tab, for the **Channel Status**, select **Active**. Click **Done**.
7. On the **Manage Channels** tab, click **Done**.

Configuring single sign-on

You can configure PingFederate to act as an identity provider for Contentful, allowing users to access Contentful through single sign-on (SSO).

About this task

> **Note**
>
> SSO integration is an optional part of this integration. If you only want to configure provisioning to Contentful, skip these steps.

Steps

1. To allow PingFederate to check user credentials against the datastore that you created in Enabling provisioning and single sign-on in PingFederate, create a password credential validator. Learn more in Password Credential Validators and Configuring the LDAP Username Password Credential Validator in the PingFederate documentation.
2. To provide users with a sign-on page, complete the steps in Creating an HTML Form Adapter instance.
3. To make it easier to create a connection to Contentful, complete the steps in Downloading the Contentful SAML metadata file.
4. To configure PingFederate for SSO to Contentful, complete the steps in Creating a single sign-on connection.
5. To make it easier to configure Contentful, complete the steps in Exporting SAML metadata from PingFederate.
6. To configure Contentful to authorize sign-on events from PingFederate,

Creating an HTML Form Adapter instance

If you don’t have an existing sign-on method set up, configure a new HTML Form Adapter instance. This will allow your users to sign on to Contentful through PingFederate.

Copyright © 2025 Ping Identity Corporation

Steps

1. In the PingFederate administrative console, create a new IdP adapter instance:

   Choose from:

   1. For PingFederate 10.1 or later: go to **Authentication > Integration > IdP Adapters**. Click **Create New Instance**.
   2. For PingFederate 10.0 or earlier: go to **Identity Provider > Adapters**. Click **Create New Instance**.

2. On the **Create Adapter Instances > Type** tab, set the basic adapter instance attributes.
   1. In the **Instance Name** field, enter a name for the adapter instance.
   2. In the **Instance ID** field, enter a unique identifier for the adapter instance.
   3. In the **Type** list, select **HTML Form IdP Adapter**. Click **Next**.
3. On the **IdP Adapter** tab, in the **Credential Validators** section, select your password credential validator.
   1. Click **Add new row to 'Credential Validators'**.
   2. In the list, select the password credential validator that you created in step 1 of Configuring single sign-on.
   3. Click **Update**.
4. Continue with the default **IdP Adapter** settings, or customize them based on Configuring an HTML Form Adapter instance in the PingFederate documentation. Click **Next**.
5. On the **Extended Contract** tab, add attributes the following attributes, or the equivalent for your datastore. Click **Next**.
   - `givenName`
   - `mail`
   - `sn`
6. On the **Adapter Attributes** tab, for **username**, select **Pseudonym**. Click **Next**.
7. On the **Adapter Contract Mapping** tab, click **Next**.
8. On the **Summary** tab, click **Done**.

Exporting your PingFederate signing certificate

To allow Contentful to trust communications from PingFederate, you need to provide the signing certificate that you'll use in your connection.

Steps

1. In the PingFederate administrative console, go to **Security > Signing & Decryption Keys & Certificates**.
2. If you don't have a signing certificate, create one.
   1. Click **Create New**.

Copyright © 2025 Ping Identity Corporation

2. On the **Create Certificate** tab, enter the certificate details. Click **Next**.

   Learn more in the **Creating new certificates** section of Managing digital signing certificates and decryption keys in the PingFederate documentation.

3. On the **Summary** tab, click **Save**.

3. In the **Action** column for the certificate that you want to use for your connection, click **Export**.
4. On the **Export Certificate** tab, select **Certificate only**. Click **Next**.
5. On the **Export & Summary** tab, click **Export**.
6. Save the `.crt` file.

   You'll use the contents of this file in Configuring single sign-on in Contentful.

7. Click **Done**.

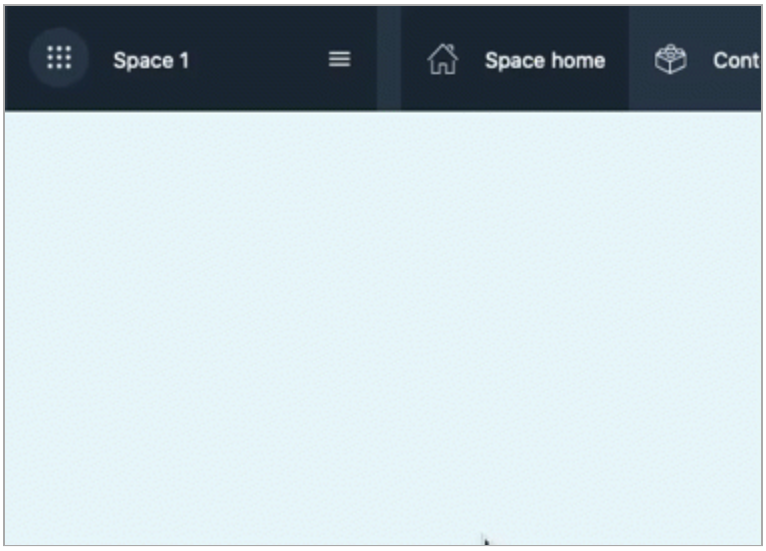
Downloading the Contentful SAML metadata file

To speed up the process of creating a service provider connection in PingFederate, download the connection details from Contentful.

Steps

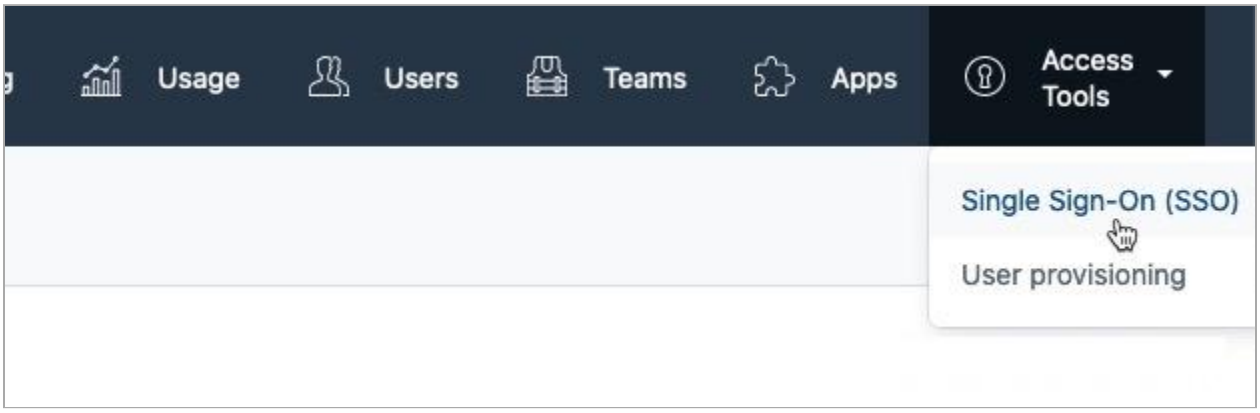
1. Sign on to Contentful as an organization administrator.
2. Go to **Organization settings > Single Sign-On (SSO)**.

   1. On the Contentful dashboard, from the **Space** menu, select **Organization settings & subscriptions**.

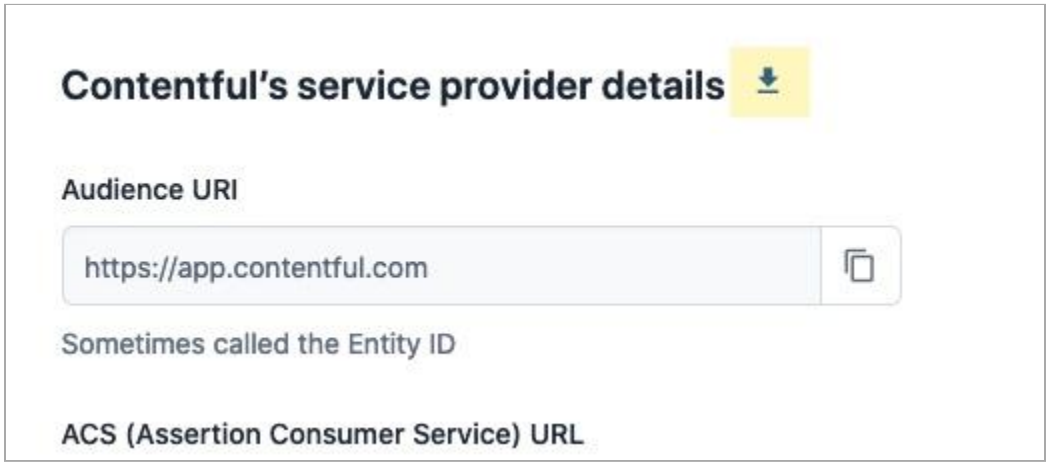


   2. From the **Access Tools** menu, select **Single Sign-On (SSO)**.

Copyright © 2025 Ping Identity Corporation



3. On the **Single Sign-On (SSO)** window, download the Contentful `metadata.xml` file.



> **Tip**
>
> Leave this page open. You'll return to it in Configuring single sign-on in Contentful.

Creating a single sign-on connection

To allow PingFederate to handle single sign-on (SSO) to Contentful, create a separate service provider (SP) SSO connection.

Steps

1. In the PingFederate administrator console, create a new SP connection:

   Choose from:

   - For PingFederate 10.1 or later: go to **Applications > Integration > SP Connections**. Click **Create Connection**.
   - For PingFederate 10.0 or earlier: go to **Identity Provider > SP Connections**. Click **Create Connection**.

2. On the **Connection Template** tab, select **Do not use a template for this connection**. Click **Next**.
3. On the **Connection Type** tab, select **Browser SSO Profiles**. From the **Protocol** list, select **SAML 2.0**. Click **Next**.
4. On the **Connection Options** tab, select only **Browser SSO**. Click **Next**.

Copyright © 2025 Ping Identity Corporation

5. On the **Import Metadata** tab, select **File**, and then upload the `metadata.xml` file that you downloaded in Downloading the Contentful SAML metadata file. Click **Next**.
6. If you see the **Metadata Summary** tab, click **Next**.
7. On the **General Info** tab, the basic connection information is populated by the metadata XML file. Click **Next**.
8. On the **Browser SSO** tab, complete the steps in Configuring Browser SSO.
9. On the **Credentials** tab, select your signing certificate.
    1. Click **Configure Credentials**.
    2. On the **Digital Signature Settings** tab, from the **Signing Certificate** list, select the certificate that you exported in Exporting your PingFederate signing certificate. Click **Next**.
    3. On the **Summary** tab, click **Done**.
    4. On the **SP Connection > Credentials** tab, click **Next**.
10. On the **Activation and Summary** tab, above the **Summary** section, turn on the connection. Click **Save**.

Configuring Browser SSO

For the single sign-on (SSO) part of your connection to Contentful, use the details below.

About this task

> **Tip**
>
> For all other settings, you can use the default or customize the configuration for your needs. Learn more in Configuring IdP Browser SSO in the PingFederate documentation.

Steps

1. On the **Browser SSO** tab, click **Configure Browser SSO**.
2. On the **Browser SSO > SAML Profiles** tab, select only **IDP-Initiated SSO** and **SP-Initiated SSO**. Click **Next**.
3. On the **Assertion Lifetime** tab, click **Next**.
4. On the **Assertion Creation** tab, configure the assertion.
    1. Click **Configure Assertion Creation**.
    2. On the **Attribute Contract** tab, set the following name formats. Click **Next**.

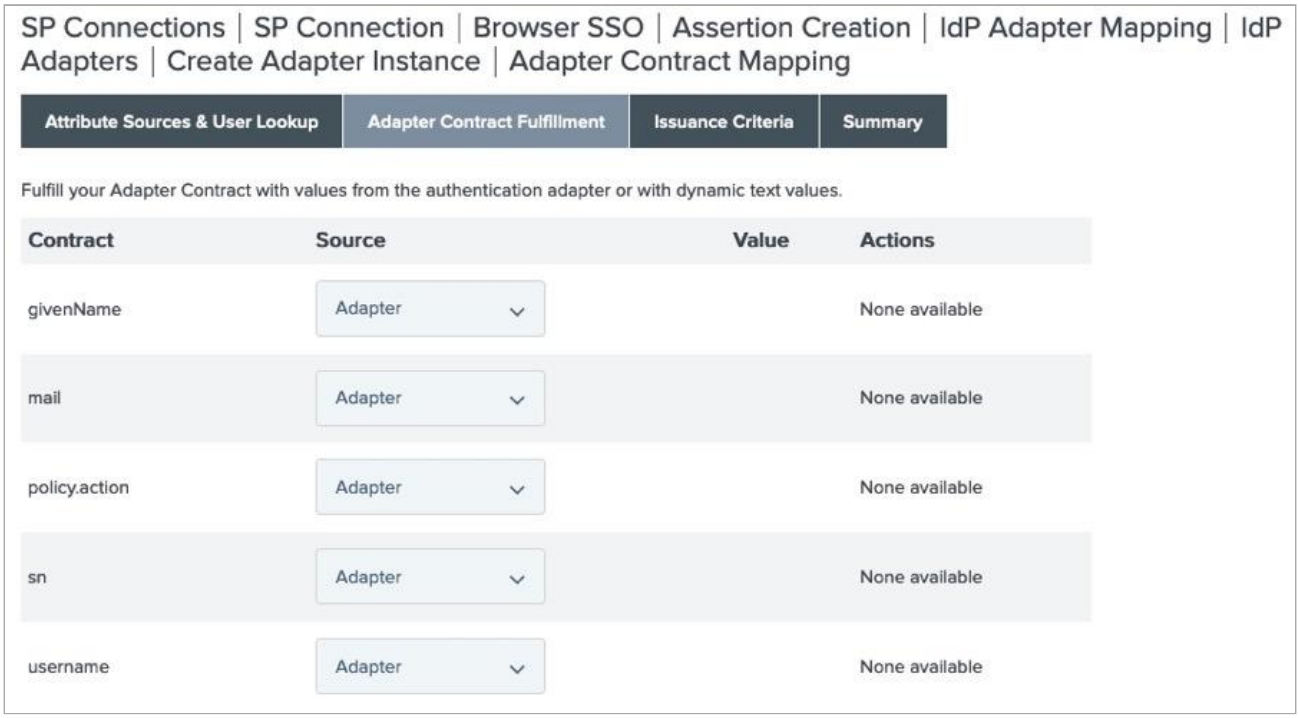
Attribute name formats

Attribute Contract	Subject Name format
SAML_SUBJECT	`urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified`

Copyright © 2025 Ping Identity Corporation

Attribute Contract	Subject Name format
email	`urn:oasis:names:tc:SAML:2.0:attrname-format:uri`
givenname	`urn:oasis:names:tc:SAML:2.0:attrname-format:basic`
surname	`urn:oasis:names:tc:SAML:2.0:attrname-format:basic`

3. On the **Authentication Source Mapping** tab, click **Map New Adapter Instance**.
4. On the **IdP Adapter Mapping > Adapter Instance** tab, in the **Adapter Instance** list, select the adapter instance that you created in Creating an HTML Form Adapter instance. Click **Next**.
5. On the **Mapping Method** tab, click **Next**.
6. On the **Attribute Contract Fulfillment** tab, select **Adapter** for all attributes. Click **Next**.



7. On the **Issuance Criteria** tab, click **Next**.
8. On the **Summary** tab, click **Done**.
9. On the **Browser SSO > Assertion Creation > Authentication Source Mapping** tab, click **Next**.
10. On the **Summary** tab, click **Done**.

5. On the **Protocol Settings** tab, configure the protocol settings.
    1. Click **Configure Protocol Settings**.
    2. On the **Assertion Consumer Service URL** tab, the default URL is populated by the Contentful `metadata.xml` file. Click **Next**.

Copyright © 2025 Ping Identity Corporation

3. On the **Allowable SAML Bindings** tab, select only **POST** and **Redirect**. Click **Next**.
4. On the **Signature Policy** tab, clear the **Require authn requests to be signed** check box. Click **Next**.
5. On the **Encryption Policy** tab, click **Next**
6. On the **Summary** tab, click **Done**.
7. On the **Browser SSO > Protocol Settings** tab, click **Next**.
8. On the **Summary** tab, click **Done**.

6. On the **SP Connection > Browser SSO** tab, click **Next**.

Exporting SAML metadata from PingFederate

Export a metadata file that describes your PingFederate identity provider configuration.

About this task

You can find general information about these steps in Metadata export in the PingFederate documentation.

Steps

1. In the PingFederate administrative console, go to the **Metadata Export** page.

   Choose from:

   - For PingFederate 10.1 or later: go to **System > Protocol Metadata > Metadata Export**.
   - For PingFederate 10.0 or earlier: go to **System > Metadata Export**.

2. If you see the **Metadata Role** tab, select **I am the identity provider (IdP)**. Click **Next**.
3. On the **Metadata Mode** tab, select **Select information to include in metadata manually**. Click **Next**.
4. On the **Protocol** tab, click **Next**.
5. On the **Attribute Contract** tab, click **Next**.
6. On the **Signing Key** tab, select a signing certificate. Click **Next**.
7. (Optional) On the **Metadata Signing** tab, select a certificate to sign the metadata XML file. Click **Next**.
8. On the **XML Encryption Certificate** tab, select the certificate that you want to use to encrypt the XML content. Click **Next**.
9. On the **Export & Summary** tab, click **Export**.
10. Save `metadata.xml`.
11. Click **Done**.

Configuring single sign-on in Contentful

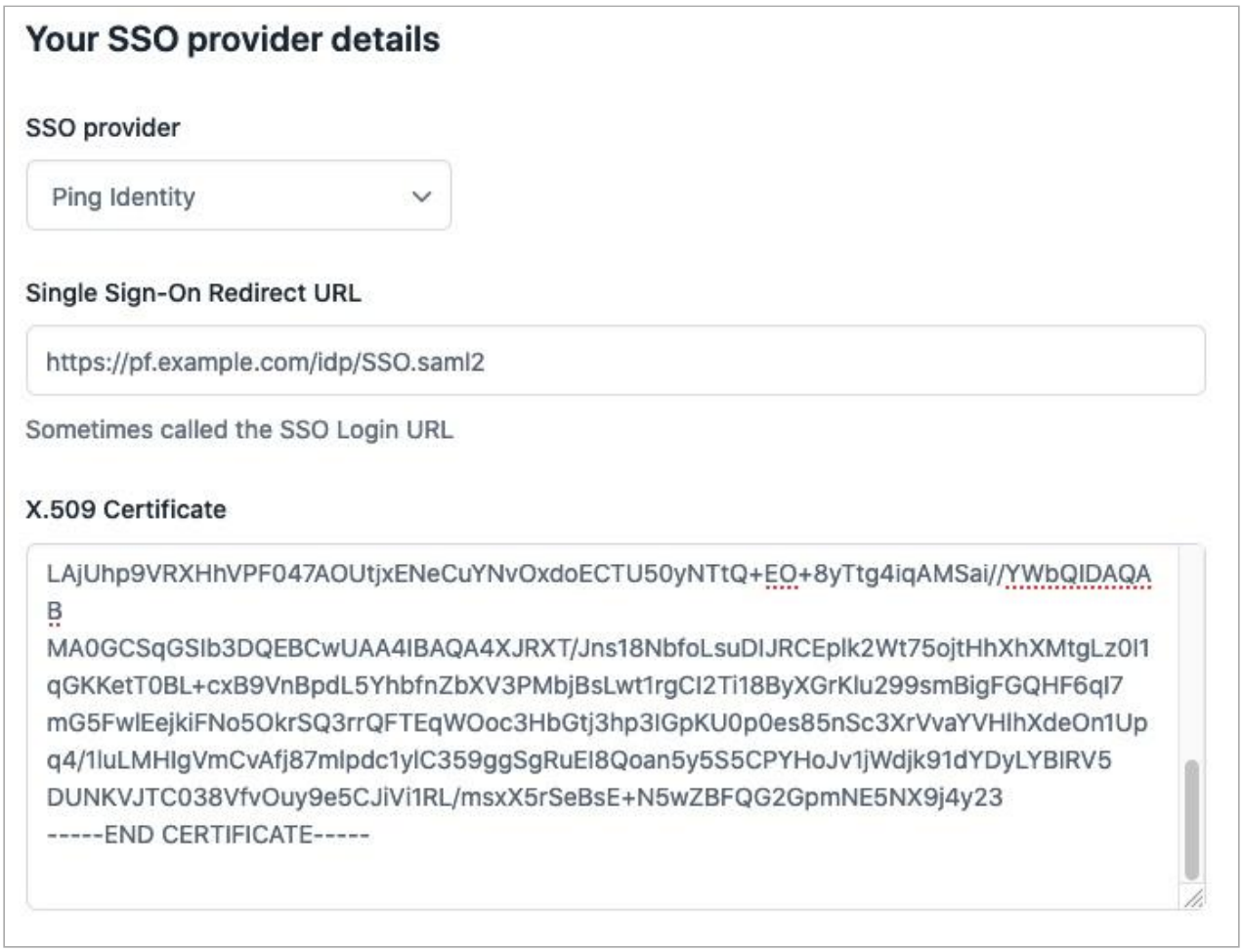
Complete the connection by entering your PingFederate URL and certificate in Contentful.

Copyright © 2025 Ping Identity Corporation

Steps

1. On the Contentful admin console, go to **Organization settings > Single Sign-On (SSO)**.
2. In the **SSO provider** list, select **Ping Identity**.
3. In the **Single Sign-On Redirect URL** field, enter `https://pf_host:pf_port/idp/SSO.saml2`.

   For example, https://pf.example.com:9031/idp/SSO.saml2.
4. Using a text editor, copy the contents of the `.crt` file that you downloaded in Exporting your PingFederate signing certificate.
5. In the **X.509 Certificate** field, paste the contents of the certificate file.



6. Click **Test connection**.
7. In the **SSO name** field, type a friendly name for the connection.
8. Check your settings. After enabling SSO in the next step, the only way to change your SSO settings is to contact Contentful.
9. Click **Enable SSO**.
10. **Optional:** If you want SSO to be the only sign-on method, contact Contentful and ask for Restricted SSO. This prevents users from signing on using email or third-party services, such as GitHub, Google, or Twitter.

Copyright © 2025 Ping Identity Corporation

CoreBlox integrations

PingFederate CoreBlox Integration Kit

The PingFederate CoreBlox Integration Kit allows PingFederate administrators to integrate their applications protected by a CoreBlox Token Service (CTS) with a PingFederate server acting as either an identity provider (IdP) or as a service provider (SP).

PingFederate CoreBlox Token Translator

The PingFederate CoreBlox Token Translator provides a token processor and token generator for use with the PingFederate's WS-Trust Security Token Service (STS).

CoreBlox Integration Kit

The CoreBlox Integration Kit allows PingFederate to integrate with applications that are protected by the CoreBlox Token Service (CTS).

Components

CoreBlox IdP Adapter

- Allows an identity provider enterprise to extend the CoreBlox domain to include partner applications with a variety of protocols, such as SAML, WS-Federation, OAuth, and OpenID Connect.
- Uses the CTS to validate and authorize tokens for use with PingFederate.

CoreBlox SP Adapter

- Allows a service provider enterprise to accept SAML or WS-Federation assertions and provide single sign-on (SSO) to applications that require a CoreBlox-issued token.
- Uses the CTS to create a CoreBlox token based on the attributes in the assertion.
- Can be configured to authorize the token against a specific resource.

Intended audience

This document is intended for PingFederate administrators.

If you need help during the setup process, learn more about available resources in:

- The following sections of the PingFederate documentation:
  - Managing IdP adapters
  - Managing SP adapters
- The following CoreBlox resources:
  - CoreBlox Token Service
  - CoreBlox Token Service Installation/Configuration from CoreBlox

Copyright © 2025 Ping Identity Corporation

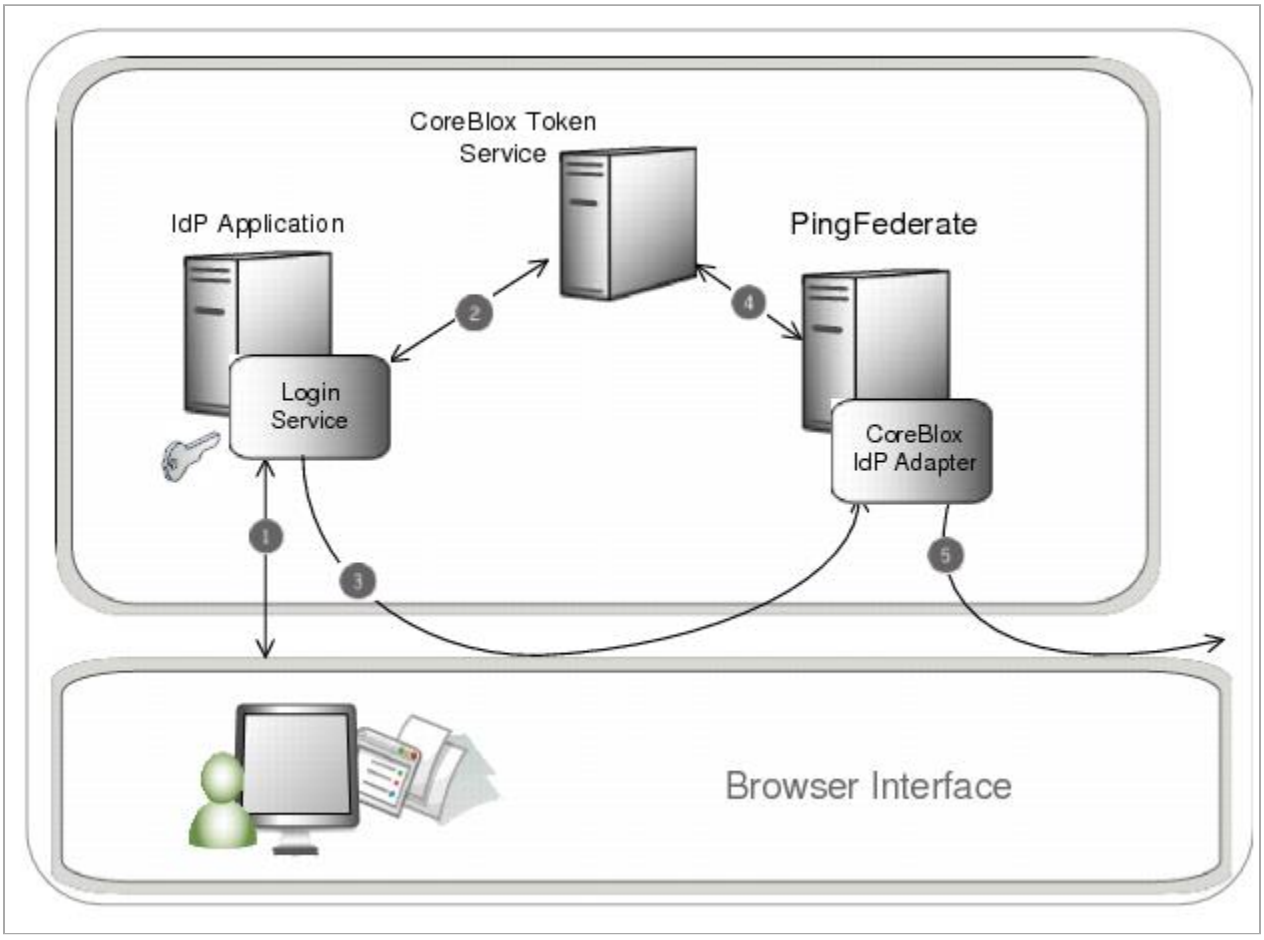
System requirements

- PingFederate 9.0 or later
- CoreBlox Token Service 1.5.4 (or later) acting as a secure token service between PingFederate and a policy server.
  - Available to download at no cost from https://coreblox.com/download/cts-1-5-4-2-war/.

Overview of the IdP SSO flow

With the CoreBlox Integration Kit, PingFederate allows the identity provider (IdP) to access user attributes from the CoreBlox Token Service (CTS).

The following figure illustrates an IdP-initiated single sign-on (SSO) scenario where PingFederate generates an assertion using a CoreBlox IdP Adapter session cookie.



Description

1. A user initiates an SSO transaction by authenticating with the IdP.
2. The login service authenticates the user with the the CoreBlox Token Service (CTS).
3. The IdP sets a session cookie in the browser and redirects the browser to PingFederate.

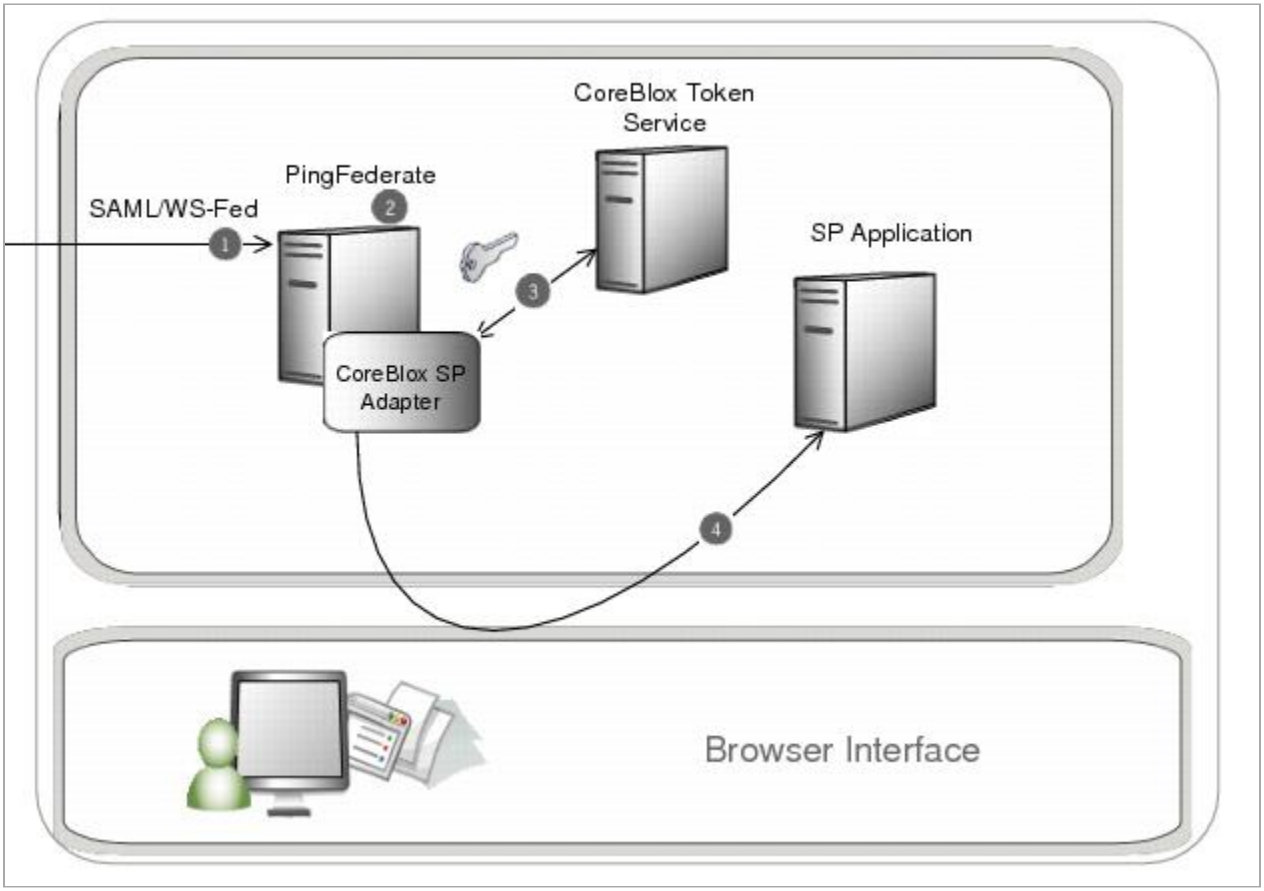
Copyright © 2025 Ping Identity Corporation

4. PingFederate uses the session cookie to query the CTS for user attributes associated with the session. The CTS returns the user attributes.
5. The adapter wraps the user attributes in an assertion. PingFederate redirects the browser to the service provider with the assertion.

Overview of the SP SSO flow

With the CoreBlox Integration Kit, PingFederate allows a service provider (SP) to start a CoreBlox session with user attributes from the identity provider (IdP).

The following figure illustrates an IdP-initiated single sign-on (SSO) scenario where PingFederate uses the CoreBlox SP Adapter to parse an assertion and create a CoreBlox session cookie.



Description

1. The PingFederate SP server receives an assertion from the IdP.
2. PingFederate parses the assertion.
3. The CoreBlox SP Adapter uses the user attributes to create and authorize a session with the CTS.
4. The adapter redirects the browser with a request that contains the session cookie.

Setup

Copyright © 2025 Ping Identity Corporation

Deploying the integration files

To get started with the integration, deploy the CoreBlox Integration Kit files to your PingFederate directory.

About this task

> **Note**
>
> If you operate PingFederate in a cluster, the following steps refer to the console node.

Steps

1. Download the CoreBlox Integration Kit `.zip` archive from the Ping Identity Integration Directory.
2. Stop PingFederate.
3. If you're upgrading an existing deployment, delete `coreblox-integration-kit-<version>.jar` from `<pf_install>/PingFederate/server/default/deploy`.
4. Extract the `.zip` archive and merge the contents of the `dist` directory with your `<pf_install>/pingfederate` directory.
5. Start PingFederate.
6. If you operate PingFederate in a cluster, repeat steps 2-5 for each engine node.

Configuring a CoreBlox IdP Adapter instance

Configure the CoreBlox IdP Adapter to determine how PingFederate communicates with the CoreBlox Token Service.

About this task

You can configure the adapter to direct authentication requests, authorization consent requests, or both to your application.

Steps

1. In the PingFederate administrative console, create a new IdP adapter instance:

   Choose from:

   - For PingFederate 10.1 or later: Go to **Authentication > Integration > IdP Adapters**. Click **Create New Instance**.
   - For PingFederate 10.0 or earlier: Go to **Identity Provider > Adapters**. Click **Create New Instance**.

2. On the **Type** tab, set the basic adapter instance attributes.
   1. In the **Instance Name** field, enter a name for the adapter instance.
   2. In the **Instance ID** field, enter a unique identifier for the adapter instance.
   3. In the **Type** list, select **CoreBlox IdP Adapter**. Click **Next**.
3. On the **IdP Adapter** tab, configure the adapter instance by referring to CoreBlox IdP Adapter settings reference. Click **Next**.

Copyright © 2025 Ping Identity Corporation

4. On the **Extended Contract** screen, add any attributes that you want to include in the assertion. Click **Next**.

> **Tip**
>
> The CoreBlox Token Service returns a `userAttributes` key that lists a set of attributes. You can use any of these attributes to extend the attribute contract.
> You can also include the `token` key in the extended attribute contract.

5. On the **Adapter Attributes** tab, set pseudonym and masking options as shown in Set pseudonym and masking options in the PingFederate documentation. Click **Next**.
6. On the **Adapter Contract Mapping** tab, configure the contract fulfillment details for the adapter as shown in Define the IdP adapter contract in the PingFederate documentation. Click **Next**.
7. On the **Summary** tab, check and save your configuration:

   Choose from:

   - For PingFederate 10.1 or later: Click **Save**.
   - For PingFederate 10.0 or earlier: Click **Done**. On the **Manage IdP Adapter Instances** tab, click **Save**.

8. Create an SP connection using this CoreBlox IdP Adapter instance. Learn more about Identity provider SSO configuration in the PingFederate documentation.

CoreBlox IdP Adapter settings reference

Field descriptions for the CoreBlox IdP Adapter configuration screen.

Standard fields

Field Name	Description
CoreBlox URL	The URL for the CoreBlox Token Service. This field is blank by default.
Validate CoreBlox Certificate Hostname	The hostname of the server certificate presented by the CoreBlox Token Service must match the hostname in the **CoreBlox URL** field. This checkbox is selected by default.
Client Certificate	The certificate that the adapter uses to authentication calls to the CoreBlox Token Service.
CoreBlox TokenType	The token type that the CoreBlox Token Service is configured to provide to the adapter. The default and only value is `SMSESSION`.
Cookie Name	The name of the cookie that contains the token used with the CoreBlox Token Service. This field is blank by default.

Copyright © 2025 Ping Identity Corporation

Field Name	Description
Cookie Domain	The domain name that the adapter uses when creating cookies. The browser compares this value to the domain of subsequent requests to determine whether the cookie should be submitted. If this field is blank, the adapter uses the domain name of the request. When sharing cookie across subdomains, this value must be prefixed with a period (`.`). This field is blank by default.
Cookie Path	The path that the adapter uses when creating cookies. The browser compares this value to the path of subsequent requests to determine whether the cookie should be submitted. The default value is `/` .
Cookie Secure Flag	The adapter writes cookies with the Secure flag. The browser only submits Secure cookies on subsequent HTTPS requests. This checkbox is selected by default.
Error URL	When an error occurs in the adapter, PingFederate redirects the browser to this URL instead of to the default error page. This URL can contain query parameters. The URL has an `errorMessage` appended to it, which contains a brief description of the error that has occurred. This field is blank by default.
Logged-Out Cookie Value	The expected value of the cookie when the user has been signed off. The default value is `LOGGEDOFF` .
HTTP Only Flag	The adapter sets a flag for the cookie. The flag indicates that the cookie should only be read through HTTP requests and disallows Javascript from accessing the cookie. This checkbox is selected by default.
Login URL	An optional URL for the authentication service. If the cookie isn't found in the request, PingFederate redirects the request to this URL along with the relative resume path. This field is blank by default.
Authentication Context	This can be any value agreed upon with your SP partner that indicates how the user was authenticated. The value is included in the SAML assertion. This field is blank by default.

Advanced fields

Field Name	Description
Perform Authorize Request	The adapter makes an authorize request to the CoreBlox Token Service before accessing the protected resource. This checkbox is cleared by default.

Copyright © 2025 Ping Identity Corporation

Field Name	Description
Resource	The resource that is protected by the agent. This field is required if **Perform Authorize Request** is selected. This field is blank by default.
Instance	Refers to the name of the agent instance. This field is required if **Perform Authorize Request** is selected. This field is blank by default.
Action	The action to take when evaluating requests against the policy server. This field is required if **Perform Authorize Request** is selected. This field is blank by default.
PingFederate Base URL	The base URL for PingFederate, such as `https://pf_host:pf_port/`. The adapter uses this value to create the return URL **Cookie Provider URL** to create the return URL. Complete this field if you're using a cookie provider to enable single sign-on (SSO) across multiple domains. This field is blank by default.
Cookie Provider URL	The URL of the cookie provider. PingFederate redirects the request to this URL if the session cookie is in a separate domain. Complete this field if you're using a cookie provider to enable SSO across multiple domains. This field is blank by default.
Cookie Provider Target Parameter	The name of parameter that contains the PingFederate return URL in the redirect to the cookie provider. Complete this field if you're using a cookie provider to enable SSO across multiple domains. This field is blank by default.
Session Cookie Prefix	The prefix to remove when **Remove Session Cookie Prefix** is selected. Complete this field if you're using a cookie provider to enable SSO across multiple domains. When using a cookie provider to enable SSO across multiple domains, tokens are prepended with the security zone name. This field works with the **Remove Session Cookie Prefix** setting to remove the security name prefix and to allow the adapter to process the token. Enter the **SSOZoneName** from your Agent Configuration Object (ACO) surrounded by `-`. For the default security zone name, enter `-SM-`. This field is blank by default.
Remove Session Cookie Prefix	The adapter removes the **Session Cookie Prefix** from the beginning of the session cookie. Select this check box if you're using a cookie provider to enable SSO across multiple domains. This checkbox is cleared by default.
Disable Refresh Session Cookie	The adapter doesn't refresh session cookies when validating them. This allows other dependent applications to manage the session cookie. This checkbox is cleared by default.

Copyright © 2025 Ping Identity Corporation

Configuring a CoreBlox SP Adapter instance

Configure the CoreBlox SP Adapter to determine how PingFederate communicates with your service provider (SP) application.

Steps

1. In the PingFederate administrative console, create a new SP adapter instance.

   Choose from:

   - For PingFederate 10.1 or later: Go to **Applications > Integration > SP Adapters**. Click **Create New Instance**.
   - For PingFederate 10.0 or earlier: Go to **Service Provider > Adapters**. Click **Create New Instance**.

2. On the **Type** tab, set the basic adapter instance attributes.
   1. In the **Instance Name** field, enter a name for the adapter instance.
   2. In the **Instance ID** field, enter a unique identifier for the adapter instance.
   3. From the **Type** list, select **CoreBlox SP Adapter**. Click **Next**.
3. (Optional) On the **Instance Configuration** tab, in the **Protected Resource Mapping Table** section, define conditions that the SAML assertion has to meet for the user to get access to a protected resource.

   > **Note**
   >
   > The CoreBlox Token Service allows you to grant permissions to specific realms in specific domains by defining the resource, instance, and action fields. These values are defined in your CTS Agent Config Object (ACO) settings. Learn more on page 29 of the the CoreBlox Token Service Install and Configuration Guide (1.0 v3) [PDF] documentation and page 27 of the the CoreBlox Token Service Installation and Configuration Guide (2.2) [PDF] documentation.

   1. Click **Add a new row to 'Protected Resource Mapping Table'**.
   2. In the **Auth Context** field, enter the authentication context that has to exist in the SAML assertion, such as `Password` or `MobileTwoFactorContract`.

      You can find the complete list of authentication contexts in the Authentication Context for the OASIS Security Assertion Markup Language (SAML) 2.0 [PDF] on oasis-open.org.
   3. (Optional) In the **Attribute Filter** field, enter an attribute that has to exist in the assertion, such as `${organization}='WidgetCo'`.

      > **Tip**
      >
      > You can use `AND` and `OR` operators to include multiple attributes or create simple rules. For example, `${organization}='WidgetCo' OR ${organization}='WidgetCoLtd'`.
   4. In the **Resource** field, enter the name of the resource that the user can access when the assertion meets the **Auth Context** and **Attribute Filters** conditions. For example, `/partner_application/partner_landing.html`.
   5. In the **Instance** field, enter the value of the `AgentName` parameter associated with the default CTS Agent Config Object. For example, `partner_site_agent`.

Copyright © 2025 Ping Identity Corporation

6. In the **Action** field, enter the action, such as `GET`, `POST`, or `PUT`.
7. In the **Action** column, click **Update**.
8. To add more attributes, repeat steps a-g.

4. On the **Instance Configuration** tab, configure the adapter instance by referring to CoreBlox SP Adapter settings reference. Click **Next**.
5. (Optional) On the **Actions** tab, if you set **Send Extended Attributes** to **OpenToken**, click **Download**, and then click **Export**. Save `agent-config.txt`. You can use this file to decode the OpenToken token that contains the extended attributes.
6. On the **Extended Contract** tab, add any attributes that you expect to retrieve other than the SAML subject. Click **Next**.
7. On the **Target App Info** tab, enter the basic information about your SP application. Click **Next**.
8. On the **Summary** tab, check and save your configuration.

   Choose from:

   - For PingFederate 10.1 or later: Click **Save**.
   - For PingFederate 10.0 or earlier: Click **Done**. On the **Manage SP Adapter Instances** tab, click **Save**.
9. Create an IdP connection using this CoreBlox SP Adapter instance. Learn more in the Service provider SSO configuration in the PingFederate documentation.

CoreBlox SP Adapter settings reference

Field descriptions for the CoreBlox SP Adapter configuration screen.

Standard fields

Field Name	Description
Validate CoreBlox Certificate Hostname	The hostname of the server certificate presented by the CoreBlox Token Service must match the hostname in the **CoreBlox URL** field. This checkbox is selected by default.
Client Certificate	The certificate that the adapter uses to authentication calls to the CoreBlox Token Service.
CoreBlox TokenType	The token type that the CoreBlox Token Service is configured to provide to the adapter. The default and only value is `SMSESSION`.
Cookie Name	The name of the cookie that contains the token used with the CoreBlox Token Service. This field is blank by default.

Copyright © 2025 Ping Identity Corporation

Field Name	Description
Cookie Domain	The domain name that the adapter uses when creating cookies. The browser compares this value to the domain of subsequent requests to determine whether the cookie should be submitted. If this field is blank, the adapter uses the domain name of the request. When sharing a cookie across subdomains, this value must be prefixed with a period (`.`). This field is blank by default.
Cookie Path	The path that the adapter uses when creating cookies. The browser compares this value to the path of subsequent requests to determine whether the cookie should be submitted. The default value is `/` .
Cookie Secure Flag	The adapter writes cookies with the Secure flag. The browser only submits Secure cookies on subsequent HTTPS requests. This checkbox is selected by default.
CoreBlox URL	The URL for the CoreBlox Token Service. This field is blank by default.
Error URL	When an error occurs in the adapter, PingFederate redirects the browser to this URL instead of to the default error page. This URL can contain query parameters. The URL has an `errorMessage` appended to it, which contains a brief description of the error that has occurred. This field is blank by default.
Logged-Out Cookie Value	The expected value of the cookie when the user has been signed off. The default value is `LOGGEDOFF` .
HTTP Only Flag	When selected, the adapter sets a flag for the cookie. The flag indicates that the cookie should be read only through http requests and disallows Javascript from accessing the cookie. This checkbox is selected by default.
Account Link URL	The URL to redirect the user to for account linking. This field is blank by default.

Advanced fields

Field Name	Description
Resource	The resource that is protected by the agent. This field is required if **Perform Authorize Request** is selected. This field is blank by default.
Instance	Refers to the name of the agent instance. This field is required if **Perform Authorize Request** is selected. This field is blank by default.

Copyright © 2025 Ping Identity Corporation

Field Name	Description
Action	The action to take when evaluating requests against the policy server. This field is required if **Perform Authorize Request** is selected. This field is blank by default.
PingFederate Base URL	The base URL for PingFederate, such as `https://pf_host:pf_port/`. The adapter uses this value to create the return URL **Cookie Provider URL** to create the return URL. Complete this field if you are using a cookie provider to enable single sign-on (SSO) across multiple domains. This field is blank by default.
Cookie Provider URL	The URL of the cookie provider. PingFederate redirects the request to this URL if the session cookie is in a separate domain. Complete this field if you're using a cookie provider to enable SSO across multiple domains. This field is blank by default.
Cookie Provider Target Parameter	The name of parameter that contains the PingFederate return URL in the redirect to the cookie provider. Complete this field if you're using a cookie provider to enable SSO across multiple domains. This field is blank by default.
Session Cookie Prefix	The prefix to remove when **Remove Session Cookie Prefix** is selected. Complete this field if you're using a cookie provider to enable SSO across multiple domains. When using a cookie provider to enable SSO across multiple domains, tokens are prepended with the security zone name. This field works with the **Remove Session Cookie Prefix** setting to remove the security name prefix and to allow the adapter to process the token. Enter the **SSOZoneName** from your Agent Configuration Object (ACO) surrounded by `-`. For the default security zone name, enter `-SM-`. This field is blank by default.
Remove Session Cookie Prefix	The adapter removes the **Session Cookie Prefix** from the beginning of the session cookie. Select this checkbox if you're using a cookie provider to enable SSO across multiple domains. This checkbox is cleared by default.
Disable Refresh Session Cookie	The adapter doesn't refresh session cookies when validating them. This allows other dependent applications to manage the session cookie. This checkbox is cleared by default.
Perform Authorize Request	The adapter makes an authorize request to the CoreBlox Token Service before accessing the protected resource. This checkbox is cleared by default.
Send Extended Attributes	The method that the adapter uses to send extended attributes.
Send Session Attributes	Includes the attribute contract as session attributes in the call to the CoreBlox Token Service. The default selection is `None`.

Copyright © 2025 Ping Identity Corporation

Field Name	Description
OpenToken Transfer Method	The method that the adapter uses to transport the OpenToken. The default selection is `Query Parameter`.
OpenToken Name	The name of the cookie or request attribute that contains the OpenToken. This name should be unique for each adapter instance.
OpenToken Password	The password that the adapter uses when encrypting extended attributes.
Send Session Attributes	When selected, the adapter includes the attribute contract as session attributes in the call to the CoreBlox Token Service. This checkbox is cleared by default.
Ignore Session Cookie	When selected, the adapter ignores any existing session cookie on the request and creates a new one. This checkbox is cleared by default.

Troubleshooting

Enabling debug logging

To help with troubleshooting or monitoring, you can turn on activity logging for the adapter.

About this task

These steps are optional. Learn more about logging in Enabling debug messages and console logging in the PingFederate documentation.

Steps

1. Open the `<pf_install>/pingfederate/server/default/conf/log4j2.xml` file for editing.
2. If you want to log activity for PingFederate and all adapters, do the following:
    1. Find the following section:

```
<AsyncRoot level="INFO" includeLocation="false">
        <!-- <AppenderRef ref="CONSOLE" /> -->
        <AppenderRef ref="FILE" />
</AsyncRoot>
```

    2. Change `INFO` to `DEBUG`.

Copyright © 2025 Ping Identity Corporation

```
<AsyncRoot level="DEBUG" includeLocation="false">
        <!-- <AppenderRef ref="CONSOLE" /> -->
        <AppenderRef ref="FILE" />
</AsyncRoot>
```

3. If you want to see the adapter activity in the console, remove the comment tags:

```
<AsyncRoot level="INFO" includeLocation="false">
        <AppenderRef ref="CONSOLE" />
        <AppenderRef ref="FILE" />
</AsyncRoot>
```

3. If you want to log activity just for the CoreBlox IdP Adapter, add the following line:

```
<Logger name="com.pingidentity.clientservices.product.coreblox" level="DEBUG"/>
```

4. Save the file.

Release notes

Changelog

The following is the change history for the CoreBlox Integration Kit.

CoreBlox Integration Kit 2.7.1 – October 2021

- Verified that the CoreBlox IdP Adapter is compliant with US Federal Information Processing Standards (FIPS). Learn more about usage in Integrating with Bouncy Castle FIPS provider in the PingFederate documentation.
- Updated internal components to better support certificate revocation checks in PingFederate 11 and later.

CoreBlox Integration Kit 2.7 – September 2020

- Fixed an issue that could cause a query string to be encoded incorrectly.
- Fixed an issue that could rarely cause a redirect loop.
- Added the **Session Cookie Prefix** and **Remove Session Cookie Prefix** settings to accommodate a change in the way the CoreBlox Token Service provides tokens.

CoreBlox Integration Kit 2.6.2 – June 2020

- Resolved an error that occurred when a user left the sign on page, then tried to open the sign on page again elsewhere.

Copyright © 2025 Ping Identity Corporation

CoreBlox Integration Kit 2.6.1 – November 2018

- Improved PingFederate node reliability when making REST calls to the the CoreBlox Token Service by changing to a bundled HTTP client.

CoreBlox Integration Kit 2.6 – October 2018

- Added an option to ignore session cookies from inbound requests to the PingFederate SP server. Refer to **Ignore Session Cookie** on the CoreBlox SP Adapter **Instance Configuration** screen.

CoreBlox Integration Kit 2.5 – April 2018

- Added 'Send Session Attributes' field to SP adapter to send attribute contract to the the CoreBlox Token Service (CTS) to include in the created session.

CoreBlox Integration Kit 2.4 – June 2017

- PingFederate 8.4 compatibility updates
- Optionally prevent an update to the session cookie upon log in
- User is redirected for authentication if session is invalid
- Session cookie is retained and set to the logged off value after SLO
- Session cookie isn’t modified on SLO error

CoreBlox Integration Kit 2.3 – September 2016

- Login URL field is made optional to support multiple CoreBlox environments
- Improved error logging in both IdP and SP adapters.

CoreBlox Integration Kit 2.2 – November 2015

- Added cookie provider functionality
- Authentication Context support added for the SP Adapter scenario

CoreBlox Integration Kit 2.1 – May 2015

- Added the capability in the CoreBlox SP adapter to download a configuration file for the OpenToken agent.
- Bug fixes.

CoreBlox Integration Kit 2.0 – April 2014

- Added additional configuration UI validation, notably for the added fields required for authorization and for the "Cookie Domain" in the SP adapter.
- Fixed Account Linking. Previously was attempting to use a 'userId-qualifier' as the attribute to link accounts. Now it’s set up to use the 'userId' attribute.

Copyright © 2025 Ping Identity Corporation

- The authentication context for the adapter is now functional. Anything that is entered in the field will appear in the SAML assertion.
- Both IdP and SP adapters can be configured to authorize a user before redirecting them to the target resource.
- The 'token' attribute returned in the JSON object from the CTS can now be added to the extended contract using the keyword 'token'.

CoreBlox Integration Kit 1.0.5 – March 2014

- Added authorization functionality, enabling users' authorization level to be scrutinized before accessing a protected resource.

CoreBlox Integration Kit 1.0 – November 2012

- CoreBlox Integration kit's debut.

Known issues and limitations

The following are known issues or limitations for the CoreBlox Integration Kit.

Known issues

There are no known issues.

Known limitations

- Cannot delete all characters of an attribute on the **Extended Contract** page during an "update" action.
- If you're running the CoreBlox Token Service (CTS) inside PingFederate and have deployed the PingID RADIUS PCV, you must use version 1.3 or later of the password credential validator.

Download manifest

The following files are included in the CoreBlox Integration Kit `.zip` archive.

- `Legal.pdf` : Copyright and license information
- `dist/pingfederate/server/default` : Contains the integration files
  - `deploy` : Contains the Java libraries
    - `coreblox-integration-kit-<version>.jar` : The CoreBlox IdP Adapter

CoreBlox Token Translator

The PingFederate CoreBlox Token Translator provides a token processor and token generator for use with the PingFederate's WS-Trust Security Token Service (STS).

Copyright © 2025 Ping Identity Corporation

The token processor allows an identity provider (IdP) STS to validate and authorize a CoreBlox session token from a Web Service Client (WSC) and then map user attributes into a SAML token to a Web Service Provider (WSP). The token generator allows a service provider (SP) STS to issue a CoreBlox session token for a WSP, including mapped attributes from an incoming SAML token.

Intended audience

This document is intended for PingFederate administrators.

If you need help during the setup process, learn more about available resources in:

- The following section of the PingFederate documentation:
  - WS-Trust STS configuration
- The following CoreBlox resources:
  - CoreBlox Token Service
  - CoreBlox Token Service Installation/Configuration [PDF] from CoreBlox

System requirements

- PingFederate 7.x or later
- A CTS installation, either local or remote

Setup

Installing the CoreBlox token translator

Steps

1. Stop the PingFederate server if it's running.
2. Remove any existing CoreBlox Token Translator files (`coreblox-token-translator-*.jar`) from the directory:

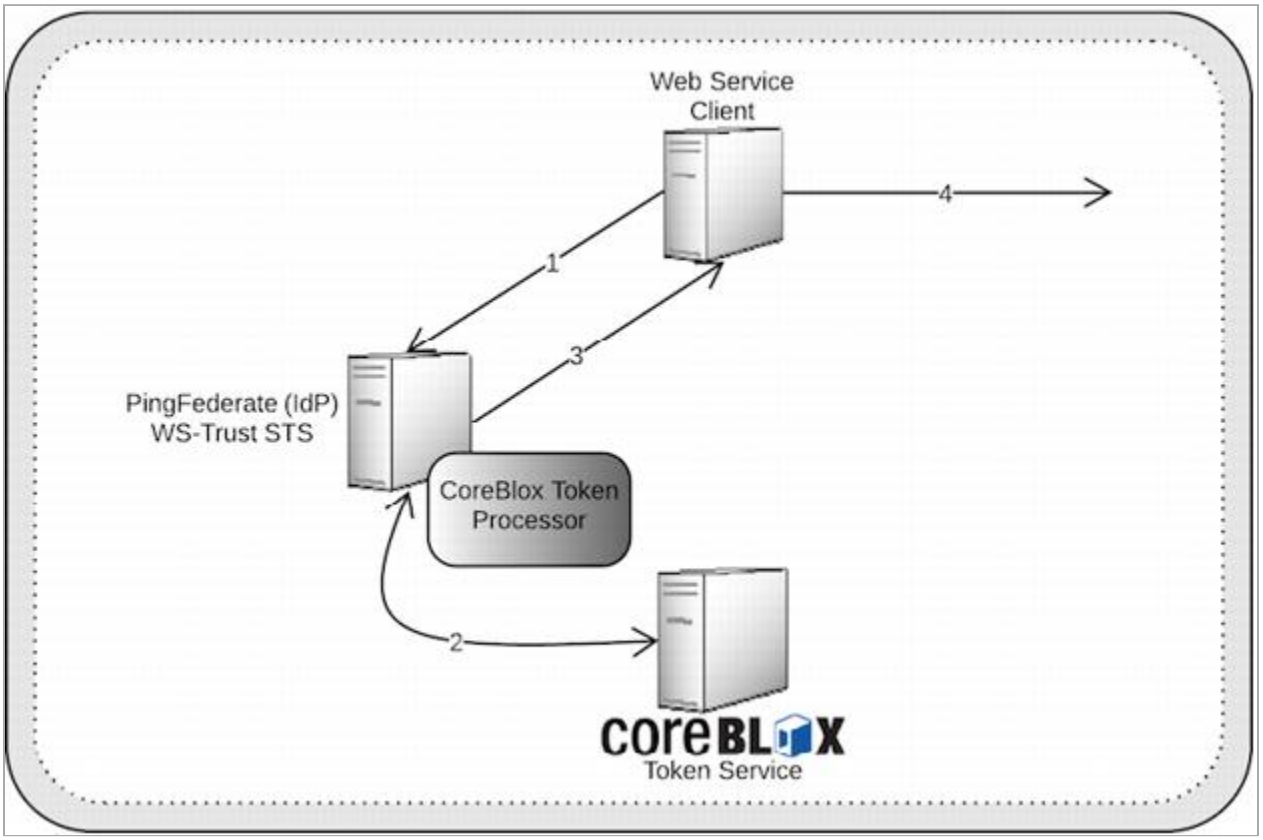
   `<PF_install>` \ `PingFederate\server\default\deploy`
3. Unzip the distribution file and copy and copy `CoreBlox-token-translator-1.2.jar` from the `/dist` directory to the PingFederate directory.

   `<PF_install>` \ `PingFederate\server\default\deploy`
4. Start or restart the PingFederate server.

Token processor (IdP) processing overview

The following figure illustrates how PingFederate and the Token Processor interacts with the CTS:

Copyright © 2025 Ping Identity Corporation



Processing Steps

1. A WSC sends a Request Security Token (RST) message containing a CoreBlox session token to the PingFederate STS IdP endpoint.
2. The CoreBlox Token Processor validates and authorizes the token from the WSC with the CTS and returns a valid token back to the Token Processor.
3. PingFederate STS embeds the mapped attributes in a SAML assertion wrapped in a Request Security Token Response (RSTR) back to the WSC.
4. The SAML assertion is sent to the WSP.
    - Configuring the IdP token processor

Configuring the IdP token processor

About this task

This section describes how to configure the CoreBlox Token Processor.

Steps

1. Sign on to the PingFederate administrative console and click **Token Processors** under **IdP Configuration** on the main menu.

Copyright © 2025 Ping Identity Corporation

> **Note**
>
> If you don't see Token Processors on the main menu, enable **WS-Trust** under **Server Settings** on the **Roles and Protocols** screen by selecting **WS-Trust** for the IdP role.

2. On the **Manage Token Processor Instances** screen, click **Create New Instance**.
3. On the **Type** screen, enter an **Instance Name** and **Instance ID**. The instance name is any name you choose for identifying this token processor instance.

> **Note**
>
> The Instance ID is used internally and cannot contain any spaces or non-alphanumeric characters. It must be uniquely named.

4. Select **CoreBlox Token Processor** as the **Type** and click **Next**.
5. Fill in the **Instance Configuration** screen as follows:

Field	Description
CoreBlox URL	The base URL for CTS requests.
Validate CoreBlox Certificate Hostname	If checked, the hostname of the server certificate presented by the CTS must match the hostname of the CoreBlox URL.
Client Certificate	The certificate used for authentication calls to the CTS.
CoreBlox Tokentype	The tokentype to be returned from the CTS. **Note** The only permissible and default value is `SMSESSION`.

6. **Optional:** Click **Show Advanced Fields** to specify the token processor's authorization configuration settings.

Field	Description
Perform Authorize Request	If selected, the token processor will make an authorize request to the CTS before accessing the protected resource. **Note** The following three fields are required for the adapter to make the authorize request.
Resource	The resource that is protected by the agent.
Instance	Refers to the name of the agent instance.
Action	The action to take when evaluating requests against the policy server.

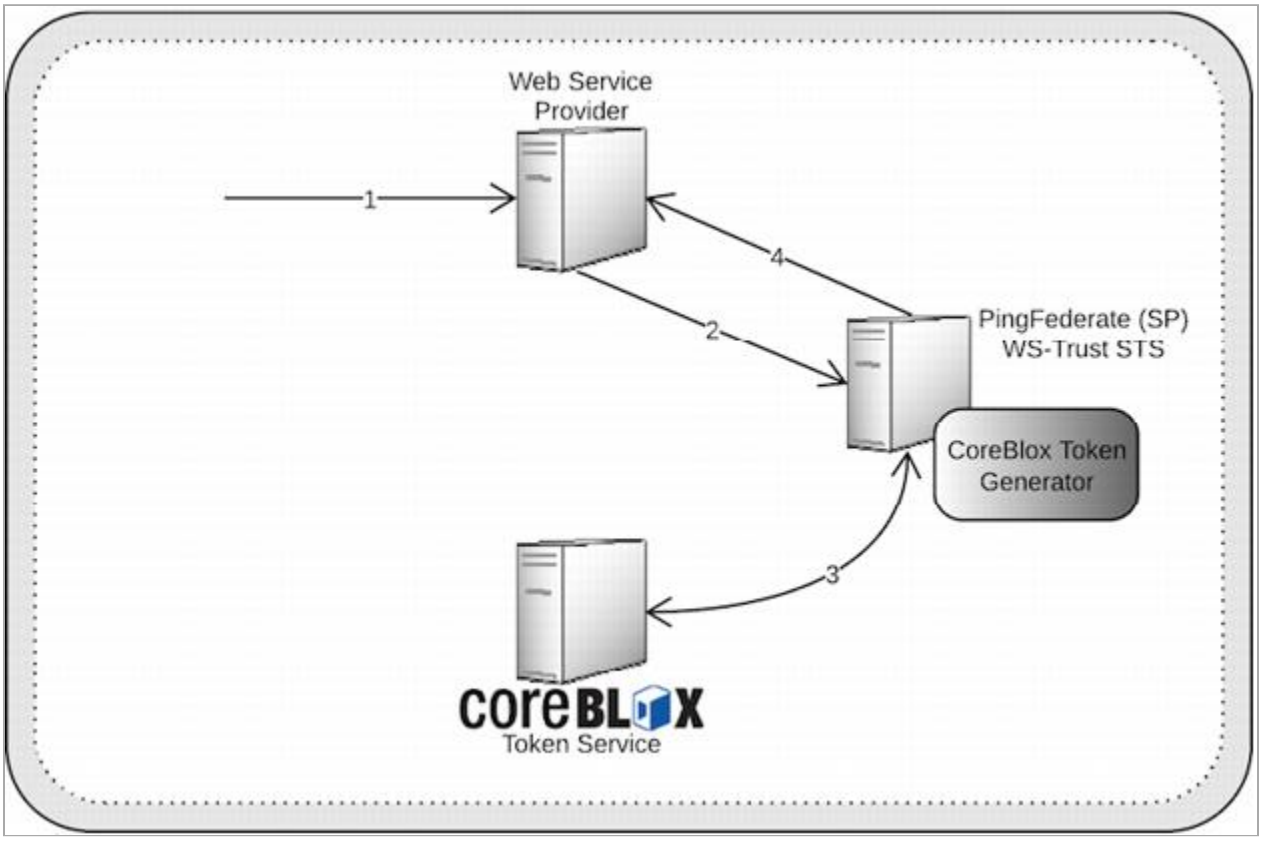
Copyright © 2025 Ping Identity Corporation

7. Click **Next**.
8. (Optional) On the **Extended Contract** screen, configure additional attributes for the adapter. Learn more about key concepts in the PingFederate Administrator's Reference Guide.
9. Click **Next**.
10. On the **Token Attributes** screen, select the **Pseudonym** checkbox for the userId attribute. You can select any extended attribute specified on the previous screen.

    Learn more about this screen in Setting Pseudonym Values and Masking in the PingFederate Administrator's Reference Guide.
11. Click **Next**.
12. On the **Summary** screen, verify that the information is correct and click **Done**.
13. On the **Manage Token Processor Instances** screen, click **Save** to complete the token generator configuration.

Token generator (SP) processing overview

The following figure illustrates how PingFederate and the Token Generator interact with the CTS:



Processing Steps

1. The WSP accepts an incoming SAML assertion.
2. The WSP makes a RST to PingFederate STS.

Copyright © 2025 Ping Identity Corporation

3. The Token Generator parses the attributes in the SAML and queries the CTS for a valid token.
4. The CoreBlox session token and mapped SAML attributes are embedded in a RSTR that is sent back to the WSP.

◦ Configuring the SP token generator

Configuring the SP token generator

About this task

This section describes how to configure the CoreBlox SP Token Generator.

Steps

1. Sign on to the PingFederate administrative console and, in the **SP Configuration** section, click **Token Generators**.
2. On the **Manage Token Generator Instances** screen, click **Create New Instance**.
3. On the **Type** screen, enter an **Instance Name** and **Instance ID**. The instance name is any name you choose for identifying this token generator instance.

> **Note**
>
> The Instance ID is used internally and cannot contain any spaces or non-alphanumeric characters. It must be uniquely named.

4. Select **CoreBlox Token Generator X.0** as the **Type** and click **Next**.
5. On the **SP Adapter** screen, click **Add a new row to 'Protected Resource Mapping Table'** and provide the following information into the table:

Field	Description
Auth Context	The attribute containing authentication context. The name of this attribute needs to be specified below as Auth Context Attribute Name.
Attribute Filter	The names and values of attributes that the assertion must contain for this Protected Resource.
Protected Resource	The protected resource to be accessed if the Authentication Context and Attribute Filters in the assertion match the provided values.

6. Click **Update**in the **Action** column. Repeat this step as needed.
7. Provide entries on the **Instance Configuration** screen, as described in the table below:

Field	Description
CoreBlox URL	The URL for the CTS.

Copyright © 2025 Ping Identity Corporation

Field	Description
Validate CoreBlox Certificate Hostname	If selected, the hostname of the server certificate presented by the CTS must match the hostname of the CoreBlox URL.
Client Certificate	The certificate used for authentication calls to the CTS.
CoreBlox Tokentype	The tokentype to be returned from the CTS. **Note** The only permissible value is `SMSESSION`.
Base64 Decode The Token	If selected, the token returned from the CTS will be base64-decoded. This prevents the token from being encoded twice.

8. (Optional) Click **Show Advanced Fields** to specify the token processor's authorization configuration settings.

Field	Description
Context Attribute Name	(Optional) Attribute Name containing Authentication Context used for mapping protected resource. This value is required if **Perform Authorize Request** is selected and the Protected Resource Mapping Table is not empty.
Perform Authorize Request	If selected, the adapter will make an authorize request to the CTS before accessing the protected resource. **Note** The following three fields, **Resource**, **Instance**, and **Action** are required for the adapter to make the authorize request.
Resource	The resource that is protected by the agent.
Instance	Refers to the name of the agent instance.
Action	The action to take when evaluating requests against the policy server.

9. Click **Next**.
10. **Optional:** On the **Extended Contract** screen for a connection, configure additional attributes for the Token Generator. Any attributes configured in this step are added to the request header.
11. Click **Next**.
12. On the **Summary** screen, verify that the information is correct and click **Done**.
13. On the **Manage Token Generator Instances** screen, click **Save**.

Copyright © 2025 Ping Identity Corporation

Deployment notes

- When configuring STS clients with PingFederate versions 6 or later, a common issue with libraries parsing XML has been documented in Invalid XML Errors While Loading STS Client Libraries with recommended steps to resolve it.

Release notes

Changelog

CoreBlox Token Translator 1.3 - August 2017 (current release)

- PingFederate 8.4 compatibility updates.

CoreBlox Token Translator 1.2 - August 2016

- Added configuration option to Base64 encode the CoreBlox token.

CoreBlox Token Translator 1.1 – May 2016

- Authentication Context support added for the Token Generator.

CoreBlox Token Translator 1.0 – April 2014

- CoreBlox Token Translator's debut.

Download manifest

The distribution `.zip` archive for the CoreBlox Integration Kit contains the following:

- `ReadMeFirst.pdf` : Contains links to this online documentation
- `/dist` : Contains libraries needed to run the adapter:
    - `CoreBlox-token-translator-1.2.jar` : CoreBlox Token Translator JAR file

Copyright © 2025 Ping Identity Corporation

Coupa Provisioner

The PingFederate Provisioner for Coupa enables an enterprise to provision its users to Coupa. This Coupa Provisioner includes a quick connection template to easily set up a single sign-on (SSO) connection requiring Coupa provisioning.

Features

- Outbound user provisioning
- Browser-based SAML SSO

Intended audience

This document is intended for PingFederate administrators.

System requirements

- PingFederate 7.2.1 or later
- Common Provisioning Layer (CPL) 2.0.1 or later (`prov-cpl-2.0.1.jar`)

Setup

Obtaining required information

Before you can configure this connector, you must complete the following steps:

- Obtaining your Coupa subdomain
- Obtaining your Coupa API key
- Obtaining your Coupa (SP) metadata file
- Synchronizing existing Coupa users

> **Tip**
>
> Some of these steps result in information that you'll need for subsequent tasks. You should copy this information to a secure location.

Obtaining your Coupa subdomain

This connector requires your Coupa subdomain to access your Coupa account for single sign-on (SSO) and outbound provisioning.

Your Coupa subdomain is the subdomain portion of the URL you visit to access your Coupa account, for example, https://*YourSubDomain*.Coupa.com.

Copyright © 2025 Ping Identity Corporation

Obtaining your Coupa API key

The Coupa Connector's outbound provisioning functionality is built using Coupa's REST API, which requires an API key for authentication.

About this task

To create and obtain the Coupa API key:

Steps

1. Sign on to your Coupa account as an administrative user.
2. Click the **Setup** tab.
3. In the **Company Setup** section, select the **API Keys** checkbox.
4. Click the **Create** button.
5. Complete the form and click the **Create** button.

Obtaining your Coupa (SP) metadata file

This connector's quick-connection template uses a metadata XML file to assist in configuring many settings in the service provider (SP) connection.

About this task

Before configuring your SP connection, you must download the `idp-metadata.xml` from Coupa.

> **Note**
>
> To enable single sign-on (SSO), you must follow the steps outlined in the Completing setup of SAML SSO to Coupa section of this guide after you have completed the Creating a connection section.

Steps

1. Sign on to Coupa as an administrative user.
2. Click the **Setup** tab.
3. In the **Company Setup** section, select the **Security controls** checkbox.
4. Click the **Coupa SP metadata** link labeled **Download and import SP metadata**.

Copyright © 2025 Ping Identity Corporation

Synchronizing existing Coupa users

> **Important**
>
> If your Coupa account already has users you want to provision with the Coupa Connector, this is possible by completing the following steps.

Ensure that the value mapped to the `email` attribute, when configuring the connector matches the existing Coupa users email address exactly as it appears in Coupa.

For example, on the **Attribute Mapping** tab, the user's **email** attribute on Coupa is mapped to the user's `mail` attribute in your LDAP. This synchronizes a user that already exists on Coupa with an `email` address in Coupa of john.smith@mydomain.com. In this case, the user's **mail** attribute in LDAP would also have to be john.smith@mydomain.com.

When the Coupa connector provisions for the first time, this address is used to synchronize the user in your LDAP datastore with the user in Coupa.

Enabling provisioning and single sign-on (SSO) in PingFederate

To use PingFederate for provisioning and single sign-on, configure an external datastore and set a SAML entity ID.

About this task

Your external datastore acts as the source of data for provisioning. PingFederate also uses an internal datastore to store the state of synchronization between the source datastore and the target datastore.

You can find more information in Datastores and Configuring outbound provisioning settings in the PingFederate documentation.

Enabling provisioning and single sign-on (SSO) in PingFederate 10.1 or later

Steps

1. In the PingFederate administrative console, configure the datastore for PingFederate to use as the source of user data.

   Learn more about Datastores in the PingFederate documentation.

2. Go to **System > Server > Protocol Settings > Federation Info**.
3. In the **SAML 2.0 Entity ID** field, enter a name for PingFederate to use when identifying itself to SAML partners.
4. On the **Outbound Provisioning** tab, in the **Provisioner Datastore** list, select the internal database that will store the synchronization state. Click **Save**.

   Learn more in Configuring outbound provisioning settings in the PingFederate documentation.

Enabling provisioning and single sign-on (SSO) in PingFederate 10.0 or earlier

Steps

1. In the PingFederate administrative console, configure the datastore for PingFederate to use as the source of user data.

Copyright © 2025 Ping Identity Corporation

Learn more in Datastores in the PingFederate documentation.

2. Go to **System > Protocol Settings > Roles & Protocols**.
3. Select the **Enable Identity Provider IdP Role and Support the Following** checkbox.
4. Select the **SAML 2.0** and **Outbound Provisioning** checkboxes. Click **Next**.
5. Click the **Federation Info** tab.
6. In the **SAML 2.0 Entity ID** field, enter a name for PingFederate to use to identify itself to SAML partners.
7. On the **Outbound Provisioning** tab, in the **Provisioner Datastore** list, select the internal database that will store the synchronization state. Click **Save**.

   Learn more in Configuring outbound provisioning settings in the PingFederate documentation.

Creating a connection

To allow PingFederate to act as an identity provider (IdP) and manage users in Coupa, create a service provider (SP) connection.

Steps

1. In the PingFederate administrator console, create a new SP connection:

   Choose from:

   - For PingFederate 10.1 or later: Go to **Applications > Integration > SP Connections**. Click **Create Connection**.
   - For PingFederate 10.0 or earlier: Go to **Identity Provider > SP Connections**. Click **Create Connection**.

2. Configure the basic connection details with the Coupa quick connection template:
   1. On the **Connection Template** tab, click **Use a template for this connection**.
   2. In the **Connection Template** list, select **Coupa Provisioner**.
   3. On the **Metadata File** row, upload the `idp-metadata.xml` file that you saved in Obtaining your Coupa (SP) metadata file. Click **Next**.
   4. On the **Connection Type** tab select the **Browser SSO Profiles** and **Outbound Provisioning** checkboxes. Click **Next**.
   5. On the **Connection Options** tab, click **Next**.
   6. On the **General Info** tab, in the **Connection Name** field, enter a name of your choosing. Click **Next**.
3. On the **Browser SSO** tab, configure your assertion creation settings and customize the defaults set by the metadata file.

   You can find more information in Configuring IdP Browser SSO in the PingFederate documentation.

4. On the **Credentials** tab, configure the connection credentials as shown in Configuring credentials in the PingFederate documentation. Click **Next**.
5. On the **Outbound Provisioning** tab, configure provisioning with the following details.

Copyright © 2025 Ping Identity Corporation

Learn more in Configuring outbound provisioning in the PingFederate documentation.

1. On the **Target** tab, enter the API key and subdomain you noted in Obtaining required information.
2. (Optional) In the **Provisioning Options** section, customize the provisioning connector behavior. Click **Next**.
3. Go to **Manage Channels > Attribute Mapping** tab and at the bottom of the attribute list, click **Refresh Fields** to get fields and specifications from your Coupa site.
4. Complete the attribute mappings by referring to the Attribute index.

   For help, reference Managing channels in the PingFederate documentation.

6. On the **Activation and Summary** tab, above the **Summary** section, click the toggle to turn on the connection. Click **Save**.

Completing setup of SAML SSO to Coupa

The following steps outline how to complete the configuration of your Coupa account and service provider (SP) connection to enable identity provider (IdP) or SP-initiated SAML single sign-on (SSO).

> **Important**
>
> Coupa supports either SP or IdP-initiated SAML SSO on a given Coupa account, but not both.

> **Tip**
>
> You can find more information in the Coupa documentation.

Obtaining the EntityDescriptor's ID from your SP connection

Steps

1. Export your service provider (SP) connection's `metadata.xml` file:
   1. In the PingFederate admin console, in the **SP Connection** section, click **Manage All SP**.
   2. Click **Export Metadata** for your Coupa SP connection.
   3. Select the signing certificate you used in that SP connection and click **Next**.
   4. Click the **Export** button to download the metadata file.
   5. Click **Done** and then click **Save**.
2. After you have the metadata from your SP connection, view the metadata file that you exported and record the ID value of the `EntityDescriptor` tag at the top of the file.

   Example:

   The following SP connection metadata file example shows the `EntityDescriptor` ID value at the top of the file, in this case `"R6HV.w-pHhv_pacAMTH7D-nFe3W"`.

Copyright © 2025 Ping Identity Corporation



idp-metadata.xml file">

Preparing the `idp-metadata.xml` file

Steps

1. Edit the `idp-metadata.xml` file that is packaged with the Coupa connector in a text editor.
2. Replace `<YOUR_ENTITY_DESCRIPTOR_ID>` with the ID you obtained in Obtaining the EntityDescriptor's ID from your SP connection.
3. Replace `<YOUR_PF_ENTITY_ID>` with the PingFederate Entity ID you obtained in Enabling provisioning and single sign-on (SSO) in PingFederate.
4. Replace `<YOUR_PF_BASE_URL>` with the PingFederate base URL you obtained in Enabling provisioning and single sign-on (SSO) in PingFederate.
5. Save your changes.

Copyright © 2025 Ping Identity Corporation

Configuring your SP connection for IdP-initiated SSO

About this task

> **Note**
>
> This task is only required if you want to enable identity provider (IdP)-initiated single sign-in (SSO).

To successfully configure your SP connection for IdP-initiated SSO, you must add a `RelayState` parameter to your SP connection's Assertion Consumer Service (ACS) URL.

Steps

1. In the PingFederate admin console, click your Coupa SP connection to edit it.
2. In the **Protocol Settings** section, click the **Assertion Consumer Service URL** link.
3. Click **Edit** for the endpoint URL you're using in your SP connection.
4. Enter the following endpoint URL, which contains the `RelayState` parameter:

   ```
   /sp/ACS.saml2?RelayState=https://YourSubDomain.coupacloud.com/sessions/saml_post
   ```

   The *YourSubDomain* variable is your Coupa subdomain, which you obtained in Obtaining required information.
5. Click **Update**.
6. Click **Done**, **Done** and **Save**.

Configuring Coupa for SAML SSO

About this task

> **Note**
>
> For assistance configuring Coupa for SAML single sign-on (SSO), Coupa recommends you send your prepared `idp-metadata.xml` to their support team or your Coupa Implementation Administrator along with:
>
> - Login page URL
> - Logout page URL
> - Timeout URL
> - A test user that exists in your identity provider (IdP)

Steps

1. Sign on to your Coupa account as an administrative user.
2. Click the **Setup** tab.

Copyright © 2025 Ping Identity Corporation

3. In the **Company Setup** section, click **Security controls**.
4. Import the `idp-metadata.xml` that you prepared above into the **Upload IdP metadata** field.

> **Note**
>
> During development and testing of the Coupa connector, Ping Identity was unable to upload the `idp-metadata.xml` into Coupa without receiving errors. Coupa Support assisted with this configuration.

5. Select the **Advanced Options** checkbox.
6. In the **Login page URL** field, enter the Login page URL:

   Choose from:

   - SP-initiated SSO: Enter `https://prdsso40.cloudcoupa.com/sp/startSSO.ping?PartnerIdpId=YOUR_PF_ENTITY_ID&TARGET=https://YOUR_COUPA_SUBDOMAIN.cloudcoupa.com/sessions/saml_post`
   - IdP-Initiated SSO: Points to the login page of your IdP.

7. In the **Logout page URL** field, enter the Logout page URL.

   The **Logout page URL** is set to where your users should be directed when they sign off of Coupa.

8. In the **Timeout URL** field, enter the Timeout URL.

   The **Timeout URL** is set to where your users should be directed if their session times out before they sign on.

9. Click **Save**.

Attribute index

The following table lists the user attributes that you can map for provisioning.

Attribute	Description
`login`	Login name for the user. **Note** This value must be unique.
`email`	User's email address. **Note** This value must be unique.
`firstName`	User's first name.
`lastName`	User's last name.

Copyright © 2025 Ping Identity Corporation

Attribute	Description
`employee-number`	User's Employee Number. **Note** This value must be unique.
`authentication-method`	The method of authentication for the user. Acceptable values include `coupa-credentials` (default), `saml`, and `ldap`. When setting a user's authentication-method to `saml` or `ldap`, you must configure the security settings of your Coupa account.
`role`	The names of roles the user is a member of. If you don't set a role, users will be assigned to the `User` role by default. The roles must exist already in Coupa before you can add a user to them. **Important** User roles can't be cleared directly by the Coupa Connector. However, you can set the role to `User`, the default user role.
`password`	The User's Coupa password. This value is ignored when `generate-password-and-notify` is set to `true`.
`generate-password-and-notify`	When creating or updating users with this field set to `true`, an email is sent to the `email` set for the user allowing the user to reset their own password. Acceptable values are `true` and `false`. If you want to set the password for a user using the `password` attribute, `generate-password-and-notify` must be set to `false`.
`SSO-identifier`	User's Single Sign-on ID (SSO ID). **Note** This value must be unique.
`phone-mobile`	User's mobile phone number, such as `1 222-333-4444 ext. 12345`. **Note** Coupa only accepts certain formats for phone numbers. If you have difficulty correctly formatting numbers, you can reach Coupa support: • Online at https://support.coupa.com. • By email at support@coupa.com. • Through your Coupa Implementation Administrator.

Copyright © 2025 Ping Identity Corporation

Attribute	Description
`phone-work`	User's work phone number, such as `1 222-333-4444 ext. 12345`. **Note** Coupa only accepts certain formats for phone numbers. If you have difficulty correctly formatting numbers, you can reach Coupa support: • Online at https://support.coupa.com. • By email at support@coupa.com. • Through your Coupa Implementation Administrator.
`manager`	The `login` attribute value of the user's manager. The manager must exist already on the Coupa account before it can be added as a manager of another user.
`Requisition Approval Limit Amount`	User's approval limit, such as `1000.00`.
`Requisition Approval Limit Currency`	User's requisition approval limit currency code, such as `CAD`. The list of valid currency codes available for your use can be found on your Coupa accounts currencies list at https://*Your_Subdomain*.coupacloud.com/currencies.
`Invoice Approval Limit Amount`	User's approval limit, such as `1000.00`.
`Invoice Approval Limit Currency`	User's invoice approval limit currency code, such as `CAD`. The list of valid currency codes available for your use can be found on your Coupa accounts currencies list at https://*Your_Subdomain*.coupacloud.com/currencies.
`Content-Groups`	The names of content groups the user is a member of. The content groups must exist already in Coupa before you can add a user to them. **Important** Content groups can't be cleared directly by the Coupa Connector. However, you can remove a user from all content groups through the Coupa admin console.
`Requisition Self-Approval Limit Amount`	User's self-Approval limit, such as `1000.00`.
`Requisition Self-Approval Limit Currency Code`	User's requisition self-approval limit currency code, such as `CAD`. The list of valid currency codes available for your use can be found on your Coupa accounts currencies list at https://*Your_Subdomain*.coupacloud.com/currencies.

Copyright © 2025 Ping Identity Corporation

Attribute	Description
`Approval Groups`	The names of approval groups the user is a member of. The approval groups must exist already in Coupa before you can add a user to them. **Important** Approval groups can't be cleared directly by the Coupa Connector. However, you can remove a user from all approval groups through the Coupa admin console.
`Default Currency`	User's Default Currency Code, such as `CAD`. The list of valid currency codes available for your use can be found on your Coupa accounts currencies list at https://*Your_Subdomain*.coupacloud.com/currencies.
`Departments`	The names of departments the user is a member of. The departments must exist already in Coupa before you can add a user to them.

Release notes

Changelog

The following changes have been implemented for the Coupa Connector:

Coupa Connector 1.0 — February 2015 (Current Release)

- Initial release.
- Added support for user provisioning.
- Added support for browser-based single sign-on (SSO).

Known issues and limitations

The following are known issues and limitations of the Coupa Connector integration:

- Because of a limitation in PingFederate 8.1 and earlier versions, when configuring two service provider (SP) connections with the same provisioner, the second connection built might be pre-populated with the channel from the first connection. To avoid conflicts, delete this pre-populated channel and create a unique channel for each connection.
- Single logout (SLO) is not supported.
- Group (department) provisioning is not supported.
- User attributes cannot be cleared when set.
- Failed requests to update a user might remove the user from their roles, approval groups, content groups or departments until the update request is successfully retried.
- Managers must exist in Coupa before they can be added as the manager of another user. Not doing so results in a failure.

Copyright © 2025 Ping Identity Corporation

- Values set for the `phone-mobile` or `phone-work` attributes must adhere to how the service provider validates phone numbers. Not doing so results in a failure.
- Currency values are expected as a number with two decimals, for example `1000.00`. Not providing the value in the correct format might result in the value being stored as something different than what you sent, for example, `1,000.00` is stored as `1.00`.

Download manifest

The distribution `.zip` archive for the connector contains the following files:

- `ReadMeFirst.pdf`: Contains links to this online documentation
- `idp-metadata.xml`: The metadata used for browser single sign-on (SSO)
- `/legal`:
  - `Legal.pdf`: Copyright and license information
- `/dist`: Contains libraries needed for the connector:
  - `pf-coupa-quickconnection-1.0.jar`: PingFederate Coupa Connector
  - `prov-cpl-2.0.1.jar`: PingFederate Common Provisioning Layer

Copyright © 2025 Ping Identity Corporation

CrowdStrike Integration Kit

When paired with the Google Chrome Enterprise Integration Kit in PingFederate, the CrowdStrike Integration Kit enables organizations to balance security and user experience.

Components

CrowdStrike IdP Adapter

When a user signs on through PingFederate, the adapter provides security posture for use in authentication policy decisions based on CrowdStrike signals received within the authentication flow. The CrowdStrike signals come from the Google Chrome Enterprise Device Trust IdP Adapter.

Intended audience

This document is intended for PingFederate administrators.

If you need help during the setup process, see the following resources:

- The following sections of the PingFederate documentation:
  - Managing IdP adapters
  - Authentication Policies

> **Note**
>
> To sign up for the CrowdStrike Falcon platform, contact CrowdStrike.

System requirements

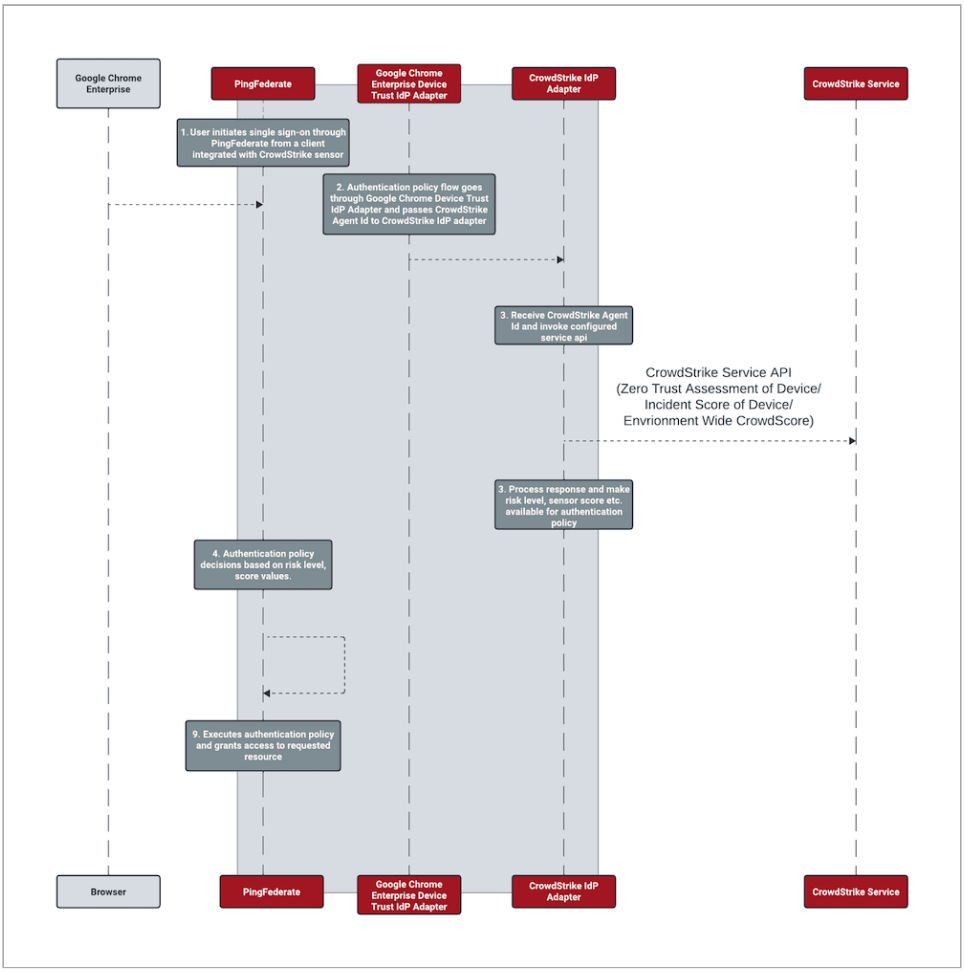
- PingFederate 10.3 or later
- To allow PingFederate to make outbound HTTPS connections, you might need to allow one of the following host names in your firewall:
  - https://api.crowdstrike.com/
  - https://api.us-2.crowdstrike.com/
  - https://api.eu-1.crowdstrike.com
  - https://api.laggar.gcw.crowdstrike.com

> **Note**
>
> The host name to allow depends on the CrowdStrike API region configuration.

Overview of the SSO flow

The following figure illustrates an example SSO process flow.

Copyright © 2025 Ping Identity Corporation



1. A user initiates the sign-on process by requesting access to a protected resource.
2. The Google Chrome Enterprise Device Trust IdP Adapter determines if the incoming request is from the Google Chrome Enterprise browser. If so, the authentication request goes through the adapter flow, which fulfills core contract attributes, such as the **CrowdStrike Agent ID**.
3. The **CrowdStrike Agent ID** is made available to the CrowdStrike IdP Adapter.
4. The adapter invokes the configured CrowdStrike Service API for the agent ID that it received.
5. If the CrowdStrike incident returns with a score, the risk level value is fulfilled based on the threshold level scores configured in the adapter. Learn more in CrowdStrike IdP Adapter settings reference.
6. The authentication policy can use the risk level value in subsequent decision making.

Setup

Deploying the integration files

To get started with the integration, deploy the CrowdStrike Integration Kit files to your PingFederate directory.

Copyright © 2025 Ping Identity Corporation

Before you begin

1. Contact CrowdStrike to sign up for the CrowdStrike Falcon platform.
2. Configure the Google Chrome Enterprise Device Trust IdP Adapter.

> **Important**
>
> As noted in Using CrowdStrike Risk Level, you must configure the Google Chrome Enterprise Device Trust IdP adapter or configure some other way to have the **CrowdStrike Agent ID** value sent to the CrowdStrike IdP Adapter for the CrowdStrike adapter flow to work.

Steps

1. Download the CrowdStrike Integration Kit `.zip` archive from the Ping Identity Integration Directory.
2. Stop PingFederate.
3. Extract the `.zip` archive and merge the contents of the `dist` directory with your `<pf_install>/pingfederate/server/default/deploy` directory.
4. Start PingFederate.
5. If you operate PingFederate in a cluster, repeat steps 2-4 for each engine node.

Next steps

Configure an adapter instance.

Configuring an adapter instance

Configure the CrowdStrike IdP Adapter to determine how PingFederate communicates with CrowdStrike.

Before you begin

Deploy the CrowdStrike Integration Kit files to your PingFederate directory.

Steps

1. In the PingFederate administrative console, go to **Authentication > Integration > IdP Adapters** and click **Create New Instance**.
2. On the **Type** tab, set the basic adapter instance attributes:
    1. In the **Instance Name** field, enter a name for the adapter instance.
    2. In the **Instance ID** field, enter a unique identifier for the adapter instance.
    3. From the **Type** list, select **CrowdStrike IdP Adapter**. Click **Next**.
3. On the **IdP Adapter** tab, in the **CrowdStrike Service API Response Mappings** section, map the attributes from the CrowdStrike Service API response to the attribute contract:
    1. Click **Add a new row to 'CrowdStrike Service API Response Mappings'**.
    2. In the **Local Attribute** field, enter a name for an attribute.

Copyright © 2025 Ping Identity Corporation

3. In the **CrowdStrike Service API Response Mapping** field, enter the JSON Pointer syntax for the source Google Verified Access API attributes, as shown in JSON Pointer syntax reference.

   Example:

   For example, the JSON pointer `/resource/0/system_serial_number` will return the system serial number.

4. In the **Action** column, click **Update**.
5. To add more attributes, repeat steps a-d.

   Result:

   These attributes are now available in your PingFederate authentication policy.

4. On the **IdP Adapter** tab, configure the adapter instance by referring to CrowdStrike IdP Adapter settings reference. Click **Next**.
5. On the **Actions** tab, test your connection to the CrowdStrike service by fetching the API access token. Resolve any issues that are reported, and then click **Next**.
6. On the **Extended Contract** tab, add any attributes that you included in the **CrowdStrike Service API Response Mappings** section of the **IdP Adapter** tab. Click **Next**.
7. On the **Adapter Attributes** tab, set pseudonym and masking options as shown in Set pseudonym and masking options in the PingFederate documentation. Click **Next**.
8. On the **Adapter Contract Mapping** tab, configure the contract fulfillment details for the adapter as shown in Define the IdP adapter contract in the PingFederate documentation. Click **Next**.
9. On the **Summary** tab, review your configuration. Click **Save**.

Next steps

Review Using CrowdStrike Risk Level.

JSON Pointer syntax reference

JavaScript Object Notation (JSON) Pointer defines a syntax for identifying a specific value within a JSON payload. Using the sample payload and JSON Pointer examples below, identify the attributes that you want to use to populate your attribute contract.

For a complete technical description of JSON Pointer syntax, see JavaScript Object Notation (JSON) Pointer on ietf.org.

Copyright © 2025 Ping Identity Corporation

Example CrowdStrike Incident service API response JSON payload

Copyright © 2025 Ping Identity Corporation

```
{
   "meta": {
      "query_time": 0.004738113,
       "powered_by": "incident-api",
       "trace_id": "b495ee9e-43e6-48bd-b0ac-19178ae304b1"
   },
   "resources": [
      {
         "incident_id": "inc:8389af03dfa04bac876ce5baf4a2dbbc:3d37a9f8c19b49b18bda301114c0861f",
         "incident_type": 1,
         "cid": "d1438ce1781d470aa639bb3374848448",
         "host_ids": [
             "8389af03dfa04bac876ce5baf4a2dbbc"
         ],
         "hosts": [
            {
               "device_id": "8389af03dfa04bac876ce5baf4a2dbbc",
               "cid": "d1438ce1781d470aa639bb3374848448",
               "agent_load_flags": "0",
               "agent_local_time": "2023-12-22T23:41:46.652Z",
               "agent_version": "6.42.15610.0",
               "bios_manufacturer": "Amazon EC2",
               "bios_version": "1.0",
               "config_id_base": "65994763",
               "config_id_build": "15610",
               "config_id_platform": "3",
               "external_ip": "34.219.249.45",
               "hostname": "EC2AMAZ-8FF6F2B",
               "first_seen": "2023-12-22T23:31:18Z",
               "last_login_timestamp": "2023-12-22T23:33:58Z",
               "last_login_user": "Administrator",
               "last_seen": "2024-01-17T22:19:59Z",
               "local_ip": "10.101.24.96",
               "mac_address": "06-66-9c-b8-bf-89",
               "machine_domain": "pfikteam.ping-eng.com",
               "major_version": "10",
               "minor_version": "0",
               "os_version": "Windows Server 2019",
               "ou": [
                   "Domain Controllers"
               ],
               "platform_id": "0",
               "platform_name": "Windows",
               "product_type": "2",
               "product_type_desc": "Domain Controller",
               "site_name": "Default-First-Site-Name",
               "status": "normal",
               "system_manufacturer": "Amazon EC2",
               "system_product_name": "t3.xlarge",
               "modified_timestamp": "2024-01-17T22:21:46Z",
               "instance_id": "i-00b24ead0c2e2bf27",
               "service_provider": "AWS_EC2_V2",
               "service_provider_account_id": "728729496554"
            }
         ],
         "created": "2024-01-17T22:38:30Z",
         "start": "2024-01-17T22:38:30Z",
         "end": "2024-01-17T22:39:04Z",
         "state": "open",
         "email_state": "START",
```


Copyright © 2025 Ping Identity Corporation

```
            "status": 20,
            "tactics": [
               "Falcon Overwatch"
            ],
            "techniques": [
            "Malicious Activity"
            ],
            "objectives": [
            "Falcon Detection Method"
            ],
            "modified_timestamp": "2024-01-17T22:39:09Z",
            "fine_score": 13
         }
      ],
      "errors": []
   }
```


JSON Pointer syntax

Description	JSON Pointer	Example value
Score of first incident detected	`/resources/0/final_sscore`	`13`
Host name of first incident	`/resources/0/hosts/0/hostname`	`EC2AMAZ-8FF6F2B`

> **Note**
>
> To populate an attribute with the entire JSON response, leave the **CrowdStrike Service API Response Mappings** field blank.

CrowdStrike IdP Adapter settings reference

Field descriptions for the CrowdStrike IdP Adapter configuration screen.

Standard fields

Field	Description
CrowdStrike Region	The CrowdStrike API region and API base URL to use.
Client ID	The client ID generated by CrowdStrike for access to the API.
Client Secret	The client secret generated by CrowdStrike for access to the API.

Copyright © 2025 Ping Identity Corporation

Field	Description
CrowdStrike Service	The CrowdStrike service to use. The CrowdStrike adapter will only call one of the following CrowdStrike services. Supported services are: • **Incident Score of Device** • **Zero Trust Assessment of Device** • **Environment Wide CrowdScore**
CrowdStrike Agent ID Attribute	The incoming chained attribute that provides the value of the **CrowdStrike Agent ID**. The default attribute name that the adapter looks for is `crowdStrikeAgentId`.

Advanced fields

Field	Description
Crowd Score Service Records Limit	The maximum number of score records to return when the **CrowdStrike Service** is **Environment Wide CrowdScore**. The default value is `20`.
High Score Threshold	If the CrowdStrike score is higher than this value, the adapter returns a `High` result.
Medium Score Threshold	If the CrowdStrike score is higher than this value but lower than or equal to the **High Score Threshold**, the adapter returns a `Medium` result.
Low Score Threshold	If the CrowdStrike score is higher than this value but lower than or equal to the **Medium Score Threshold**, the adapter returns a `Low` result.
API Request Timeout	The amount of time in milliseconds thatPingFederateallows when establishing a connection with the Google Verified Access API or waiting for a response to a request. A value of `0` disables the timeout. The default value is `5000`.
Proxy Settings	Defines proxy settings for outbound HTTP requests. The default value is **System Defaults**.
Custom Proxy Host	The proxy server host name to use when **Proxy Settings** is set to **Custom**. This field is blank by default.

Copyright © 2025 Ping Identity Corporation

Field	Description
Custom Proxy Port	The proxy server port to use when **Proxy Settings** is set to **Custom**. This field is blank by default.

Using CrowdStrike Risk Level

Because the CrowdStrike IdP Adapter relies on the **CrowdStrike Agent ID** that is automatically detected and returned by the Google Chrome Enterprise Device Trust IdP Adapter, the CrowdStrike IdP Adapter is designed to work with the Google Chrome Enterprise Device Trust IdP Adapter flow.

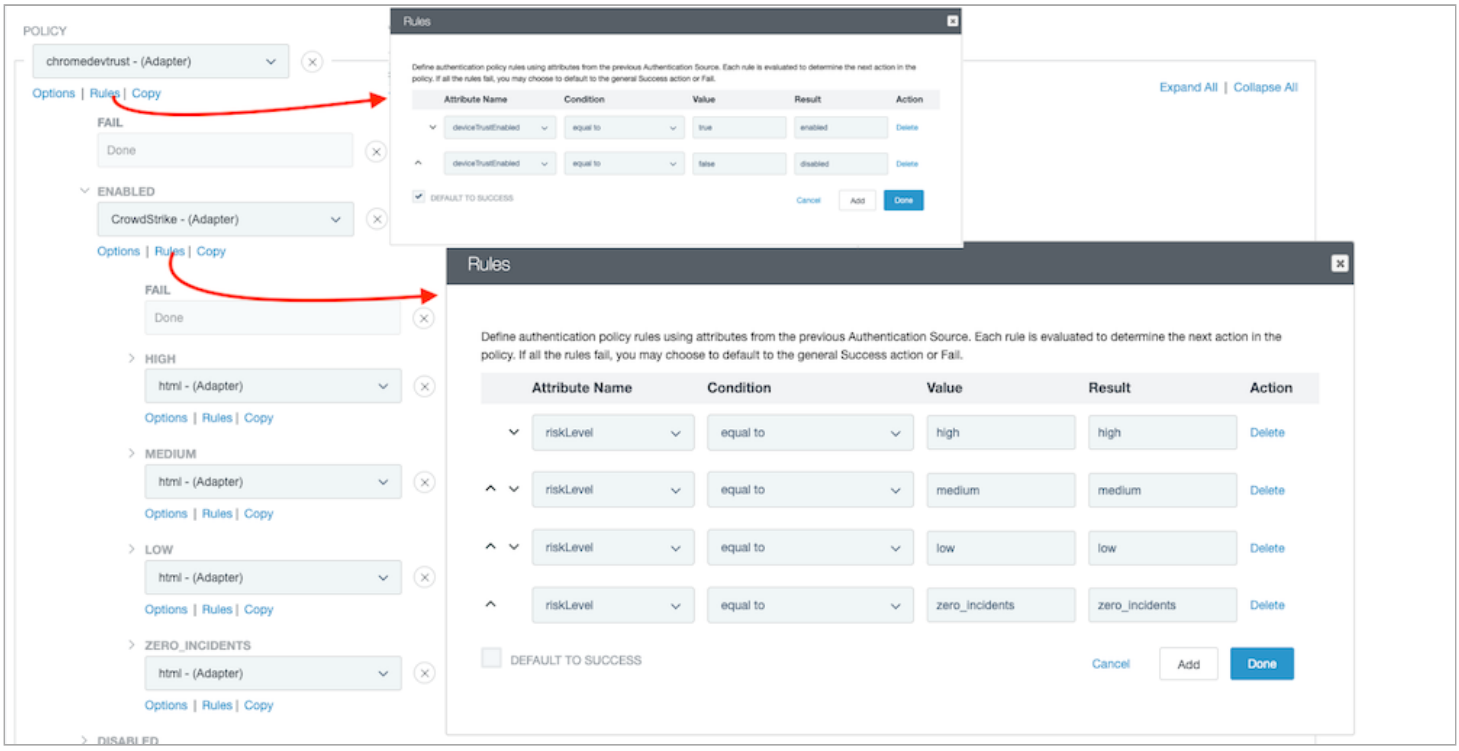
> **Note**
>
> If you can make the **CrowdStrike Agent ID** of the authenticating user available to the CrowdStrike IdP Adapter without using the Google Chrome Enterprise Device Trust IdP adapter, then you don't need to configure the Google Chrome Enterprise Device Trust IdP adapter in the PingFederate authentication policy.

The CrowdStrike IdP Adapter fulfills the `riskLevel` core contract attribute value based on the response it receives from the Google Chrome Enterprise Device Trust IdP Adapter and the threshold adapter's configuration. For example:

- If there are no incidents, or the returned score is less than the **Low Score Threshold** adapter configuration value, the `riskLevel` contract value is set to the value `zero_incidents`.
- If an incident is detected with a risk score in the range of the **High Score Threshold**, **Medium Score Threshold**, or **Low Score Threshold** field values as described in the CrowdStrike IdP Adapter settings reference, the `riskLevel` contract value is set to a value of `high`, `medium`, or `low` accordingly.

The following policy configuration shows a typical use case where a request coming from a trusted device goes through the Google Chrome Enterprise Device Trust IdP adapter and reaches the CrowdStrike IdP Adapter. The authentication policy is subsequently configured to handle and branch out based on different risk level scores.

Copyright © 2025 Ping Identity Corporation



Troubleshooting

Enabling debug logging

To help with troubleshooting or monitoring, you can turn on activity logging for PingFederate, the CrowdStrike IdP Adapter, or both.

About this task

You can use logging for both troubleshooting and analytics.

For general information about logging, see Enabling debug messages and console logging in the PingFederate documentation.

Steps

1. Open the `<pf_install>/pingfederate/server/default/conf/log4j2.xml` file for editing.
2. To log activity for PingFederate and all adapters:
    1. Find the following section:

```
<AsyncRoot level="INFO" includeLocation="false">
        <!-- <AppenderRef ref="CONSOLE" /> -->
        <AppenderRef ref="FILE" />
</AsyncRoot>
```

    2. Change `INFO` to `DEBUG`:

Copyright © 2025 Ping Identity Corporation

```
<AsyncRoot level="DEBUG" includeLocation="false">
        <!-- <AppenderRef ref="CONSOLE" /> -->
        <AppenderRef ref="FILE" />
</AsyncRoot>
```

3. **Optional:** To see the adapter activity in the console, remove the comment tags that surround the `CONSOLE` line:

```
<AsyncRoot level="INFO" includeLocation="false">
        <AppenderRef ref="CONSOLE" />
        <AppenderRef ref="FILE" />
</AsyncRoot>
```

3. To log activity relating to the CrowdStrike IdP Adapter:

   Choose from:

   - To log activity for the CrowdStrike IdP Adapter as well as its HTTPS and component activity, add the following line:

```
<Logger name="com.pingidentity.adapters.crowdstrike" level="DEBUG"/>
```

   - To log activity for the adapter's HTTPS activity and other components but not for the adapter itself, add the following line:

```
<Logger name="com.pingidentity.adapters.crowdstrike.shade"  level="DEBUG"/>
```

   - To log activity for the CrowdStrike IdP Adapter but not its HTTPS or component activity, add the following lines:

```
<Logger name="com.pingidentity.adapters.crowdstrike" level="DEBUG"/>
<Logger name="com.pingidentity.adapters.crowdstrike.shade" level="INFO"/>
```

4. Save the file.

Release Notes

Changelog

The following is the change history for the CrowdStrike Integration Kit.

CrowdStrike Integration Kit 1.0 – February 2024

- Initial release

Copyright © 2025 Ping Identity Corporation

Known issues and limitations

The following are known issues or limitations for the CrowdStrike Integration Kit.

Known limitations

- The adapter does not implement support for the PingFederate Authentication API because it is not applicable in the CrowdStrike IdP Adapter flow. The adapter determines the score and risk level for the **CrowdStrike Agent ID** it receives from the Google Chrome Enterprise Device Trust IdP Adapter and according to the configured CrowdStrike service, and these determinations are immediately used in authentication policy decisions.

Download manifest

The following files are included in the CrowdStrike Integration Kit `.zip` archive.

- `Legal.pdf` – copyright and license information
- `dist/pingfederate/server/default` – contains the integration files
    - `deploy` – contains the Java libraries
        - `pf-crowdstrike-adapter-<version>.jar` – The CrowdStrike IdP Adapter

Copyright © 2025 Ping Identity Corporation

Dropbox Provisioner

The PingFederate provisioner for Dropbox allows an enterprise to provision its users to Dropbox. This Dropbox Provisioner includes a quick connection template to easily setup a single sign-on (SSO) connection requiring Dropbox provisioning.

This guide provides instructions for quickly configuring PingFederate to connect to Dropbox for secure internet SSO. This configuration also lays the foundation for implementing optional outbound provisioning (formerly software as a service (SaaS) provisioning).

Features

- Outbound user provisioning
- Outound group provisioning
- Ability to add users to groups
- Browser-based service provider (SP)- and identity provider (IdP)-initiated SSO

Intended audience

This document is intended for PingFederate administrators.

If you need help during the setup process, learn more in the following sections of the PingFederate documentation:

- SP connection management
- Identity provider SSO configuration
- Datastores
- Configuring outbound provisioning
- Configuring outbound provisioning settings

System requirements

- PingFederate 7.2.1 or later
- You might need to add the following endpoints to the allow list on the firewall to allow outbound connections:
  - https://api.dropboxapi.com/

User and group management

The Dropbox Provisioner synchronizes users and groups from your datastore to Dropbox. The following sections describe the behavior of each provisioning capability.

Copyright © 2025 Ping Identity Corporation

Synchronizing existing users

PingFederate synchronizes users based on the `Email` attribute in Dropbox. If a user already exists in your datastore and Dropbox, mapping this attribute correctly links the two records together.

For example:

- In Dropbox, Janet's `Email` is `jsmith@domain.com`.
- In your datastore, Janet's `mail` is `jsmith@domain.com`.
- On the **Attribute Mapping** tab of your provisioning connection configuration, map the `Email` attribute to `mail`.
- When the provisioning connector runs, the datastore user is provisioned with a `Email` of `jsmith@domain.com`. That matches Janet's existing `Email` in Dropbox, so her information in the datastore is synchronized to her Dropbox account.

User provisioning

PingFederate provisions users when any of the following happens:

- A user is added to the datastore group or filter that is targeted by the provisioning connector.
- A user with "disabled" status is added to the datastore group or filter that is targeted by the provisioning connector, and the **Provision disabled users** provisioning option is enabled.

User updates

PingFederate updates users when a user attribute changes in your datastore.

The **Attribute Mapping** tab of your provisioning connection configuration defines which attributes PingFederate monitors for changes.

Synchronizing existing groups

PingFederate synchronizes groups from the datastore to the target service based on the group name.

For example:

- In Dropbox, there is a group is named `Accounting`.
- In your datastore, there is a group with a `CN` of `Accounting`.
- When the provisioning connector runs, the two groups are synchronized.

Group provisioning

PingFederate provisions groups when a group is added to the datastore filter that is targeted by the provisioning connector.

Group name updates

PingFederate renames groups when they are renamed in the datastore.

Copyright © 2025 Ping Identity Corporation

Group membership updates

PingFederate updates group memberships when memberships change in the datastore, whether the change is in the group's properties or a user's properties.

Group memberships in the datastore overwrite the group memberships in Dropbox.

Group deletion

PingFederate deletes groups when any of the following happen:

- The group is deleted in the datastore.
- The group is removed from the datastore group or filter that is targeted by the provisioning connector.

Group deletions are permanent and cannot be undone.

Setup

Obtaining your OAuth access token

For this procedure you will need to perform two steps:

1. Obtain an App Key and Secret from Dropbox.
2. Generate Your OAuth 2.0 Access Token.

Obtain an App Key and Secret from Dropbox

About this task

The Dropbox Connectors Outbound Provisioning functionality is built using Dropbox's REST API, which requires an OAuth 2.0 access token for authentication. To obtain the access token, you must obtain an app key and secret from Dropbox.

Steps

1. Sign on to Dropbox as an administrative user.
2. Go to **My Apps for Dropbox**.
3. Click the **Create app** button.
4. Click **Dropbox Business API** as the API to use.
5. Click **Team member management** as the type of access you need.
6. Give your application a name, such as `PingFederate Provisioning`.
7. Click **Create app**.
8. Copy the **App Key** and **App secret** values to use in the next section.

Copyright © 2025 Ping Identity Corporation

9. Add the following URL to the **Redirect URIs**.

   https://oauth.pingone.com/ocs/ppm/rest/v1/oauth/oastempcredresponse/

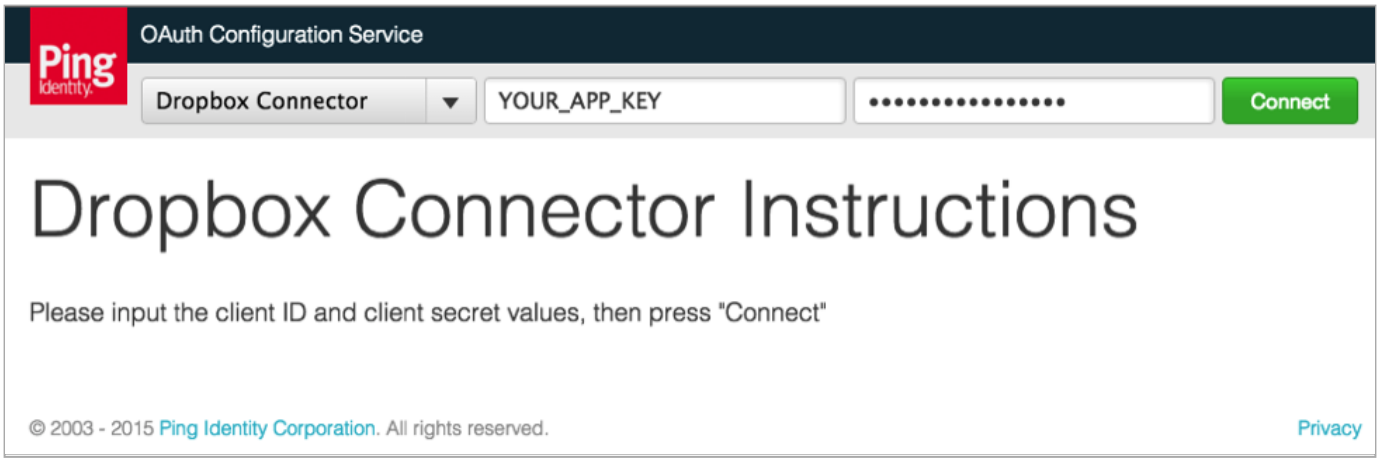
10. (Optional) Click the **Apply for production** button to open your app to more users.

> **Note**
>
> Development status only allows your app to be accessed by up to 100 users.

Generate Your OAuth 2.0 Access Token

Steps

1. Go to the Ping Identity OAuth Configuration Service.
2. In the connector menu, select **Dropbox Connector**.
3. In the **ClientID** field, enter the **App Key** you obtained previously.
4. In the **Client Secret** field, enter the **App Secret** you obtained previously and click **Connect**.

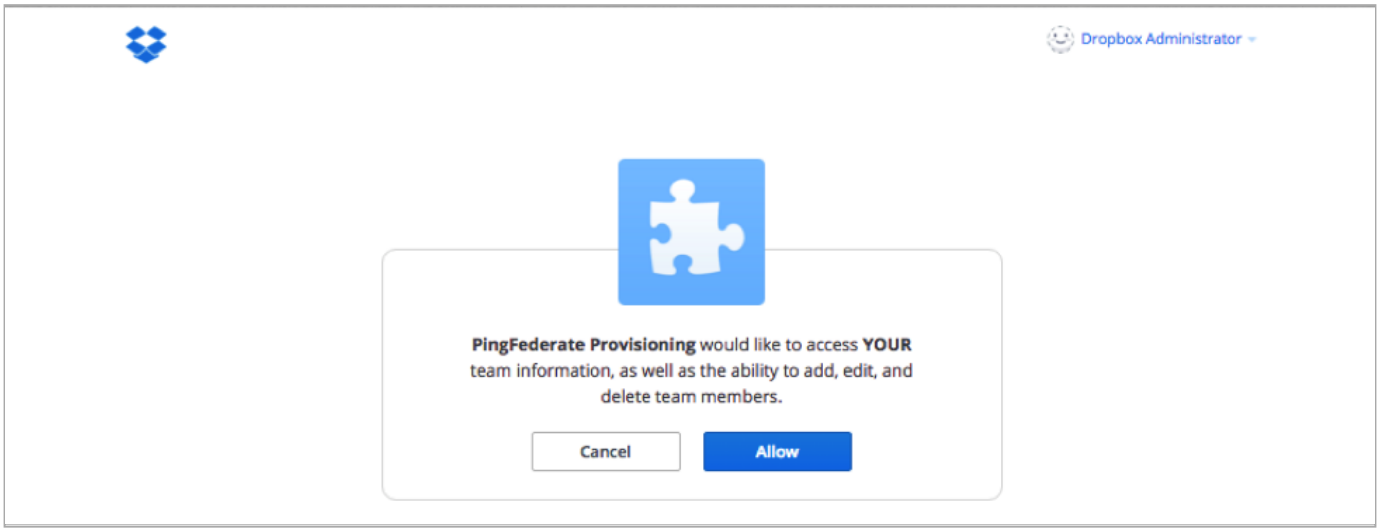


5. Sign on to Dropbox as an administrative user.

> **Note**
>
> If you are already signed on to Dropbox, you will not be asked to sign on again. Make sure that the account you are signed on as is an administrative account.

6. Click **Allow** to generate your Access Token.

Copyright © 2025 Ping Identity Corporation



7. Copy the **Access Token** to use when configuring the Dropbox Connector.

Preparing the Dropbox SAML 2.0 metadata XML file

This Connector's quick-connection template uses a metadata XML file to assist in configuring many settings in the service provider (SP) connection.

About this task

When asked during the connection configuration steps, import the `saml-metadata.xml` packaged with this connector.

Steps

1. Open the `saml-metadata.xml` file contained with your connector with a text editor of your choice.
2. No changes are required, unless you have an SP connection already created with the same entity ID.
3. When you have updated the `saml-metadata.xml` file, save your changes.

Upgrading existing Dropbox connectors

Steps

1. Before stopping the PingFederate server to upgrade the Dropbox Connector, on the **Attribute Mapping** tab for existing channel configurations, record the current configuration.

> ⚠ **Warning**
>
> The upgrade process removes existing mappings and defaults on the **Attribute Mapping** tab. You must configure these again before activating the channel configuration.

2. Disable the existing service provider (SP) connection where the Dropbox Connector is configured.
3. Delete the existing Dropbox Connector SP connection and save.
4. Stop the PingFederate server if it is running.

Copyright © 2025 Ping Identity Corporation

5. Extract the Dropbox Connector distribution `.zip` archive into a holding directory.
6. Remove any versions of `pf-dropbox-quickconnection-.x.jar` from the `<pf_install>/pingfederate/server/default/deploy` directory.
7. Remove the following files from the same directory if they are present:
    - `pf-dropbox-oauth-helper.war`
    - `gson-2.2.4.jar`
8. From the `/dist` directory of the new version of the connector, copy the `pf-dropbox-quickconnection-2.0.jar` to the `<pf_install>/pingfederate/server/default/deploy` directory.

> **Important**
>
> Make sure to remove existing versions of Dropbox Connector files.

9. Start the PingFederate server.
10. Create a new SP connection using Dropbox as the connection template.
11. Follow the instructions in the Creating a connection section to configure Metadata and OAuth.
12. Access the Attribute Mapping for existing channel configurations and click **Refresh Fields**.
13. Ensure all new required fields, if any, are mapped appropriately or have a default value.
14. When you have finished the attribute configuration, click **Done**, **Done**, and **Save**.
15. Click the toggle to activate the SP Connection to resume outbound provisioning.

Deploying the integration files

To get started with the integration, deploy the Dropbox Provisioner files to your PingFederate directory.

Steps

1. Download the Dropbox Provisioner `.zip` archive from the Ping Identity Integration Directory.
2. Stop PingFederate.
3. If you're upgrading an existing deployment, delete the `pf-dropbox-quickconnection-<version>.jar` file from your `<pf_install>/pingfederate/server/default/deploy` directory.
4. Extract the `.zip` archive and merge the contents of the `dist` directory with your `<pf_install>/pingfederate/server/default/deploy` directory.
5. Enable the PingFederate provisioning engine:
    1. Open your `<pf_install>/pingfederate/bin/run.properties` file.
    2. Change `pf.provisioner.mode` to `STANDALONE`. Save the file.

Copyright © 2025 Ping Identity Corporation

> **Note**
>
> Learn more about configuring the `FAILOVER` mode instead in Deploying provisioning failover in the PingFederate Server Clustering Guide.

6. Start PingFederate.
7. If you operate PingFederate in a cluster, repeat steps 2 - 4 and step 6 for each engine node.

Enabling provisioning in PingFederate

To use PingFederate for provisioning, configure an external datastore.

About this task

Your external datastore acts as the source of data for provisioning. PingFederate also uses an internal datastore to store the state of synchronization between the source datastore and the target datastore.

Learn more in Datastores in the PingFederate documentation.

Steps

1. Configure the datastore for PingFederate to use as the source of user data.

   Learn more in Datastores in the PingFederate documentation.

   > **Tip**
   >
   > When targeting users and groups for provisioning, exclude the user account that you will use to administer users in your connection to Dropbox. This prevents the PingFederate provisioning engine from interfering with the account that provisions users and groups.

2. Do one of the following:

   Choose from:

   - For PingFederate 10.1 or later: Go to **System > Server > Protocol Settings**.
   - For PingFederate 10.0 or earlier: Enable the identity provider (IdP) and outbound provisioning roles:
     1. Go to **System > Protocol Settings > Roles & Protocols**.
     2. Select **Enable Identity Provider IdP Role and Support the Following**.
     3. Select **Outbound Provisioning**. Click **Next**.

3. On the **Outbound Provisioning** tab, select the PingFederate internal datastore. Click **Save**.

   Learn more in Configuring outbound provisioning settings in the PingFederate documentation.

Creating a connection

To allow PingFederate to act as an identity provider (IdP) and manage users in Dropbox, create a service provider (SP) connection.

Copyright © 2025 Ping Identity Corporation

Steps

1. In the PingFederate administrator console, create a new SP connection:

   Choose from:

   - For PingFederate 10.1 or later: go to **Applications > Integration > SP Connections**. Click **Create Connection**.
   - For PingFederate 10.0 or earlier: go to **Identity Provider > SP Connections**. Click **Create Connection**.

2. Configure the basic connection details with the Dropbox quick connection template:

   1. On the **Connection Template** tab, click **Use a template for this connection**.
   2. In the **Connection Template** list, select **Dropbox Provisioner**.
   3. On the **Metadata File** row, upload the `saml-metadata.xml` file that you saved in Preparing the Dropbox SAML 2.0 metadata XML file. Click **Next**.
   4. On the **Connection Type** tab select the **Browser SSO Profiles** and **Outbound Provisioning** checkboxes. Click **Next**.
   5. On the **Connection Options** tab, click **Next**.
   6. On the **General Info** tab, the default values are taken from the metadata file you uploaded earlier. Click **Next**.

3. On the **Browser SSO** tab, configure browser single sign-on (SSO).

   Learn more in Configuring IdP Browser SSO in the PingFederate documentation.

   1. On the **Browser SSO > SAML Profiles** tab, select the **IDP-Initiated SSO** and **SP-Initiated SSO** checkboxes.
   2. On the **Browser SSO > Assertion Creation > Attribute Contract** tab, in the **SAML_SUBJECT** row, select **urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress** in the **Subject Name Format** list.
   3. On the **Browser SSO > Protocol Settings > Allowable SAML Bindings** tab, select the **Post** and **Redirect** checkboxes. Click **Next**.

4. On the **Browser SSO > Protocol Settings > Signature Policy** tab, select the **Always sign the SAML Assertion** checkbox. Click **Next**.

5. On the **Credentials** tab, configure the connection credentials as shown in Configuring credentials in the PingFederate documentation. Click **Next**.

6. On the **Outbound Provisioning** tab, configure provisioning with the following details.

   Learn more in Configuring outbound provisioning in the PingFederate documentation.

   1. On the **Target** tab, complete the fields as follows:

Field Name	Description
OAuth 2 Access Token	The OAuth 2.0 access token for authentication. To obtain the access token, you must first Obtain an App Key and Secret from Dropbox. When you have obtained the app key and secret from Dropbox, you can Generate Your OAuth 2.0 Access Token.

Copyright © 2025 Ping Identity Corporation

Field Name	Description
User Create Enabled	■ `True` (default): Users will be created in Dropbox through PingFederate. ■ `False` : Users will not be created in Dropbox. **Note** The provisioner.log displays a warning within the create user workflow that the user was not created in Dropbox.
User Update Enabled	■ `True` (default): Users will be updated in Dropbox through PingFederate. ■ `False` : Users will not be updated in Dropbox. **Note** The `provisioner.log` will display a warning within the update user workflow that the user was not updated in Dropbox.

2. (Optional) In the **Provisioning Options** section, customize the provisioning connector behavior. Click **Next**.
3. On the **Manage Channels > Attribute Mapping** tab, at the bottom of the attribute list, click **Refresh Fields** to get fields and specifications from your Dropbox site.
4. Complete the attribute mappings by referring to Supported attributes reference.

Learn more in Managing channels in the PingFederate documentation.

7. On the **Activation and Summary** tab, above the **Summary** section, click the toggle to enable the connection. Click **Save**.

Supported attributes reference

Attribute	Description
`Given Name`	The user's first name.
`Surname`	The user's last name.
`Email`	The user's email address.
`Role`	The user's role in Dropbox. Valid options are: • `member_only` • `support_admin` • `team_admin` • `user_management_admin`

Copyright © 2025 Ping Identity Corporation

Attribute	Description
`Send Welcome Email`	Whether to send a newly created user a welcome email. Valid options are `true` or `false`. **Note** If set to false, no email invitation is sent to the user.
`Wipe Data on Delete`	Whether to delete data on user devices when they are suspended (deleted or disabled). Valid options are `true` or `false`. If not specified, Dropbox assumes it should delete data on user devices.
`External ID`	A unique external identifier for the user.

Completing setup of SAML SSO to Dropbox

The following section describes the steps for configuring service provider (SP)- and identity provider (IdP)-initiated single sign-on (SSO) to Dropbox.

About this task

Note

This section requires two pieces of information from PingFederate:

- The PingFederate SSO Application Endpoint, which can be found on the **Activation & Summary** page, in the **SP Connection** section
- The exported certificate used to sign the SAML assertion (configured in Creating a connection)

Steps

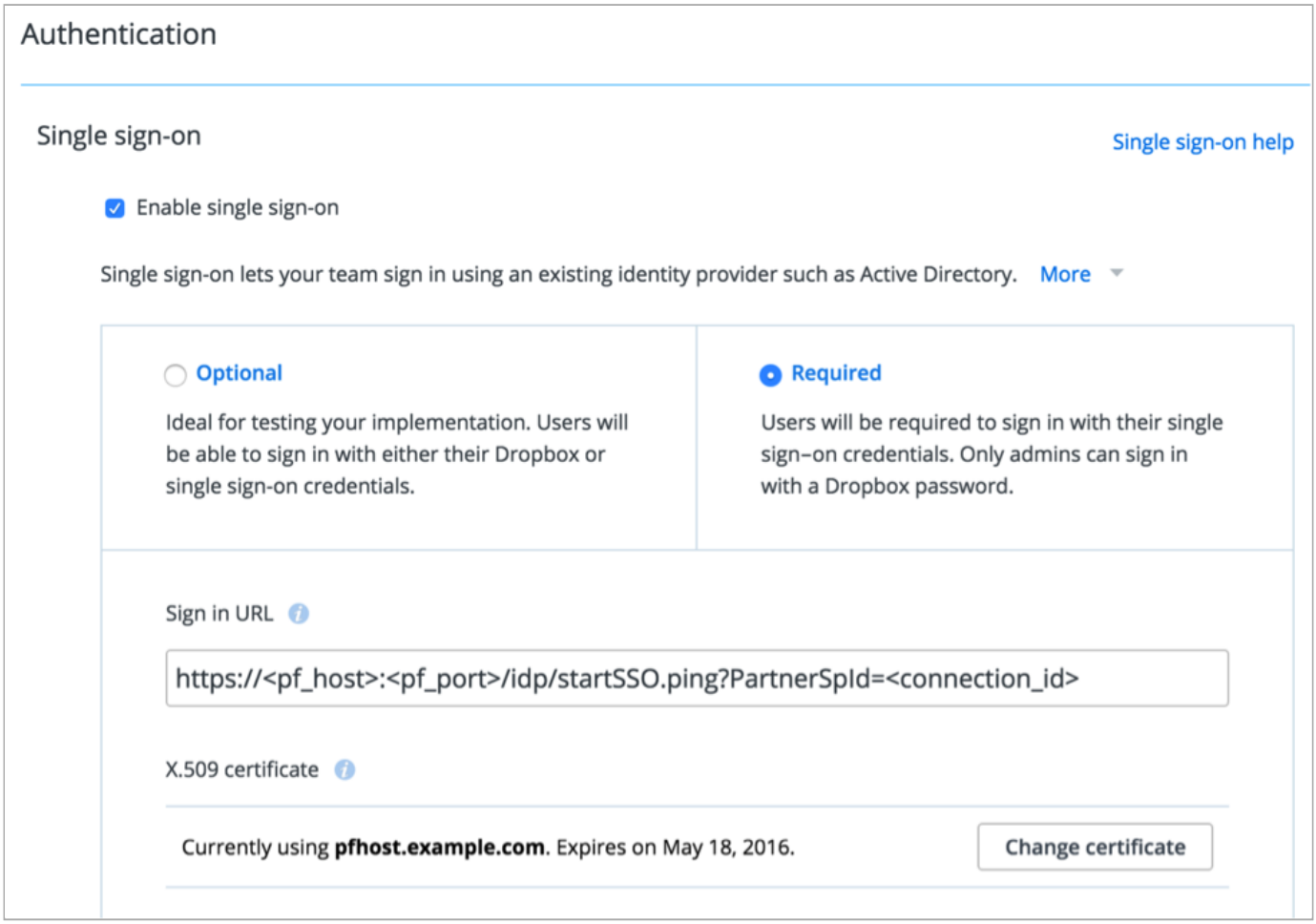
1. Go to https://www.dropbox.com/team/admin/ and sign on with your team owner credentials.
2. Go to **Authentication** to view the **Single sign-on** section.
3. Select the **Enable single sign-on** checkbox and click **Optional** or **Required** based on your SSO requirements.
4. In the **Sign in URL** field, enter the PingFederate SSO Application endpoint.

   https://*pf_host*:*pf_port*/idp/startSSO.ping?PartnerSpId=*connection_id*

   where:

   - *pf_host* is the machine running the PingFederate server.
   - *pf_port* is the PingFederate port (default value: 9031).
   - *connection_id* is the connection ID of the SP connection, for example, https://www.dropbox.com/.
5. Import the signing certificate into the X.509 certificate field.

Copyright © 2025 Ping Identity Corporation



6. Click **Save Changes** to complete the Dropbox SSO setup.

   Result:

   When saved, emails are sent to team members to instruct them on how to initiate signing on.

Release notes

Changelog

Dropbox Connector 2.1 – April 2017 (current release)

- Added configuration options for CRUD capabilities

Dropbox Connector 2.0.1 – April 2016

- Added Support for proxy connections

Dropbox Connector 2.0 – September 2015

- Added Support for additional User attributes

Copyright © 2025 Ping Identity Corporation

- Added Support for Group Provisioning
- Added Support for adding Users to Groups

Dropbox Connector 1.0 – August 2013

- Initial Release
- Added Support for User Provisioning

Known issues and limitations

The following are known issues and limitations of the Dropbox Connector.

- Single logout (SLO) is not supported.
- User attributes cannot be cleared after they are set.
- Because of API limitations, a user's email cannot be updated until the user has activated their account.
- Because of API limitations, a user cannot suspend or un-suspend until the user has activated their account.
- Because of API limitations, if a user's given name or surname fails to update due to the new value containing unsupported characters (*|:"<>?), an error might not be reported in the provisioning logs.
- Because of API limitations, a user cannot be added to a group until the user has activated their account.
- The `externalId` attribute is not supported for group provisioning

Download manifest

The distribution `.zip` archive for the Dropbox Connector contains the following files:

- `ReadMeFirst.pdf` : Contains links to this online documentation
- `saml-metadata.xml` : The metadata used for browser single sign-on (SSO)
- `/legal` : Contains the following document:
  - `Legal.pdf` : Copyright and license information
- `/dist` : Contains libraries needed for the connector:
  - `pf-dropbox-quickconnection-2.1.jar` : The PingFederate Dropbox Connector

Copyright © 2025 Ping Identity Corporation

Duo Security integrations

Duo Security Integration Kit (latest)

The latest Duo Security Integration Kit allows PingFederate to use Duo Universal Prompt for multi-factor authentication (MFA). This sign-on experience uses OpenID Connect standards and does not require iFrames.

Duo Security Integration Kit 2.2

This documentation is available for customers with existing deployments of Duo Security Integration Kit 2.2 and earlier. This sign-on experience uses an iFrame and works with the Duo Auth API.

This version of the Duo Security Integration Kit integrated with Duo MFA using a PingFederate-based template.

Duo Security Integration Kit 3.x

The Duo Security Integration Kit allows PingFederate to use Duo Universal Prompt for multi-factor authentication (MFA). This sign-on experience uses OpenID Connect standards and does not require or support iFrames.

> **Note**
>
> This documentation is for Duo Security Integration Kit 3.0 and later. For earlier versions of the integration kit, see the Duo Security Integration Kit 2.2 documentation.

Upgrade considerations

The Duo Security Integration Kit 3.0 and later no longer support the session state feature or the adapter session feature. If you were using the adapter session feature previously, you can configure Authentication sessions instead.

> **Note**
>
> The authentication sessions feature doesn't have complete feature parity with the adapter session feature. For example, the authentication sessions feature doesn't support session sharing between composite adapters. To use authentication sessions with the Duo Security Integration Kit 3.0 and later, remove any interfering composite adapters.

Features

- Supports Duo's Standard Editions and Federal Editions
- Supports Duo's OpenID Connect-enabled APIs.
- Supports the Duo Universal Prompt authentication experience.
- Supports the PingFederate PingFederate authentication API.
- Supports the JavaScript Widget for the PingFederate Authentication API.

Copyright © 2025 Ping Identity Corporation

Components

The Duo Security IdP Adapter:

- Allows PingFederate to redirect users to Duo Universal Prompt for a multi-factor authentication (MFA) challenge during sign on.
- Allows for MFA when single sign-on (SSO) is initiated at the identity provider (IdP) or service provider (SP).

Intended audience

This document is intended for PingFederate administrators.

If you need help during the setup process, see the following resources:

- The following sections of the Duo Universal Prompt documentation:
  - Getting Started with Duo Security
  - What is the Duo Universal Prompt?
  - Application Options - Universal Prompt
  - Duo Universal Prompt Update Guide
  - Troubleshooting - Universal Prompt Update Progress
  - PingFederate
- The following sections of the PingFederate documentation:
  - Managing IdP adapters
  - Authentication policies

System requirements

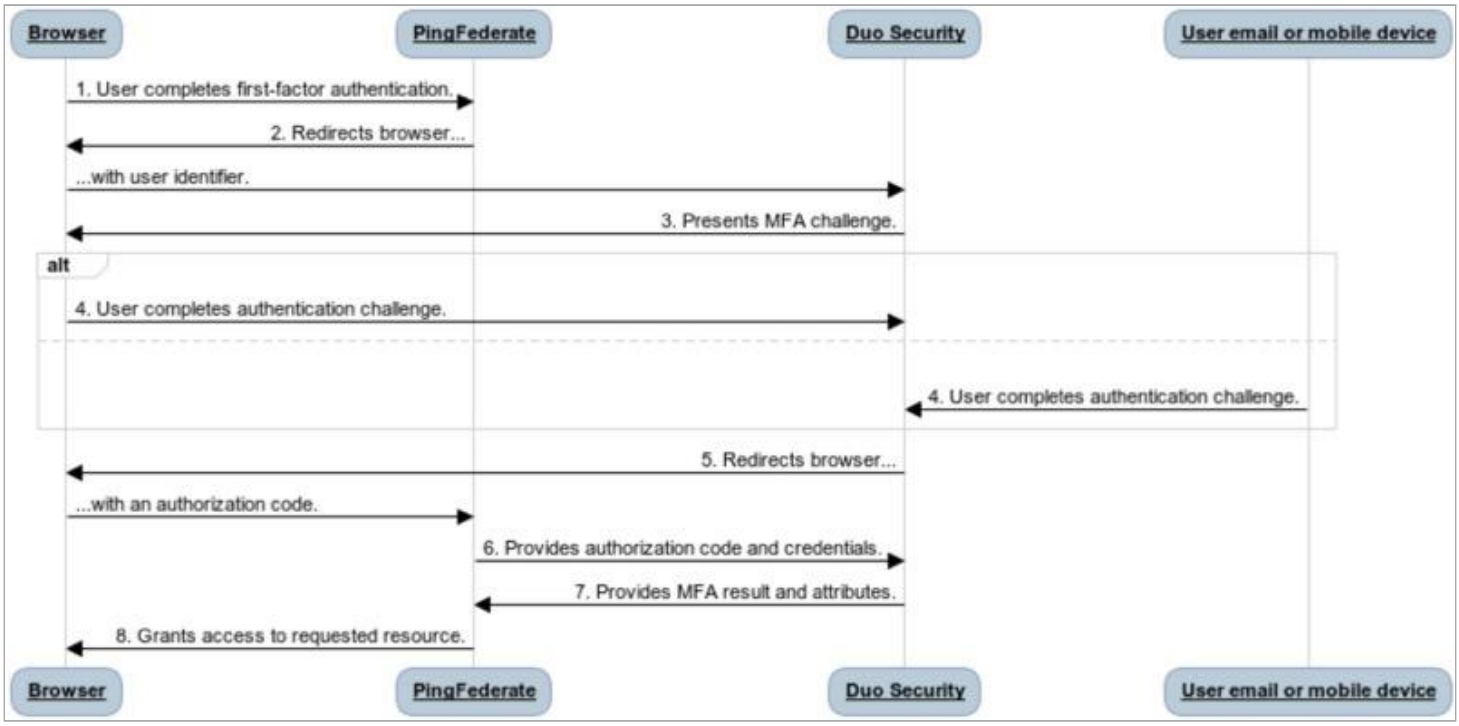
- PingFederate 10.3 or later
- A Duo Universal Prompt administrator account

Overview of the SSO flow

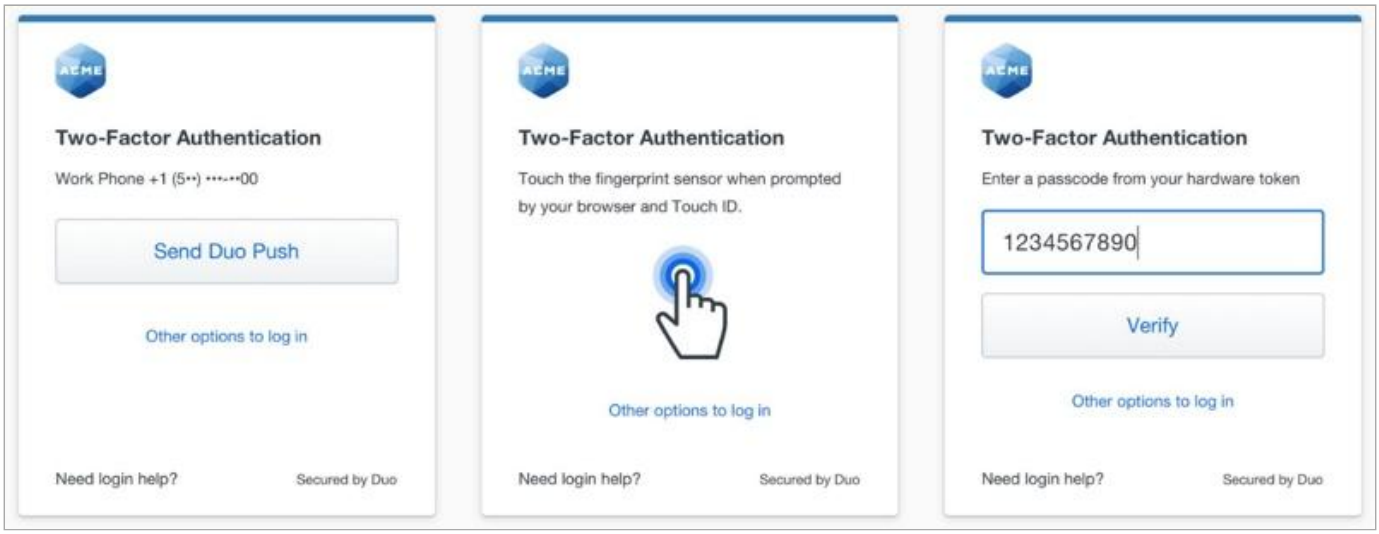
With the Duo Security Integration Kit, PingFederate includes Duo Universal Prompt in the sign-on flow.

The following figure illustrates a single sign-on (SSO) scenario in which PingFederate authenticates users to an SP application using Duo Universal Prompt.

Copyright © 2025 Ping Identity Corporation



1. The user initiates SSO with PingFederate and completes the first-factor authentication step, such as an HTML Form Adapter instance.
2. PingFederate redirects the user to Duo Universal Prompt with a user identifier.
3. Duo Security presents the user's second-factor authentication options.



4. The user completes the MFA challenge in their browser or on their mobile device.
5. Duo Universal Prompt redirects the browser to PingFederate with an authorization code.
6. PingFederate provides its credentials and the authorization code to Duo Universal Prompt.
7. Duo Universal Prompt provides the authentication result and other user attributes.
8. If the user authenticated successfully, PingFederate provides access to the requested resource.

Copyright © 2025 Ping Identity Corporation

Authentication API support

You can use the PingFederate authentication API to integrate the Duo Security IdP Adapter into your application.

The PingFederate Authentication API provides access to the current state of the authentication flow as a user steps through the PingFederate authentication policy. Learn more in PingFederate authentication API in the PingFederate documentation.

To integrate the Duo Security IdP Adapter into your authentication flow, configure your application based on the states, actions, and models available in the PingFederate Authentication API Explorer. Learn more in Exploring the Authentication API in the PingFederate documentation.

Setup

Deploying the integration files

To get started with the integration, deploy the Duo Security Integration Kit files to your PingFederate directory.

Steps

1. Download the Duo Security Integration Kit `.zip` archive from the **Add-ons** tab of the PingFederate downloads page.
2. Stop PingFederate.
3. If you are upgrading an existing deployment, delete earlier versions of the integration files.

   Choose from:

   - For upgrades from Duo Security Integration Kit 3.0 or later: delete `pf-duo-security-integration-kit-<version>.jar` from `<pf_install>/pingfederate/server/default/deploy`.
   - For upgrades from Duo Security Integration Kit 2.2 or earlier:
     1. Delete the `duo.adapter.template.html` file from `<pf_install>/pingfederate/server/default/conf/template`.
     2. Delete the following files from `<pf_install>/pingfederate/server/default/deploy`:
        - `pf-duo-security-adapter-<version>.jar`
        - `duo-web/js/Duo-Web-v2.min.js`
4. Extract the `.zip` archive and merge the contents of the `dist` directory with your `<pf_install>/pingfederate` directory.
5. If there is more than one version of the `pf-authn-api-sdk-<version>.jar` file in your `<pf_install>/pingfederate/server/default/lib` directory, delete all but the latest version of the file.
6. Start PingFederate.
7. If you operate PingFederate in a cluster, repeat steps 2 - 6 for each engine node.

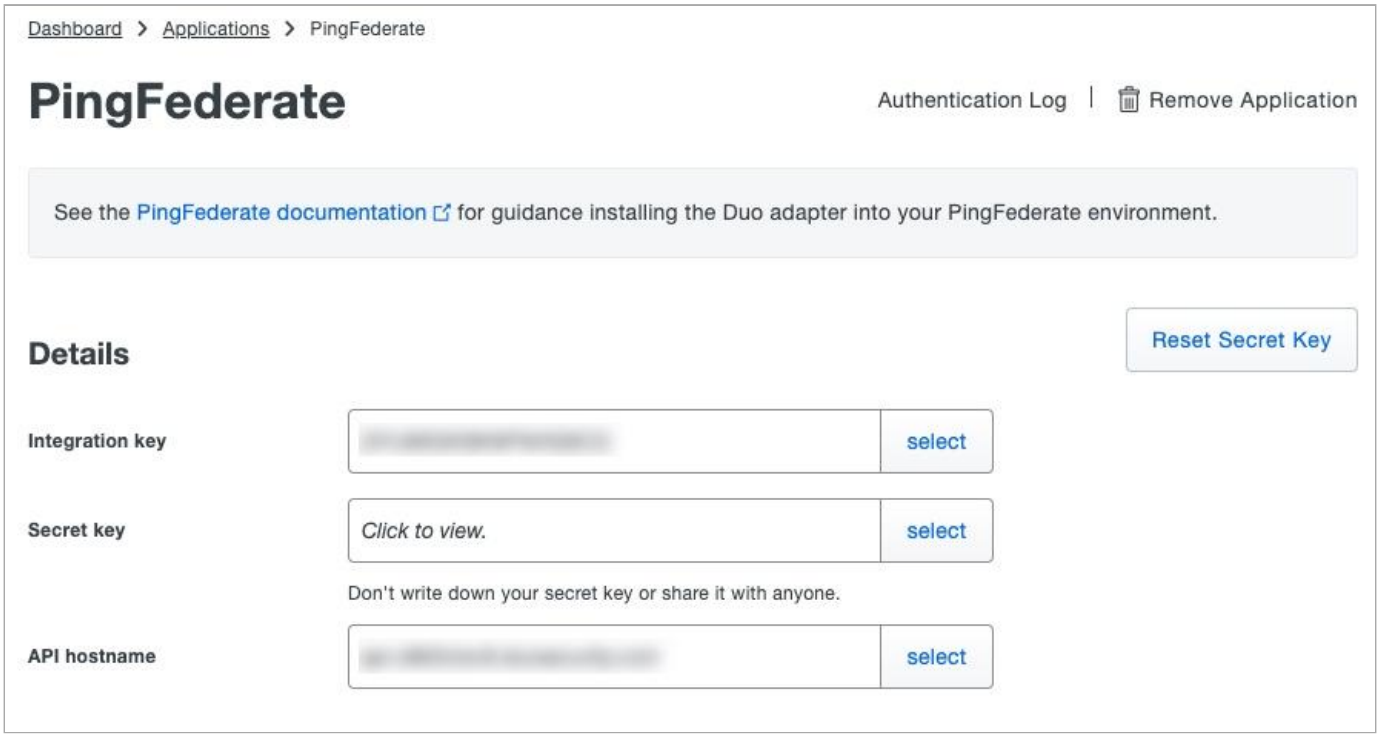
Copyright © 2025 Ping Identity Corporation

Getting your Duo Security keys and API hostname

To allow PingFederate to communicate with Duo Security, add it as a protected application.

Steps

1. Log into Duo Security as an administrator.
2. Go to **Applications > Protect an Application**. For **PingFederate**, click **Protect this Application.**
3. Note the **Client ID**, **Client Secret**, and **API hostname**. You will use these in Configuring an adapter instance.



Configuring an adapter instance

Configure the Duo Security IdP Adapter to determine how PingFederate communicates with Duo Universal Prompt.

Steps

1. In the PingFederate administrative console, create a new IdP adapter instance:

   Choose from:

   - For PingFederate 10.1 or later: go to **Authentication > Integration > IdP Adapters**. Click **Create New Instance**.
   - For PingFederate 10.0 or earlier: go to **Identity Provider > Adapters**. Click **Create New Instance**.

2. On the **Type** tab, set the basic adapter instance attributes.

   1. In the **Instance Name** field, enter a name for the adapter instance.
   2. In the **Instance ID** field, enter a unique identifier for the adapter instance.

Copyright © 2025 Ping Identity Corporation

3. From the **Type** list, select **Duo Security IdP Adapter**. Click **Next**.

3. **Optional:** On the **IdP Adapter** tab, in the **Duo API Response Mappings** section, map attributes from the Duo Universal Prompt response to the attribute contract. These attributes become available in your PingFederate authentication policy.

    1. Click **Add a new row to 'Duo API Response Mappings'**.
    2. In the **Local Attribute** field, enter a name of your choosing for an attribute.
    3. In the **Duo API Attribute Mapping** field, enter the JSON Pointer syntax for the value of the matching Duo Security attributes as shown in JSON Pointer syntax reference.
    4. In the **Action** column, click **Update**.
    5. To add more attributes, repeat steps a - d.

4. On the **IdP Adapter** tab, configure the adapter instance by referring to Duo Security IdP Adapter settings reference.
5. On the **Extended Contract** tab, add any attributes that you want to include in the contract. Click **Next**.
6. On the **Adapter Attributes** tab, set pseudonym and masking options as shown in Set pseudonym and masking options in the PingFederate documentation. Click **Next**.
7. On the **Adapter Contract Mapping** tab, configure the contract fulfillment details for the adapter as shown in Define the IdP adapter contract in the PingFederate documentation. Click **Next**.
8. On the **Summary** tab, check and save your configuration:

    Choose from:

    - For PingFederate 10.1 or later: click **Save**.
    - For PingFederate 10.0 or earlier: click **Done**. On the **Manage IdP Adapter Instances** tab, click **Save**.

JSON Pointer syntax reference

JavaScript Object Notation (JSON) Pointer defines a syntax for identifying a specific value within a JSON payload. Using the sample payload and JSON Pointer examples below, identify the attributes that you want to use to populate your attribute contract.

For a complete technical description of JSON Pointer syntax, see JavaScript Object Notation (JSON) Pointer on ietf.org.

Copyright © 2025 Ping Identity Corporation

Example JSON payload


```
{
  "aud": "FGHI1J2K34L5MN6OPQ7R",
  "auth_context": {
    "access_device": {
      "ip": "192.0.0.1",
      "hostname": null,
      "location": {
        "state": "New York",
        "city": "New York",
        "country": "United States"
      }
    },
    "factor": "duo_push",
    "txid": "8adfb30c-bde3-97a6-b753-3abbcca321654",
    "auth_device": {
      "ip": "192.0.0.1",
      "location": {
        "state": "New York",
        "city": "New York",
        "country": "United States"
      },
      "name": "mydevice"
    },
    "reason": "user_approved",
    "event_type": "authentication",
    "user": {
      "key": "FGHI1J2K34H2MDLAZGVO",
      "name": ""
    },
    "timestamp": 1581993342,
    "result": "success",
    "alias": "",
    "application": {
      "key": "FGHI1J2K34C3PC0HBW0M",
      "name": "Web SDK"
    }
  },
  "auth_result": {
    "status_msg": "Login Successful",
    "status": "allow",
    "result": "allow"
  },
  "auth_time": 1581993342,
  "exp": 1581993642,
  "iat": 1581993376.883131,
  "iss": "https://api-a123bcd4.duosecurity.com/oauth/v1/token",
  "sub": "janetsmith"
}
```


JSON Pointer syntax:

Copyright © 2025 Ping Identity Corporation

Description	JSON Pointer	Example value
Access device location (country)	/auth_context/access_device/location/country	`United States`
Status message	/auth_result/status_msg	`Login Successful`
Authentication result	/auth_result/result	`allow`

Duo Security IdP Adapter settings reference

Field descriptions for the Duo Security IdP Adapter configuration screen.

Standard fields

Field	Description
Client ID	The client ID that you noted in Getting your Duo Security keys and API hostname. This field is blank by default.
Client Secret	The client secret that you noted in Getting your Duo Security keys and API hostname. This field is blank by default.
API Hostname	The API hostname that you noted in Getting your Duo Security keys and API hostname. This field is blank by default.
Failure Mode	When Duo Universal Prompt is unavailable or an error occurs, this setting determines whether the user's sign-on attempt should fail or continue. • Select **secure** to deny all sign-on attempts when Duo Universal Prompt is unavailable. • Select **safe** to allow all sign-on attempts when Duo Universal Prompt is unavailable. The default selection is **secure**.

Advanced fields

Field	Description
Duo Health Check Endpoint	The endpoint used to check the availability of the Duo Universal Prompt API. The default value is `/oauth/v1/health_check`.
Duo Authorization Endpoint	The endpoint used to perform second-factor authentication to Duo Universal Prompt. The default value is `/oauth/v1/authorize`.
Duo Access Token Endpoint	The endpoint used to get an access token from Duo Universal Prompt. The default value is `/oauth/v1/token`.

Copyright © 2025 Ping Identity Corporation

Field	Description
API Request Timeout	The amount of time in milliseconds that PingFederate allows when establishing a connection with Duo Universal Prompt or waiting for a response to a request. A value of 0 disables the timeout. The default value is `5000`.
Send Application Name	Send Duo the application name configured in the SP connection for use in push notifications. This check box is selected by default.
Proxy Settings	Defines proxy settings for outbound HTTP requests. The default value is **System Defaults**.
Custom Proxy Host	The proxy server host name to use when **Proxy Settings** is set to **Custom**. This field is blank by default.
Custom Proxy Port	The proxy server port to use when **Proxy Settings** is set to **Custom**. This field is blank by default.

Adding Duo Security to your authentication policy

By modifying your PingFederate authentication policy to include the Duo Security IdP Adapter, you can challenge users to complete a multi-factor authentication (MFA) step.

About this task

These steps are designed to help you add to an existing authentication policy. You can find general information about configuring authentication policies in Authentication policies in the PingFederate documentation.

Steps

1. In the PingFederate administrative console, go to the **Policies** tab.

   Choose from:

   - For PingFederate 10.1 or later: go to **Authentication > Policies > Policies**.
   - For PingFederate 10.0 or earlier: go to **Identity Provider > Authentication Policies > Policies**.

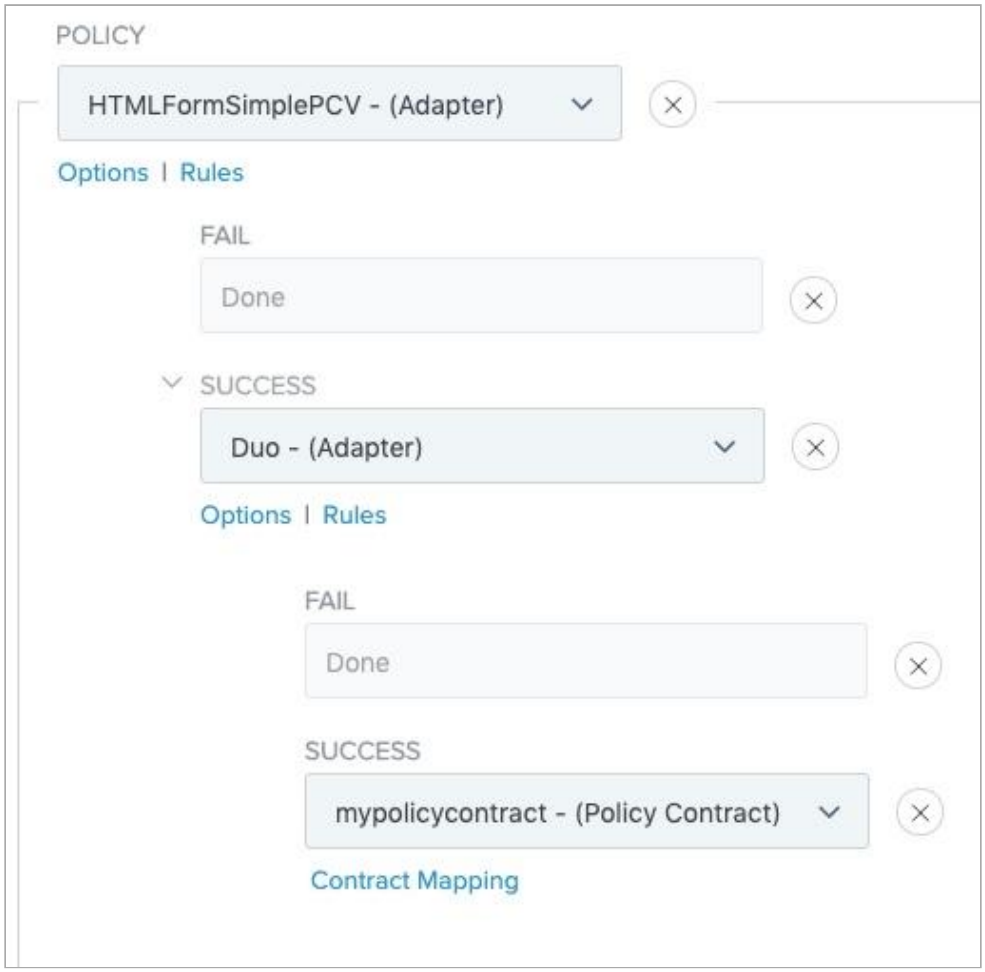
2. Select the **IdP Authentication Policies** check box.
3. Open an existing authentication policy, or click **Add Policy**. Learn more in Defining authentication policies in the PingFederate documentation.
4. In the **Policy** area, from the **Select** list, select a Duo Security IdP Adapter instance.
5. Map the Duo Security user ID or username into the Duo Security IdP Adapter instance.

   1. Under the Duo Security IdP Adapter instance, click **Options**.

Copyright © 2025 Ping Identity Corporation

2. On the **Options** dialog, from the **Source** list, select a previous authentication source that collects the Duo Security user ID or username.
3. From the **Attribute** list, select the user ID. Click **Done**.

6. Configure each of the authentication paths.



7. Click **Done**.
8. In the **Policies** window, click **Save**.

Troubleshooting

Enabling debug logging

To help with troubleshooting or monitoring, you can turn on activity logging for the adapter.

About this task

These steps are optional. For general information about logging, see Enabling debug messages and console logging in the PingFederate documentation.

Copyright © 2025 Ping Identity Corporation

Steps

1. Open the `<pf_install>/pingfederate/server/default/conf/log4j2.xml` file for editing.
2. If you want to log activity for PingFederate and all adapters, do the following.
    1. Find the following section.

```
<AsyncRoot level="INFO" includeLocation="false">
        <!-- <AppenderRef ref="CONSOLE" /> -->
        <AppenderRef ref="FILE" />
</AsyncRoot>
```

    2. Change `INFO` to `DEBUG`.

```
<AsyncRoot level="DEBUG" includeLocation="false">
        <!-- <AppenderRef ref="CONSOLE" /> -->
        <AppenderRef ref="FILE" />
</AsyncRoot>
```

    3. If you want to see the adapter activity in the console, remove the comment tags.

```
<AsyncRoot level="INFO" includeLocation="false">
        <AppenderRef ref="CONSOLE" />
        <AppenderRef ref="FILE" />
</AsyncRoot>
```

3. If you want to log activity just for the Duo Security IdP Adapter, add the following line.

```
<Logger name="com.pingidentity.clientservices.product.integrationkit.duo" level="DEBUG"/>
```

4. Save the file.

Release notes

Changelog

The following is the change history for the Duo Security Integration Kit.

Duo Security Integration Kit 3.1.2 - May 2024

- Fixed a issue that caused the adapter to throw an error instead of making an authentication request to Duo if an OIDC call preceded the adapter call.

Copyright © 2025 Ping Identity Corporation

Duo Security Integration Kit 3.1.1 - July 2023

- Fixed an issue with how the adapter collects the application name configured in the SP connection and sends it to Duo for use in mobile push notifications.
- Added the ability to select whether to send application names to Duo for use in push notifications. When the **Send Application Name** check box is selected, the adapter sends Duo the application's name if it's configured in the SP connection.

> **Note**
>
> This check box is selected by default. If you haven't set an application name or if you clear the **Send Application Name** check box, push notifications use the application name configured in Duo instead.

Duo Security Integration Kit 3.1 - May 2023

- Added the ability for the adapter to send the application name to Duo where it can be displayed in the Push notification in the Duo mobile application. This provides the user more context for which application is requesting a push notification.

Duo Security Integration Kit 3.0.3 - December 2022

- Fixed a defect in adapter when PingFederate is deployed under non-default context path.

Duo Security Integration Kit 3.0.2 - September 2021

- Verified that the Duo Security IdP Adapter is compliant with US Federal Information Processing Standards (FIPS). For usage, see Integrating with Bouncy Castle FIPS provider in the PingFederate documentation.

Duo Security Integration Kit 3.0.1 - March 2021

- Fixed an issue that caused transactions to fail when many users were signing into Duo Security at the same time.

Duo Security Integration Kit 3.0 - October 2020

- Added support for the new Duo Universal Prompt multi-factor authentication (MFA) flow, including:
  - OpenID Connect authentication protocol
  - Duo Universal Prompt
  - New endpoints for checking API availability, authorizing users, and getting authentication results
- Added the ability to map any Duo Universal Prompt API response attribute to an attribute in the PingFederate authentication policy.
- Added support for the PingFederate authentication API.
- Added support for the JavaScript Widget for the PingFederate Authentication API.
- Added a setting for API connection and request timeout.
- Added settings to override the PingFederate system-default proxy settings.

Copyright © 2025 Ping Identity Corporation

- Improved security by disabling device registration during password reset flows.
- Removed the template and JavaScript files that were needed in previous versions.

Duo Security Integration Kit 2.2.2 - January 2020

- Fixed an issue that allowed sessions to exceed the `max_age` parameter in the authentication request.
- Improved input validation for the **Session Timeout** field in the adapter configuration.

Duo Security Integration Kit 2.2.1 - July 2017

- Fixed an issue using the Duo adapter with **Failmode** set to **safe**.
- Upgraded to v2 of the Duo Web SDK.

Duo Security Integration Kit 2.2 - September 2016

- Added **Failmode** option, allowing users to bypass second factor authentication in the event that the Duo service is down.

Duo Security Integration Kit 2.1 - April 2016

- Added clustering support without the use of sticky sessions.
- Added option to configure login template for each adapter instance.

Duo Security Integration Kit 2.0 - September 2015

- Initial release.
- Added support for SSO.

Known issues and limitations

The following are known issues or limitations for the Duo Security Integration Kit.

Known issues

There are no known issues.

Known limitations

There are no known limitations.

Download manifest

The following files are included in the Duo Security Integration Kit `.zip` archive:

- `Legal.pdf` – copyright and license information

Copyright © 2025 Ping Identity Corporation

- `dist/pingfederate/server/default` – contains the integration files
  - `deploy` – contains the Java libraries
- `lib/pf-authn-api-sdk-<version>.jar` – a JAR file that contains the PingFederate Authentication API SDK

Duo Security Integration Kit 2.2

The Duo Security Integration Kit allows PingFederate to use the Duo Security service for multi-factor authentication (MFA). This sign-on experience uses an iFrame and works with the Duo Auth API.

> **Note**
>
> This documentation is for Duo Security Integration Kit 2.2 and earlier. For later versions of the integration kit, see the latest Duo Security Integration Kit 3.x documentation.

Components

- Duo Security IdP Adapter
  - Allows PingFederate to trigger a multi-factor authentication (MFA) challenge during sign on.
  - Allows for MFA when single sign-on (SSO) is intiated at the identity provider (IdP) or service provider (SP).
- Template and script files
  - When a user signs on through PingFederate and the authentication policy triggers the Duo Security Adapter, these files present the Duo Security authentication page.

Intended audience

This document is intended for PingFederate administrators.

If you need help during the setup process, see the following resources:

- PingFederate in the Duo Security documentation
- The following sections of the PingFederate documentation:
  - Managing IdP adapters
  - Authentication policies

System requirements

- PingFederate 8.0 or later
- A Duo Security administrator account

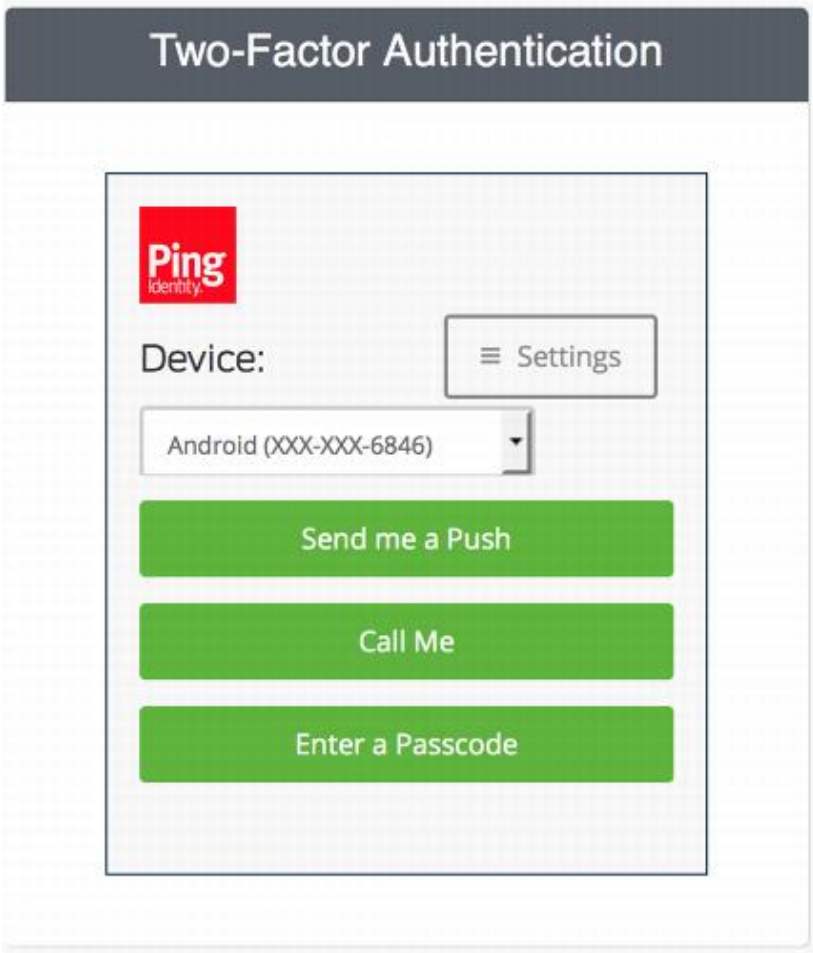
Copyright © 2025 Ping Identity Corporation

Overview of the SSO flow

With the Duo Security Integration Kit, PingFederate includes Duo Security in the sign-on flow.

The following figure illustrates a single sign-on (SSO) scenario in which PingFederate authenticates users to an SP application using PingOne MFA.

1. The user initiates SSO with PingFederate and completes the first-factor authentication step, such as an HTML Form Adapter instance.
2. PingFederate contacts Duo Security and provides the user identifier.
3. Duo Security provides the user's MFA challenge options.
4. The Duo Security IdP Adapter presents the authentication challenge options in the browser.



5. Depending on the authentication method, one of the following occurs:
    - For push notification, Duo Security sends a push notification to the user's mobile app. PingFederate polls the API until Duo Security provides the authentication result.
    - For call authentication, Duo Security sends the one-time passcode (OTP) to the user by phone. In the browser, PingFederate shows a form requesting the OTP. The user enters the OTP in the form.
    - For passcode authentication, the user enters a passcode in the form.

Copyright © 2025 Ping Identity Corporation

6. If the user authenticates successfully, PingFederate provides access to the requested resource. Otherwise, it shows the user an optional page with the reason authentication failed.

Setup

Deploying the integration files

To get started with the integration, deploy the Duo Security Integration Kit files to your PingFederate directory.

About this task

> **Note**
>
> If you operate PingFederate in a cluster, the following steps refer to the console node.

Steps

1. Download the Duo Security Integration Kit `.zip` archive from the Ping Identity Integration Directory.
2. Stop PingFederate.
3. If you are upgrading an existing deployment, delete `pf-duo-security-adapter-<version>.jar` from `<pf_install>/pingfederate/server/default/deploy`.
4. Extract the `.zip` archive and merge the contents of the `dist` directory with your `<pf_install>/pingfederate` directory.
5. Start PingFederate.
6. If you operate PingFederate in a cluster, repeat steps 2 - 5 for each engine node.

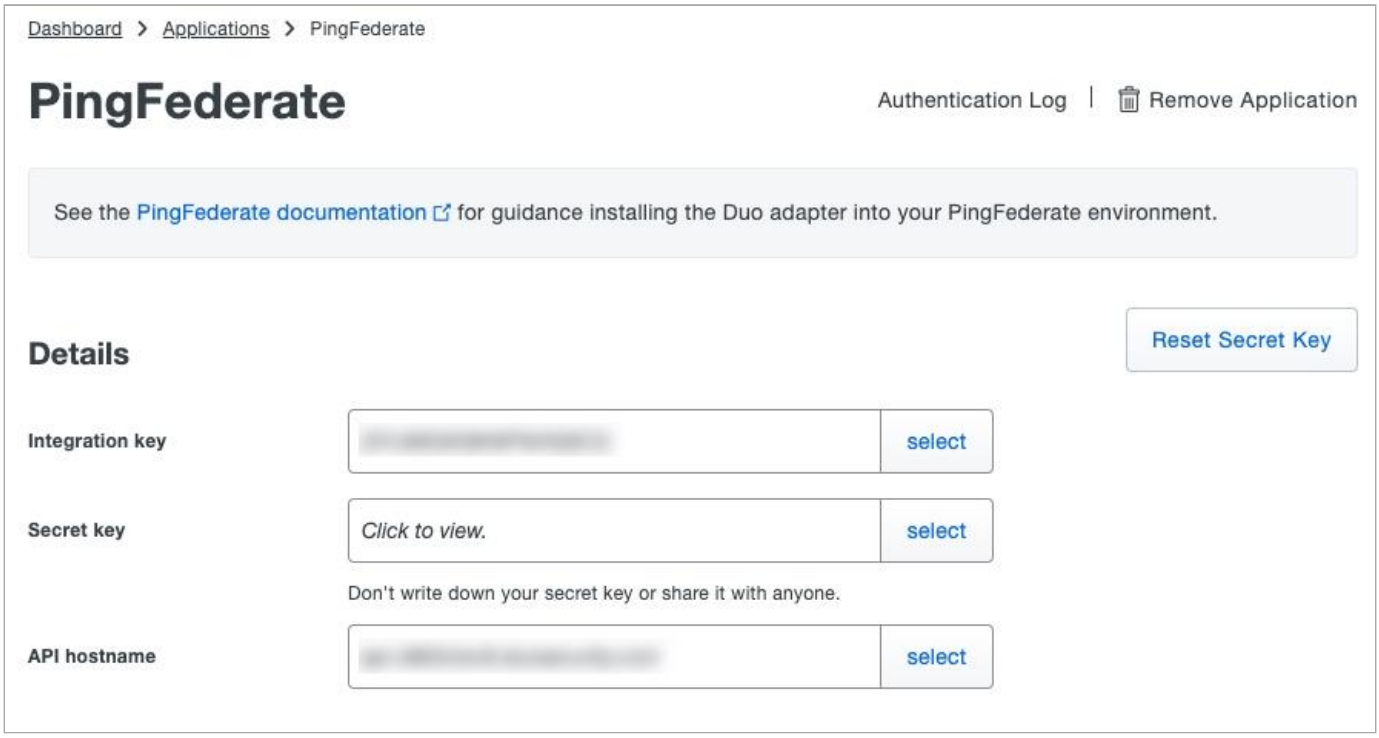
Getting your Duo Security keys and API hostname

To allow PingFederate to communicate with Duo Security, add it as a protected application.

Steps

1. Log into Duo Security as an administrator.
2. Go to **Applications > Protect an Application**. For **PingFederate**, click **Protect this Application.**
3. Note the **Client ID**, **Client Secret**, and **API hostname**. You will use these in Configuring an adapter instance.

Copyright © 2025 Ping Identity Corporation



Configuring an adapter instance

Configure the Duo Security IdP Adapter to determine how PingFederate communicates with Duo Security.

Steps

1. In the PingFederate administrative console, create a new IdP adapter instance:

   Choose from:

   - For PingFederate 10.1 or later: go to **Authentication > Integration > IdP Adapters**. Click **Create New Instance**.
   - For PingFederate 10.0 or earlier: go to **Identity Provider > Adapters**. Click **Create New Instance**.

2. On the **Type** tab, set the basic adapter instance attributes.

   1. In the **Instance Name** field, enter a name for the adapter instance.
   2. In the **Instance ID** field, enter a unique identifier for the adapter instance.
   3. From the **Type** list, select **Duo Security IdP Adapter**. Click **Next**.

3. On the **IdP Adapter** tab, configure the adapter instance by referring to Duo Security Adapter settings reference. Click **Next**.
4. On the **Actions** tab, test your connection to Duo Security. Resolve any issues that are reported, and then click **Next**.
5. On the **Extended Contract** tab, add any attributes that you want to include in the contract. Click **Next**.
6. On the **Adapter Attributes** tab, set pseudonym and masking options as shown in Set pseudonym and masking options in the PingFederate documentation. Click **Next**.

Copyright © 2025 Ping Identity Corporation

7. On the **Adapter Contract Mapping** tab, configure the contract fulfillment details for the adapter as shown in Define the IdP adapter contract in the PingFederate documentation. Click **Next**.
8. On the **Summary** tab, check and save your configuration:

   Choose from:

   - For PingFederate 10.1 or later: click **Save**.
   - For PingFederate 10.0 or earlier: click **Done**. On the **Manage IdP Adapter Instances** tab, click **Save**.

Duo Security Adapter settings reference

Field descriptions for the Duo Security IdP Adapter configuration screen.

Field	Description
Integration Key	The integration key that you noted in Getting your Duo Security keys and API hostname.
Secret Key	The secret key that you noted in Getting your Duo Security keys and API hostname.
API Host	The API hostname that you noted in Getting your Duo Security keys and API hostname.
Session State	Determines whether this Duo Security Adapter instance maintains adapter sessions and shares adapter sessions with other Duo Security Adapter instances. ***Globally*** Adapter sessions from this Duo Security Adapter instance are shared among other Duo Security Adapter instances that have **Session State** set to **Globally**. ***None*** This Duo Security Adapter does not maintain adapter sessions for this Duo Security Adapter instance. **Note** If you intend to enable PingFederate authentication sessions globally or individually for this adapter instance, select **None**. You can find more information about PingFederate authentication sessions in Sessions in the PingFederate documentation. The default selection is **None**.
Session Timeout	The amount of time in minutes of inactivity that a session is valid. A value of 0 disables the timeout. Applies only when **Session State** is set to **Globally**. The default value is 60.
Login Template	The filename of the Duo Security login template. You can use this to replace the provided template with your own. The custom template must be in `<pf_install>/pingfederate/default/conf/template`. The default value is `duo.form.login.template.html`.

Copyright © 2025 Ping Identity Corporation

Field	Description
Failmode	When Duo Security is unavailable or an error occurs, this setting determines whether the user's sign-on attempt should fail or continue. Select **secure** to deny all sign-on attempts when Duo Security is unavailable. Select **safe** to allow all sign-on attempts when Duo Security is unavailable. The default selection is **secure**.

Adding Duo Security to your authentication policy

By modifying your PingFederate authentication policy to include the Duo Security IdP Adapter, you can challenge users to complete a multi-factor authentication (MFA) step.

About this task

These steps are designed to help you add to an existing authentication policy. You can find general information about configuring authentication policies in Authentication policies in the PingFederate documentation.

Steps

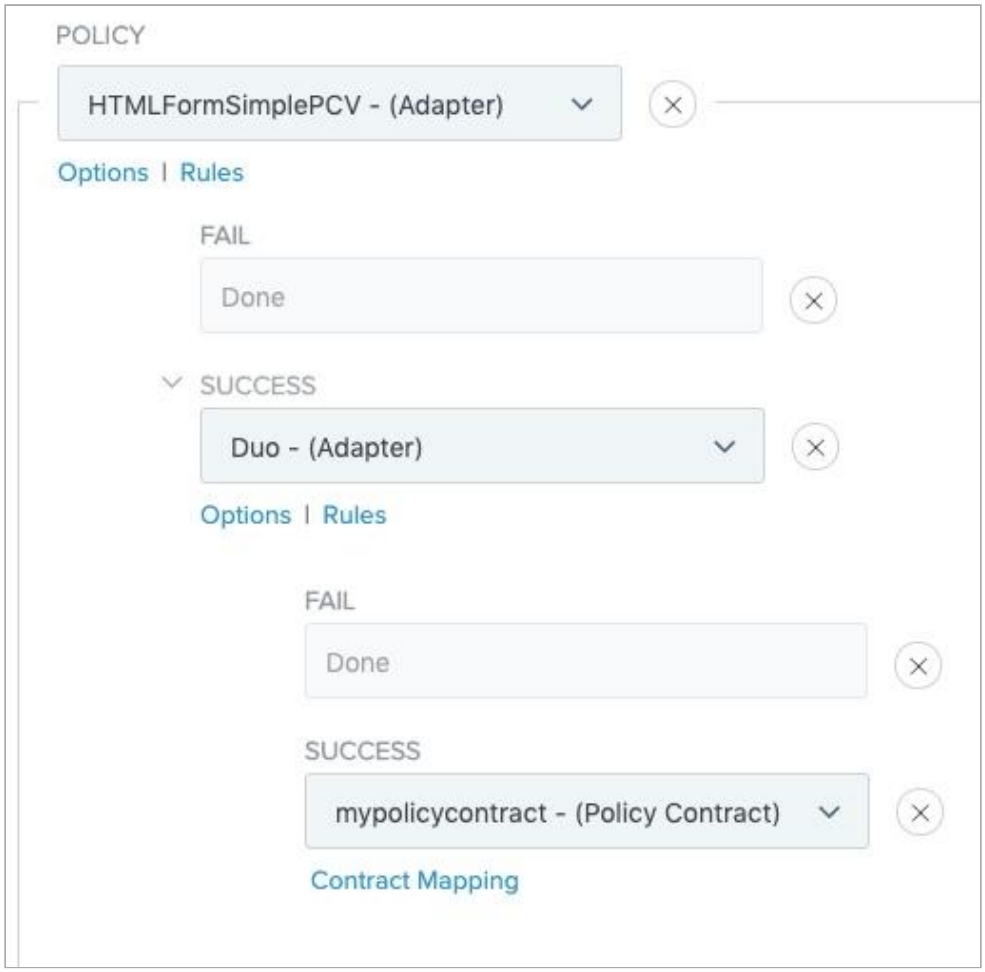
1. In the PingFederate administrative console, go to the **Policies** tab.

   Choose from:

   - For PingFederate 10.1 or later: go to **Authentication > Policies > Policies**.
   - For PingFederate 10.0 or earlier: go to **Identity Provider > Authentication Policies > Policies**.

2. Select the **IdP Authentication Policies** check box.
3. Open an existing authentication policy, or click **Add Policy**. Learn more in Defining authentication policies in the PingFederate documentation.
4. In the **Policy** area, from the **Select** list, select a Duo Security IdP Adapter instance.
5. Map the Duo Security user ID or username into the Duo Security IdP Adapter instance.
   1. Under the Duo Security IdP Adapter instance, click **Options**.
   2. On the **Options** dialog, from the **Source** list, select a previous authentication source that collects the Duo Security user ID or username.
   3. From the **Attribute** list, select the user ID. Click **Done**.
6. Configure each of the authentication paths.

Copyright © 2025 Ping Identity Corporation



7. Click **Done**.
8. In the **Policies** window, click **Save**.

Troubleshooting

Enabling debug logging

To help with troubleshooting or monitoring, you can turn on activity logging for the adapter.

About this task

These steps are optional. For general information about logging, see Enabling debug messages and console logging in the PingFederate documentation.

Steps

1. Open the `<pf_install>/pingfederate/server/default/conf/log4j2.xml` file for editing.
2. If you want to log activity for PingFederate and all adapters, do the following.
    1. Find the following section.

Copyright © 2025 Ping Identity Corporation

```
<AsyncRoot level="INFO" includeLocation="false">
        <!-- <AppenderRef ref="CONSOLE" /> -->
        <AppenderRef ref="FILE" />
</AsyncRoot>
```

2. Change `INFO` to `DEBUG`.

```
<AsyncRoot level="DEBUG" includeLocation="false">
        <!-- <AppenderRef ref="CONSOLE" /> -->
        <AppenderRef ref="FILE" />
</AsyncRoot>
```

3. If you want to see the adapter activity in the console, remove the comment tags.

```
<AsyncRoot level="INFO" includeLocation="false">
        <AppenderRef ref="CONSOLE" />
        <AppenderRef ref="FILE" />
</AsyncRoot>
```

3. If you want to log activity just for the Duo Security IdP Adapter, add the following line.

```
<Logger name="com.pingidentity.clientservices.product.integrationkit.duo" level="DEBUG"/>
```

4. Save the file.

Release notes

Changelog

The following is the change history for the Duo Security Integration Kit 2.2.

> **Note**
>
> This documentation is for Duo Security Integration Kit 2.2 and earlier. You can find later versions of the integration kit in the latest Duo Security Integrations documentation.

Duo Security Integration Kit 2.2.2 - January 2020

- Resolved an issue that allowed sessions to exceed the `max_age` parameter in the authentication request.
- Improved input validation for the **Session Timeout** field in the adapter configuration.
- Defect fixes to resolve a potential security vulnerability.

Copyright © 2025 Ping Identity Corporation

Duo Security Integration Kit 2.2.1 - July 2017

- Resolved issue using the Duo adapter with **Failmode** set to **safe**.
- Upgraded to v2 of the Duo Web SDK.

Duo Security Integration Kit 2.2 - September 2016

- Added **Failmode** option, allowing users to bypass second factor authentication in the event that the Duo service is down.

Duo Security Integration Kit 2.1 - April 2016

- Added clustering support without the use of sticky sessions.
- Added option to configure login template for each adapter instance.

Duo Security Integration Kit 2.0 - September 2015

- Initial release.
- Added support for SSO.

Known issues and limitations

The following are known issues or limitations for the Duo Security Integration Kit.

Known issues

There are no known issues.

Known limitations

There are no known limitations.

Download manifest

The following files are included in the Duo Security Integration Kit `.zip` archive:

- `ReadMeFirst.pdf` – contains links to this online documentation.
- `legal` – a directory that contains this document:
  - `Legal.pdf` – copyright and license information.
- `dist` – a directory that contains the Java libraries needed to run the adapter:
  - `pf-duo-security-adapter-<version>.jar` – JAR file that contains the Duo Security Adapter.
  - `duo-web/js` – a directory that contains a script that works with the Duo Security authentication page.
    - `Duo-Web-v2.min.js` – a script that communicates with the Duo Security API.

Copyright © 2025 Ping Identity Corporation

- `conf/template` – a directory that contains the HTML template and script that send device data to the Duo Security API.
    - `duo.adapter.template.html` – a page that presents the Duo Security authentication options during sign on.

Copyright © 2025 Ping Identity Corporation

Dynamics CRM Integration Guide

The purpose of this document is to guide a PingFederate and Microsoft Dynamics CRM 2011 (Dynamics CRM) administrator through the configuration of both products so that federated users can log on to Dynamics CRM using claims-based authentication. The document outlines how to configure an SP connection compatible with Dynamics CRM within PingFederate and how to configure Dynamics CRM to consume the federation metadata provided by the SP connection.

> **Note**
>
> Knowledge of Dynamics CRM is assumed as well as familiarity with PingFederate.

Prerequisites

The following must be installed and configured in order to complete the configuration:

- Install (in your JDK) the Unlimited Strength Java TM Cryptographic Extension (JCE) Policy Files in order to use the AES-256 encryption algorithm used by Dynamics CRM.
- Install PingFederate 6.11 or higher.

  > **Note**
  >
  > If you need to support active clients, such as native desktop applications, for use with Dynamics CRM, ensure that PingFederate is installed with a license that enables the WS-Trust Security Token Service (STS).

- Obtain two certificates for use by PingFederate and Dynamics CRM to establish trust between the two services. The first certificate is an encryption certificate that encrypts data between PingFederate and Dynamics CRM. The second certificate is a signing certificate used to digitally sign the SAML assertions returned from PingFederate.
- Configure a connection to the Active Directory LDAP data store (see Configuring an LDAP Connection).
- Configure the HTML Form IdP Adapter with an LDAP Username Password Credential Validator (see Configuring an HTML Form Adapter instance).
- If you are configuring the connection for active federation, install and configure the Username Token Translator 1.1 (or higher) to use LDAP bind as the processing scheme with the Active Directory LDAP data store created above. Contact Ping Identity support for information about this plug-in.

  > **Note**
  >
  > For PingFederate 7.2 or higher, Username Token Processor is part of the product and does not require a separate download or installation.

- In order for the connector to work properly you must configure PingFederate to "Omit Line Breaks in Digital Signatures" by adding the following java startup option to your run.sh, run.bat and/or PingFederateService.conf file:

  ```
  -Dorg.apache.xml.security.ignoreLineBreaks=true
  ```

- For this release, Microsoft Dynamics CRM 2011 for active login with Microsoft Office TM Outlook versions 2007 or later were tested.

Copyright © 2025 Ping Identity Corporation

PingFederate configuration

Configure PingFederate to include an SP Connection (see SP connection management).

Create an SP connection

Steps

1. On the **Connection Type** screen, select the **Browser SSO Profiles** check box and select **WS-Federation** as the Protocol. If you are configuring the connection for active federation (for native client cases such as the Dynamics CRM plug-in for Outlook), select the **WS-Trust STS** check box. You must select **SAML 1.1** as the Default Token Type.
2. On the **General Info** screen, enter the CRM Web site in the **Partner's Realm** field.

   For example, `https://ping.crm.com/default.aspx`
3. Enter a value in the **Connection Name** field.
4. On the **Browser SSO** screen, click **Configure Browser SSO**.
5. On the **Assertion Lifetime** screen, update the lifetime of the assertion as needed. Note that Dynamics CRM uses the SAML token to determine the session lifetime; you may increase the **Minutes After** value to extend the CRM session lifetime beyond 5 minutes (the default value).
6. On the **Assertion Creation** screen, click **Configure Assertion Creation**.
7. On the **Identity Mapping** screen, select **User Principal Name**.
8. On the **Attribute Contract** screen, extend the contract by adding `upn` and selecting **http://schemas.xmlsoap.org/ws/2005/05/identity/claims** as the attribute name format.

Map an IdP adapter

Steps

1. Click **Map New Adapter Instance** and select the HTML Form IdP Adapter as the Adapter Instance.
2. On the **Assertion Mapping** screen, select the option button to retrieve additional attributes from a data store including options to use alternate data stores and/or a failsafe mapping.
3. Click **Add Attribute Source** and configure the LDAP data source, adding `userPrincipalName` as an additional attribute and including a filter value such as `sAMAccountName=${username}`.
4. On the **Attribute Contract Fulfillment** screen, select **Text** as the Source for SAML_SUBJECT and enter an unused value. Select **LDAP** as the Source for upn and select **userPrincipalName** as the value.
5. On the **Failsafe Attribute Source** screen, select the **Abort the SSO Transaction** option.

Copyright © 2025 Ping Identity Corporation

Configure protocol settings

Steps

1. Click **Configure Protocol Settings** on the Protocol Settings screen
2. Enter the CRM Web site as the Endpoint URL.

   For example:

   ```
   https://ping.crm.com/default.aspx
   ```

Configure WS-Trust STS

About this task

> **Note**
>
> If you are not using active federation (for native client cases such as the Dynamics CRM plug-in for Outlook), then you do not need to configure WS-Trust STS settings: skip ahead to Configure credentials. If the task bar is showing `WS-Trust STS`, return to the **Connection Type** screen and clear the WS-Trust check box. Then go to the **Credentials** screen.

Steps

1. Click **Configure WS-Trust STS**.
2. Enter the base URL of your Dynamics CRM Web site on the **Protocols Settings** screen.

   For example:

   ```
   https://ping.crm.com
   ```

3. Select the **Generate Key for SAML Holder of Key Subject Confirmation Method** check box.
4. When configuring token creation, extend the attribute contract on the **Attribute Contract** screen by adding `upn` and selecting `http://schemas.xmlsoap.org/ws/2005/05/identity/claims` as the attribute name format.

Configure IdP token processor mapping

Steps

1. Click **Map New Token Processor Instance** and select a configured Username Token Processor as the Token Processor Instance.
2. On the Attribute Retrieval screen, select the option to retrieve additional attributes from data stores to fulfill the attribute contract.

Copyright © 2025 Ping Identity Corporation

3. In the **Attribute Sources & User Lookup** screen, configure the LDAP data store that will return the upn attribute for the corresponding user, adding userPrincipalName as an additional attribute and including a filter value such as `sAMAccountName=${username}`.
4. On the Attribute Contract Fulfillment screen, select **Text** as the Source for SAML_SUBJECT and enter an unused value. Select **LDAP** as the Source for upn and select **userPrincipalName** as the value.
5. Configure issuance criteria, if necessary.

Configure credentials

Steps

1. Click **Configure Credentials**.
2. Select the signing certificate you want to use.
3. Select **RSA SHA1** or **RSA SHA256** as the signing algorithm on the **Digital Signature Settings** screen.

> **Note**
>
> This certificate can be self-signed and can be exported for use by the Dynamics CRM server.

4. If the Dynamics CRM server is configured for token signature validation, export the signing certificate.

> **Note**
>
> You will import the certificate into the Dynamics CRM server (see Dynamics CRM configuration).

Share the signing certificate with the Dynamics CRM server

Steps

1. Click **Manage Certificates**.
2. Click **Export** on the **Manage Digital Signing Certificates** screen for the signing certificate you want to export.
3. On the Export & Summary screen, click **Export**. Transfer the certificate to the Dynamics CRM server and import it into the appropriate Certificate store using the Microsoft Management Console Certificates Snap-in.

> **Note**
>
> See Enabling ADFS 2.0 Token Signing for more information on token signature validation (technet.microsoft.com/en-us/library/gg188574.aspx).

4. Click **Done** until you return to the Digital Signature Settings screen.

Select WS-Trust encryption algorithm

About this task

Copyright © 2025 Ping Identity Corporation

> **Note**
>
> This step is only necessary if WS-Trust is being used.

Steps

1. On the **Select XML Encryption Certificate** screen, select **AES-256** as the Block Encryption Algorithm.
2. Select the encryption certificate for Dynamics CRM claims encryption from the drop-down list.

   This is the same encryption certificate used in Dynamics CRM configuration.

   > **Note**
   >
   > If AES-256 is unavailable, make sure the high-level encryption libraries were imported correctly.

Save and activate the connection

Steps

1. Save and activate the connection on the **Activation & Summary** screen.

Dynamics CRM configuration

About this task

Use the following steps to configure Dynamics CRM to consume the federation metadata provided by the PingFederate SP connection.

Steps

1. Access the Dynamics CRM server.
2. If Dynamics CRM is configured for token signature validation, run mmc.exe and attach the Certificates (Local Computer) Snap-in.

   Import the signature verification certificate used in PingFederate or the certificate's CA certificate into the appropriate certificate store. See Enabling ADFS 2.0 Token Signing for more information on token signature validation.
3. If WS-Trust STS was configured for the CRM connection in PingFederate, import the encryption certificate used in PingFederate (see Select WS-Trust encryption algorithm) along with the certificate's private key into the Dynamics CRM server's personal certificate store. The Dynamics CRM server searches this store when configuring claims-based authentication.
4. On the Dynamics CRM server, run the Microsoft Dynamics CRM Deployment Manager.
5. Select Configure Claims-based Authentication and click **Next**.
6. Enter the following URL for the Federation metadata URL and click **Next**:

Copyright © 2025 Ping Identity Corporation

```
https://<pf_host>:<pf_port>/pf/federation_metadata.ping?
PartnerSpId=<SPConnectionID>&forceIssuedTokenPolicy
```

where:

- <pf_host> is the host name or IP address where PingFederate is running.
- <pf_port> is the port number for PingFederate.
- <SPConnectionID> is the ID for the PingFederate SP Connection you configured above – for example:

```
https://ping.crm.com/default.aspx
```

> **Note**
>
> If an error appears stating that the Federation URL is unavailable, add PingFederate's server certificate (signed by the domain controller) to the Dynamics CRM server to establish trust with PingFederate's SSL server certificate.

7. When prompted for the encryption certificate, use the same certificate shared with PingFederate (see Select WS-Trust encryption algorithm).
8. Save the configuration and run iisreset from the command line so the Dynamics CRM server recognizes the changes.

Change signing certificates

About this task

The Dynamics CRM server may cache the signing certificate, which can cause trust errors when changing the PingFederate signing certificate. Reconfiguring claims-based authentication is recommended when you change the PingFederate signing certificate. After changing the signing certificate in the Dynamics CRM connection, do the following in the Dynamics CRM server:

Steps

1. In the Microsoft Dynamics CRM Deployment Manager, disable claims-based authentication.
2. Run `iisreset` from the command line on the Dynamics CRM Web server.
3. In the Microsoft Dynamics CRM Deployment Manager, reconfigure claims-based authentication.
4. In the Microsoft Dynamics CRM Deployment Manager, if previously enabled, reconfigure Internet-Facing Deployment.
5. Run `iisreset` from the command line on the Dynamics CRM Web server.

Known issues

- If you configure Internet-Facing Deployment in the Dynamics CRM Deployment Manager, you can ignore the following warning and still successfully complete the configuration: "The Discovery Web Service could not be accessed. The domain is unavailable or does not exist."

Copyright © 2025 Ping Identity Corporation

- Active login for Dynamics CRM Outlook clients is supported only when the claims provider for the Dynamics CRM Outlook client is the same claims provider as the Dynamics CRM server. This means all Outlook client computers need to be configured to login to the SAML claims provider (which is the default behavior). In other words, the HKEY_LOCAL_MACHINE\Software\Policies\Microsoft\MSCRMClient\HomeRealmUrl registry key on the Outlook client computer must not be set. See Configure Microsoft Dynamics CRM for Outlook to use Claims-based Authentication for more information (technet.microsoft.com/en-us/library/gg188615.aspx).

Copyright © 2025 Ping Identity Corporation

Egnyte Provisioner

The PingFederate Egnyte Provisioner enables an enterprise to provision users to Egnyte. A quick connection template is also included to simplify the configuration of single sign-on (SSO).

Features

- Browser-based SP and IdP-initiated SSO
- Includes support for user life cycle management (including creates, updates, and disables).

Intended audience

This document is intended for PingFederate administrators.

If you need help during the setup process, see the following sections of the PingFederate documentation:

- SP connection management
- Identity provider SSO configuration
- Datastores
- Configuring outbound provisioning
- Configuring outbound provisioning settings

System requirements

- PingFederate 7.3 or later with Java 8
- Might require the following endpoints to be whitelisted on the firewall to allow outbound connections:
  - The URL used to access your organization's Egnyte. For example, https://*myCompany*.egnyte.com/

For additional details on Egnyte, see the Egnyte website.

User management

The Egnyte Provisioner synchronizes users from your datastore to Egnyte. The behavior of each provisioning capability is described below.

You can configure these capabilities in the Configure provisioning step of the setup process.

Synchronizing existing users

PingFederate synchronizes users based on the `Username` attribute in Egnyte. If a user already exists in your datastore and Egnyte, mapping this attribute correctly links the two records together.

For example:

- In Egnyte, Janet's `Username` is `bjensen@example.com`.

Copyright © 2025 Ping Identity Corporation

- In your datastore, Janet's `mail` is `bjensen@example.com`.
- On the **Attribute Mapping** tab of your provisioning connection configuration, you map the `Username` attribute to `mail`.
- When the provisioning connector runs, the datastore user is provisioned with a `Username` of `bjensen@example.com`. That matches Janet's existing `Username` in Egnyte, so her information in the datastore is synchronized to her Egnyte account.

User provisioning

PingFederate provisions users when one of the following happens:

- A user is added to the datastore group or filter that is targeted by the provisioning connector.

The **Source Location** tab of your provisioning connection configuration defines which users PingFederate targets for provisioning.

User updates

PingFederate updates users when a user attribute changes in your datastore.

The **Attribute Mapping** tab of your provisioning connection configuration defines which attributes PingFederate monitors for changes.

User deprovisioning

PingFederate deprovisions users when one of the following happens:

- A user is deleted from the user store.
- A user is disabled in the user store.
- A user is removed from the datastore group or filter that is targeted by the provisioning connector.

Setup

Obtain API key

About this task

The Egnyte Connector's outbound provisioning functionality is built using Egnyte's REST API, which requires an authorized OAuth 2.0 access token. To generate your access token in Obtain access token, you will first need to obtain an API key.

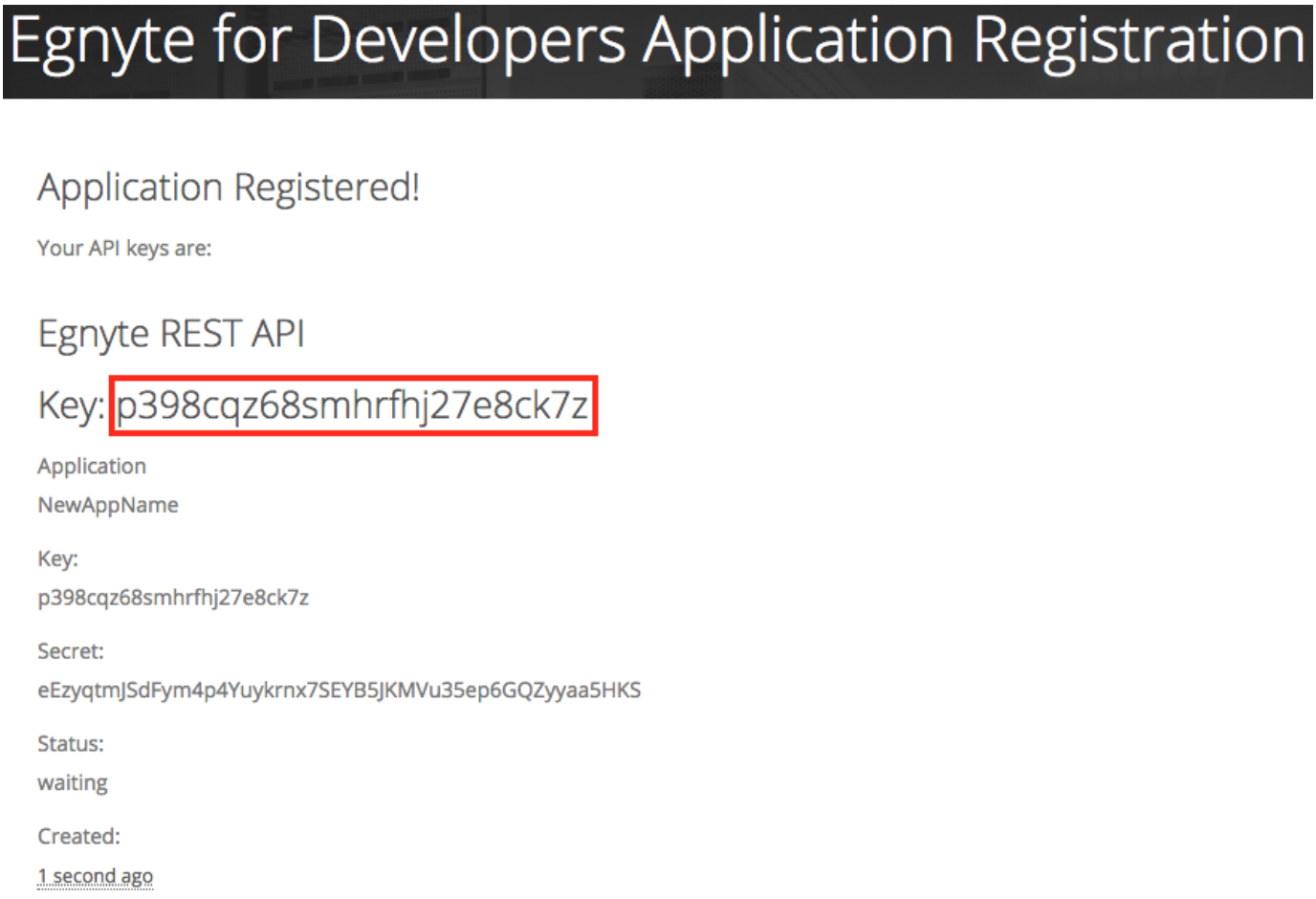
To obtain an API key, you will need to have a registered account on Egnyte's developer site. The developer site will be used to create an application which will contain the API key. The following steps will be used to create your application and obtain the API key:

Steps

1. Log in to Egnyte's developer site with an administrative developer account.
2. Go to **My Account > Applications > Create a New Application**.

Copyright © 2025 Ping Identity Corporation

3. Complete the form to register your new application with the following options:
    - Enter a name for the application.
    - Set the **Type** to **Internal Application (own company use only)**.
    - Set the **Current User Base** to **New App**.
    - Set the **Platform** to **Other**.
    - Set the **Egnyte domain you will use for testing** to your Egnyte subdomain. This is the subdomain portion of the URL used to access your organization's Egnyte. For example, if the URL used to access Egnyte is https://*myCompany*.egnyte.com/ then the subdomain is *myCompany*.
    - Ensure the **Issue a new key for Egnyte REST API** check box is selected.
    - Select the check box, indicating that you have read and agree to the Egnyte API Terms of Service and click the **Register Application** button.
4. Once your application is registered, note the **Key** for later use in Obtain access token.



> **Note**
>
> After creating your application, Egnyte must approve it before you can use the API key to generate your access token in Obtain access token. For more information, see Frequently Asked Questions.

Copyright © 2025 Ping Identity Corporation

Obtain access token

About this task

Once you have obtained your API key, you are able to use it to generate an access token using cURL.

> **Note**
>
> After creating your application, Egnyte must approve it before you can use the API key to generate your access token. For more information, see Frequently Asked Questions on the Egnyte developers site.

Steps

1. From an application or terminal capable of running cURL commands, run the following cURL command:

```
curl -v --request POST -H "Content-Type: application/x-www-form-urlencoded"
-d grant_type=password -d username=YOUR_USERNAME -d password=YOUR_PASSWORD
-d client_id=YOUR_API_KEY https://YOUR_SUBDOMAIN.egnyte.com/puboauth/token
```

   where:

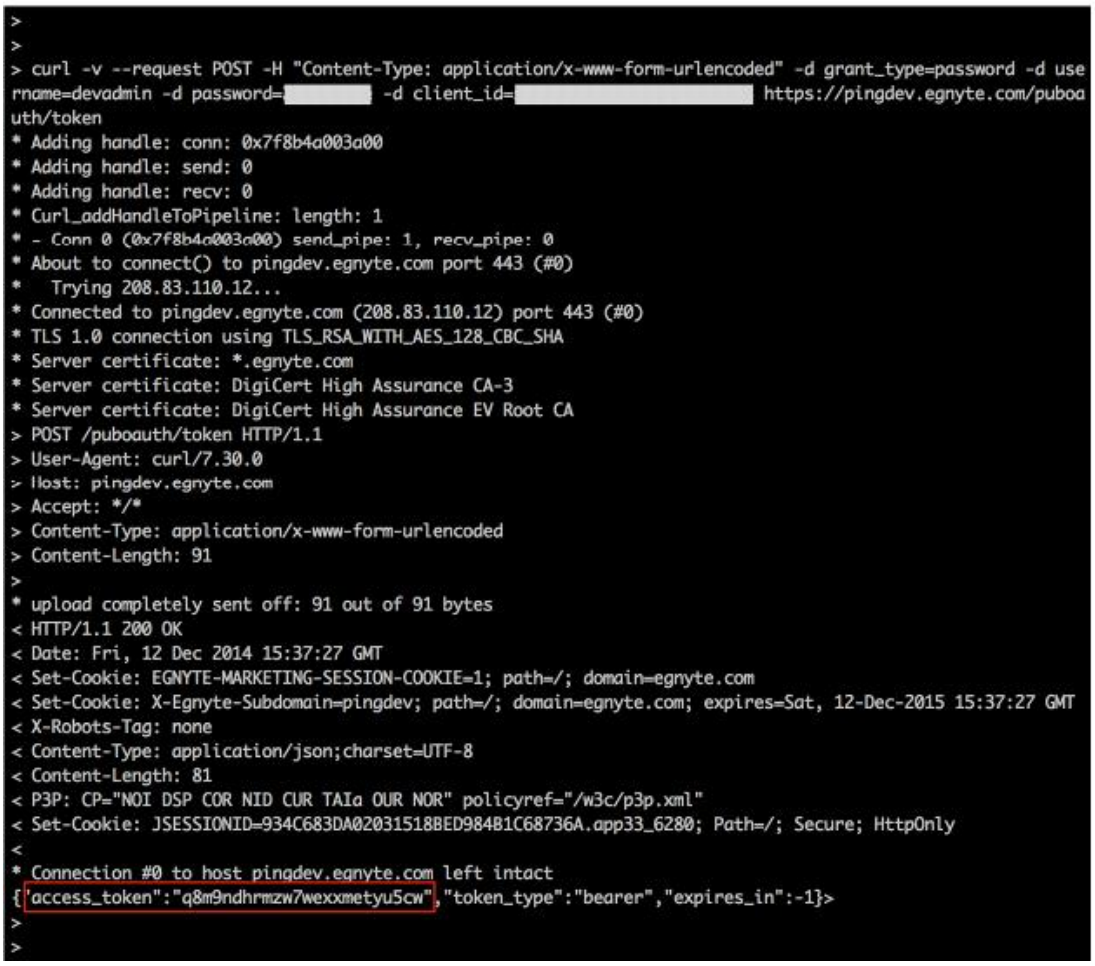
   - *YOUR_USERNAME* is the username of an administrative user on the Egnyte subdomain you wish to provision users.

     > **Note**
     >
     > The administrative user for you Egnyte subdomain is not the same account that you used to generate the application on Egnyte's developer site.

   - *YOUR_PASSWORD* is the password for the administrative user of your Egnyte subdomain.
   - *YOUR_API_KEY* is the API key identified in Obtain API key.
   - *YOUR_SUBDOMAIN* is the subdomain portion of the URL used to access your organization's Egnyte. For example, if the URL used to access Egnyte is https://*myCompany*.egnyte.com/ then the subdomain is *myCompany*.

Copyright © 2025 Ping Identity Corporation

2. Note the presented access token for later use in Configure provisioning.



Obtain Egnyte SAML 2.0 metadata

About this task

The connector's quick-connection template uses a metadata XML file to assist in configuring many settings in the SP connection. When asked during the connection configuration steps, import the `egnyte-saml-metadata.xml` packaged with this connector.

To prepare your metadata.xml:

Steps

1. Open the `egnyte-saml-metadata.xml` file packaged with the connector in a text editor of your choice.
2. Replace all instances of *SUBDOMAIN* with the subdomain portion of the URL used to access your organization's Egnyte.

   For example, if the URL used to access Egnyte is https://*myCompany*.egnyte.com/ then the *SUBDOMAIN* is *myCompany*.
3. Once you have updated the *SUBDOMAIN*, save the metadata file changes.

Copyright © 2025 Ping Identity Corporation

Install the connector

About this task

This section describes the common steps required to install the PingFederate Egnyte Connector.

Steps

1. Stop the PingFederate server if it is running.
2. Unzip the Egnyte Connector distribution `.zip` archive.
3. Copy the contents of the `dist` directory into the directory: `<pf_install>/pingfederate/server/default/deploy`
4. **Optional:** If you plan to use the connector for outbound provisioning, edit the `run.properties` file located in `<pf_install>/pingfederate/bin`, changing the property `pf.provisioner.mode` to `STANDALONE`. For example: `pf.provisioner.mode=STANDALONE`

   > **Note**
   >
   > For information about using the FAILOVER setting for runtime deployment, see Deploying provisioning failover.

5. Start the PingFederate server.

Enable outbound provisioning

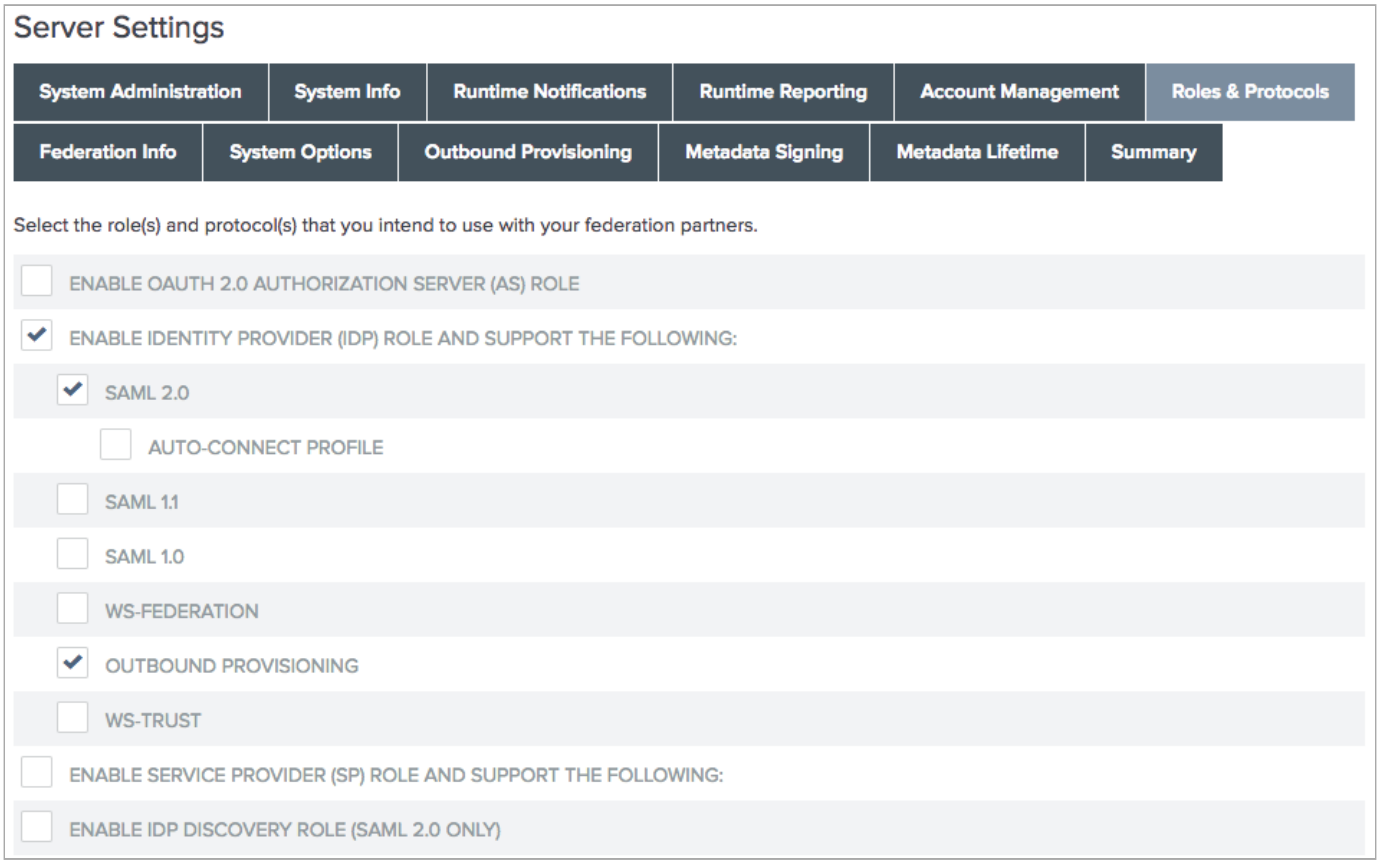
About this task

After enabling outbound provisioning in the `<pf_install>/pingfederate/bin/run.properties` file, you must also activate the outbound provisioning role in the administrative console.

Steps

1. Go to the **Server Configuration > Server Settings > Roles & Protocols** screen.
2. Select the **Outbound Provisioning** check box.

Copyright © 2025 Ping Identity Corporation



> **Note**
>
> Enabling outbound provisioning adds the outbound provisioning screen, requiring the selection of a database to facilitate provisioning. For more information, see Configuring outbound provisioning settings or click **Help** from the configuration screen.

Configure provisioning

About this task

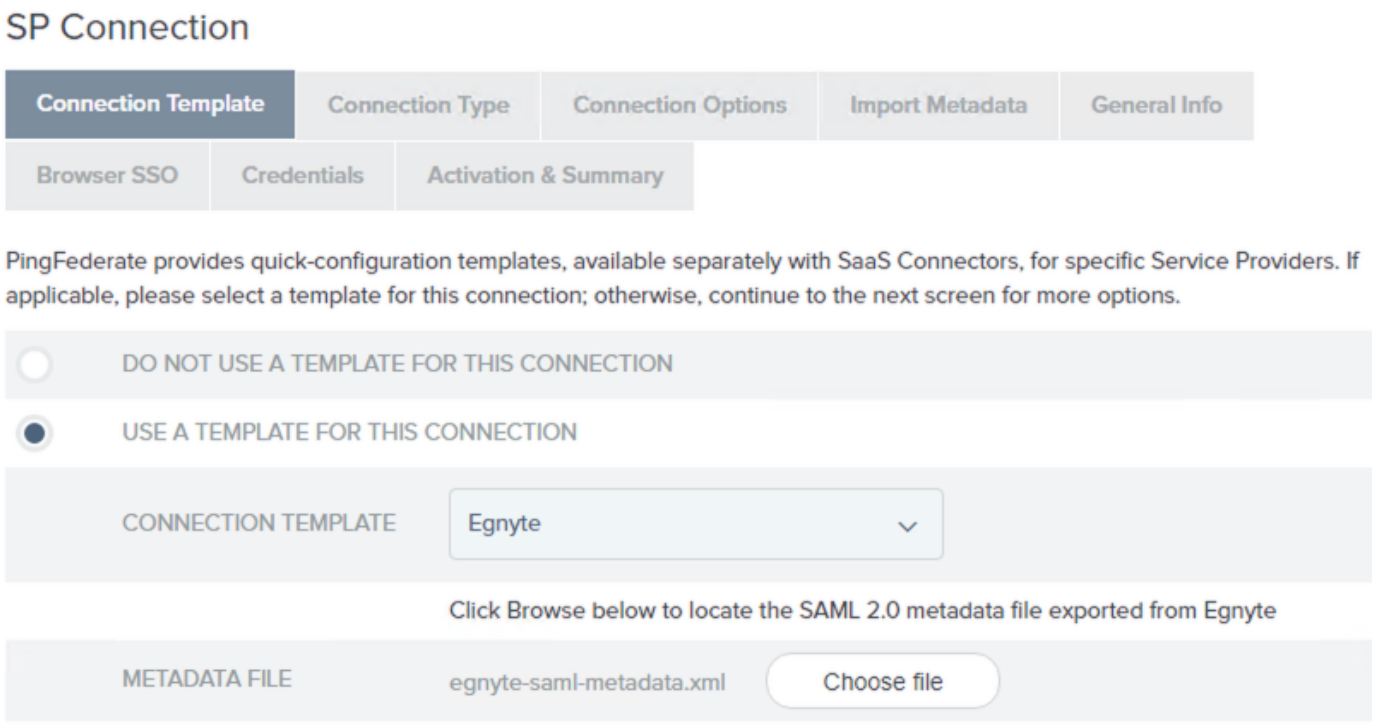
To configure a connection for outbound provisioning to Egnyte, follow the instructions in this section. Outbound provisioning details are managed within an SP connection and may be added to an existing SP connection.

Steps

1. In the PingFederate administrator console, configure the data store that PingFederate will use as the source of user data. For instructions, see Datastores in the PingFederate documentation.
    - When targeting users and groups for provisioning, exclude the user account that you will use to administer users in your connection to Egnyte. This prevents the PingFederate provisioning engine from interfering with the account that provisions users and groups.
2. Create a new SP connection or select an existing SP connection from the **SP Configuration** menu.

Copyright © 2025 Ping Identity Corporation

3. On the **Connection Template** screen, select **Use a template for this connection** and choose **Egnyte** from the **Connection Template** drop-down list. When asked during the connection configuration steps, import the `egnyte-saml-metadata.xml` you prepared earlier in Obtain Egnyte SAML 2.0 metadata.

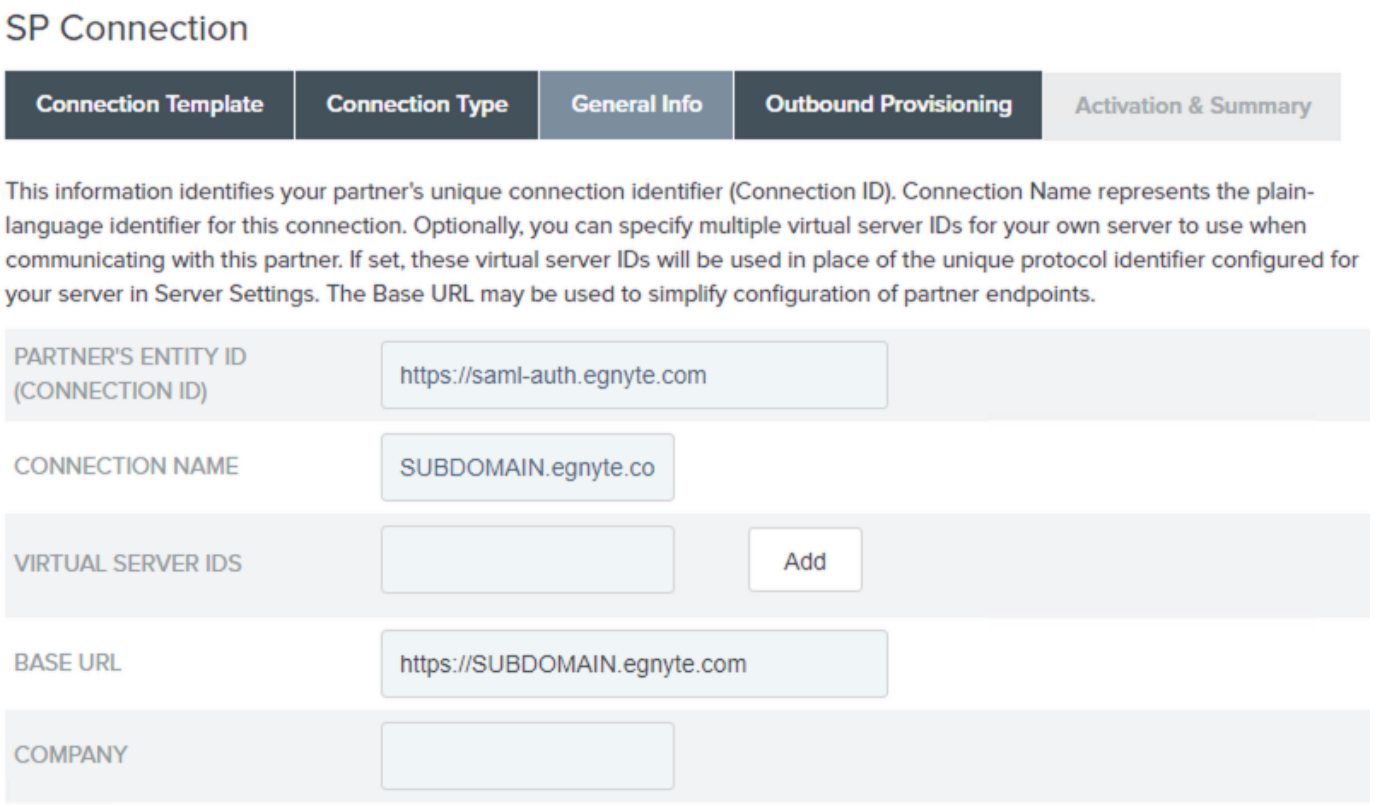


> **Tip**
>
> If this selection is not available, verify the connector installation and restart PingFederate.

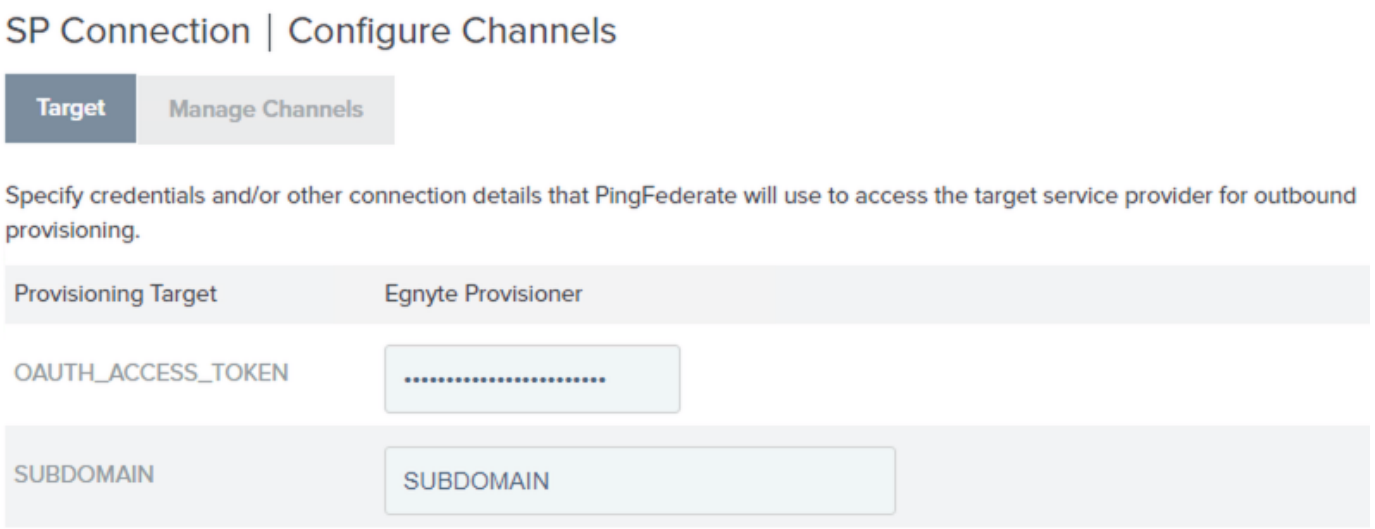
4. On the **Connection Type** screen, ensure the **Outbound Provisioning** check box is selected, and the **Browser SSO Profiles** check box is cleared (if appropriate).

5. On the **General Info** screen, the default values are taken from the metadata file you selected in step 2.

Copyright © 2025 Ping Identity Corporation



6. Follow the connection wizard to configure the connection.
7. On the **Outbound Provisioning** screen, click **Configure Provisioning**.
8. On the **Target** screen, enter the values for each field as required by the Egnyte Connector.



Copyright © 2025 Ping Identity Corporation

Target screen options

Field Name	Description
OAuth_Access_Token	The OAuth access token for the Egnyte account. For more information, see Obtain access token.
Subdomain	The subdomain portion of the URL used to access your organization's Egnyte. For example, if the URL used to access Egnyte is https://*myCompany*.egnyte.com/ then the subdomain is *myCompany*.

9. Click **Next** to continue the provisioning configuration. For more information, see the following sections under Outbound provisioning for IdPs in the PingFederate documentation:
    - Managing channels
    - Specifying channel information
    - Identifying the source datastore
    - Modifying source settings
    - Specifying a source location
    - Mapping attributes
    - Reviewing channel settings

> **Note**
>
> Credentials will be verified when the channel and SP connection is set to Active and provisioning is initiated.

> **Tip**
>
> If you are not ready to complete the provisioning configuration, you can click **Save** and return to the configuration page later. To return to the configuration page, select the connection from **Identity Provider > SP Connections > Manage All**.

Supported attributes reference

The following table consists of the attributes that can be mapped on a user during provisioning. For more information, see the Egnyte User Management API.

Attribute	Description
email	The Egnyte username for the user. This attribute is required.

Copyright © 2025 Ping Identity Corporation

authType	The authentication type for the user. This attribute is required. Available options include: • ad (Active Directory) – Users with their `authType` set to this option will be able to authenticate through Active Directory. This requires your Egnyte domain to be configured for **Active Directory Authentication** under **Settings > Configuration > Security & Authentication**. When setting this option on a user, the user must also have a userPrincipalName set. • sso (SAML SSO) – Users with their `authType` set to this option will be able to authenticate using single sign-on (SSO). This requires your Egnyte domain to be configured for **Single Sign-on Authentication** under **Settings > Configuration > Security & Authentication**. When setting this option on a user, the user must also have an idpUserId set. • egnyte (Internal Egnyte) – Users with their authType set to this option will be able to log in to Egnyte using their Egnyte account credentials.
externalId	An immutable unique identifier for the user. This can be any plain text value identifier. This attribute is required. WARNING: This field is immutable meaning once set, it can never be updated.
familyName	The last name of the user. This attribute is required.
givenName	The first name of the user. This attribute is required.
sendInvite	If set to true when creating a user, an invitation email will be sent if the user is created in active state in the LDAP data store. If the user is disabled in the LDAP data store when they are created, no email is sent. This attribute is required. Available options include: • true • false
userName	The Egnyte username for the user. This attribute is required. **Note** Usernames must start with a letter or digit. Special characters are not supported (with the exception of periods, hyphens, and underscores). **Important** It is not possible to change a user's userName after they are created.
userType	The type of the user. This attribute is required. Acceptable options include: • admin • power • standard

Copyright © 2025 Ping Identity Corporation

idpUserId	Only required if the user's authType is set to 'sso'. This is how the user is identified within the SAML response from an SSO identity provider. For example, the SAML subject 'bjensen' or 'bjensen@example.com'. **Tip** The idpUserId of a user must match the **Default user mapping** configured in Egnyte. For more information, see Configure Egnyte for SSO.
role	The role assigned to the user. Available options include 'default' or other existing custom role names. **Note** This option is only available to users with a userType of 'power'.
userPrincipalName	Only required if the user's authType is set to 'ad'. This field is used to bind child authentication policies to a user when using Active Directory authentication in a multi-domain setup. For example, 'bjensen@example.com'.

Configure SAML SSO

The following section describes the steps for configuring single sign-on (SSO) to Egnyte.

Note

Configuring SSO is optional for outbound provisioning.

- Configure Egnyte for SSO
- Configure PingFederate for SSO

Configure Egnyte for SSO

About this task

To proceed with configuring Egnyte for SSO, you will require the below details from PingFederate.

- Identity provider login URL
  - Example: https://<*pf_hostname*>:<*pf_port*>/idp/SSO.saml2
- Identity provider entity ID
  - This is the SAML 2.0 Entity ID from PingFederate, which can be found on the **Server Configuration > Server Settings > Federation Info** screen. For more information, see Specifying federation information.

Copyright © 2025 Ping Identity Corporation

- Identity provider certificate
    - This is the public signing certificate from PingFederate used to sign the SAML assertion (configured in your SP connection). For more information, see Manage digital signing certificates and decryption keys.

The following section describes the steps to configure SSO in Egnyte. For more information, see Single Sign-On (SSO) Configuration.

> **Note**
>
> For users to sign in through SAML SSO, their authType attribute must be set to 'sso'. The authType attribute may be set when the user is provisioned or by an administrator on the user's profile under **Settings > Configuration > Users & Groups**. For more information, see Supported attributes reference.

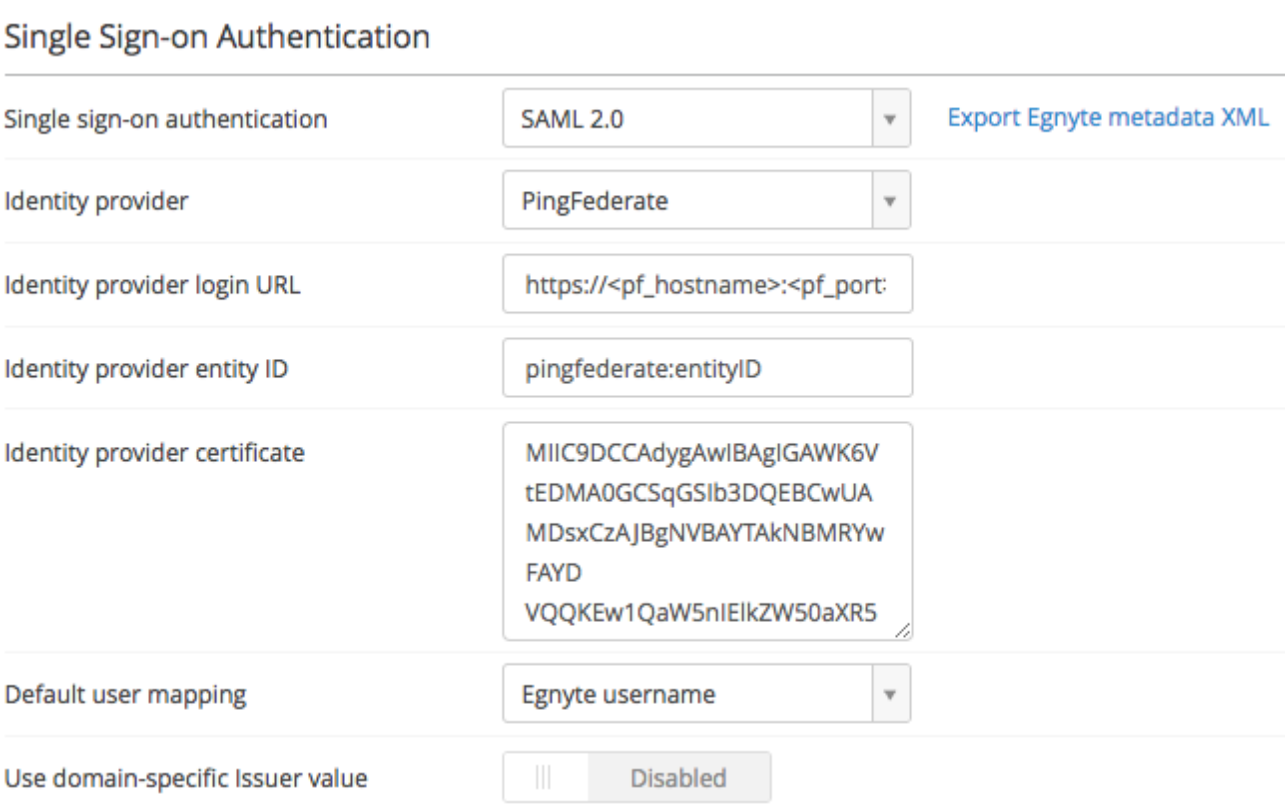
Steps

1. Log in to your Egnyte subdomain as an administrative user for your organization.
2. Navigate to **Settings > Configuration > Security & Authentication**.
3. Under the **Single Sign-on Authentication** section:
    - Select SAML 2.0 as the **Single sign-on authentication** type.
    - Select PingFederate as the **Identity provider**.
    - Set the **Identity provider login URL**, **Identity provider entity ID**, and **Identity provider certificate**.

      > **Tip**
      >
      > The **Identity provider certificate** is the PEM-encoded X.509 certificate from PingFederate. Paste the entire certificate, but remove the BEGIN and END CERTIFICATE lines.

    - Set the **Default user mapping**. The value of the SAML_SUBJECT sent in your SP connection's attribute contract must match this mapping.
    - Optional: Set the **User domain-specific Issuer value** based on your business use.

Copyright © 2025 Ping Identity Corporation



4. Click the **Save** button.

Configure PingFederate for SSO

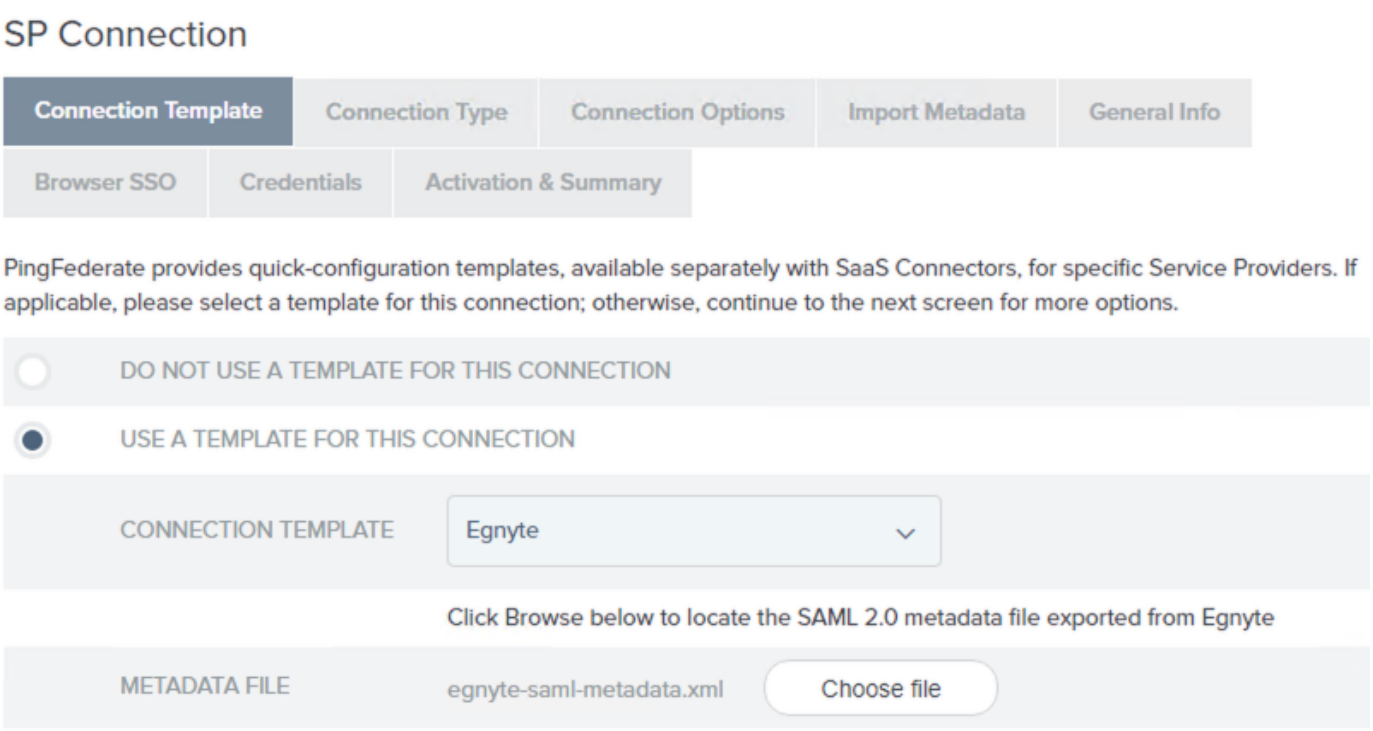
About this task

The following section describes the steps for configuring single sign-on (SSO) to Egnyte. Configuring SAML SSO involves configuring both the PingFederate SP connection and Egnyte SSO screens. NOTE: Configuring SSO is optional for outbound provisioning.

Steps

1. Create a new SP connection or select an existing SP connection from the **SP Configuration** menu.
2. On the **Connection Template** screen, select **Use a template for this connection** and choose **Egnyte** from the **Connection Template** drop-down list. When asked during the connection configuration steps, import the `egnyte-saml-metadata.xml` you prepared earlier in Obtain Egnyte SAML 2.0 metadata.

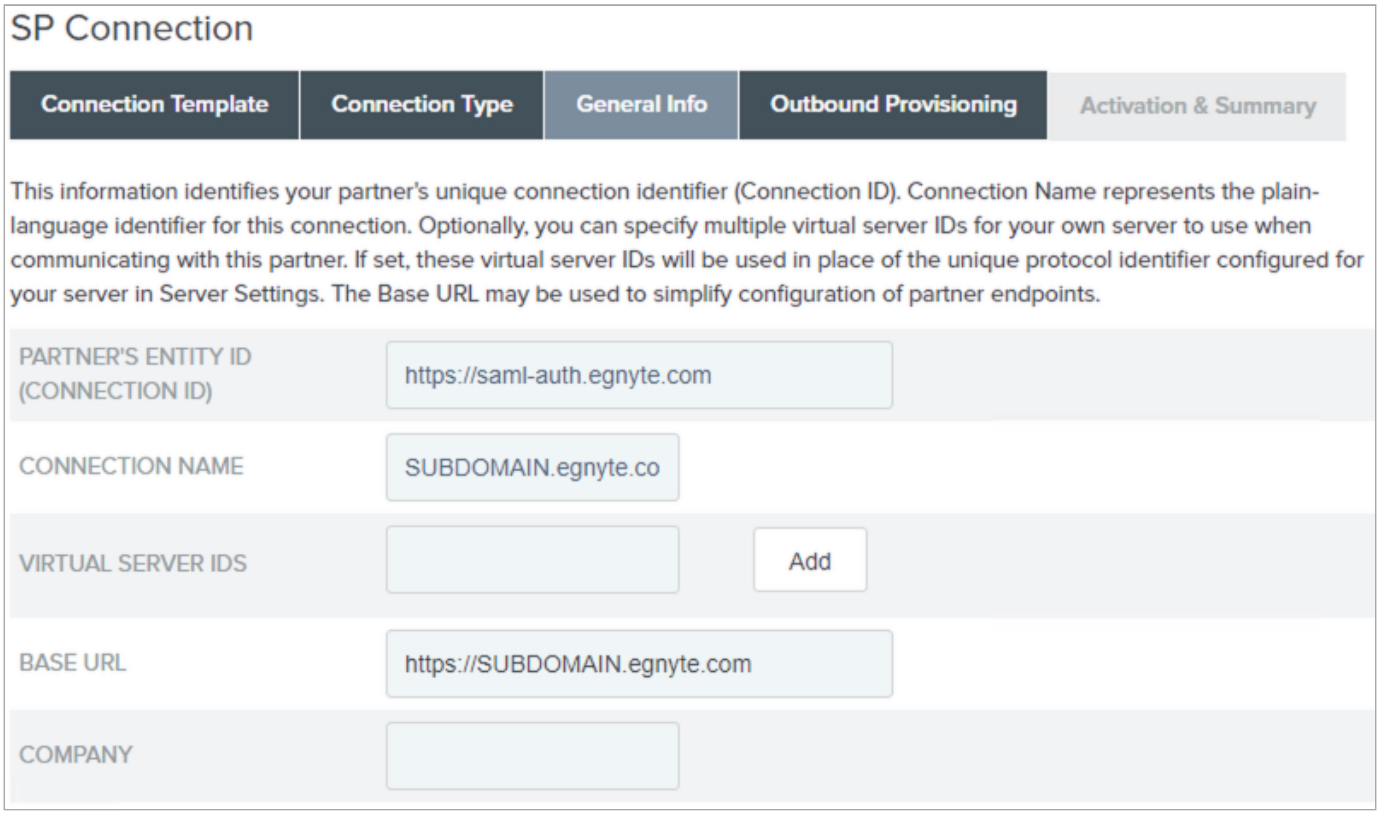
Copyright © 2025 Ping Identity Corporation



> **Tip**
>
> If this selection is not available, verify the connector installation and restart PingFederate.

3. On the **Connection Type** screen, ensure the **Outbound Provisioning** check box is selected, and the **Browser SSO Profiles** check box is cleared (if appropriate).
4. On the **General Info** screen, the default values are taken from the metadata file you selected in step 2.

Copyright © 2025 Ping Identity Corporation



5. Click **Next** to continue the Browser SSO configuration. For more information, see the following sections under Identity provider SSO configuration:
    - Managing IdP adapters
    - Configure IdP Browser SSO
    - Configuring credentials

    > **Important**
    >
    > The SAML_SUBJECT configured on the **IdP Adapter Mapping > Attribute Contract Fulfillment** screen must match the **Default user mapping** configured in Egnyte. For more information, see Configure Egnyte for SSO.

6. On the **Protocol Settings > Allowable SAML Bindings** screen, ensure that both **POST** and **Redirect** are selected.
7. On the **Credentials > Digital Signature Settings** screen, select the signing certificate.
8. On the **Activation & Summary** screen, set **Connection Status** to Active, then click **Save**.

> **Tip**
>
> If you are not ready to complete the SSO configuration, you can click **Save** and return to the configuration page later. To return to the configuration page, select the connection from **Identity Provider > SP Connections > Manage All**.

Copyright © 2025 Ping Identity Corporation

Release notes

Changelog

Egnyte Connector 1.0 – January 2015 (current release)

- Initial release
- Added support for user provisioning
- Added support for browser-based single sign-on

Known issues and limitations

- Due to a limitation with PingFederate 8.1 and earlier versions, when configuring two SP connections with the same provisioner, the second connection built may be pre-populated with the channel from the first connection. To avoid conflicts, delete this pre-populated channel and create a unique channel for each connection.
- When an LDAP user is deleted in a targeted group distinguished name (DN), the provisioning connector does not propagate the deletion until a new user is added to the group. This limitation is compounded when the **User Create** provisioning option is disabled. For solutions, see SaaS provisioner does not remove the user in the Knowledge Base.
- Due to PingFederate limitations, user attributes cannot be cleared once set.
- Due to an issue that has been reported to the service provider, users that have had their authType set to 'sso' will no longer be able to log in with their standard username and password.
- Due to an issue that has been reported to the service provider, the service provider returns an HTTP status code of 500 on some requests to update a user.
- Due to a limitation with the service provider's API, a user's role can only be updated through the service provider's administrative console.

Performance limitations

- Due to Egnyte API limitations, API calls are rate limited. This will prevent requests from being completed. For more information, see Frequently Asked Questions on the Egnyte developers site.

Download manifest

The distribution `.zip` archive for the connector contains the following:

- `ReadMeFirst.pdf` – contains links to this online documentation.
- `egnyte-saml-metadata.xml` – SAML metadata file for use with an Egnyte connection. For more information, see Obtain Egnyte SAML 2.0 metadata.
- `/legal`:
  - `Legal.pdf` – copyright and license information.

Copyright © 2025 Ping Identity Corporation

- `/dist` – contains libraries needed for the connector:
  - `pf-egnyte-quickconnection-1.0.0.2.jar` – PingFederate Egnyte Connector
  - `prov-cpl-2.0.1.jar` – PingFederate Common Provisioning Layer
  - `gson.2.2.4.jar` – Used to serialize and deserialize objects in the Egnyte Connector

Copyright © 2025 Ping Identity Corporation

Entrust Identity Enterprise Integration Kit

The Entrust Identity Enterprise Integration Kit allows PingFederate to use the Entrust service for multi-factor authentication (MFA).

Features

- Supports the PingFederate PingFederate authentication API.
- Supports the JavaScript Widget for the PingFederate Authentication API.

Components

Entrust IdP Adapter

- Allows PingFederate to communicate with the Entrust Identity Enterprise for MFA.

Templates

- Allows you to modify the appearance of the authentication prompts.

Language packs

- Allows you to customize or localize the messages returned by the PingFederate authentication API and shown on the templates during authentication. For help, see Localizing messages for end users in the PingFederate documentation.

Intended audience

This document is intended for PingFederate administrators.

If you need help during the setup process, see the following resources:

- The following sections of the PingFederate documentation:
  - Identity provider SSO configuration
  - Managing IdP adapters
  - Authentication policies

System requirements

- PingFederate 10.3 or later

Overview of the SSO flow

With the Entrust Identity Enterprise Integration Kit, PingFederate includes Entrust in the sign-on flow.

Description

1. The user initiates SSO from an SP application through the PingFederate SP server.

Copyright © 2025 Ping Identity Corporation

> **Note**
>
> This SP-initiated scenario represents the optimal use case, where both the identity provider (IdP) and SP use PingFederate. If your SP partner does not support this scenario, however, PingFederate accepts any valid SAML authentication request.
> You can also enable IdP-initiated SSO. In this case, the SSO flow would not include this step or the next one.

2. The PingFederate SP server generates a SAML `AuthnRequest` and sends it to the PingFederate IdP server.
3. The IdP requests authentication from the adapter and it asks for the User ID.
4. The adapter sends the User ID to Entrust.
5. Entrust responds with a list of authenticators configured for the user.

> **Note**
>
> If the **Default to Primary Authenticator** setting is enabled, the adapter displays the appropriate window. This setting is not enabled by default.

6. The adapter displays the list of authenticators and the user selects the one they want to use.
7. The adapter sends the selected authenticator to Entrust.
8. Entrust responds with the next step to the user (sends SMS OTP, TOTP, Entrust soft token push notification, mobile smart credential push authentication, Grid card selection, KBA) and the adapter.
9. The adapter presents the appropriate screen (enter OTP, KBA, grid selection, enter temporary access tokens) and sends the user response to Entrust.
10. Entrust validates the credentials sent and responds to PingFederate.
11. If the validation fails, the user is denied. If validation succeeds, the PingFederate IdP server generates a SAML assertion with the username as the Subject and passes it to the PingFederate SP server.

Authentication API Support

You can use the PingFederate authentication API to integrate the Entrust IdP Adapter into your application.

The PingFederate Authentication API provides access to the current state of the authentication flow as a user steps through the PingFederate authentication policy. For more information, see PingFederate authentication API in the PingFederate documentation. To integrate the Entrust IdP Adapter into your authentication flow, configure your application based on the states, actions, and models available in the PingFederate Authentication API Explorer. For help, see Exploring the Authentication API in the PingFederate documentation.

Models, objects, and error codes

When using the Entrust IdP Adapter through the PingFederate authentication API, the adapter uses the following state models, action models, objects, and error codes.

Copyright © 2025 Ping Identity Corporation

State models

Status	Response Model	Action	Description
`USER_ID_REQUIRED`	This state has no model.	• `checkuserId` • `cancel`	The user must select the one-time device from the list to proceed with the authentication flow.
`AUTHENTICATOR_SELECTION_REQUIRED`	• `authenticators:` The list of authenticators available to the user.	• `selectAuthenticator` • `cancel`	The user must select an authenticator for authentication.
`INPUT_REQUIRED`	• `authenticator (string):` The authenticator that the user selected. • `gridChallenge (GridChallenge):` The information that's needed for grid authentication. • `kbaChallenge (KbaChallenge):` The information that's needed for KBA authentication. • `tokenPushMutualChallenge (string):` The optional token push mutual challenge that's received.	• `checkInput` • `showAlternativeAuthentication` • `resendAuthenticationRequest` • `cancel`	The user must complete authentication via an input.
`ENTRUST_FAILED`	• `code (string):` The error code. • `message (string):` The developer-facing error message. • `userMessage (string):` The user-facing error message.	• `cancel`	Entrust authentication failed.

Copyright © 2025 Ping Identity Corporation

Action models

Action	Request Model	Errors	Description
`checkuserId`	• `userId (string):` The user ID that's used for authentication.	This action has no errors.	Submits the user ID for authentication.
`cancel`	This action has no model.	This action has no errors.	Cancels the current operation.
`selectAuthenticator`	• `authenticator (string):` The authenticator that the user selected.	• Error: `VALIDATION_ERROR`	Selects an authenticator as an authentication method.
`checkInput`	• `input (string):` The response that's required for authentication. • `answers(array[KbaAnswer]):` The KBA response that's required for authentication.	• Error: `VALIDATION_ERROR`	Validates the input from the user.
`showAlternativeAuthentication`	This action has no model.	This action has no errors.	Shows the list of alternative authentication options.
`resendAuthenticationRequest`	This action has no model.	• Error: `VALIDATION_ERROR`	Resends the current authentication request.

Objects

GridChallenge

Parameter Name	Type	Description
`cells`	String array	The cells used for the input.
`numCharsPerCell`	Integer	The number of characters per cell.

Copyright © 2025 Ping Identity Corporation

Parameter Name	Type	Description
`serialNumbers`	String array	The list of serial numbers of the grid cards that can be used.

KbaChallenge

Parameter name	Type	Description
`id`	String	The ID used for the challenge.
`userQuestions`	Array of `UserQuestion`	The questions used for the input.

UserQuestion

Parameter name	Type	Description
`id`	String	The ID for the user question.
`question`	String	The question value for the user question.
`answer`	String	The answer value for the user question.

KbaAnswer

Parameter name	Type	Description
`id`	String	The ID of the question.
`answer`	String	The answer value for the question.

Error Codes

An error code is returned if the call flow state has not reached a dead end, and the user can still authenticate with a device.

Top level error codes

Error code	Message	HTTP status
`VALIDATION_ERROR`	One or more validation errors occurred.	400

Copyright © 2025 Ping Identity Corporation

Detail level error codes

Error code	Message	userMessageKey	Parent code
`INVALID_AUTHENTICATOR`	Selected authenticator is not a valid form of authentication.	`invalid.authenticator`	`VALIDATION_ERROR`
`INVALID_INPUT`	The input entered is incorrect.	`invalid.input`	`VALIDATION_ERROR`
`INVALID_INPUT_FORMAT`	The format of input is incorrect.	`invalid.input.format`	`VALIDATION_ERROR`
`GENERAL_ERROR`	Authentication error.	`general.error`	`VALIDATION_ERROR`
`ACCOUNT_LOCKED_OUT`	The user account is locked.	`account.locked.out`	`VALIDATION_ERROR`
`PUSH_CANCELLED`	User canceled push notification transaction.	`push.cancel`	`VALIDATION_ERROR`
`PUSH_SUSPICIOUS`	User marked the push notification transaction as suspicious.	`push.suspicious`	`VALIDATION_ERROR`

Setup

Deploying the integration files

To get started with the integration, deploy the Entrust Identity Enterprise Integration Kit files to your PingFederate directory.

About this task

> **Note**
>
> If you operate PingFederate in a cluster, the following steps refer to the console node.

Steps

1. Download the Entrust Identity Enterprise Integration Kit `.zip` archive from the Ping Identity Integration Directory.
2. Stop PingFederate.
3. If you are upgrading an existing deployment, back up your customizations and delete previous versions of the integration files.
    1. Back up any Entrust Identity Enterprise Integration Kit files that you customized in `<pf_install>/pingfederate/server/default/conf/`.
    2. Delete `pf-entrust-adapter-<version>.jar` from `<pf_install>/pingfederate/server/default/deploy`.

Copyright © 2025 Ping Identity Corporation

4. Extract the `.zip` archive and merge the contents of the `dist` directory with your `<pf_install>/pingfederate` directory.
5. If there is more than one version of the `pf-authn-api-sdk-<version>.jar` file in your `<pf_install>/pingfederate/server/default/lib` directory, delete all but the latest version of the file.
6. If you backed up any customized files, modify the new files with your customizations.
7. Start PingFederate.
8. If you operate PingFederate in a cluster, repeat steps 2-8 for each engine node.

Registering PingFederate as an application in Entrust

To process sign-on requests, PingFederate requires access to the Entrust Authentication API. You can grant access in the Entrust admin console.

About this task

For more information about registering authentication agents, see Integrate API applications in the Entrust documentation.

Steps

1. Sign on to the Entrust admin console.
2. Go to **Resources > Applications**.
3. In the **Applications** window, click **Add**.
4. In the **Select an Application Template** window, in the **API Applications** section, click **Authentication API**.
5. In the **Add Authentication API** window, in the **Application Name** field, enter a name for your application.
6. In the **Application Description** field, enter a description for your application.
7. **Optional:** Add a custom application logo.
    1. Next to **Application logo**, click the + icon.
    2. Click the **Upload** icon.
    3. Browse to your chosen file and click **Open**.
    4. Resize your image if required, then click **OK**.
8. Click **Next**.
9. In the **General Settings** window, for Source of the Client IP Address for Risk Conditions, select one of the following.

Option	Description
Not Provided	The IP address is not provided and is not extracted from the connection.
Provided in the API	The IP address is provided in the request body of the API.

Copyright © 2025 Ping Identity Corporation

Option	Description
From the incoming HTTP connection	The IP address is extracted from the connection.

10. Click Submit.

    Result:

    The **Application ID** is generated.

11. Copy and paste the **Application ID** into PingFederate.
12. Click **Done**.

Configuring an adapter instance

Configure the Entrust IdP Adapter to determine how PingFederate communicates with Entrust.

Steps

1. In the PingFederate administrative console, create a new IdP adapter instance.
    1. Go to **Authentication > Integration > IdP Adapters**.
    2. Click **Create New Instance**.
2. On the **Type** tab, set the basic adapter instance attributes.
    1. In the **Instance Name** field, enter a name for the adapter instance.
    2. In the **Instance ID** field, enter a unique identifier for the adapter instance.
    3. From the **Type** list, select **Entrust IdP Adapter**. Click **Next**.
3. On the **IdP Adapter** tab, configure the adapter instance by referring to Entrust IdP Adapter settings reference. Click **Next**.

On the **Extended Contract** tab, add any attributes that you want to include in the contract. Click **Next**.

1. On the **Adapter Attributes** tab, set pseudonym and masking options as shown in Set pseudonym and masking options in the PingFederate documentation. Click **Next**.
2. On the **Adapter Contract Mapping** tab, configure the contract fulfillment details for the adapter as shown in Define the IdP adapter contract in the PingFederate documentation. Click **Next**.
3. On the **Summary** tab, check and save your configuration. Click **Save**.
4. Create or modify a connection to the service provider using the Entrust IdP Adapter instance. See SP connection management in the PingFederate documentation.

Entrust IdP Adapter settings reference

Field descriptions for the Entrust IdP Adapter configuration screen.

Copyright © 2025 Ping Identity Corporation

Standard Fields

Field	Description
Service Domain	The domain of the Entrust service. Format is `<customer>.<region>.trustedauth.com` For example: `mycompany.us.trustedauth.com`
Application ID	The application ID for the Identity as a Service application.

Advanced Fields

Field	Description
Authenticators Endpoint	The endpoint used to get user authenticators. The default endpoint is `api/web/v2/authentication/users`.
Select Authenticator Endpoint	The endpoint used to select the authenticator. The default endpoint is `/api/web/v2/authentication/users/authenticate/%s` The `%s` will be replaced by the authenticator used.
Complete Authentication Endpoint	The endpoint used to complete authentication actions. The default endpoint is `api/web/v1/authentication/users/authenticate/%s/complete`. The `%s` will be replaced by the authenticator used.
Logout Endpoint	The endpoint used to complete logout. The default endpoint is `/api/web/v1/authentication/logout`.
HTML Template Prefix	Identifies the set of HTML templates that the adapter uses. If you customized the template file names in `/server/default/conf/template`, enter the new prefix here. The default value is `entrust`.
Messages Files	Identifies the customizable language-pack file that the adapter uses. If you customize the `languaged-pack` file name in the `/server/default/conf/language-packs` directory, enter the new name here. The default value is `entrust-messages`.
Error Message Key Prefix	Prefix for error messages in the language pack. The default value is `entrust.error`.

Copyright © 2025 Ping Identity Corporation

Field	Description
API Request Timeout	The amount of time in milliseconds that PingFederate allows when establishing a connection with Entrust or waiting for a response to a request. A value of 0 disables the timeout. The default value is `5000`.
Default to Primary Authenticator	If selected, the user will default to the primary authenticator set in Entrust.
Enable Token Push With Mutual Challenge	If selected, token push requests from the adapter enable a mutual authentication challenge unless the chained attribute `enable-token-push-with-mutual-challenge` is set to `false` and overrides this value, which is specific to the adapter instance. This check box is cleared by default.
Show Failed Screen	If selected, the adapter displays an authentication failed page when it encounters an error. This check box is selected by default.
Branch on Error	A comma-separated list of error codes from Entrust that can be used for conditional branching in the policy. The error codes listed are handled as a success.
Proxy Settings	Defines proxy settings for outbound HTTP requests. The default value is **System Defaults**.
Custom Proxy Host	The proxy server host name to use when **Proxy Settings** is set to **Custom**. This field is blank by default.
Custom Proxy Port	The proxy server port to use when **Proxy Settings** is set to **Custom**. This field is blank by default.

Troubleshooting

Enabling debug logging

To help with troubleshooting or monitoring, you can turn on activity logging for the adapter.

About this task

These steps are optional. For general information about logging, see Enabling debug messages and console logging in the PingFederate documentation.

Copyright © 2025 Ping Identity Corporation

Steps

1. Open the `<pf_install>/pingfederate/server/default/conf/log4j2.xml` file for editing.
2. If you want to log activity for PingFederate and all adapters, do the following.
    1. Find the following section.

```
<AsyncRoot level="INFO" includeLocation="false">
        <!-- <AppenderRef ref="CONSOLE" /> -->
        <AppenderRef ref="FILE" />
</AsyncRoot>
```

    2. Change `INFO` to `DEBUG`.

```
<AsyncRoot level="DEBUG" includeLocation="false">
        <!-- <AppenderRef ref="CONSOLE" /> -->
        <AppenderRef ref="FILE" />
</AsyncRoot>
```

    3. If you want to see the adapter activity in the console, remove the comment tags.

```
<AsyncRoot level="INFO" includeLocation="false">
        <AppenderRef ref="CONSOLE" />
        <AppenderRef ref="FILE" />
</AsyncRoot>
```

3. If you want to log activity just for the Entrust IdP Adapter, add the following line.

```
<Logger name="com.pingidentity.adapters.entrust" level="DEBUG"/>
```

4. Save the file.

Release notes

Changelog

The following is the change history for the Entrust Identity Enterprise Integration Kit.

Entrust Identity Enterprise Integration Kit 1.5 - February 2024

- Added the ability for users to resend authentication requests for supported authenticators. For more information, see Models, objects, and error codes.
- Fixed knowledge-based authentication UI display issues.

Copyright © 2025 Ping Identity Corporation

- Added the ability to include the subject, when available, in the audit log framework.

Entrust Identity Enterprise Integration Kit 1.4 - September 2023

- Added support for token push notifications that use mutual authentication. For more information, see https://entrust.us.trustedauth.com/help/authentication/posts/5.26/ and Entrust IdP Adapter settings reference.

Entrust Identity Enterprise Integration Kit 1.3 - November 2022

- Added the ability to handle Entrust API error codes so that it can be used for policy branching.
- Added the ability to display custom error messages based on the Entrust API error codes returned.
- Fixed an issue that caused `errorMessage` and `errorMessageDetails` to not populate in a clustered deployment.

Entrust Identity Enterprise Integration Kit 1.2 – June 2022

- Added support for `TEMP_ACCESS_CODE` (Temporary Access Tokens).
- Added support for `SMARTCREDENTIALPUSH` (Mobile Smart Credential Push).
- Updated authenticator and message names for all authenticators in the template files.

Entrust Identity Enterprise Integration Kit 1.1 – April 2022

- Added support for customizable error template.
- Added support for TOKEN and KBA authentication options.
- Added the ability for administrators to skip authentication selection.
- Added the ability for users to choose another authentication option in the input required state.
- Updated to the latest jackson-databind libraries.
- Updated push notifications to do better error handling for canceled or suspicious use cases.

Entrust Identity Enterprise Integration Kit 1.0 – January 2022

- Initial Release

Known issues and limitations

The following are known issues or limitations for the Entrust Identity Enterprise Integration Kit.

Known issues

- For `TOKENPUSH` and `SMARTCREDENTIALPUSH` use-cases, if the user does not act on the push notification received on their Entrust application, the input required page will keep attempting to complete the transaction.

Known limitations

- The adapter supports resource rules with an authentication decision of **Skip Password** for first factor for low risk cases.

Copyright © 2025 Ping Identity Corporation

- When using the PingFederate authentication API and a user completes a `TOKENPUSH` or `SMARTCREDENTIALPUSH` authenticator, the user is taken to the `INPUT_REQUIRED` state. The user can submit the default template to continue and finish the flow.

Download manifest

The following files are included in the Entrust Identity Enterprise Integration Kit `.zip` archive:

- `Legal.pdf` – copyright and license information
- `dist/pingfederate/server/default` – contains the integration files
    - `deploy` – contains the Java libraries
        - `pf-entrust-adapter-<version>.jar` – JAR file that contains the Entrust IdP Adapter.
    - `conf` – contains the HTML template that presents the Entrust Identity Enterprise sign-on form.
    - `template` – contains user-facing HTML template files
        - `entrust-user-id-required.html`
        - `entrust-input-required.html`
        - `entrust-authenticator-selection-required.html`
        - `entrust-failed.html`
            - `assets` – contains functional scripts and files used by the template
        - `scripts` – contains script files used to collect and send information
            - `profileDevice.js`
            - `machineSecret.js`
            - `jquery-<version>.min.js` – A JavaScript file that contains the jQuery library
            - `deviceProfileEmbedded.js`
        - `images` – contains template image files
            - `ping-logo.svg` – an image file with company branding
            - `spinner.svg` – an image file used in a spinner animation
        - `css` – contains CSS files for the templates.
            - `entrust.css` – a CSS file that customizes the appearance of the template files.
            - `end-user/<version>/end-user.css` – a CSS file that customizes the appearance of the template files.
        - `fonts/end-user/icons` – contains template icons
            - `language-packs`

Copyright © 2025 Ping Identity Corporation

        - `entrust-messages.properties`

- `lib/pf-authn-api-sdk-<version>.jar` – a JAR file that contains the PingFederate Authentication API SDK

Copyright © 2025 Ping Identity Corporation

Evernote Provisioner

The PingFederate Evernote Provisioner enables an enterprise to provision users to Evernote. A quick connection template is also included to simplify the configuration of single sign-on (SSO).

Features

- Browser-based SP and IdP-initiated SSO
- Includes support for user life cycle management (including create, update, and disable).
- Includes configuration options for workflow capabilities (for example, the ability to disable updates).

Intended audience

This document is intended for PingFederate administrators.

Learn more about the setup process in the following sections of the PingFederate documentation:

- SP connection management
- Identity provider SSO configuration
- Datastores
- Configuring outbound provisioning
- Configuring outbound provisioning settings

System requirements

- PingFederate 7.3 or later with Java 8
- To allow outbound connections, you might need to allow the following endpoints on your firewall:
  - https://m1.svc.evernote.com

SSO is required for SCIM provisioning with Evernote. SCIM is available only in Evernote Teams accounts created on or after September 15, 2017. Learn more in How to determine when an Evernote Teams account was created in the Evernote documentation.

You can find more information on Evernote in the Evernote website.

User management

The Evernote Provisioner synchronizes users from your datastore to Evernote. The behavior of each provisioning capability is described below.

You can configure these capabilities in the Configure PingFederate for provisioning step of the setup process.

Copyright © 2025 Ping Identity Corporation

Synchronizing existing users

PingFederate synchronizes users based on the `Username` attribute in Evernote. If a user already exists in your datastore and Evernote, mapping this attribute correctly links the two records together.

For example:

- In Evernote, Janet's `Username` is `bjensen@example.com`.
- In your datastore, Janet's `mail` is `bjensen@example.com`.
- On the **Attribute Mapping** tab of your provisioning connection configuration, you map the `Username` attribute to `mail`.
- When the provisioning connector runs, the datastore user is provisioned with a `Username` of `bjensen@example.com`. That matches Janet's existing `Username` in Evernote, so her information in the datastore is synchronized to her Evernote account.

User provisioning

PingFederate provisions users when one of the following happens:

- A user is added to the datastore group or filter that is targeted by the provisioning connector.

The **Source Location** tab of your provisioning connection configuration defines which users PingFederate targets for provisioning.

User updates

PingFederate updates users when a user attribute changes in your datastore.

The **Attribute Mapping** tab of your provisioning connection configuration defines which attributes PingFederate monitors for changes.

User deprovisioning

PingFederate deprovisions users when one of the following happens:

- A user is deleted from the user store.
- A user is disabled in the user store.
- A user is removed from the datastore group or filter that is targeted by the provisioning connector.

Setup

Obtain PingFederate signing certificate

About this task

Copyright © 2025 Ping Identity Corporation

The following are the steps to obtain the signing certificate that is used for the digital signature verification of the PingFederate identity provider (IdP). When asked during Configure Evernote for SSO, provide the certificate that you have downloaded from PingFederate. Learn more in Managing digital signing certificates and decryption keys in the PingFederate documentation.

To export an existing certificate:

Steps

1. Sign on to the PingFederate Administrative Console as an administrative user for your organization.
2. Go to **Server Configuration > Certificate Management > Signing & Decryption Keys & Certificates**.
3. Click **Export** under **Action** for the certificate you want to export.
4. On the **Export Certificate** screen, select **Certificate Only** export and then click **Next**.
5. On the **Export & Summary** screen, click **Export** to save the certificate file on your system.
6. Click **Done**.
7. Click **Cancel** to go back to the **Server Configuration** screen.

Upgrade an existing connector

Steps

1. Before stopping the PingFederate server to upgrade the Evernote Connector, go to the **Attribute Mapping** screen for existing channel configurations and note the current configuration.

   > **Warning**
   >
   > The upgrade process removes existing mappings and defaults on the **Attribute Mapping** screen. These must be configured again before activating the channel configuration.

2. Disable the existing service provider (SP) connection where the Evernote Connector is configured.
3. Delete the existing Evernote Connector SP connection and save.
4. Stop the PingFederate server if it is running.
5. Extract the new Evernote Connector distribution `.zip` archive into a holding directory.
6. Remove any older versions of `pf-evernote-quickconnection-[version].jar` from:

   `<pf_install>/pingfederate/server/default/deploy`

7. Remove the following files from the same directory if they are present:
   - `evernote-business-provisioning-api-1.28.jar`
   - `pf-evernote-oauth-helper.war`
8. From the `dist` directory of the new version of the connector, copy the file:

   `pf-evernote-quickconnection-2.0.jar`

Copyright © 2025 Ping Identity Corporation

into the directory:

`<pf_install>/pingfederate/server/default/deploy`

9. Start the PingFederate server.
10. Create a new SP connection, using **Evernote Connector** as the connection template.
11. Follow instructions in Configure PingFederate for SSO and Configure PingFederate for provisioning in order to configure metadata.
12. Go to the **Attribute Mapping** screen for existing channel configurations and click **Refresh Fields**.
13. Ensure all new required fields (if any), are mapped appropriately or have a default value.
14. When you have completed the attribute configuration, click **Done**, **Done**, and **Save**.
15. Activate the SP connection to resume outbound provisioning.

Install the connector

About this task

This section describes the common steps required to install the PingFederate Evernote Provisioner Connector.

Steps

1. Stop the PingFederate server if it is running.
2. Extract the Evernote Connector distribution `.zip` archive.
3. Copy the contents of the `dist` directory into the directory:

   `<pf_install>/pingfederate/server/default/deploy`
4. (Optional) If you plan to use the connector for outbound provisioning, edit the `run.properties` file located in `<pf_install>/pingfederate/bin`, changing the property `pf.provisioner.mode` to `STANDALONE`.

   For example: `pf.provisioner.mode=STANDALONE`

   > **Note**
   >
   > Learn more about using the FAILOVER setting for runtime deployment in Deploying provisioning failover in the PingFederate documentation.
5. Start the PingFederate server.

Enable outbound provisioning

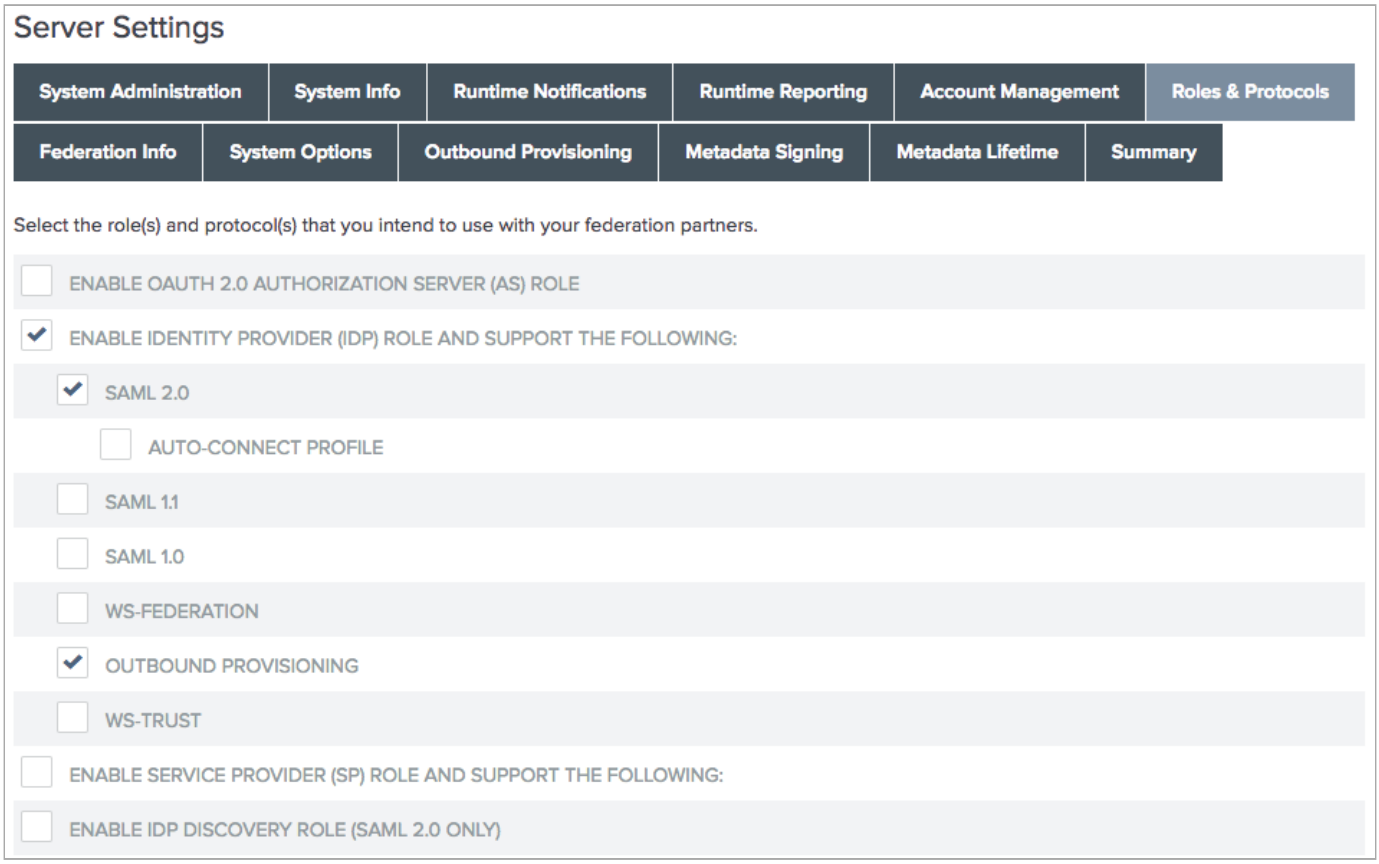
About this task

After enabling outbound provisioning in the `<pf_install>/pingfederate/bin/run.properties` file, you must also activate the outbound provisioning role in the administrative console.

Copyright © 2025 Ping Identity Corporation

Steps

1. Go to the **Server Configuration > Server Settings > Roles & Protocols** screen.
2. Select the **Outbound Provisioning** checkbox.



> **Note**
>
> Enabling outbound provisioning adds the outbound provisioning screen, requiring the selection of a database to facilitate provisioning. Learn more in Configuring outbound provisioning settings in the PingFederate documentation or click **Help** from the configuration screen.

Configure PingFederate for SSO

About this task

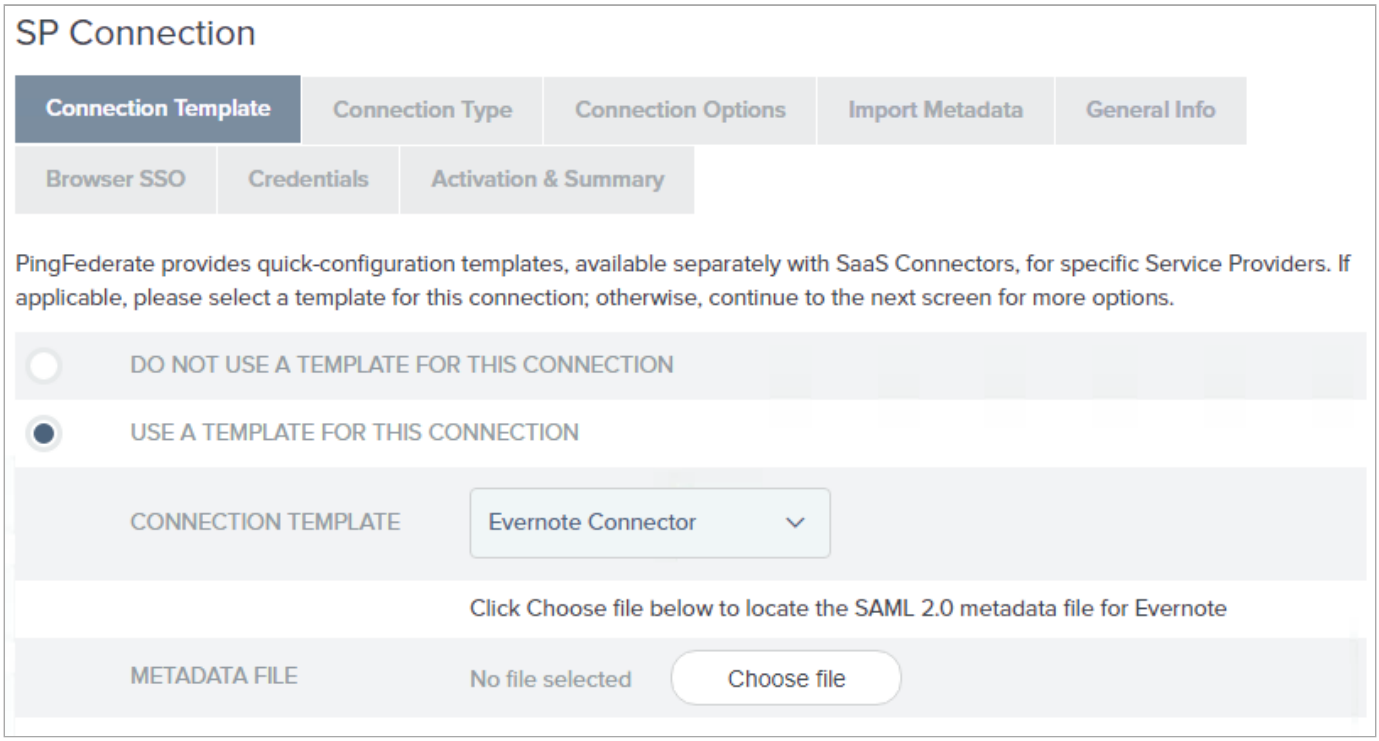
To configure a connection for single sign-on (SSO) to Evernote, follow the instructions in this section. Outbound provisioning details are managed within a service provider (SP) connection and can be added to an existing SP connection.

> **Important**
>
> SSO is required for SCIM provisioning with Evernote. SCIM is available only in Evernote Teams accounts created on or after September 15, 2017. Learn more in How to determine when an Evernote Teams account was created in the Evernote documentation.

Copyright © 2025 Ping Identity Corporation

Steps

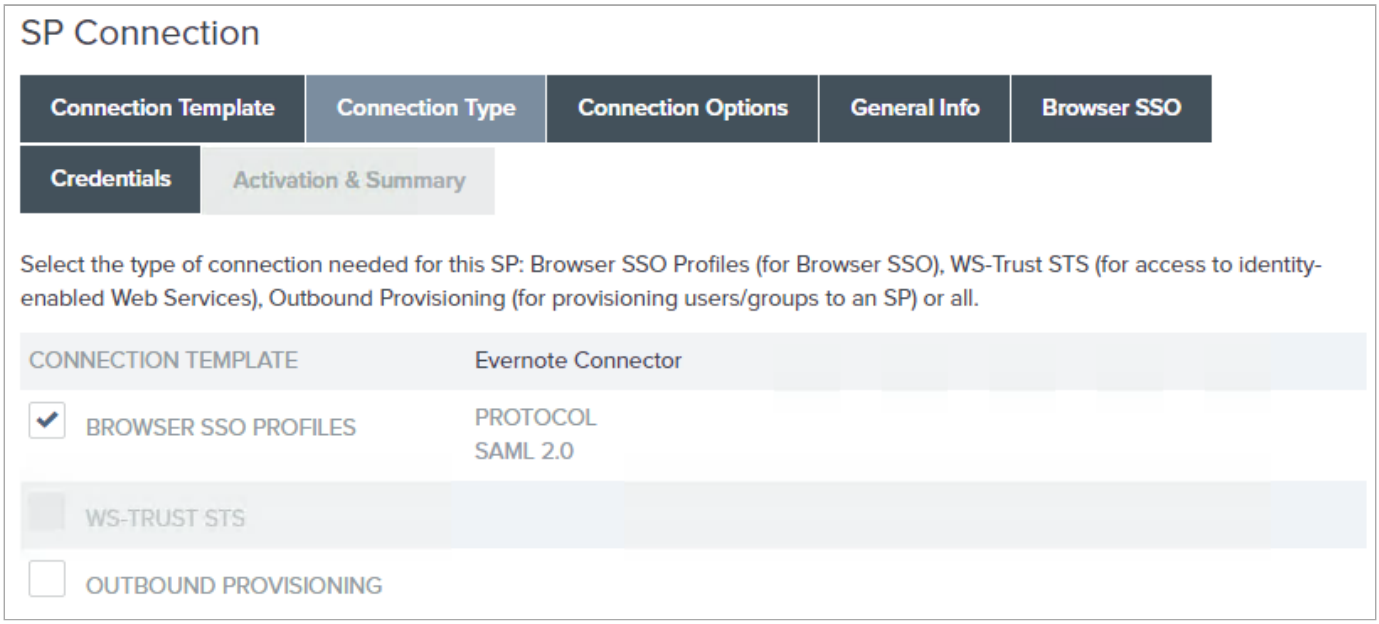
1. Create a new SP connection or select an existing SP connection from the **SP Configuration** menu.
2. On the **Connection Template** screen, select **Use a template for this connection** and choose **Evernote Connector** in the **Connection Template** list. When asked during the connection configuration steps, import the `evernote-saml-metadata.xml` packaged with this connector.



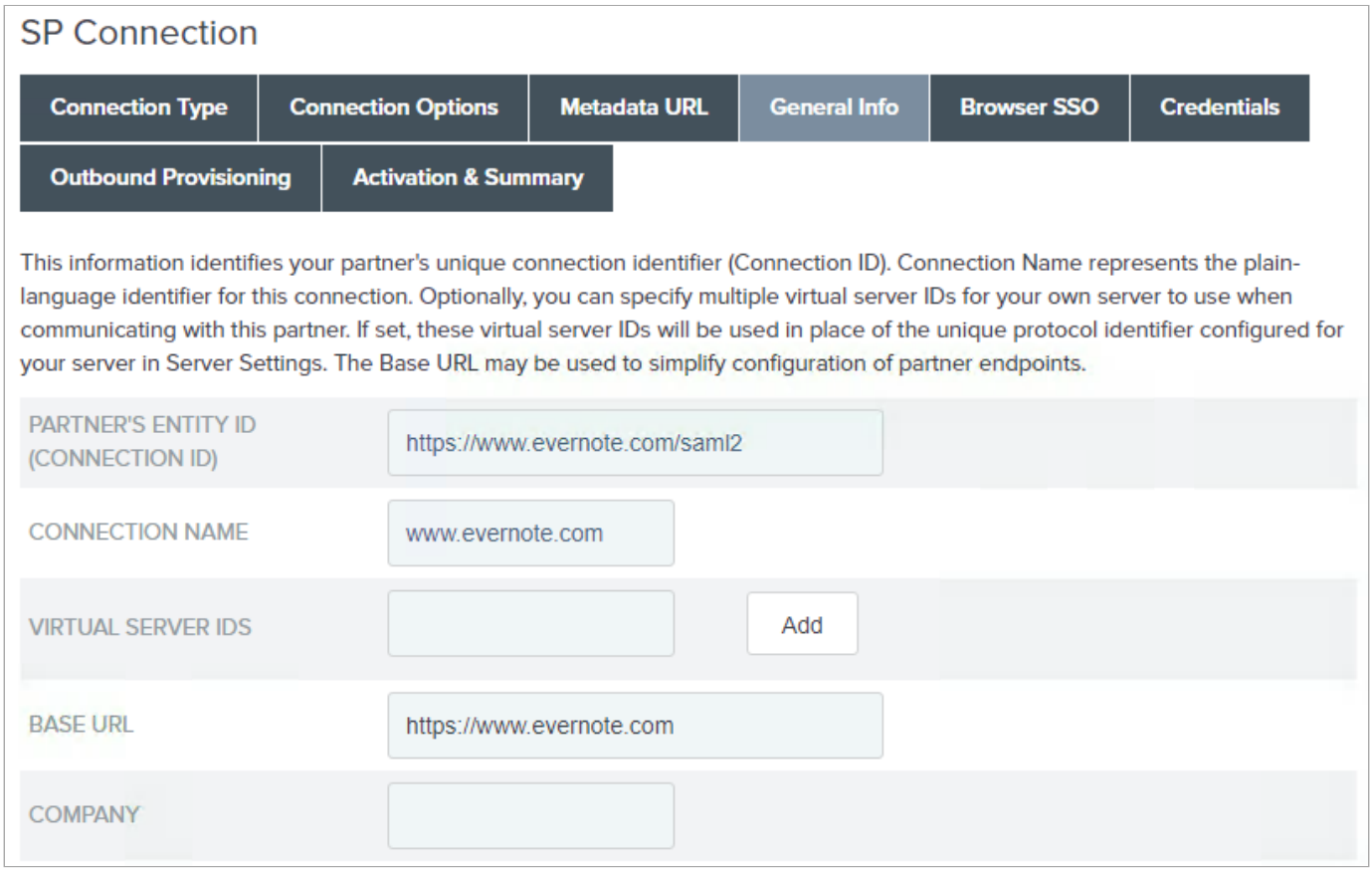
> **Tip**
>
> If this selection is not available, verify the connector installation and restart PingFederate.

3. On the **Connection Type** screen, ensure that the **Browser SSO Profiles** checkbox is selected and the **Outbound Provisioning** checkbox is cleared.

Copyright © 2025 Ping Identity Corporation



4. On the **General Info** screen, the default values are taken from the metadata file you selected in an earlier step. We recommend using the metadata default values.



5. Click **Next** to continue the Browser SSO configuration.

   Learn more in the following sections under Identity provider SSO configuration in the PingFederate documentation:

   - Managing IdP adapters

Copyright © 2025 Ping Identity Corporation

- Configure IdP Browser SSO
- Configuring credentials

6. On the **Attribute Contract** screen, set the **Subject Name Format** for SAML_SUBJECT to the below value:
   - urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress
7. On the authentication adapter's **Attribute Contract Fulfillment** screen, map SAML_SUBJECT to email address. Evernote requires SAML_SUBJECT to contain the user's email address, which must match the Evernote user's business email address.
8. On the **Credentials > Digital Signature Settings** screen, select the signing certificate.
9. On the **Activation & Summary** screen, set **Connection Status** to **ACTIVE**, then click **Save**.

> **Tip**
>
> If you are not ready to complete the SSO configuration, you can click **Save** and return to the configuration page later. To return to the configuration page, select the connection from **Identity Provider > SP Connections > Manage All**.

Configure Evernote for SSO

About this task

To configure Evernote for single sign-on (SSO), you must have the signing certificate from your PingFederate identity provider (IdP) setup. Learn more in Obtain PingFederate signing certificate.

> **Note**
>
> Only Evernote Teams account admins are authorized to configure SSO. Learn more about Evernote SSO configuration in Configure SSO for your Evernote Teams account in the Evernote documentation.

Steps

1. Sign on to your Evernote account as an administrative user for your organization.
2. Go to the Evernote Business Admin Console.
3. On the left sidebar, go to **Security > Single Sign-On**.
4. In the **SAML HTTP Request URL** field, enter the HTTPS endpoint of your IdP for SSO requests.

   For example, https://<*pf_hostname*>:<*pf_port*>/idp/SSO.saml2
5. Open your PingFederate signing certificate file using a text editor.
6. Under **X.509 Certificate**, paste your PingFederate signing certificate used to verify SAML responses.
7. Enter a **Session Duration**. This is the number of hours the SSO token will remain valid before employees must re-authorize their token.

   The default value is 24 hours.

Copyright © 2025 Ping Identity Corporation

8. Click **Save & Enable**

> **Important**
>
> Clicking **Save & Enable** will require you to successfully perform SSO to enable this functionality. SSO must be enabled by an Evernote Business account admin.

Configure Evernote for provisioning

About this task

The following are the tasks to enable SCIM provisioning on the Evernote Admin Portal. Learn more about configuring Evernote for SCIM in Configure SCIM with your Evernote Teams account in the Evernote documentation.

> **Important**
>
> To enable SCIM provisioning, you must enable single sign-on (SSO). Learn more in Configure Evernote for SSO. SCIM is only available in Evernote Teams accounts created on or after September 15, 2017. Learn more in How to determine when an Evernote Teams account was created.

Steps

1. Sign on to your Evernote account as an administrative user for your organization.
2. Go to the Evernote Business Admin Console.
3. On the left sidebar, go to **Integrations > SCIM**.
4. Generate a bearer token by clicking the **Generate token** link.
5. Copy the bearer token value for later use.

> **Important**
>
> You must copy the token immediately, as it will disappear on your next visit.

Configure PingFederate for provisioning

About this task

To configure a connection for outbound provisioning to Evernote, follow the instructions in this section. Outbound provisioning details are managed within a service provider (SP) connection and can be added to an existing SP connection.

> **Important**
>
> Single sign-on (SSO) is required for SCIM provisioning with Evernote. SCIM is available only in Evernote Teams accounts created on or after September 15, 2017. Learn more in How to determine when an Evernote Teams account was created in the Evernote documentation.

Steps

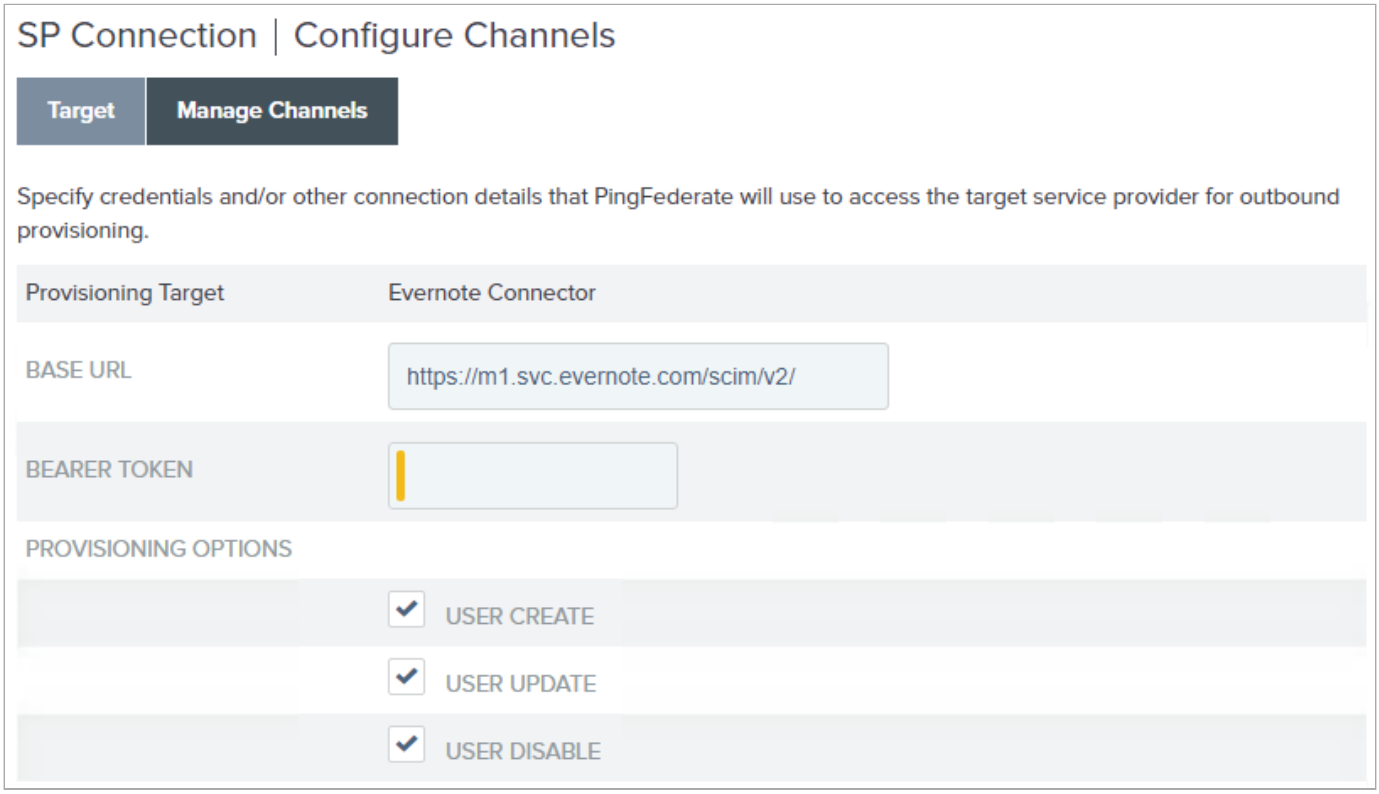
1. In the PingFederate administrative console, configure the datastore that PingFederate will use as the source of user data.

Copyright © 2025 Ping Identity Corporation

Learn more in Datastores in the PingFederate documentation.

- When targeting users and groups for provisioning, exclude the user account that you will use to administer users in your connection to Evernote. This prevents the PingFederate provisioning engine from interfering with the account that provisions users and groups.

2. Select the existing SP connection created in Configure PingFederate for SSO from the **Identity Provider > SP Connections > Manage All** menu.
3. On the **Activation & Summary** screen, go to **Connection Type** and select the **Outbound Provisioning** and **Browser SSO Profiles** checkboxes.
4. On the **Outbound Provisioning** screen, click **Configure Provisioning**.
5. On the **Target** screen, enter the values for each field as required by the Evernote Connector.



Target screen options

Field Name	Description
Base URL	The base URL for Evernote. ◦ https://m1.svc.evernote.com/scim/v2/
Bearer Token	The bearer token used by the connector to make authenticated API calls to Evernote. Learn more about obtaining the bearer token in Configure Evernote for provisioning.
Provisioning Options	

Copyright © 2025 Ping Identity Corporation

Field Name	Description
User Create	**True** (default): Users will be created in Evernote. **Note** Due to Evernote API limitations, new users cannot be created with the same username as a previously deactivated user. **False**: Users will not be created in Evernote. **Note** The `provisioner.log` will display a warning within the create user workflow that the user was not created in Evernote.
User Update	**True** (default): Users will be updated in Evernote. **False**: Users will not be updated in Evernote. **Note** The `provisioner.log` will display a warning within the update user workflow that the user was not updated in Evernote.
User Disable	**True** (default): Users will be disabled in Evernote. **Note** Due to Evernote API limitations, once a user is deactivated, they cannot be reactivated with the provisioner. A deactivated user can only be reactivated in the Evernote Teams Admin Console. **False**: Users will not be disabled in Evernote. **Note** The `provisioner.log` will display a warning within the user workflow that the user was not disabled in Evernote.

6. Click **Next** to continue the provisioning configuration.

   Learn more in the following sections under Outbound provisioning for IdPs in the PingFederate documentation:

   - Managing channels
   - Specifying channel information
   - Identifying the source datastore
   - Modifying source settings
   - Specifying a source location
   - Mapping attributes
   - Reviewing channel settings

Copyright © 2025 Ping Identity Corporation

> **Note**
>
> Credentials will be verified when the channel and SP connection is set to Active and provisioning is initiated.

> **Tip**
>
> If you are not ready to complete the provisioning configuration, you can click **Save** and return to the configuration page later. To return to the configuration page, select the connection from **Identity Provider > SP Connections > Manage All**.

Supported attributes reference

The following table consists of the attributes that can be mapped for user provisioning.

Attribute	Description
Username	Evernote's unique identifier for the user. The expected format is user@domain (for example, 'bjensen@example.com'). This attribute is required.
Display Name	The name of the user, suitable for display to users. This attribute is required.
External ID	A string that is an identifier for the resource as defined by the provisioning client.

Release notes

Changelog

Evernote Connector 2.0 – July 2018 (current release)

- Updated support for user life cycle management (including create, update and disable)
- Added configuration options for workflow capabilities (for example, the ability to disable updates)
- Added support for the below user attributes:
  - Display Name
  - External ID
- Added support for the Evernote SCIM 2.0 API.
- Removed support for hard delete (feature deprecated by Evernote)
- Removed support for reactivating a disabled user (feature deprecated by Evernote)

Evernote Connector 1.0 – July 2014

- Initial Release

Copyright © 2025 Ping Identity Corporation

Known issues and limitations

- Due to a limitation with PingFederate 8.1 and earlier versions, when configuring two SP connections with the same provisioner, the second connection built may be pre-populated with the channel from the first connection. To avoid conflicts, delete this pre-populated channel and create a unique channel for each connection.
- When an LDAP user is deleted in a targeted group distinguished name (DN), the provisioning connector does not propagate the deletion until a new user is added to the group. This limitation is compounded when the **User Create** provisioning option is disabled. Learn more in SaaS provisioner does not remove the user in the Ping Identity Knowledge Base.
- Due to PingFederate limitations, user attributes cannot be cleared after they're set.
- Provisioning disabled users from the User Store (LDAP) to Evernote is not supported.
- Due to Evernote API limitations, after a user is deactivated, they cannot be reactivated with the provisioner. A deactivated user can only be reactivated in the Evernote Teams Admin Console.
- Due to Evernote API limitations, new users cannot be created with the same username as a previously deactivated user.

Download manifest

The distribution `.zip` archive for the connector contains the following:

- `ReadMeFirst.pdf` : Contains links to this online documentation.
- `evernote-saml-metadata.xml` : SAML metadata file for use with an Evernote connection. Learn more in Configure PingFederate for SSO.
- `/legal` :
  - `Legal.pdf` : Copyright and license information.
- `/dist` : Contains libraries needed for the connector:
  - `pf-evernote-quickconnection-[version].jar` : PingFederate Evernote Connector

Copyright © 2025 Ping Identity Corporation

Facebook integrations

PingFederate Facebook Cloud Identity Connector

This PingFederate Cloud Identity Connector allows a Service Provider (SP) to leverage Facebook as an Identity Provider (IdP) for access to Internet applications in the SP domain.

PingFederate Workplace from Facebook Connector

The PingFederate Workplace from Facebook Connector (formerly known as Facebook At Work and Workplace by Facebook) enables an enterprise to provision users to Workplace from Facebook.

Facebook Login Integration Kit

The Facebook Login Integration Kit allows PingFederate to use Facebook as an identity provider (IdP). This allows users to access service provider (SP) applications by signing on with Facebook.

Components

Facebook IdP Adapter

- This adapter allows PingFederate to communicate with the Facebook API to process sign-on requests and get user information.

Intended audience

This document is intended for PingFederate administrators.

If you need help during the setup process, see the following resources:

- The following sections of the PingFederate documentation:
  - Managing IdP adapters
  - Authentication policies
- Facebook Login Overview in the Facebook documentation

System requirements

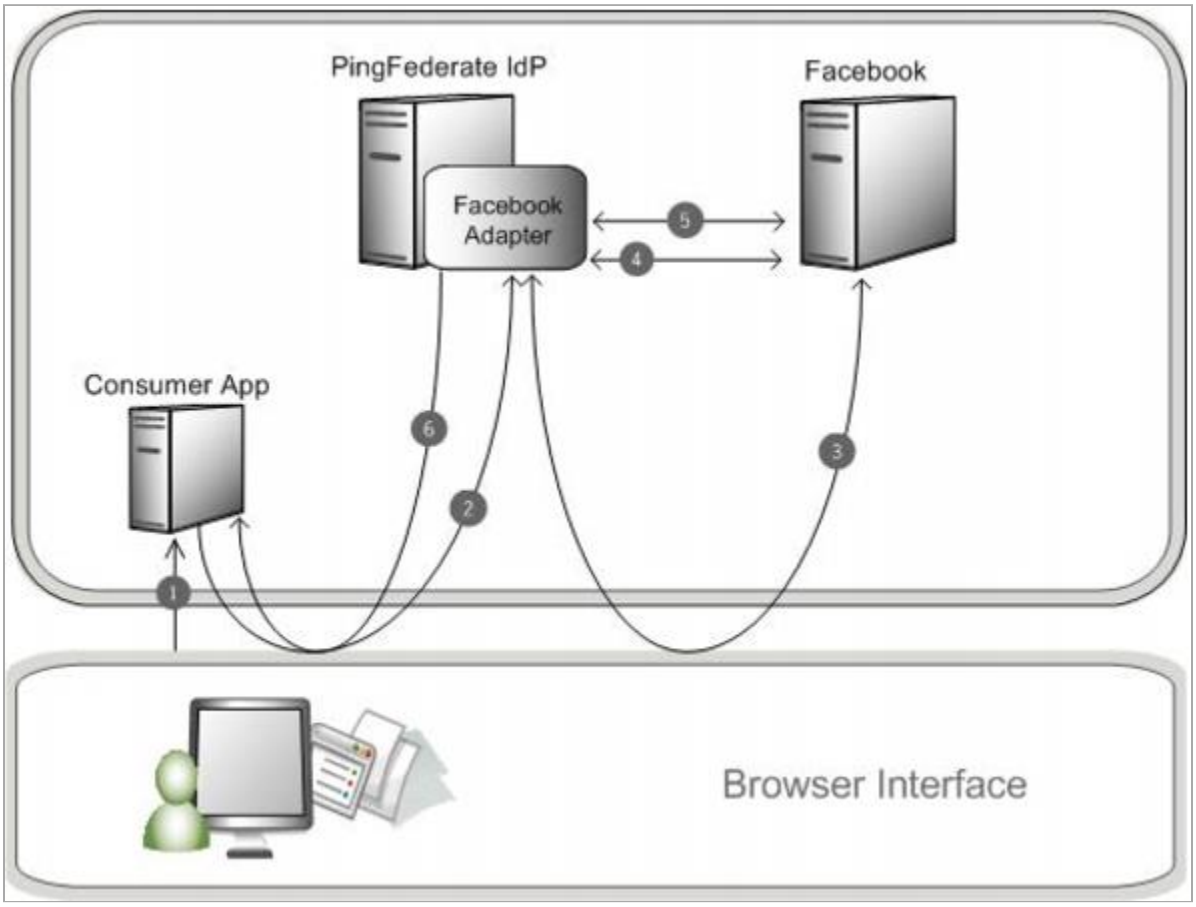
- PingFederate 9.0 or later

Overview of the SSO flow

With the Facebook Login Integration Kit, PingFederate includes the Facebook API in the sign-on flow.

The following figure illustrates a service provider (SP)-initiated single sign-on (SSO) scenario in which PingFederate authenticates users to an SP application using the Facebook IdP Adapter.

Copyright © 2025 Ping Identity Corporation



Description

1. The user opens a web application and chooses the **Login with Facebook** option.
2. The sign-on link points to the Facebook IdP Adapter, which redirects the browser...
3. ...to Facebook for authentication with a list of requested permissions.

   The user authenticates their identity and then authorizes the requested permissions. Facebook redirects the browser to the PingFederate Facebook IdP Adapter authorization callback endpoint with an authorization code.

   If the user fails to authenticate or does not authorize the request, the response includes an error code instead.
4. The Facebook IdP Adapter makes an HTTP request to the Facebook API to obtain an access token. It provides the app ID and secret, and the authorization code. The Facebook API returns an access token.
5. The Facebook IdP Adapter requests user information from the Facebook API. It provides the access token and an "app secret proof".

   > **Note**
   >
   > For information about the app secret proof, see Securing Graph API Requests in the Facebook documentation.
6. PingFederate redirects the user to the web application with the user information from Facebook.

Copyright © 2025 Ping Identity Corporation

Facebook permissions

Facebook uses permissions to determine which user attributes and data an application has access to. You can configure the Facebook IdP Adapter to request additional permissions on behalf of your application.

By default, the Facebook Login Integration Kit gives your application access to the user's public profile, including their name, profile picture, gender, and friends. If your application needs to access other user data, such as their email address or photos, you must request additional permissions. You can enter permissions when you get to the Configuring an adapter instance part of the setup process.

When a user signs on, Facebook asks them to give your application access to their photos. If they agree, the Facebook IdP Adapter sends an access token to your application along with the standard user attributes. When your application makes an HTTP request to the Facebook API, it can include this access token to get the photos from Facebook.

For more information and a list of permissions, see Permissions with Facebook Login and Permissions Reference in the Facebook documentation.

Setup

Upgrading an existing deployment

If you're upgrading from a previous version of the Facebook Login Integration Kit, back up your PingFederate configuration and note your adapter configuration.

Steps

1. Back up your current PingFederate configuration as shown in Exporting an archive in the PingFederate documentation.
2. On the PingFederate administrative console, note the following from your adapter instance configuration:
    - Scope permissions
    - Extended contract attributes
    - Attribute mappings
3. Complete the steps in Deploying the integration files.
4. In the PingFederate administrative console, open your existing IdP adapter instance.

   Choose from:

    - For PingFederate 10.1 or later: go to **Authentication > Integration > IdP Adapters**. Select your IdP adapter instance.
    - For PingFederate 10.0 or earlier: go to **Identity Provider > Adapters**. Select your IdP adapter instance.
5. Restore the configuration settings that you noted.
6. If any of the following scope permissions in the **Scope Permissions** section are blank, configure it again, and then click **Update**:
    - `email`

Copyright © 2025 Ping Identity Corporation

- `user_hometown`
- `user_location`

7. In **Advanced Configuration Fields** section, update the API version numbers in the following endpoints:
   - **Facebook Authentication URL**
   - **Facebook Access Token URL**
   - **Facebook User Data URL**

     For example, change `https://graph.facebook.com/v2.2/me` to `https://graph.facebook.com/v8.0/me`.

     For a list of available API versions, see the Graph API Changelog in the Facebook documentation.

     To automatically use the oldest available version of the API, remove the `/v2.2` or equivalent portion of the URL. For example, `https://graph.facebook.com/me`.
8. Remove any query parameters from the **Facebook User Data URL** field.

   For example, change `https://graph.facebook.com/me?fields=id,birthday,email` to `https://graph.facebook.com/me`.
9. On the **Extended Contract** screen, add any non-core attributes that you want to include in the contract, such as those associated with additional permissions that you configured.

   If you upgraded from a previous version of the Facebook IdP Adapter, add any attributes that were previously configured but are no longer part of the core contract. For a list of core contract attributes, see Core contract attributes reference.
10. On the **Adapter Attributes** tab, set pseudonym and masking options as shown in Set pseudonym and masking options in the PingFederate documentation. Click **Next**.
11. On the **Adapter Contract Mapping** tab, configure the contract fulfillment details for the adapter as shown in Define the IdP adapter contract in the PingFederate documentation. Remap any attributes that Facebook has deprecated that you want to continue using. Click **Next**.
12. Click **Save**.

Core contract attributes reference

The Facebook IdP Adapter has a set of core contract attributes. The core contract attributes have changed across versions to accommodate changes in the Facebook API.

Core contract attributes in the Facebook IdP Adapter

Version 1.3	Version 2.0	Version 2.1
about		
access_token	access_token	access_token
	age_range	

Copyright © 2025 Ping Identity Corporation

Version 1.3	Version 2.0	Version 2.1
bio[1]		
birthday[1]		
	cover	
education[1]		
first_name	first_name	first_name
gender	gender	
hometown[1]		
id	id	id
last_name	last_name	last_name
link	link	
locale	locale	
location[1]		
		middle_name
name	name	name
		name_format
	picture	picture
quotes[1]		
		short_name
timezone	timezone	
token_expires	token_expires	token_expires
updated_time	updated_time	
verified	verified	
work[1]		

Copyright © 2025 Ping Identity Corporation

1. Attribute requires additional permissions. If configured in 1.3, it will need to be added as an extended attribute after the upgrade.

Deploying the integration files

To get started with the integration, deploy the Facebook Login Integration Kit files to your PingFederate directory.

Steps

1. Download the integration `.zip` archive from the Ping Identity Integration Directory.
2. Stop PingFederate.
3. If you are upgrading from a previous deployment, back up your customizations and delete older versions of the integration files.
    1. Back up any Facebook Login Integration Kit files that you customized in `<pf_install>/pingfederate/server/default/conf/`.
    2. Delete the following files from `<pf_install>/pingfederate/server/default/deploy`:
        - `pf-facebook-adapter-<version>.jar`
        - `json-simple-<version>.jar`
4. Delete `pf-authn-api-sdk-<version>.jar` from `<pf_install>/pingfederate/server/default/lib` if it is older than the version in the integration files.
5. Extract the `.zip` archive and merge the contents of the `dist` directory with your `<pf_install>/pingfederate` directory.
6. If you backed up any customized files, modify the new files with your customizations.
7. Start PingFederate.
8. If you operate PingFederate in a cluster, repeat steps 2-7 for each engine node.

Creating an app in Facebook

To allow PingFederate to process sign-on requests using Facebook, add PingFederate as a sign-on provider in Facebook.

Steps

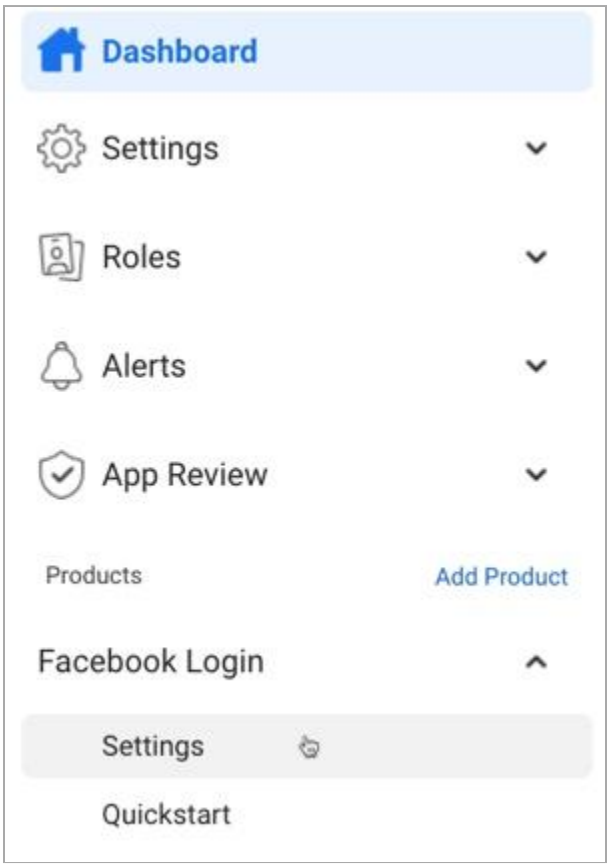
1. Sign on to the Facebook developer apps page.
2. Create the app.
    1. Click **Create App**.
    2. On the **Select an app type** modal, in the **Which products, permissions and features does your app need?** section, select **Consumer**.
    3. Configure the basic app details. Click **Create App**.
    4. Confirm your password. Click **Submit**.

Copyright © 2025 Ping Identity Corporation

3. On the **Add a Product** page, for **Facebook Login**, click **Set Up**.
4. Click **Web**.
5. On the **Web** tab, in the **Tell Us about Your Website** section, in the **Site URL** field, enter your PingFederate Facebook authentication URL based on the following. Click **Save**. Click **Continue**.

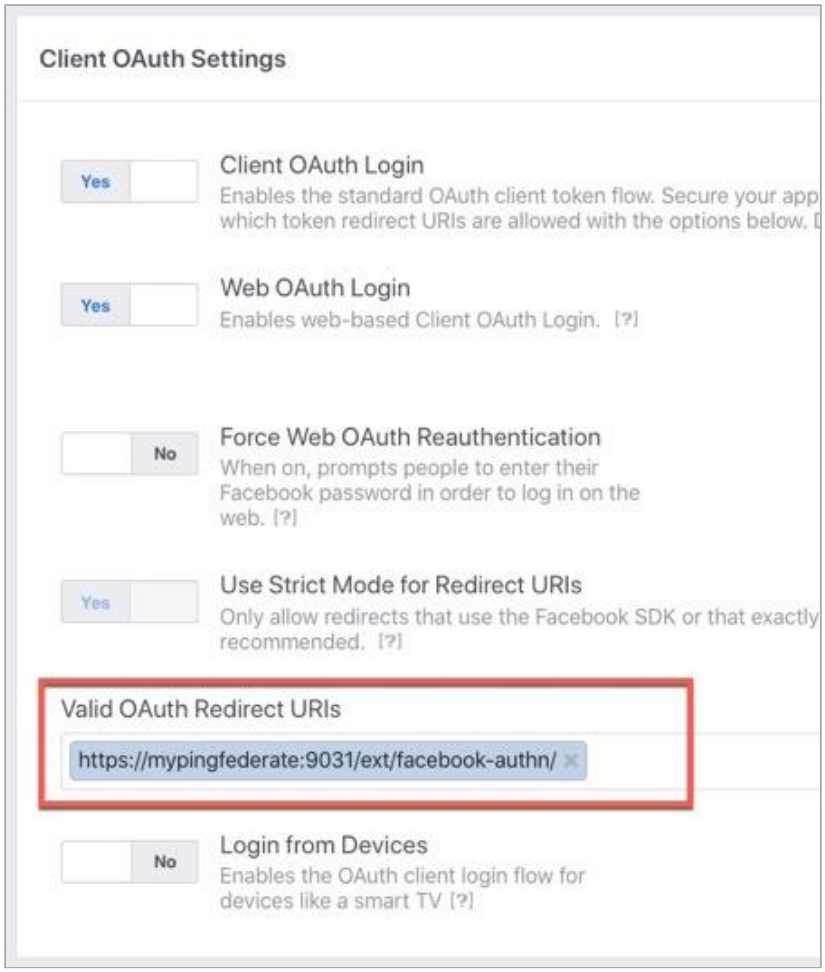
```
https://pf_host:pf_port/ext/facebook-authn/
```

6. Disregard the remaining sections on this page, your settings are saved automatically.
7. Go to **Products** > **Facebook Login** > **Settings**.

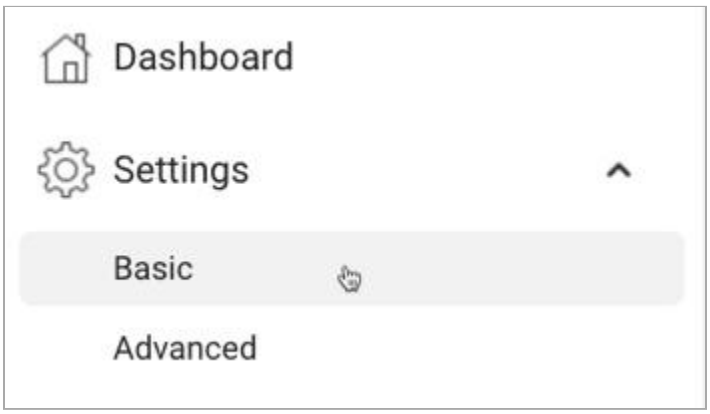


8. In the **Valid OAuth Redirect URIs** field, enter the same PingFederate URL.

Copyright © 2025 Ping Identity Corporation



9. Click **Save Changes**.
10. Go to **Settings > Basic**. Note your **App ID** and **App Secret**.



You will use these in Configuring an adapter instance.

11. **Optional:** Go to **Settings > Advanced**. In the **Security** section, turn on **Require App Secret**. Click **Save Changes**.

Copyright © 2025 Ping Identity Corporation

Configuring an adapter instance

Configure the Facebook IdP Adapter to determine how PingFederate communicates with the Facebook API.

Steps

1. In the PingFederate administrative console, create a new IdP adapter instance:

   Choose from:

   - For PingFederate 10.1 or later: go to **Authentication > Integration > IdP Adapters**. Click **Create New Instance**.
   - For PingFederate 10.0 or earlier: go to **Identity Provider > Adapters**. Click **Create New Instance**.

2. On the **Type** tab, set the basic adapter instance attributes.

   1. In the **Instance Name** field, enter a name for the adapter instance.
   2. In the **Instance ID** field, enter a unique identifier for the adapter instance.
   3. In the **Type** list, select **Facebook IdP Adapter**. Click **Next**.

3. **Optional:** On the **IdP Adapter** screen, in the **Scope Permissions** section, add any additional Facebook scopes that you want to request.

   For more information, see Facebook permissions.

   1. Click **Add a new row to 'Scope Permissions'**.
   2. In the **Additional Facebook Scope Permissions** field, enter the name of a scope.
   3. In the **Action** column, click **Update**.
   4. To add more scopes, repeat steps a-c.

4. On the **IdP Adapter** tab, configure the adapter instance by referring to Facebook IdP Adapter settings reference. Click **Next**.

5. On the **Extended Contract** tab, add any attributes that you want to include in the contract. Click **Next**.

6. On the **Adapter Attributes** tab, set pseudonym and masking options as shown in Set pseudonym and masking options in the PingFederate documentation. Click **Next**.

7. On the **Adapter Contract Mapping** tab, configure the contract fulfillment details for the adapter as shown in Define the IdP adapter contract in the PingFederate documentation. Click **Next**.

8. On the **Summary** tab, check and save your configuration:

   Choose from:

   - For PingFederate 10.1 or later: click **Save**.
   - For PingFederate 10.0 or earlier: click **Done**. On the **Manage IdP Adapter Instances** tab, click **Save**.

Copyright © 2025 Ping Identity Corporation

Facebook IdP Adapter settings reference

Field descriptions for the Facebook IdP Adapter configuration screen.

Standard fields

Field	Description
Application ID	The **App ID** that you noted in Creating an app in Facebook.
Application Secret	The **App Secret** that you noted in Creating an app in Facebook.
Site URL	The PingFederate Facebook authentication endpoint that you entered as the **Site URL** in Creating an app in Facebook: `https://pf_host:pf_port/ext/facebook-authn`
Error Redirect URL	When an error occurs in the adapter, PingFederate redirects the browser to this URL instead of the default error page. This field is blank by default.
Logout URL	The Facebook logout URL. The adapter redirects users to this address to sign them out. The default value is `https://www.facebook.com/logout.php`.
Unauthorized redirect URL	When the user does not authorize Facebook to share their user information, PingFederate redirects the browser to this URL instead of the default error page. This field is blank by default.

Advanced fields

Field	Description
Facebook Authentication URL	The Facebook endpoint used for authentication. If Facebook changes this endpoint, enter the new URL. The default value is `https://www.facebook.com/dialog/oauth`.
Facebook Access Token URL	The Facebook endpoint used to retrieve an OAuth access token. If Facebook changes this endpoint, enter the new URL. The default value is `https://graph.facebook.com/oauth/access_token`.
Facebook User Data URL	The Facebook endpoint used to retrieve user data. If Facebook changes this endpoint, enter the new URL. The default value is `https://graph.facebook.com/me`.

Copyright © 2025 Ping Identity Corporation

Field	Description
Facebook Sign-On Presentation	Determines how the user is directed to Facebook for authentication. If automatic redirects are blocked in your environment, select **Pop-up window**. If you are not using PingFederate in authentication API mode, this alternative presents a template file that requires user interaction. The default value is **Redirect**.
Facebook Template	The template file that presents the Facebook sign-on form. Applies only when **Facebook sign-on presentation** is set to **Pop-up window**. The default value is `facebook-pop-up-template.html`.
Facebook Post Auth Template	The template that adapter presents after the user signs on. This template signals to the main template or authentication widget to continue the authentication flow. Applies only when **Facebook sign-on presentation** is set to **Pop-up window**. The default value is `facebook-post-auth-template.html`.
Retry Request	Determines whether the adapter retries requests when the Facebook API responds with an error code listed in the **Retry Error Codes** field. This check box is cleared by default.
Maximum Retries Limit	Determines the number of times the adapter retries a request. The default value is `5`.
Retry Error Codes	The list of error codes that cause the adapter to retry a request. Separate multiple error codes with a comma. For a list of error codes, see Handling Errors in the Facebook documentation. The default value is `4, 7, 341`.
API Request Timeout	The amount of time in milliseconds that PingFederate waits for the Facebook API to respond to requests. A value of `0` disables the timeout. The default value is `5000`.
Connection Timeout	The amount of time in milliseconds that PingFederate allows to establish a connection with the Facebook API. A value of 0 disables the timeout. The default value is `5000`.
Proxy Settings	Defines proxy settings for outbound HTTP requests. The default value is **System Defaults**.

Copyright © 2025 Ping Identity Corporation

Field	Description
Custom Proxy Host	The proxy server host name to use when **Proxy Settings** is set to **Custom**. This field is blank by default.
Custom Proxy Port	The proxy server port to use when **Proxy Settings** is set to **Custom**. This field is blank by default.

Configuring SSO to an application within the PingFederate domain

To enable SSO to an application within the PingFederate domain, configure an adapter-to-adapter mapping.

Steps

1. On the PingFederate administrative console, create an SP adapter instance to integrate with your web application.

   For an overview of SP adapters, see Managing SP adapters in the PingFederate documentation. For most use cases, we recommend the Reference ID Adapter included in the Agentless Integration Kit.

2. Map the attributes from your Facebook IdP Adapter instance to the SP adapter instance by creating an adapter-to-adapter mapping.

   For instructions, see Adapter-to-adapter mappings in the PingFederate documentation.

3. In your web application, add sign-on hyperlink based on the following URL:

   ```
   https://pf_host:pf_port/pf/adapter2adapter.ping?
   IdpAdapterId=IdpAdapterId&SpSessionAuthnAdapterId=SpAdapterId
   ```

pf_host	The host name or IP address of the PingFederate server.
pf_port	The port number of the PingFederate server.
IdpAdapterId	The instance ID of the IdP adapter instance.
SpAdapterId	The instance ID of the SP adapter instance.

Configuring SSO to an SP outside the PingFederate domain

To enable SSO to a service provider (SP) outside the PingFederate domain, configure an SP connection and add a sign-on link to the SP application.

Steps

1. Create an SP connection and select your Facebook IdP Adapter instance on the **Authentication Source Mapping** tab.

Copyright © 2025 Ping Identity Corporation

For detailed instructions, see SP connection management in the PingFederate documentation.

2. In your web application, add sign-on hyperlink based on the following URL:

```
https://pf_host:pf_port/idp/startSSO.ping?PartnerSpId=ConnectionId
```

pf_host	The host name or IP address of the PingFederate server.
pf_port	The port number of the PingFederate server.
ConnectionId	The ID of the SP connection.

Troubleshooting

Enabling debug logging

To help with troubleshooting or monitoring, you can turn on activity logging for the adapter.

About this task

These steps are optional. For general information about logging, see Enabling debug messages and console logging in the PingFederate documentation.

Steps

1. Open the `<pf_install>/pingfederate/server/default/conf/log4j2.xml` file for editing.
2. If you want to log activity for PingFederate and all adapters, do the following.
    1. Find the following section.

    ```
    <AsyncRoot level="INFO" includeLocation="false">
            <!-- <AppenderRef ref="CONSOLE" /> -->
            <AppenderRef ref="FILE" />
    </AsyncRoot>
    ```

    2. Change `INFO` to `DEBUG`.

    ```
    <AsyncRoot level="DEBUG" includeLocation="false">
            <!-- <AppenderRef ref="CONSOLE" /> -->
            <AppenderRef ref="FILE" />
    </AsyncRoot>
    ```

    3. If you want to see the adapter activity in the console, remove the comment tags.

Copyright © 2025 Ping Identity Corporation

```
<AsyncRoot level="INFO" includeLocation="false">
        <AppenderRef ref="CONSOLE" />
        <AppenderRef ref="FILE" />
</AsyncRoot>
```

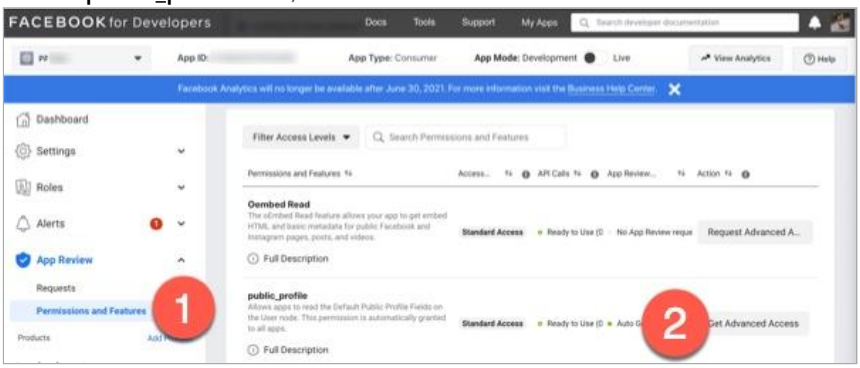
3. If you want to log activity just for the Facebook IdP Adapter, add the following line.

```
<Logger name="com.pingidentity.adapters.idp.facebook" level="DEBUG"/>
```

4. Save the file.

Troubleshooting

The following table lists situations that you could encounter with the Facebook Login Integration Kit, along with possible solutions.

Problem	Possible Cause/Solution
Users see the following error: `Feature Unavailable: Facebook Login is currently unavailable for this app, since this app does not have Advanced public_profile permission`	If you created your app on the Facebook developer site before February 2021, you need to manually request Advanced Access for the public_profile permission. 1. Sign on to the Facebook developer apps page. 2. Select the app you created in Creating an app in Facebook. 3. Go to **App Review > Permissions and Features**. 4. On the **public_profile** row, click **Get Advanced Access**.  5. Select the **I agree** check box. Click **Confirm**. 6. Confirm your password. Click **Submit**.
When attempting to sign on, the following error occurs: `Can't Load URL: The domain of this URL isn't included in the app's domains.`	For information on how to resolve this issue, see Facebook CIC can't load URL method error in the Ping Identity Knowledge Base.

Copyright © 2025 Ping Identity Corporation

Problem	Possible Cause/Solution
The sign-on hyperlink in your application does not reach the PingFederate endpoint, and you are running the PingFederate server behind a reverse proxy.	Extend the existing proxy rules within your network to allow network traffic to the following endpoint: `https://pf_host:pf_port/ext/facebook-authn/`
The user is redirected to the **Unauthorized Redirect URL** in your adapter instance configuration with an `error_msg` parameter appended to the URL.	• During authentication, the user did not agree to the requested permissions, or • There is an invalid permission configured in the adapter instance.
The `HTTP Error 400 - Bad Request` error message occurs.	The user attempted to access the site URL endpoint directly from the browser.

Release notes

Changelog

The following is the change history for the Facebook Login Integration Kit.

Cloud Identity Connector for Facebook 2.1 – August 2020

- Fixed an error that could occur when permissions were added to a new adapter instance.
- Fixed an issue that prevented single logout from completing correctly.
- Updated the core contract attributes to accommodate changes in the Facebook API.
- Added support for the Facebook **Require App Secret** option.
- Added the ability to show the Facebook sign-on prompt in a pop-up window for environments that block the automatic redirect.
- Added support for the JavaScript Widget for the PingFederate Authentication API.
- Added support for the PingFederate authentication API.
- Added settings for retrying failed requests.
- Added settings for API connection and request timeouts.
- Added settings to override the PingFederate system-default proxy settings.

Cloud Identity Connector for Facebook 2.0.1 – November 2018

- Improved HTTP client package names to avoid class-loading collisions with other add-ons, such as the CoreBlox Integration Kit.

Copyright © 2025 Ping Identity Corporation

Cloud Identity Connector for Facebook 2.0 – February 2018

- Improved user experience for adding additional scope permissions and attributes
- Added support for multivalued attributes, which are returned from the adapter as a json string.
- Improved error handling for a canceled sign-on
- Bug fixes

Cloud Identity Connector for Facebook 1.3.2 – March 2017

- Upgraded to v2.3 of the Facebook Graph API. v2.8 is also supported
- Resolved SLO issue when running this adapter behind a Load Balancer

Cloud Identity Connector for Facebook 1.3 – March 2015

- Upgraded to v2.2 of the Facebook Graph API
- URL query parameters can now be included in the Facebook User Data URL. E.g.

```
https://graph.facebook.com/v2.2/me?fields=id,birthday,email,first_name,gender,last_name,name,picture
```

Cloud Identity Connector for Facebook 1.2.2 – July 2013

- Added Masking to the application Id and application Secret fields in the Adapter GUI
- Removed access token, application Id, and application secret from being displayed in the server.log when log level is set to debug mode.

Cloud Identity Connector for Facebook 1.2.1 – May 2013

- Addressed potential security vulnerabilities found since version 1.2

Cloud Identity Connector for Facebook 1.2 – February 2013

- Added Logout URL entry field in IdP Adapter screen

Cloud Identity Connector for Facebook 1.1 – March 2011

- Modified error handling capabilities.
- Added configurable Facebook API endpoints in Advanced Fields.

Cloud Identity Connector for Facebook 1.0 – December 2010

Initial Release

Known issues and limitations

The following are known issues or limitations for the Facebook Login Integration Kit.

Copyright © 2025 Ping Identity Corporation

Known issues

There are no known issues.

Known limitations

Logout functionality is not supported with the adapter-to-adapter mapping configuration shown in Configuring SSO to an application within the PingFederate domain. Logout functionality is supported only for SP-connection configurations.

Download manifest

The following files are included in the Facebook Login Integration Kit `.zip` archive:

- `ReadMeFirst.pdf` : Contains links to this online documentation
- `Legal.pdf` : Copyright and license information
- `dist` : Contains the integration files
  - `deploy` : Contains the Java libraries
    - `pf-facebook-adapter-<version>.jar` : JAR file that contains the Facebook IdP Adapter.
  - `conf` : Contains the HTML template that presents the Facebook sign-on form.
    - `language-packs` : Contains files with customizable user-facing messages
      - `authn-api-messages.properties` : A version of the common PingFederate authentication API variable file that includes adapter-specific messages
      - `facebook-messages.properties` : A variable file that customizes the messages that appear on the template file.
    - `template` : Contains user-facing HTML template files
      - `facebook.template.html` : A page that presents the Facebook sign-on window in the **Pop-up window** sign-on presentation mode.
      - `facebook.post.auth.template.html` : A page that signals the main template or authentication widget to continue the authentication flow in the **Pop-up window** sign-on presentation mode.
      - `assets` : Contains functional scripts and files used by the template
        - `css/facebook.css` : A CSS file that customizes the appearance of the template file.
        - `fonts/end-user` : Contains template fonts and icons
          - `icons` : Contains template icons
          - `proxima-nova` : Contains template fonts
        - `images` : Contains template image files
          - `ping-logo.svg` : An image file with company branding

Copyright © 2025 Ping Identity Corporation

- `lib/pf-authn-api-sdk-<version>.jar` : A JAR file that contains the PingFederate Authentication API SDK

Workplace from Facebook Provisioner

The Workplace from Facebook Provisioner allows PingFederate to integrate with Workplace for user provisioning and single sign-on (SSO).

Components

Workplace from Facebook provisioning and SSO connector:

- Allows PingFederate to manage users in Workplace based on changes in an external user datastore.
- Optional configuration allows PingFederate to create an SSO connection to Workplace.
- Includes a quick-connection template that pre-populates some configuration settings.

Intended audience

This document is intended for PingFederate administrators. Before you start, you should be familiar with the following:

- The following sections of the PingFederate documentation:
  - Identity provider SSO configuration
  - Datastores
  - Managing IdP adapters
  - Managing digital signing certificates and decryption keys
  - SP connection management
  - Configuring outbound provisioning
  - Configuring outbound provisioning settings
- The following sections in the Workplace from Facebook documentation:
  - Getting Started
  - Workplace Help Center
  - Technical Resources
  - Account management
  - Configuring single sign-on (SSO) for Workplace
  - Account Management API
  - Custom Integrations
  - Custom integration deactivation policy

Copyright © 2025 Ping Identity Corporation

- Known issues and limitations

System requirements

- PingFederate 9.0 or later.
- A Workplace system administrator account.
- To allow PingFederate to make outbound connections to the Workplace API, you might need to whitelist the following domain in your firewall: https://scim.workplace.com/

Features

- Manages users in Workplace based on changes in an external datastore that is attached to PingFederate
  - Creates, updates, and disables users
  - Allows you to enable the create, update, and disable capabilities independently
  - Provisions disabled users
- Browser-based SSO initiated by the service provider (SP) or identity provider (IdP)

User management

The Workplace from Facebook Provisioner synchronizes users from your datastore to Workplace from Facebook. The following describes the behavior of each provisioning capability.

> **Tip**
>
> You can configure the following capabilities and specify which users to provision when you get to the Creating a provisioning connection part of the setup process.

Synchronizing existing users

PingFederate synchronizes users based on the `UserName` attribute in Workplace from Facebook. If a user already exists in your datastore and Workplace from Facebook, mapping this attribute correctly links the two records together.

For example:

- In Workplace from Facebook, Janet's `UserName` is `jsmith`.
- In your datastore, Janet's `sAMAccountName` is `jsmith`.
- On the **Attribute Mapping** tab of your provisioning connection configuration, map the `UserName` attribute to `sAMAccountName`.
- When the provisioning connector runs, the datastore user is provisioned with a `UserName` of `jsmith`. That matches Janet's existing `UserName` in Workplace from Facebook, so her information in the datastore is synchronized to her Workplace from Facebook account.

Copyright © 2025 Ping Identity Corporation

User provisioning

PingFederate provisions users when any of the following happens:

- A user is added to the datastore group or filter that is targeted by the provisioning connector.
- A user with `disabled` status is added to the datastore group or filter that is targeted by the provisioning connector, and the**Provision disabled users** provisioning option is enabled. This feature is not available in all provisioning connectors.

> **Tip**
>
> You can define which users PingFederate targets for provisioning on the **Source Location** tab of your provisioning connection configuration.

User updates

PingFederate updates users when a user attribute changes in your datastore.

> **Tip**
>
> You can define which attributes PingFederate monitors for changes on the **Attribute Mapping** tab of your provisioning connection configuration.

User deprovisioning

PingFederate deprovisions users when any of the following happens:

- A user is deleted from the user store.
- A user is disabled in the user store.
- A user is removed from the datastore group or filter that is targeted by the provisioning connector.

Setup

Upgrading an existing deployment

If you're upgrading from a previous version of the Workplace from Facebook Provisioner, note your existing service provider (SP) connection configuration and create a new connection.

Upgrading an existing deployment in PingFederate 10.1 or later

Steps

1. Back up your current PingFederate configuration as shown in Configuration archive in the PingFederate documentation.
2. In the PingFederate administrative console, go to **Applications > Integration > SP Connections** and select your connection.
3. Note the attribute mappings for your existing SP connection.

   For help, see Mapping attributes in the PingFederate documentation.

Copyright © 2025 Ping Identity Corporation

4. Delete your existing SP connection.
    1. Go to **Applications > Integration > SP Connections**.
    2. For your existing connection, click **Select action**, and then click **Delete**. Click **Confirm.**
5. Complete the steps in Deploying the integration files.
6. Complete the steps in Creating a provisioning connection.
    - From **Outbound Provisioning > Manage Channels > Channel**, on the **Attribute Mapping** tab, configure the attribute mappings based on your notes.
7. **Optional:** Complete the steps in Adding SSO to a connection.

Upgrading an existing deployment in PingFederate 10.0 or earlier

Steps

1. Back up your current PingFederate configuration as shown in Configuration archive in the PingFederate documentation.
2. In the PingFederate administrative console, go to **Identity Provider > SP Connections** and select your connection.
3. Note the attribute mappings for your existing SP connection.

   For help, see Mapping attributes in the PingFederate documentation.
4. Delete your existing SP connection.
    1. Go to **Identity Provider > SP Connections > Manage All**.
    2. For your existing connection, click **Select action**, and then click **Delete**. Click **Save.**
5. Complete the steps in Deploying the integration files.
6. Complete the steps in Creating a provisioning connection.
    - From **Outbound Provisioning > Manage Channels > Channel**, on the **Attribute Mapping** tab, configure the attribute mappings based on your notes.
7. **Optional:** Complete the steps in Adding SSO to a connection.

Deploying the integration files

To get started with the integration, deploy the Workplace from Facebook Provisioner files to your PingFederate directory.

Steps

1. Download the Workplace from Facebook Provisioner `.zip` archive from the Ping Identity Integration Directory.
2. Stop PingFederate.
3. If you are upgrading from an earlier version of the integration, delete the following files from `<pf_install>/pingfederate/server/default/deploy`:
    - `pf-workplacebyfacebook-quickconnection-<version>.jar`

Copyright © 2025 Ping Identity Corporation

- `pf-workplacebyfacebook-quickconnect-<version>.jar`

4. From the `.zip` archive, copy the contents of `dist` to `<pf_install>/pingfederate/server/default/deploy`.
5. Enable the PingFederate provisioning engine.
    1. In `<pf_install>/pingfederate/bin`, open `run.properties` for editing.
    2. Change `pf.provisioner.mode` to `STANDALONE`. Save the file.

    > **Note**
    >
    > To configure the `FAILOVER` mode instead, see Deploying provisioning failover in the PingFederate Server Clustering Guide.

6. Start PingFederate.
7. If you operate PingFederate in a cluster, repeat steps 2-6 for each engine node.

Enabling provisioning and single sign-on in PingFederate

To use PingFederate for provisioning and single sign-on, configure an external datastore and set a SAML entity ID.

About this task

Your external datastore acts as the source of data for provisioning. PingFederate also uses an internal datastore to store the state of synchronization between the source datastore and the target datastore.

For more information, see Datastores and Configuring outbound provisioning settings in the PingFederate documentation.

Enabling provisioning and single sign-on in PingFederate 10.1 or later

Steps

1. In the PingFederate administrative console, configure the datastore for PingFederate to use as the source of user data.

    For help, see Datastores in the PingFederate documentation.
2. Go to **System > Server > Protocol Settings > Federation Info**.
3. In the **SAML 2.0 Entity ID** field, enter a name for PingFederate to use when identifying itself to SAML partners.
4. On the **Outbound Provisioning** tab, in the **Provisioner Data Store** list, select the internal database that will store the synchronization state. Click **Save**.

    For more information, see Configuring outbound provisioning settings in the PingFederate documentation.

Enabling provisioning and single sign-on in PingFederate 10.0 or earlier

Steps

1. In the PingFederate administrative console, configure the datastore for PingFederate to use as the source of user data.

    For help, see Datastores in the PingFederate documentation.

Copyright © 2025 Ping Identity Corporation

2. Go to **System > Protocol Settings > Roles & Protocols**.
3. Select the **Enable Identity Provider IdP Role and Support the Following** check box.
4. Select the **SAML 2.0** and **Outbound Provisioning** check boxes. Click **Next**.
5. Click the **Federation Info** tab.
6. In the **SAML 2.0 Entity ID** field, enter a name for PingFederate to use to identify itself to SAML partners.
7. On the **Outbound Provisioning** tab, in the **Provisioner Data Store** list, select the internal database that will store the synchronization state. Click **Save**.

   For more information, see Configuring outbound provisioning settings in the PingFederate documentation.

Getting SAML details from Workplace

To streamline your PingFederate configuration, copy SAML details from Workplace from Facebook to the provided metadata file.

Steps

1. Sign on to Workplace as an administrator.
2. From the **Admin Panel**, go to **Security**.
3. On the **Authentication** tab, select the **Single-sign on (SSO)** check box.

> **Note**
>
> Turning on single sign-on (SSO) lets you get the details for the outbound provisioning configuration. Once you have the configuration details, you can turn off the Workplace from Facebook SSO setting or finish configuring SSO in PingFederate. For instructions, see Configuring single sign-on.

4. In the **SSO Providers** section, click **Add new SSO Provider** or open preferences for an existing one.
5. Note the company ID from the **Audience URL** field. For example, if your audience URL is https://www.workplace.com/company/1234567891234567, your ID is `1234567891234567`.
6. Note the company subdomain from the **Recipient URL** field. For example, if your recipient URL is https://mycompany.workplace.com/work/saml.php, your subdomain is `mycompany`.



> **Note**
>
> You will use the ID and subdomain details again in Creating a provisioning connection.

7. Close the **Single Sign-On (SSO) Setup** dialog.

Copyright © 2025 Ping Identity Corporation

8. Clear the **Single-sign on (SSO)** check box.
9. Copy the SAML details to the `.xml` metadata file used to configure the connection in PingFederate.
    1. In a text editor, open the `workplacebyfacebook-saml-metadata.xml` file that is included in the Workplace from Facebook Provisioner `.zip` archive.
    2. In the `.xml` file, replace *COMPANY_ID* with the company ID that you noted.
    3. In the `.xml` file, replace *COMPANY_SUBDOMAIN* with the subdomain that you noted.
    4. Save the changes to `workplacebyfacebook-saml-metadata.xml`.

       You will use this file in Configuring provisioning.

Configuring provisioning

To allow PingFederate to manage users in Workplace from Facebook, configure SAML in Workplace from Facebook, and then create a connection in PingFederate.

Steps

1. Complete the steps in Creating a custom integration and access token in Workplace.
2. Complete the steps in Creating a provisioning connection.

Creating a custom integration and access token in Workplace

To allow PingFederate to access the Workplace from Facebook API, create a custom integration in Workplace from Facebook that will allow you to generate an access token.

About this task

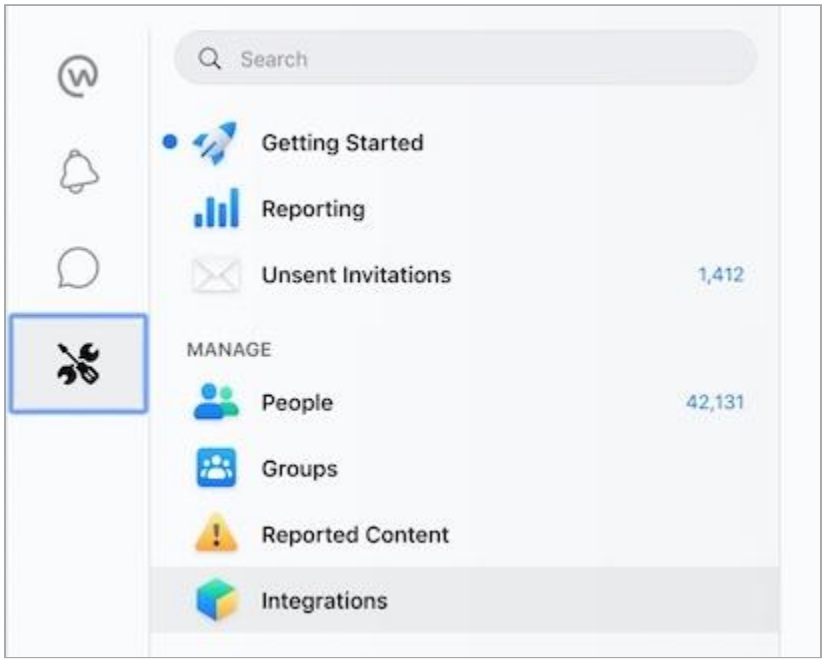
By creating a custom integration to represent PingFederate in Workplace from Facebook, you can generate an access token.

> **Note**
>
> Workplace from Facebook automatically deactivates custom integrations that have not been active for 60 days. For details, see Custom integration deactivation policy in the Workplace documentation.

Steps

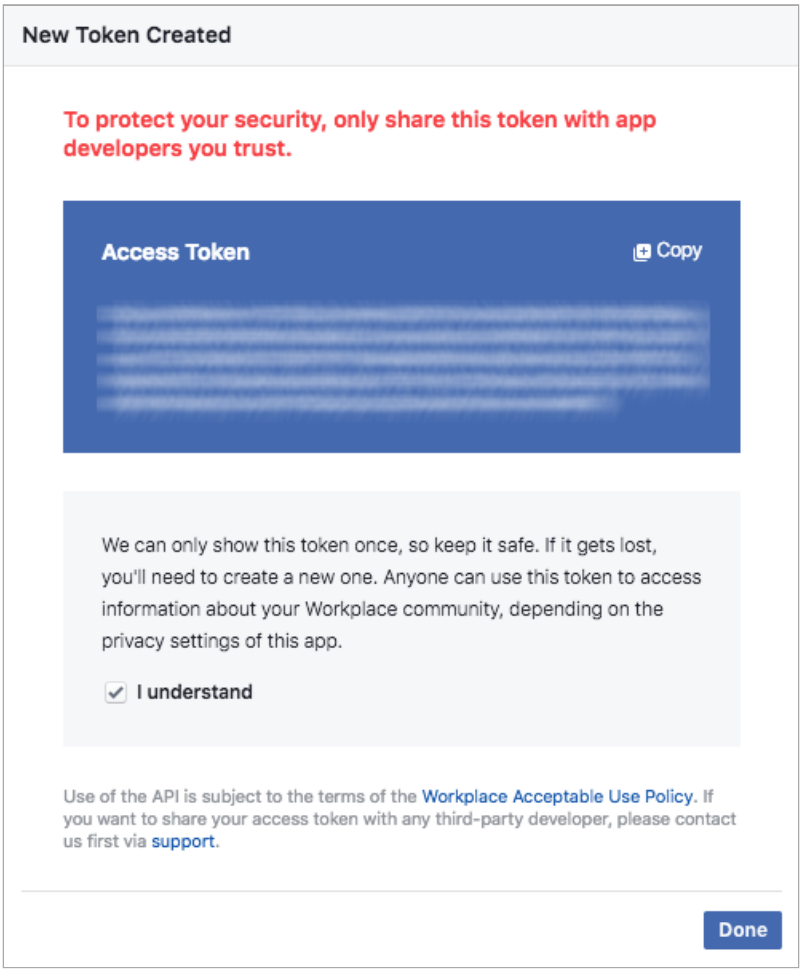
1. Sign on to Workplace as an administrator.
2. From the **Admin Panel**, go to **Integrations**.

Copyright © 2025 Ping Identity Corporation



3. Click **Create Custom Integration**.
4. On the **Create Custom Integration** dialog, enter a name and description of your choosing. Click **Create**.
5. Click **Create Access Token**.
6. On the **New Token Created** dialog, note the access token value. You will use this in Configuring provisioning.

Copyright © 2025 Ping Identity Corporation



7. Select **I understand**, and then click **Done**.
8. On the **Integration Details** window, in the **Integration Permissions** section, select **Manage accounts**.
9. **Optional:** Select **Automatically invite people to Workplace as soon as they're added using this integration**.
10. **Optional:** Enable the App secret proof feature.
    1. In the **Integration Details** section, note the **App Secret** value.

       You will use this in Configuring provisioning.
    2. In the **Security Settings** section, turn on **Require App Secret Proof**.
11. Click **Save**.

App secret proof feature

The "app secret proof" feature can protect your integration from being compromised or misused.

Copyright © 2025 Ping Identity Corporation

The provisioning connector uses an access token to communicate with Workplace from Facebook. Because access tokens are portable, they can be stolen by malicious software on a person's computer or by using a "man in the middle" attack. That access token can then be used from an unauthorized third party to generate spam or steal data.

To ensure that API calls are only ever made from authorized server-side code, Workplace from Facebook recommends that you set your custom integration to require an "app secret proof".

When the feature is enabled and configured in PingFederate, the provisioning connector uses the app secret to send a short-lived app secret proof along with the access token when making API calls to Workplace from Facebook. This can have a minor effect on performance.

For official documentation on this feature, see App Secret Proof and Securing Graph API Requests in the Workplace from Facebook documentation.

Creating a provisioning connection

To allow PingFederate to manage users in Workplace from Facebook, create a service provider (SP) connection.

Steps

1. In the PingFederate administrator console, create a new SP connection:

   Choose from:

   - For PingFederate 10.1 or later: go to **Applications > Integration > SP Connections**. Click **Create Connection**.
   - For PingFederate 10.0 or earlier: go to **Identity Provider > SP Connections**. Click **Create Connection**.

2. Configure the basic connection details with the Workplace from Facebook quick connection template.
   1. On the **Connection Template** tab, select **Use a template for this connection**.
   2. From the **Connection Template** list, select **Workplace from Facebook Provisioner**.
   3. On the **Metadata File** row, upload the workplacebyfacebook-saml-metadata.xml file that you saved in Getting SAML details from Workplace. Click **Next**.
   4. On the **Connection Type** tab select **Outbound Provisioning**. Click **Next**.
   5. On the **Connection Options** tab, click **Next**.
   6. On the **General Info** tab, in the **Connection Name** field, enter a name of your choosing. Click **Next**.
3. On the **Outbound Provisioning** tab, configure provisioning with the following details.

   For help, see Configuring outbound provisioning in the PingFederate documentation.

   1. On the **Target** screen, enter the **OAuth Access Token** value that you noted in Creating a custom integration and access token in Workplace.

      > **Note**
      >
      > PingFederate verifies the access token when you activate the channel and SP connection.

Copyright © 2025 Ping Identity Corporation

2. If you want to enable the App secret proof feature, enter the **App Secret** value that you noted in Creating a custom integration and access token in Workplace.
3. In the **SCIM URL** field, update the pre-populated value with the one of the following URLs depending on which version of SCIM you are using:
   - For SCIM 1.1 (Provisioner JAR 1.8.1 or earlier) use `https://www.facebook.com/scim/v1/`.
   - For SCIM 2.0 (Provisioner JAR 1.10 and later) use `https://scim.workplace.com/`.
4. Under **Provisioning Options**, customize the provisioning connector behavior by referring to the Provisioning options reference. Click **Next**.
5. From the **UserName Attribute Mapping** list, select the attribute that you want to use to synchronize users between the datastore and Workplace from Facebook. For details, see User management.
6. On the **Manage Channels** tab, create a channel as shown.

   For help, see Managing channels in the PingFederate documentation.

   > **Note**
   >
   > If you changed the **UserName Attribute Mapping** in **Provisioning Options**, make the same change in **Manage Channels > Channel > Attribute Mapping > userName**.

   > **Note**
   >
   > For more information about the attributes available in your channel configuration, see Supported attributes reference.

4. On the **Activation and Summary** tab, above the **Summary** section, click the toggle to turn on the connection. Click **Save**.

Provisioning options reference

The following table lists the main provisioning capabilities available in the Workplace from Facebook connection configuration.

Field Name	Description
User Create	**Selected** (default) – PingFederate creates users in Workplace. **Cleared** – PingFederate does not create users in Workplace.
User Update	**Selected** (default) – PingFederate updates existing users in Workplace. PingFederate can also re-enable disabled users. **Cleared** – PingFederate does not update existing users in Workplace.
User Disable	**Selected** (default) – PingFederate disables users in Workplace. **Cleared** – PingFederate does not disable users in Workplace.

Copyright © 2025 Ping Identity Corporation

Field Name	Description
	Note If any of the above options are cleared, PingFederate logs a warning in the user workflow section of `provisioner.log` when the related action fails.
Provision Disabled Users	**Selected** (default) – PingFederate creates users in Workplace with a "disabled" status. **Cleared** – If a user has a "disable" status, PingFederate does not create the user in Workplace.
UserName Attribute Mapping	The Active Directory attribute to map to the `UserName` attribute in Workplace. When provisioning, PingFederate uses this to match each user's `manager` between Active Directory and Workplace. The default value is **mail**. This selection must match the mapping on **Manage Channels > Channel > Attribute Mapping > userName** in your provisioning connection configuration.

Supported attributes reference

The following table lists the attributes that can be mapped for user provisioning to Workplace by Facebook.

For detailed attribute information, see Account Management API (SCIM) in the Workplace documentation.

Attribute	Description
`userName`	Unique identifier for the user, used by the user to directly authenticate with the service provider. Must be unique. This attribute is required. **Note** This value must be in the format of an email address.
`givenName`	The given name of the user, or first name in most Western languages. For example, Barbara given the full name Ms. Barbara Jane Jensen, III. This attribute is required.
`familyName`	The family name of the user, or last name in most Western languages. For example, Jensen given the full name Ms. Barbara Jane Jensen, III. This attribute is required.
`addressType`	The type of address that the `streetAddress`, `city`, `state`, `postalCode` and `country` for the user describes. For example, work, home or other. The default value is set to work.

Copyright © 2025 Ping Identity Corporation

Attribute	Description
`city`	The city of the user's full formatted address.
`country`	The country of the user's full formatted address. For example, the United States and Sweden are US and SE respectively. **Note** The value must be in ISO 3166-1 alpha 2 short code format.
`department`	The user's department.
`displayName`	The name of the user, suitable for display to end-users. The name should be the full name of the user being described if known. For example, Ms. Barbara Jane Jensen, III.
`formattedName`	The user's full name formatted for display. If the `formattedName` attribute is unmapped, the user's first name, middle name, last name, honorific prefix and honorific suffix will be sent if available. For example, Ms. Barbara Jane Jensen.
`homePhone`	The home phone number for the user. For example, +1-201-555-0123.
`honorificPrefix`	The honorific prefix of the user, or title in most Western languages. For example, Ms. given the full name Ms. Barbara J Jensen, III.
`honorificSuffix`	The honorific suffix in a user's full name. For example, III given the full name Ms. Barbara Jane Jensen, III.
`imHandle1`	The user's handle for the instant messaging service type set in `imType1`. For example, bjensen.
`imHandle2`	The user's handle for the instant messaging service type set in `imType2`. For example, bjensen.
`imHandle3`	The user's handle for the instant messaging service type set in `imType3`. For example, bjensen.
`imType1`	The instant messaging service type for `imHandle1`. For example, Skype.
`imType2`	The instant messaging service type for `imHandle2`. For example, Skype.
`imType3`	The instant messaging service type for imHandle3. For example, Skype.

Copyright © 2025 Ping Identity Corporation

Attribute	Description
`locale`	Used to indicate a default location for the user for the purposes of localizing items such as currency, date time format and numerical representations. For example, en-US. **Note** Valid values are concatenation of the ISO 639-1 two-letter language code plus a hyphen plus the ISO 3166-1 two letter country code.
`manager`	The display name or email representing the manager of this user. If using the LDAP manager attribute, a search is performed in the LDAP directory to find the matching manager's email based on the **UserName Attribute Mapping** selection on the **Target** screen of your channel configuration. After retrieving the attribute, the connector searches Workplace for this user's manager, and uses the manager's attributes to update this user's profile in Workplace. If using a custom attribute, the email of the manager is required. For example, bjensen@example.com. For more information, see Configuring provisioning. **Note** During initial synchronization, you might notice numerous errors due to the manager not being found in Workplace. The manager must match an already provisioned user or the provisioning update will fail and be retried during the next provisioning cycle. The user will still be created or updated without the manager attribute details until the retry is successful. This attribute is not supported in PingFederate 9.0 and 9.0.1.
`middleName`	The middle name or names of the user. For example, Jane given the full name Ms. Barbara Jane Jensen, III.
`mobilePhone`	The mobile phone number for the user. For example, +1-201-555-0123.
`nickName`	The casual way to address the user in real life. For example, Bob or Bobby instead of Robert.
`postalCode`	The postal code of the user's full formatted address.
`preferredLanguage`	Indicates the preferred written or spoken language for this user. Generally used for selecting a localized user interface. For example, en-US. **Note** Valid values are concatenation of the ISO 639-1 two letter language code plus a hyphen plus the ISO 3166-1 two letter country code.
`profilePhotoUrl`	The URL of a user's profile photo. Must be a valid URL to an image file ending in `.jpg`, `.gif`, or `.png`.

Copyright © 2025 Ping Identity Corporation

Attribute	Description
`profileUrl`	A fully qualified URL to a page representing the user online profile. **Note** This attribute is not longer updateable in version 1.10 and later.
`roles`	One or more roles the user holds. For example, Student or Faculty.
`state`	The state of the user's full formatted address.
`streetAddress`	The full street address component, which can include house number, street name, P.O. box, and multi-line extended street address information.
`timeZone`	The timezone of the user in the Olson timezone database format (for example, America/Los_Angeles).
`title`	The title of the user (for example, Vice President).
`userType`	Used to identify the organization to user relationship (for example, Contractor, Employee, Intern, Temp, External and Unknown).
`workEmail`	The work email for the user. For example, bjensen@example.com.
`workPhone`	The work phone number for the user. For example, +1-201-555-0123.
`employeeNumber`	The unique alphanumeric identifier assigned to a person. This is typically based on hiring order or association with an organization.
`costCenter`	The name of the user's cost center.
`organization`	The name of the user's organization.
`division`	The name of the user's division.

Configuring single sign-on

To allow PingFederate to act as an identity provider for Workplace from Facebook, exchange signing certificates and create a connection.

About this task

Note

Single sign-on (SSO) is not required for provisioning. If you only want to use the Workplace from Facebook Provisioner for provisioning, skip these steps.

Copyright © 2025 Ping Identity Corporation

Steps

1. Complete the steps in Adding SSO to a connection.
2. Complete the steps in Registering PingFederate as an identity provider in Workplace.

Adding SSO to a connection

To enable single sign-on (SSO) to Workplace from Facebook, modify the provisioning connection that you created in Creating a provisioning connection.

Steps

1. On the PingFederate administrator console, open your existing SP connection.

   Choose from:

   - For PingFederate 10.1 or later: go to **Applications > Integration > SP Connections**. Select your connection.
   - For PingFederate 10.0 or earlier: go to **Identity Provider > SP Connections**. Select your connection.

2. On the **Connection Type** tab select **Browser SSO Profiles**. Click **Next**.
3. On the **Browser SSO** tab, configure your SSO settings.

   1. Go to **Browser SSO > SAML Profiles** and select only **IdP-Initiated SSO** and **SP-Initiated SSO**.
   2. Go to **Browser SSO > Assertion Creation > Attribute Contract**. For **SAML_SUBJECT**, select **urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress**.
   3. On the **Authentication Policy Mapping** tab, select or create an authentication source that maps `SAML_SUBJECT` to an email attribute that matches the email addresses used in Workplace from Facebook.

   For configuration help, see Configuring IdP Browser SSO in the PingFederate documentation.

4. On the **Credentials** tab, configure the connection credentials as shown in Configuring credentials in the PingFederate documentation.

   - On the **Digital Signature Settings** tab, from the **Signing Certificate** list, select the certificate that you want to use with Workplace from Facebook.

5. On the **Activation and Summary** tab, above the **Summary** section, click the toggle to turn on the connection. Click **Save**.

Registering PingFederate as an identity provider in Workplace

To allow PingFederate to coordinate authentication for Workplace from Facebook, export your signing certificate and configure SAML in Workplace from Facebook.

About this task

To configure Workplace from Facebook for single sign-on (SSO), you require the following details from PingFederate.

- SAML URL, such as https://*pf_host*:*pf_port*/idp/SSO.saml2
- SAML 2.0 Entity ID that you set in Enabling provisioning and single sign-on in PingFederate

Copyright © 2025 Ping Identity Corporation

- SAML certificate
  - This is the public signing certificate from PingFederate used to sign the SAML assertion (configured in your SP connection). For more information, see Managing digital signing certificates and decryption keys in the PingFederate documentation.

The following section describes the steps to configure SSO in Workplace from Facebook. For more information, see Single Sign On Authentication in the Workplace from Facebook documentation.

Steps

1. In PingFederate, export your signing certificate.
   1. On the PingFederate admin console, go to **Security > Signing & Decryption Keys & Certificates**.
   2. For the certificate that you want to use, in the **Action** column, click **Export**.
   3. On the **Export Certificate** tab, click **Next**.
   4. On the **Export & Summary** tab, click **Export**.
   5. Open the `**********.crt` file in a text editor and copy the contents.
2. Sign on to Workplace from Facebook as an administrator.
3. Go to **Admin Panel > Security**.
4. On the **Authentication** tab, select the **Single-sign on (SSO)** check box.
5. If you want to prevent users from signing on without SSO, clear the **Password** check box.
6. In the **SSO Providers** section, click **Add new SSO Provider** or open preferences for an existing one.
7. In the **Name** field, enter a name of your choosing.
8. In the **SAML URL** field, enter your PingFederate SSO endpoint. For example, `https://pf_host:pf_port/idp/SSO.saml2`.
9. In the **SAML Issuer URL** field, enter the PingFederate **SAML 2.0 Entity ID** that you set in Enabling provisioning and single sign-on in PingFederate.
10. In the **SAML Certificate** field, paste the contents of the `**********.crt` file that you exported from PingFederate.
11. Click **Test SSO**. Resolve any issues that are reported, and then click **Save**.

Release notes

Changelog

The following is the change history for the Workplace from Facebook Provisioner.

Workplace from Facebook Provisioner 1.10 - October 2022

- Added support for SCIM 2.0.
- Changed the default SCIM URL to https://scim.workplace.com/.

Copyright © 2025 Ping Identity Corporation

Workplace from Facebook Provisioner 1.8.1 - January 2021

- Rebranded to new name (formerly known as Workplace by Facebook).

Workplace from Facebook Provisioner 1.8.1 - March 2020

- Changed the default SCIM URL to https://www.facebook.com/scim/v1/.

Workplace from Facebook Provisioner 1.8 - January 2020

- Added support for the app secret security feature.
- Added the following attributes: `employeeNumber`, `costCenter`, `organization`, `division`.
- Fixed an issue that caused a JSON error when updating users that were awarded a badge or added to a People Set or Work Team.

Workplace from Facebook Provisioner 1.7.1 - September 2018

- Improved error handling and reporting when workplace users contain no ID.
- Improved check connection call by not retrieving a list of users.

Workplace from Facebook Provisioner 1.7 - February 2018

- Added support for `formattedName` attribute.

Workplace from Facebook Provisioner 1.6.2 - October 2017

- Fixed LDAP lookup issue when multiple SP connections use different LDAP data stores.

Workplace from Facebook Provisioner 1.6.1 - July 2017

- Fixed LDAP lookup issue when multiple SSO connections exist along side provisioning connections.

Workplace from Facebook Provisioner 1.6 - June 2017

- Improved Manager attribute lookup by connecting to the LDAP directory to determine manager's email address for an optimized Workplace query.

Workplace from Facebook Provisioner 1.5 - June 2017

- Support added for the profileUrl attribute
- Added a "disable" configuration option for CRUD capabilities
- Added configuration options for provisioning disabled users
- Fixed attributes being cleared in Facebook when attributes are unmapped in PingFederate
- Fixed disabling a user after initial synchronization when the user was not created via the provisioner

Copyright © 2025 Ping Identity Corporation

Workplace from Facebook Provisioner 1.4.1 - March 2017

- Fixed synchronization on update of users, that were previously created with "User Create Enabled" set to false in configurable options

Workplace from Facebook Provisioner 1.4 - October 2016

- Rebranded to new name (formerly known as Facebook At Work)
- Removed suppressEmail attribute

Workplace from Facebook Provisioner 1.3 - August 2016

- Added configuration options for CRUD capabilities
- Improved exception handling and reporting

Workplace from Facebook Provisioner 1.2 - June 2016

- Added support for the Manager attribute.

Workplace from Facebook Provisioner 1.1 - January 2016

- Added support for provisioning additional user attributes
- Minor bug fixes
- Added support for user provisioning

Workplace from Facebook Provisioner 1.0 - March 2015

- Initial release

Known issues and limitations

The following are known issues or limitations with the Workplace from Facebook Provisioner.

Known issues

- Values for the `roles` attribute are not shown in the Workspace UI or API response.

Known limitations

- Due to a limitation in PingFederate, once a user attribute is set, it can be updated but not cleared.
- Due to a limitation with PingFederate 9.0 and 9.0.1, the `manager` attribute cannot be mapped and used. There is no impact to other functionality.
- In PingFederate 8.1 and earlier, when you configure a second SP connection with the provisioning connector, the second connection might be pre-populated with the channel from the first connection. To avoid conflicts, remove this pre-populated channel and create a unique channel for each connection.

Copyright © 2025 Ping Identity Corporation

- Workplace does not support single logout (SLO).
- When using the default mapping for the manager attribute, the process for adding a manager involves an Active Directory search, followed by a call to Workplace by Facebook. This might diminish initial synchronization provisioning performance. To improve performance, you can use a custom attribute mapping to link the manager attribute to a manager's email.

Download manifest

The following files are included in the Workplace from Facebook Provisioner `.zip` archive:

- `legal/Legal.pdf` – copyright and license information.
- `dist`: Contains the integration files
  - `pf-workplacebyfacebook-quickconnection-<version>.jar` – The Workplace from Facebook Provisioner quick connection template.
- `workplacebyfacebook-saml-metadata.xml` – a modifiable metadata file used to create the connection.

Copyright © 2025 Ping Identity Corporation

ForgeRock Intelligent Access Integration Kit

The ForgeRock Intelligent Access Integration Kit allows PingFederate to communicate with ForgeRock Intelligent Access and provide a rich sign-on orchestration experience.

ForgeRock Intelligent Access can deploy at scale and across any platform. It's a software as a service (SaaS) application with self-managed options for:

- Public cloud deployment
- Private cloud deployment
- On-premises virtual machine deployment
- Bare metal server deployment

PingFederate outsources user sign-on experience to ForgeRock Intelligent Access.

Components

ForgeRock Intelligent Access IdP Adapter

When a user signs on through PingFederate, the adapter validates whether there's a valid session cookie and whether it needs to create a new session with ForgeRock Intelligent Access.

Intended audience

This document is intended for PingFederate administrators.

If you need help during the setup process, see the following resources:

- https://backstage.forgerock.com/docs/am/7.4/eval-guide/step-2-deploy-am.html

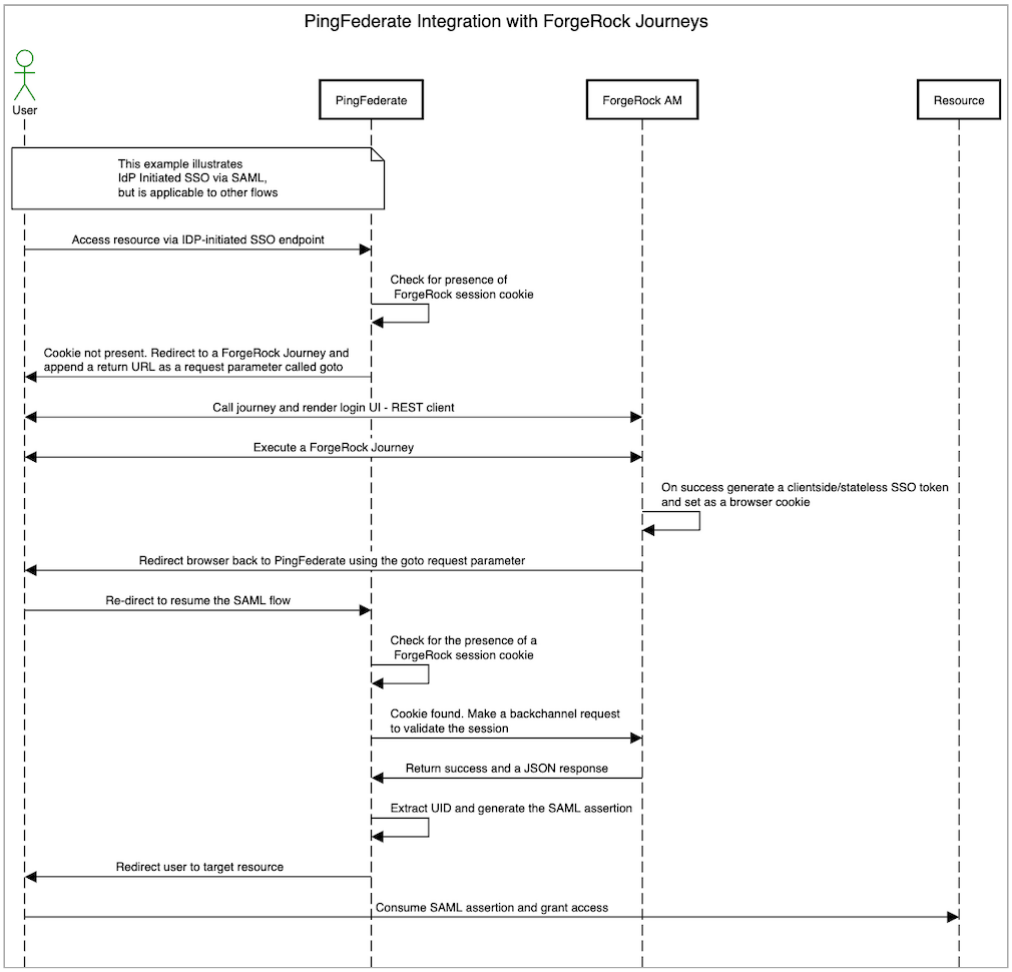
System requirements

- PingFederate 10.3 or later.
- You must deploy PingFederate and ForgeRock Intelligent Access on the same cookie domain because the current implementation relies on the presence and validation of a ForgeRock Intelligent Access single sign-on (SSO) cookie.

Overview of the SSO flow

The following figure illustrates an example SSO process flow.

Copyright © 2025 Ping Identity Corporation



1. A user initiates the sign-on process by requesting access to a protected resource.
2. If PingFederate detects that the ForgeRock Intelligent Access cookie is not present, it redirects to a configurable Intelligent Access journey to orchestrate authentication. It also appends a PingFederate URL as a request parameter to resume the flow post-login.

   If a session cookie is present, the ForgeRock solution checks the session cookie against ForgeRock Intelligent Access to ensure its validity.

Setup

Deploying the integration files

To get started with the integration, deploy the ForgeRock Intelligent Access Integration Kit files to your PingFederate directory.

Before you begin

You must sign up for a ForgeRock Intelligent Access account.

Copyright © 2025 Ping Identity Corporation

Steps

1. Download the ForgeRock Intelligent Access Integration Kit `.zip` archive from the **Add-ons** tab of the PingFederate Downloads page.
2. Stop PingFederate.
3. Extract the `.zip` archive and merge the contents of the `dist` directory with your `<pf_install>/pingfederate/server/default/deploy` directory.
4. Start PingFederate.
5. If you operate PingFederate in a cluster, repeat steps 2-4 for each engine node.

Next steps

Configure an adapter instance.

Configuring an adapter instance

Configure the ForgeRock Intelligent Access IdP Adapter to determine how PingFederate communicates with ForgeRock Intelligent Access.

About this task

To begin the integration, deploy the ForgeRock Intelligent Access Integration Kit files to your PingFederate directory.

Steps

1. In the PingFederate administrative console, go to **Authentication > Integration > IdP Adapters** and click **Create New Instance**.
2. On the Type tab, set the basic adapter instance attributes:
    1. In the Instance Name field, enter a name for the adapter instance.
    2. In the Instance ID field, enter a unique identifier for the adapter instance.
    3. From the Type list, select ForgeRock Intelligent Access IdP Adapter. Click Next.
3. **Optional:** On the **IdP Adapter** tab, in the **Response Mappings** section, map attributes from the ForgeRock Intelligent Access response to the attribute contract:
    1. In the **Local Attribute** field, enter a name of your choosing for an attribute.
    2. In the **Action** column, click **Update**.
    3. To add more attributes, repeat steps a-b.

        Result:

        These attributes become available in your PingFederate authentication policy.
4. On the IdP Adapter tab, configure the adapter instance by referring to ForgeRock Intelligent Access IdP Adapter settings reference. Click Next.

Copyright © 2025 Ping Identity Corporation

5. On the Extended Contract tab, add any attributes that you included in the ForgeRock Intelligent Access Response Mappings section of the IdP Adapter tab. Click Next.
6. On the Adapter Attributes tab, set pseudonym and masking options as shown in Set pseudonym and masking options in the PingFederate documentation. Click Next.
7. On the Adapter Contract Mapping tab, configure the contract fulfillment details for the adapter as shown in Define the IdP adapter contract in the PingFederate documentation. Click Next.
8. On the **Summary** tab, review your configuration. Click **Save**.

Next steps

Review Using ForgeRock Intelligent Access Journey.

ForgeRock Intelligent Access IdP Adapter settings reference

Field descriptions for the ForgeRock Intelligent Access IdP Adapter configuration screen.

Standard fields

Field	Description
ForgeRock Base URL	The ForgeRock Intelligent Access base URL, including context for the sign-on redirect.
Realm	The ForgeRock Realm to call.
Journey	The ForgeRock Journey or Tree to call.
Cookie Name	The ForgeRock session cookie name.

Advanced fields

Field	Description
API Request Timeout	The amount of time in milliseconds that PingFederate allows when establishing a connection with the ForgeRock Intelligent Access API or waiting for a response to a request. A value of `0` disables the timeout. The default value is `5000`.
Proxy Settings	Defines proxy settings for outbound HTTP requests. The default value is **System Defaults**.
Custom Proxy Host	The proxy server host name to use when **Proxy Settings** is set to **Custom**. This field is blank by default.

Copyright © 2025 Ping Identity Corporation

Field	Description
Custom Proxy Port	The proxy server port to use when **Proxy Settings** is set to **Custom**. This field is blank by default.

Using ForgeRock Intelligent Access Journey

To use a ForgeRock Intelligent Access Journey, you must install ForgeRock Access Management and perform the additional configuration steps described in this document.

Before you begin

Complete the steps in the ForgeRock Access Management (ForgeRock AM) installation guide.

About this task

Perform the following configuration steps in the ForgeRock Access Management admin console:

Steps

1. If you do not have an alpha realm in your environment yet:
    1. Go to **Realms** and click **New Realm**.
    2. In the **Name** field, enter `alpha`.
    3. Click **Use Client-Side Sessions**.
    4. Click **Create** to save your configuration.

    > **Important**
    >
    > You should have an alpha realm because it's best practice to reserve the root realm for administrative operations. Learn more about configuring a realm in Create a new realm.

2. Enable goto and redirects for the validation service.

    By default, ForgeRock AM denies all goto and redirects after the sign on flow is complete.

    1. In the alpha realm, go to **Services**.
    2. If the **Validation Service** is not in the list of services, click **Add a Service** and in the **Choose a service type** drop-down list, select **Validation Service.**
    3. In the **Valid goto URL Resources** field, enter one or more valid URL patterns to allow.

        Example:

        - https://*<my-ping_url>*:*/*
        - https://*<my-ping_url>*:*/**?**

3. Configure push authentication journeys by completing the steps in the push authentication journeys guide.

Copyright © 2025 Ping Identity Corporation

4. **Optional:** To adjust authentication session lifetimes:

    1. In the alpha realm, go to **Services**, click **Session**, then click **Create**.
    2. On the **Dynamic Attributes** tab, enter the desired values in the **Maximum Session Time** and **Maximum Idle Time** fields.
    3. Click **Save Changes**.

Troubleshooting

Enabling debug logging

To help with troubleshooting or monitoring, you can turn on activity logging for PingFederate, the ForgeRock Intelligent Access IdP Adapter, or both.

About this task

You can use logging for troubleshooting and analytics.

Learn more about logging in Enabling debug messages and console logging in the PingFederate documentation.

Steps

1. Open the `<pf_install>/pingfederate/server/default/conf/log4j2.xml` file for editing.
2. To log activity for PingFederate and all adapters:
    1. Find the following section:

```
<AsyncRoot level="INFO" includeLocation="false">
        <!-- <AppenderRef ref="CONSOLE" /> -->
        <AppenderRef ref="FILE" />
</AsyncRoot>
```

    2. Change `INFO` to `DEBUG`:

```
<AsyncRoot level="DEBUG" includeLocation="false">
        <!-- <AppenderRef ref="CONSOLE" /> -->
        <AppenderRef ref="FILE" />
</AsyncRoot>
```

    3. **Optional:** To see the adapter activity in the console, remove the comment tags that surround the `CONSOLE` line:

```
<AsyncRoot level="INFO" includeLocation="false">
        <AppenderRef ref="CONSOLE" />
        <AppenderRef ref="FILE" />
</AsyncRoot>
```

Copyright © 2025 Ping Identity Corporation

3. To log activity relating to the ForgeRock Intelligent Access IdP Adapter:

   Choose from:

   - To log activity for the ForgeRock Intelligent Access IdP Adapter as well as its HTTPS and component activity, add the following line:

     ```
     <Logger name="com.pingidentity.adapters.forgerock" level="DEBUG"/>
     ```

   - To log activity for the adapter's HTTPS activity and other components but not for the adapter itself, add the following line:

     ```
     <Logger name="com.pingidentity.adapters.forgerock.shade" level="DEBUG"/>
     ```

   - To log activity for the ForgeRock Intelligent Access IdP Adapter but not its HTTPS or component activity, add the following lines:

     ```
     <Logger name="com.pingidentity.adapters.forgerock" level="DEBUG"/>
     <Logger name="com.pingidentity.adapters.forgerock.shade" level="INFO"/>
     ```

4. Save the file.

Release notes

Changelog

The following is the change history for the ForgeRock Intelligent Access Integration Kit.

ForgeRock Intelligent Access Integration Kit 1.0 – February 2024

- Initial release

Known issues and limitations

The following are known issues or limitations for the ForgeRock Intelligent Access Integration Kit.

Known limitations

- The adapter does not implement support for the PingFederate Authentication API, as it isn't feasible to externally generate verified access challenge response values.
- The adapter does not yet provide the ability to map simple and advanced attributes from PingFederate to send to ForgeRock. This was included in version 1.0 of the integration kit but was removed before the release because of security concerns that will be addressed in the next release.

Copyright © 2025 Ping Identity Corporation

Download manifest

The following files are included in the ForgeRock Intelligent Access Integration Kit `.zip` archive.

- `Legal.pdf` – copyright and license information
- `dist/pingfederate/server/default` – contains the integration files
    - `deploy` – contains the Java libraries
        - `pf-forgerock-intelligent-access-integration-kit-<version>.jar` – The ForgeRock Intelligent Access IdP Adapter

Copyright © 2025 Ping Identity Corporation

Overview of GitHub integrations

PingFederate GitHub Cloud Identity Connector

The PingFederate GitHub Cloud Identity Connector allows PingFederate to use GitHub as an identity provider (IdP). This allows users to access service provider (SP) applications by signing onto GitHub.

PingFederate GitHub Connector

The PingFederate GitHub Connector enables an enterprise to provision users to GitHub. A quick connection template is also included to simplify the configuration of single sign-on (SSO).

GitHub Login Integration Kit

The GitHub Login Integration Kit allows PingFederate to use GitHub as an identity provider (IdP). This allows users to access service provider (SP) applications by signing on to GitHub.

Features

- Supports the PingFederate PingFederate authentication API.
- Supports the JavaScript Widget for the PingFederate Authentication API.

Components

GitHub IdP Adapter

- Allows PingFederate to communicate with the GitHub API to process sign-on requests and get user information.

Templates

- Allows the adapter to prompt the user to sign on. The template can be presented with a browser redirect or as a pop-up window.
- Allows you to modify the appearance of the sign-on prompt.

Language packs

- Allows you to customize or localize the messages returned by the PingFederate authentication API and shown on the templates during authentication. Learn more in Localizing messages for end users in the PingFederate documentation.

Intended audience

This document is intended for PingFederate administrators.

If you need help during the setup process, see the following sections of the PingFederate documentation:

- Identity provider SSO configuration
- Managing IdP adapters

Copyright © 2025 Ping Identity Corporation

- Authentication policies
- SP connection management

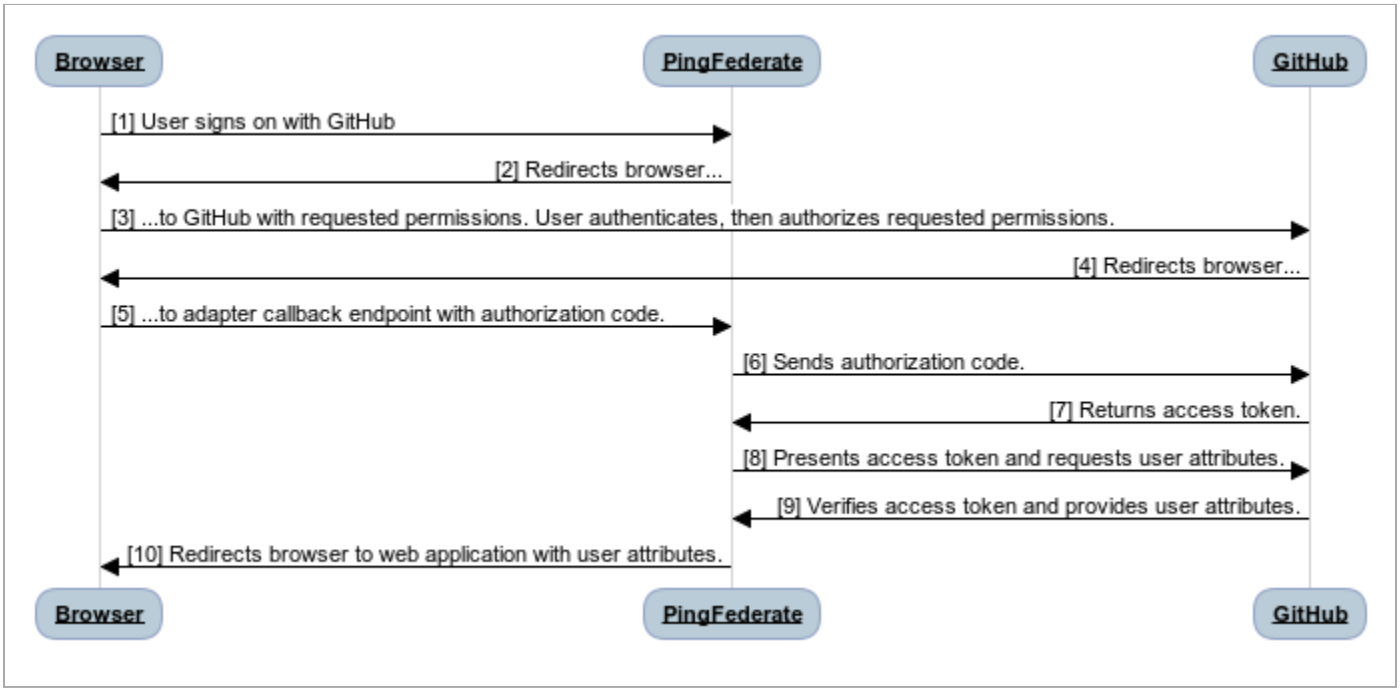
System requirements

- PingFederate 9.0 or later, or 9.3 to use the authentication API
- A standard GitHub account

Overview of the SSO flow

With the GitHub Login Integration Kit, PingFederate includes the GitHub authentication API in the sign-on flow.

The following figure illustrates a service provider (SP)-initiated single sign-on (SSO) scenario in which PingFederate authenticates users to an SP application using the GitHub IdP Adapter.



Description

1. The user opens a web application and chooses the GitHub sign-on option.
2. The sign-on link points to the GitHub IdP Adapter, which redirects the browser...
3. ...to GitHub with the client ID and a list of requested permissions. On GitHub, the user authenticates their identity and then authorizes the requested permissions.
4. GitHub redirects the browser...
5. ...to the GitHub IdP Adapter authorization callback endpoint with an authorization code.

   If the user fails to authenticate or does not authorize the request, the response includes an error code instead.

Copyright © 2025 Ping Identity Corporation

6. PingFederate sends GitHub the client ID, client secret, authorization code, and the PingFederate authorization callback URL.
7. GitHub returns an access token.
8. PingFederate sends GitHub a request for user attributes and presents the access token.
9. GitHub verifies the access token and provides the user information.
10. PingFederate redirects the user to the web application with the user attributes.

Authentication API Support

You can use the PingFederate authentication API to integrate the GitHub IdP Adapter into your application.

The PingFederate Authentication API provides access to the current state of the authentication flow as a user steps through the PingFederate authentication policy. For more information, see PingFederate authentication API in the PingFederate documentation.

To integrate the GitHub IdP Adapter into your authentication flow, configure your application based on the states, actions, and models available in the PingFederate Authentication API Explorer. For help, see Exploring the Authentication API in the PingFederate documentation.

Setup

Deploying the integration files

To get started with the integration, deploy the GitHub Login Integration Kit files to your PingFederate directory.

About this task

> **Note**
>
> If you operate PingFederate in a cluster, the following steps refer to the console node.

Steps

1. Download the GitHub Login Integration Kit `.zip` archive from the Ping Identity Integration Directory.
2. Stop PingFederate.
3. If you're upgrading an existing deployment, back up your customizations and delete earlier versions of the integration files:
    1. Back up any GitHub Login Integration Kit files that you customized in the `<pf_install>/pingfederate/server/default/conf/` directory.
    2. Delete the `pf-github-idp-adapter-<version>.jar` file from your `<pf_install>/pingfederate/server/default/deploy` directory.
4. Extract the `.zip` archive and merge the contents of the `dist` directory with your `<pf_install>/pingfederate` directory.

Copyright © 2025 Ping Identity Corporation

5. If there is more than one version of the `pf-authn-api-sdk-<version>.jar` file in your `<pf_install>/pingfederate/server/default/lib` directory, delete all but the latest version of the file.
6. If you backed up any customized files, modify the new files with your customizations.
7. Start PingFederate.
8. If you operate PingFederate in a cluster, repeat steps 2-7 for each engine node.

Registering PingFederate as an OAuth application in GitHub

To allow PingFederate to process social sign-on requests with GitHub, add PingFederate as an OAuth application in the GitHub administrative console.

Steps

1. Sign on to the GitHub app settings page using a GitHub account.
2. Click OAuth Apps.
3. Click Register a new OAuth application.
4. On the Register a new OAuth application screen, in the Application name field, enter a name, such as `PingFederate`.
5. In the Homepage URL `https://pf.example.com:9031`.
6. In the Authorization callback URL field, enter the hostname (or virtual hostname) and port of your PingFederate server followed by the GitHub IdP Adapter instance endpoint, such as `https://pf.example.com:9031/ext/github-authn`.

   > **Note**
   >
   > The default endpoint is `/github-authn`. If you set a custom endpoint here, enter the matching value in the **Authorization Callback Endpoint** field of your adapter instance configuration in Configuring an adapter instance.

7. Click Register application.
8. On the OAuth application page, note the Client ID field, enter the hostname (or virtual and Client Secret) to use in Configuring an adapter instance.

Configuring an adapter instance

Configure the GitHub IdP Adapter to determine how PingFederate communicates with GitHub.

Steps

1. In the PingFederate administrative console, create a new IdP adapter instance:

   Choose from:

   - For PingFederate 10.1 or later: go to **Authentication > Integration > IdP Adapters**. Click Create New Instance.
   - For PingFederate 10.0 or earlier: go to **Identity Provider > Adapters**. Click Create New Instance.

Copyright © 2025 Ping Identity Corporation

2. On the Type tab, set the basic adapter instance attributes.
    1. In the Instance Name field, enter a name for the adapter instance.
    2. In the Instance ID field, enter a unique identifier for the adapter instance.
    3. In the Type list, select GitHub IdP Adapter. Click Next.
3. (Optional) On the IdP Adapter page, in the Optional GitHub Attributes section, override or add to the default mappings between local attributes and the attributes from the core or extended contract.

    You can find a list of available attributes in Available user attributes and the example JSON payload in JSON Pointer syntax reference.
    1. Click Add a new row to 'Optional GitHub Attributes'.
    2. In the Local Attribute field, enter the name of a local attribute.
    3. In the GitHub Attribute field, enter the JSON Pointer syntax for the value of the matching GitHub attribute.

        Learn more in JSON Pointer syntax reference.
    4. In the Action column, click Update.
    5. To add more attributes, repeat steps a - d.
4. On the IdP Adapter page, configure the adapter instance by referring to GitHub IdP Adapter settings reference. Click Next.
5. On the Actions screen, click Test Connection. Resolve any reported issues, then click Next.
6. On the Extended Contract tab, add any attributes that you want to include in the contract. Click Next.
7. On the Adapter Attributes tab, set pseudonym and masking options as shown in Set pseudonym and masking options in the PingFederate documentation. Click Next.
8. On the Adapter Contract Mapping tab, configure the contract fulfillment details for the adapter as shown in Define the IdP adapter contract in the PingFederate documentation. Click Next.
9. On the Summary tab, check and save your configuration:

    Choose from:
    - For PingFederate 10.1 or later: click Save.
    - For PingFederate 10.0 or earlier: click Done. On the Manage IdP Adapter Instances tab, click Save.

Available user attributes

GitHub supports a multitude of attributes. The most important ones are described below.

For additional attributes, see an example payload in JSON Pointer syntax reference.

Attribute	Description
access_token	The access token that GitHub provides to your application.

Copyright © 2025 Ping Identity Corporation

Attribute	Description
email	The user's public email address, as shown their GitHub profile page.
emails	The complete list of email addresses associated with the user, including primary, public, and other addresses. This data is retrieved from the email endpoint and requires the `email` scope. The result is a JSON array that is stored as a string.
id	The unique ID for the user. This ID is only used in the context of your application.
name	The user's name, as shown on the user's GitHub profile page.
primary_email	The user's email address that is marked as primary. This value is only available with the `email` scope.
username	The user's GitHub username.

JSON Pointer syntax reference

JavaScript Object Notation (JSON) Pointer defines a syntax for identifying a specific value within a JSON payload. Using the sample payload and JSON Pointer examples below, identify the attributes that you want to use to populate your attribute contract.

You can find a complete technical description of JSON Pointer syntax in JavaScript Object Notation (JSON) Pointer on ietf.org.

Copyright © 2025 Ping Identity Corporation

Example JSON payload - `user`


```
{
  "login": "octocat",
  "id": 1,
  "node_id": "MDQ6VXNlcjE=",
  "avatar_url": "https://github.com/images/error/octocat_happy.gif",
  "gravatar_id": "",
  "url": "https://api.github.com/users/octocat",
  "html_url": "https://github.com/octocat",
  "followers_url": "https://api.github.com/users/octocat/followers",
  "following_url": "https://api.github.com/users/octocat/following{/other_user}",
  "gists_url": "https://api.github.com/users/octocat/gists{/gist_id}",
  "starred_url": "https://api.github.com/users/octocat/starred{/owner}{/repo}",
  "subscriptions_url": "https://api.github.com/users/octocat/subscriptions",
  "organizations_url": "https://api.github.com/users/octocat/orgs",
  "repos_url": "https://api.github.com/users/octocat/repos",
  "events_url": "https://api.github.com/users/octocat/events{/privacy}",
  "received_events_url": "https://api.github.com/users/octocat/received_events",
  "type": "User",
  "site_admin": false,
  "name": "monalisa octocat",
  "company": "GitHub",
  "blog": "https://github.com/blog",
  "location": "San Francisco",
  "email": "octocat@github.com",
  "hireable": false,
  "bio": "There once was...",
  "public_repos": 2,
  "public_gists": 1,
  "followers": 20,
  "following": 0,
  "created_at": "2008-01-14T04:33:35Z",
  "updated_at": "2008-01-14T04:33:35Z",
  "private_gists": 81,
  "total_private_repos": 100,
  "owned_private_repos": 100,
  "disk_usage": 10000,
  "collaborators": 8,
  "two_factor_authentication": true,
  "plan": {
    "name": "Medium",
    "space": 400,
    "private_repos": 20,
    "collaborators": 0
  }
}
```


JSON Pointer syntax

Description	JSON Pointer	Example value
Name	`/name`	`monalisa octocat`

Copyright © 2025 Ping Identity Corporation

Description	JSON Pointer	Example value
Plan Name	`/plan/name`	`Medium`
login	`/login`	`octocat`

Example JSON payload - `emails`

```
[
  {
    "email": "octocat@octocat.org",
    "primary": false,
    "verified": false,
    "visibility": "public"
  },
  {
    "email": "octocat@github.com",
    "primary": false,
    "verified": false,
    "visibility": null
  },
  {
    "email": "support@github.com",
    "primary": false,
    "verified": false,
    "visibility": null
  }
]
```

GitHub IdP Adapter settings reference

Field descriptions for the GitHub IdP Adapter configuration screen.

Table 1. Standard fields

Field	Description
Client ID	The client ID that you noted in Registering PingFederate as an OAuth application in GitHub.
Client Secret	The client secret that you noted in Registering PingFederate as an OAuth application in GitHub.

Copyright © 2025 Ping Identity Corporation

Field	Description
Authorization Callback Endpoint	The PingFederate endpoint that GitHub uses to respond to authorization requests. If you set a custom endpoint in the Authorization callback URL field in Registering PingFederate as an OAuth application in GitHub, change this field to match. This default value is `/github-authn`.
Error Redirect URL	When an error occurs in the adapter, PingFederate redirects the browser to this URL instead of the default error page. This field is blank by default.

Advanced fields

Field	Description
GitHub Authorization URL	The URL that PingFederate uses to send authorization requests to GitHub. If GitHub changes this endpoint, enter the new URL. The default value is `https://github.com/login/oauth/authorize`.
GitHub Access Token URL	The URL that PingFederate uses to retrieve access tokens from GitHub. If GitHub changes this endpoint, enter the new URL. The default value is `https://github.com/login/oauth/access_token`.
GitHub User Info URL	The URL that PingFederate uses to retrieve user data from GitHub. The default value is `https://api.github.com/user`.
GitHub User Email URL	The URL that PingFederate uses to retrieve user email data from GitHub. The default value is `https://api.github.com/user/emails`.
GitHub User Logout URL	The URL that PingFederate redirects the browser to after a user logs out of the application. By default, the user is not logged out of GitHub. If you also want PingFederate to log the user out of GitHub, enter `http://github.com/logout`. **Note** The GitHub logout page does not redirect the user back to PingFederate or the application. This field is blank by default.
Scopes	The scopes that you want to request from GitHub. Separate scopes with a space. The default value is `user:email read:user`.

Copyright © 2025 Ping Identity Corporation

Field	Description
GitHub Sign-on Presentation	Determines how the adapter presents the GitHub sign-on form. ***Redirect (default)*** The adapter redirects the browser to the GitHub sign-on form. ***Pop-up window*** The adapter opens a new window with the GitHub sign-on form on a PingFederate template. Use this option if automatic redirects are blocked by your users' browsers. This setting has no effect when using the adapter through the PingFederate authentication API.
GitHub Pop-Up Template	The template file that presents the GitHub sign-on form. Applies only when GitHub Login Presentation is set to Pop-up window. The default value is `github-pop-up-template.html`.
GitHub Post-Auth Template	The template file that the adapter presents after the user signs on. Applies only when GitHub Login Presentation is set to Pop-up window. The default value is `github-post-auth-template.html`.
GitHub Messages File	The language-pack file associated with the GitHub pop-up template. The default value is `pingfederate-github-adapter-messages`.
Retry Request	Determines whether PingFederate will retry requests after it receives a response with a failure code. This checkbox is cleared by default.
Maximum Retries Limit	Determines how many times PingFederate retries a request. The default value is `5`.
Retry Error Codes	A list of response codes that you want to trigger a retry. Separate response codes with a comma. The default value is `403`.
API Request Timeout	The amount of time in milliseconds that PingFederate allows when establishing a connection with GitHub or waiting for a response to a request. A value of 0 disables the timeout. The default value is `5000`.
Proxy Settings	Defines proxy settings for outbound HTTP requests. The default value is System Defaults.
Custom Proxy Host	The proxy server host name to use when Proxy Settings is set to Custom. This field is blank by default.
Custom Proxy Port	The proxy server port to use when Proxy Settings is set to Custom. This field is blank by default.

Copyright © 2025 Ping Identity Corporation

Integrating social sign-on into your application

To complete your GitHub sign-on integration, add a sign-on hyperlink to your application.

Steps

1. Create an SP connection to your application as shown in SP connection management in the PingFederate documentation. Use your IdP adapter instance and Mapping an adapter instance in the PingFederate documentation.
    - Use the GitHub IdP Adapter instance as an authentication source. You can find help in Mapping an adapter instance in the PingFederate documentation.
2. If your application is outside the PingFederate domain, configure a service provider (SP) connection.
    1. Create an SP connection that uses your IdP adapter instance as shown in SP connection management and Mapping an adapter instance in the PingFederate documentation.
    2. In your web application, create a hyperlink to allow users to sign on to the SP application. Use the following URL and replace the variables based on the descriptions in the following table.

        ```
        https://<pf_host>:<pf_port>/idp/startSSO.ping?PartnerSpId=<ConnectionId>
        ```

Variable	Description
<pf_host>	The host name or IP address of the PingFederate server.
<pf_port>	The port number for PingFederate.
<ConnectionId>	The federation identifier of the SP for the connection that uses the GitHub IdP Adapter instance.

3. If your application is inside the PingFederate domain, configure an adapter-to-adapter mapping.
    1. On the **System > Protocol Settings > Roles & Protocols** screen, select **Enable Identity Provider (IdP) role and support for the following** and **Enable Service Provider (SP) role and support for the following**.
    2. In both the **Enable Identity Provider** and **Enable Service Provider** sections, select any protocol, such as **SAML 2.0**. Click **Save**.

        > **Note**
        >
        > PingFederate requires a protocol selection to activate the roles. The protocol that you select is not used for this integration.

    3. On the **Service Provider > Adapters** screen, create or select an adapter instance that is integrated with the application as shown in SP application integration settings in the PingFederate documentation.
    4. On the **Identity Provider > Adapter-to-Adapter Mappings** screen, configure the IdP-to-SP adapter mapping as shown in Adapter-to-adapter mappings in the PingFederate documentation.
    5. In your web application, create a hyperlink to allow users to sign on to the SP application. Use the following URL and replace the variables based on the descriptions in the table below:

Copyright © 2025 Ping Identity Corporation

`https://<pf_host>:<pf_port>/pf/adapter2adapter.ping?IdpAdapterId=<IdpAdapterId>&SpSessionAuthnAdapterId=<SpAdapterId>`

Variable	Description
<pf_host>	The host name or IP address of the PingFederate server.
<pf_port>	The port number for PingFederate.
<IdpAdapterId>	The instance ID of the GitHub IdP Adapter instance.
<SpAdapterId>	The instance ID of the SP adapter instance that has been integrated with the application.

Troubleshooting

Enabling debug logging

To help with troubleshooting or monitoring, you can turn on activity logging for the adapter.

About this task

These steps are optional. For general information about logging, see Enabling debug messages and console logging in the PingFederate documentation.

Steps

1. Open the `<pf_install>/pingfederate/server/default/conf/log4j2.xml` file for editing.
2. If you want to log activity for PingFederate and all adapters, do the following.
    1. Find the following section.

    ```
    <AsyncRoot level="INFO" includeLocation="false">
            <!-- <AppenderRef ref="CONSOLE" /> -->
            <AppenderRef ref="FILE" />
    </AsyncRoot>
    ```

    2. Change `INFO` to `DEBUG`.

    ```
    <AsyncRoot level="DEBUG" includeLocation="false">
            <!-- <AppenderRef ref="CONSOLE" /> -->
            <AppenderRef ref="FILE" />
    </AsyncRoot>
    ```

    3. If you want to see the adapter activity in the console, remove the comment tags.

Copyright © 2025 Ping Identity Corporation

```
<AsyncRoot level="INFO" includeLocation="false">
        <AppenderRef ref="CONSOLE" />
        <AppenderRef ref="FILE" />
</AsyncRoot>
```

3. If you want to log activity just for the GitHub IdP Adapter, add the following line.

```
<Logger name="com.pingidentity.adapter.idp.github" level="DEBUG"/>
```

4. Save the file.

Release notes

Changelog

The following is the change history for the GitHub Login Integration Kit.

GitHub Cloud Identity Connector 1.1 – November 2020

- Added a setting to use browser redirect or pop-up window for the sign-on presentation.
- Added customizable sign-on templates for the pop-up window presentation.
- Added customizable user-facing language-pack messages.
- Added support for the PingFederate authentication API.
- Added support for the JavaScript Widget for the PingFederate Authentication API.

GitHub Cloud Identity Connector 1.0 – August 2019

- Initial release.
- Added support for social login using GitHub credentials.
- Added support for retrieving GitHub user and email information.
- Added support for overriding or adding default mappings between local attributes and GitHub attributes from the core or extended contract.
- Added the ability to configure when to retry a failed request.
- Added the ability to configure the maximum amount of retries to perform.
- Added the ability to override the system-default proxy settings.
- Added the ability to configure API connection timeout settings.
- Added the ability to override GitHub API endpoints.

Copyright © 2025 Ping Identity Corporation

Known issues and limitations

The following are known issues or limitations for the GitHub Login Integration Kit.

Known issues

There are no known issues.

Known limitations

- The GitHub logout page does not redirect back to PingFederate. Learn more in **GitHub User Logout URL** in GitHub IdP Adapter settings reference.

Download manifest

The following files are included in the GitHub Login Integration Kit `.zip` archive:

- `ReadMeFirst.pdf` : Contains links to this online documentation
- `legal/Legal.pdf` : Copyright and license information.
- `dist` : Contains the integration files
    - `deploy` : Contains the Java libraries
        - `pf-github-idp-adapter-<version>.jar` : A `.jar` file that contains the GitHub IdP Adapter.
    - `conf` : Contains the HTML template that presents the GitHub sign-on form.
        - `language-packs` : Contains files with customizable user-facing messages
            - `authn-api-messages.properties` : A version of the common PingFederate authentication API variable file that includes adapter-specific messages
            - `pingfederate-github-adapter-messages.properties` : A variable file that customizes the messages that appear on the template files.
        - `template` : Contains user-facing HTML template files
            - `github-pop-up-template.html` : A template that opens a pop-up window to prompt the user to sign on.
            - `github-post-auth-template.html` : A template that returns the user to the PingFederate sign-on flow after they sign on with GitHub.
            - `assets` : Contains functional scripts and files used by the template
                - `css/github.css` : A CSS file that customizes the appearance of the template files.
                - `fonts/end-user` : Contains template fonts and icons
                    - `icons` : Contains template icons
                    - `proxima-nova` : Contains template fonts

Copyright © 2025 Ping Identity Corporation

- `images`: Contains template image files
  - `ping-logo.svg`: An image file with company branding
- `lib/pf-authn-api-sdk-<version>.jar`: A `.jar` file that contains the PingFederate Authentication API SDK

GitHub Provisioner

The PingFederate GitHub Provisioner enables an enterprise to provision users to GitHub. A quick connection template is also included to simplify the configuration of single sign-on (SSO).

For more information on GitHub, see the GitHub website.

Features

- Browser-based SP and IdP-initiated SSO
- Includes support for user life cycle management (including creates, updates and deletes).
- Includes configuration options for workflow capabilities (for example, the ability to disable updates).

Intended audience

This document is intended for PingFederate administrators.

If you need help during the setup process, see the following sections of the PingFederate documentation:

- SP connection management
- Identity provider SSO configuration
- Datastores
- Configuring outbound provisioning
- Configuring outbound provisioning settings

System requirements

- PingFederate 7.3 or later with Java 8.
- Might require the following endpoints to be whitelisted on the firewall to allow outbound connections:
  - https://api.github.com/scim/v2/organizations/*<organization_name>*

> **Note**
>
> To determine your organization name, see Accessing an organization.

- A GitHub "Enterprise" or "One" plan with SAML single sign-on enabled

Copyright © 2025 Ping Identity Corporation

User management

The GitHub Provisioner synchronizes users from your datastore to GitHub. The behavior of each provisioning capability is described below.

You can configure these capabilities in the Configure PingFederate for provisioning and SSO step of the setup process.

Synchronizing existing users

PingFederate synchronizes users based on the `Username` attribute in GitHub. If a user already exists in your datastore and GitHub, mapping this attribute correctly links the two records together.

For example:

- In GitHub, Janet's `Username` is `bjensen@example.com`.
- In your datastore, Janet's `mail` is `bjensen@example.com`.
- On the **Attribute Mapping** tab of your provisioning connection configuration, you map the `Username` attribute to `mail`.
- When the provisioning connector runs, the datastore user is provisioned with a `Username` of `bjensen@example.com`. That matches Janet's existing `Username` in GitHub, so her information in the datastore is synchronized to her GitHub account.

User provisioning

PingFederate provisions users when one of the following happens:

- A user is added to the datastore group or filter that is targeted by the provisioning connector.

The **Source Location** tab of your provisioning connection configuration defines which users PingFederate targets for provisioning.

User updates

PingFederate updates users when a user attribute changes in your datastore.

The **Attribute Mapping** tab of your provisioning connection configuration defines which attributes PingFederate monitors for changes.

User deprovisioning

PingFederate deprovisions users when one of the following happens:

- A user is deleted from the user store.
- A user is disabled in the user store.
- A user is removed from the datastore group or filter that is targeted by the provisioning connector.

Setup

Obtain client ID and secret from GitHub

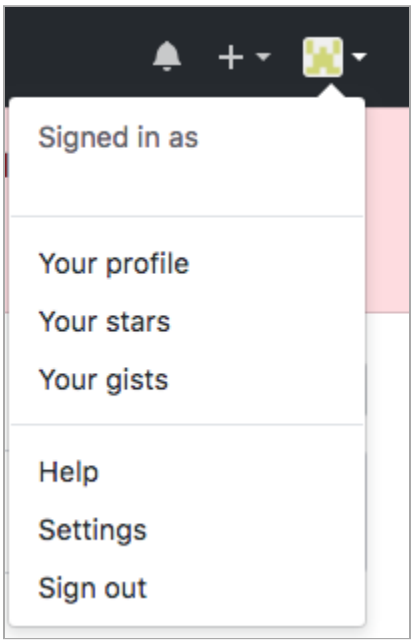
About this task

Copyright © 2025 Ping Identity Corporation

An OAuth app must be created in GitHub to access protected content in an organization. Upon OAuth app creation, client ID and secret will be generated.

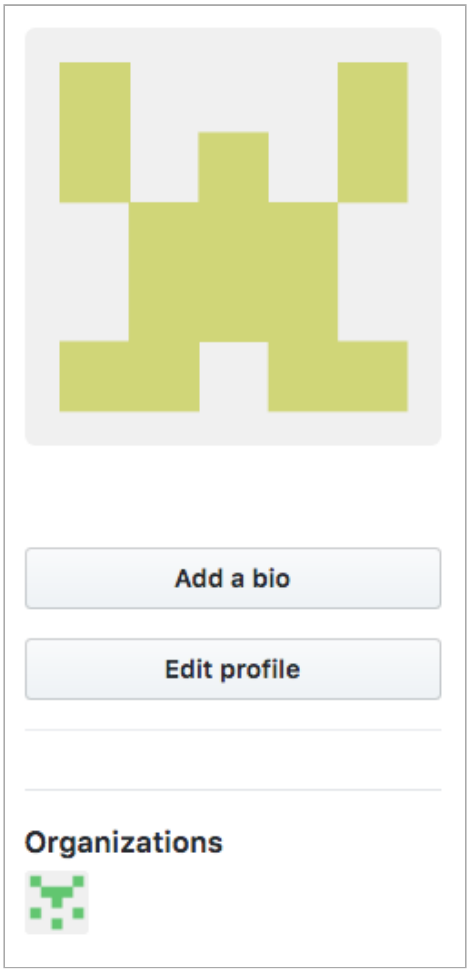
Steps

1. Log into your GitHub account as an administrative user for your organization.
2. In the top right corner of GitHub, click your profile photo, then click **Your profile**.

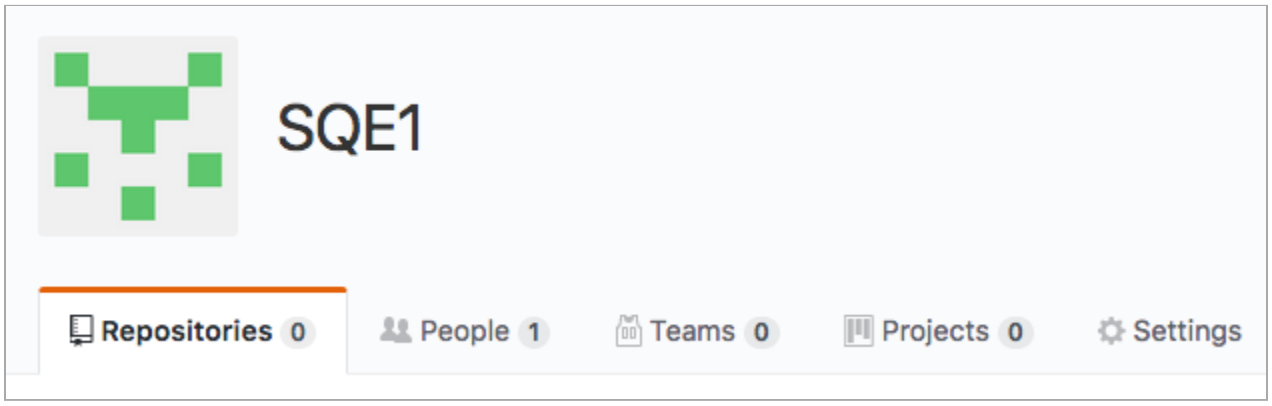


3. On the left side of your profile page, under **Organizations**, click the icon for your organization.

Copyright © 2025 Ping Identity Corporation

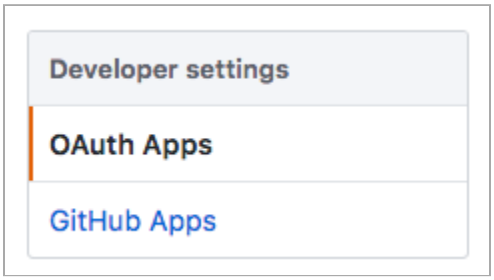


4. Under your organization name, click **Settings**.



5. In the developer settings sidebar, click **OAuth Apps**.

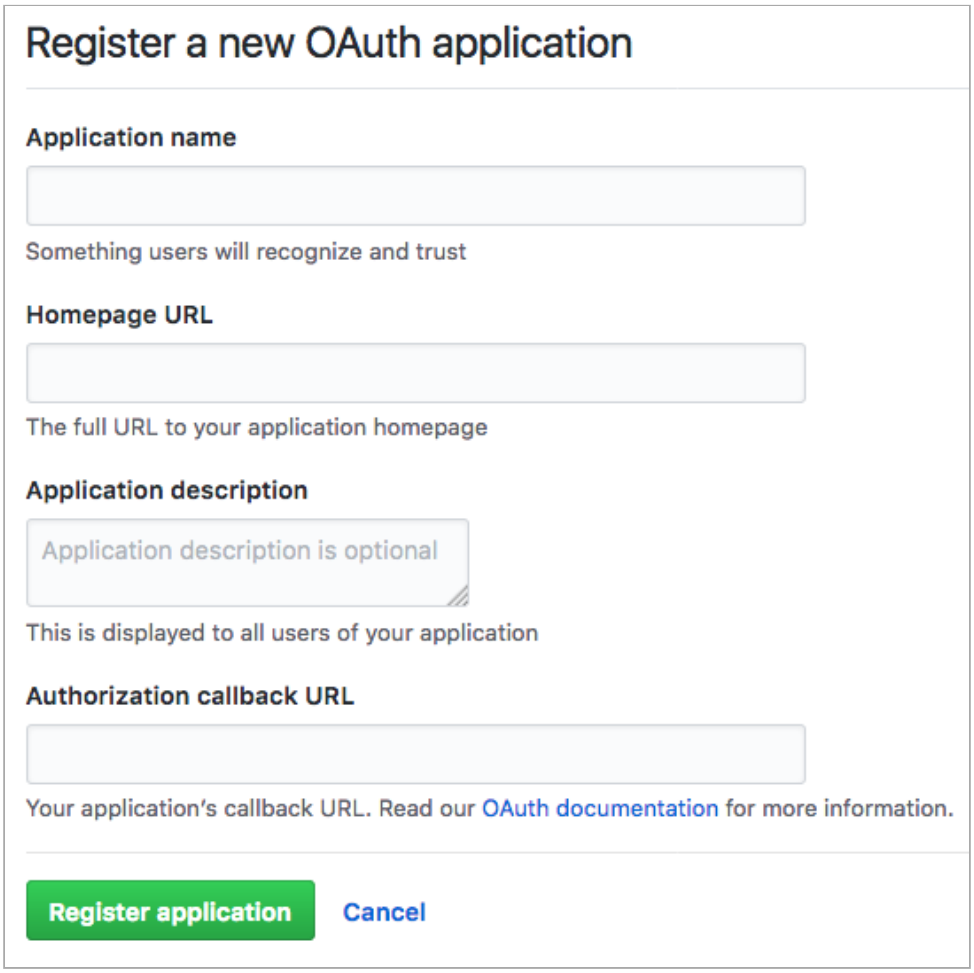
Copyright © 2025 Ping Identity Corporation



6. On the **OAuth Apps** screen, click **Register an application** or **New OAuth App**.
    - Give your application a name, such as "PingFederate Provisioning".
    - Add the following URL to the **Homepage URL** field:

      `https://www.pingidentity.com`
    - Add the following URL to the **Authorization callback URL** field:

      `https://oauth.pingone.com/ocs/ppm/rest/v1/oauth/oastempcredresponse/`



7. Click the **Register application** button to save your changes.
8. Copy the **Client ID** and **Client Secret** values to use in the next section.

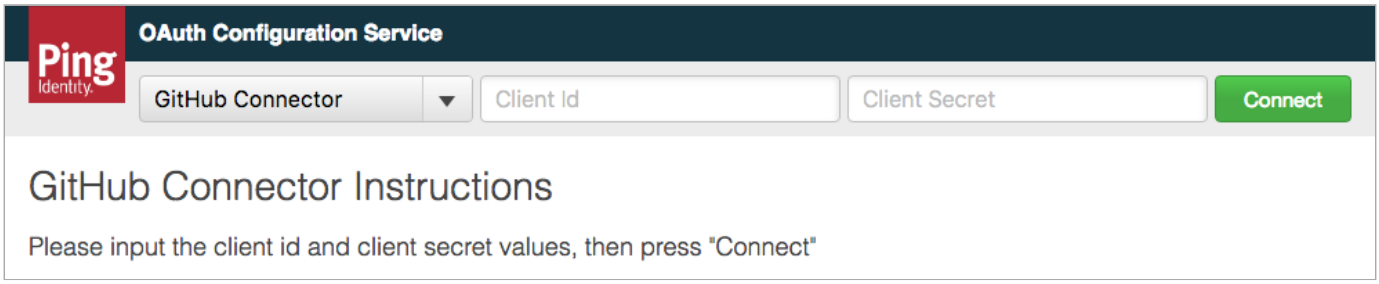
Copyright © 2025 Ping Identity Corporation

> **Note**
>
> Applications that are owned by the organization are automatically given access to the organization's resources. For more information, see About OAuth App access restrictions in the GitHub documentation.

Generate OAuth access tokens

Steps

1. Go to the Ping Identity OAuth Configuration Service.
2. Select **GitHub Connector** from the drop down menu.

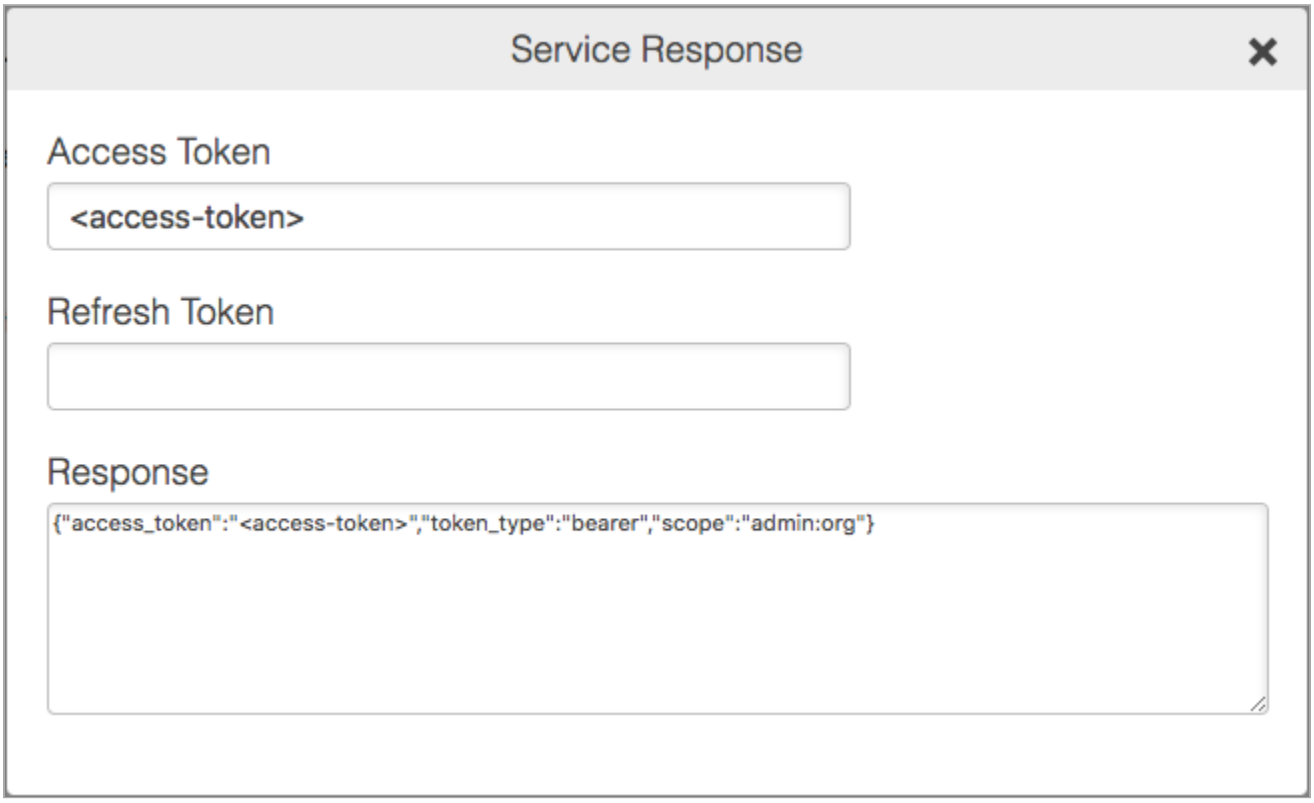


3. Enter the **Client ID** value you previously obtained into the **ClientID** field.
4. Enter the **Client Secret** value you previously obtained into the **Client Secret** field.
5. Click the **Connect** button.
6. Log into GitHub as an administrative user for your organization.

> **Note**
>
> If you are already signed into GitHub you will not be asked to log in again. Please be sure that the account you are signed in under is an administrative account.

7. Click the **Authorize** button to generate your access token.
8. You should be redirected to the OAuth Configuration Service and presented with an access token.

Copyright © 2025 Ping Identity Corporation



9. Copy the **Access Token** to use when configuring the GitHub Connector.

Obtaining PingFederate SAML 2.0 metadata

About this task

To configure GitHub for provisioning and SSO, you will require the below information. These values can be obtained from your PingFederate Identity Provider (IdP) metadata. Learn more in Exporting connection-specific SAML metadata

- PingFederate SAML Endpoint
  - Example: `https://<pf_hostname>:<pf_port>/idp/SSO.saml2`
- Identity Provider Issuer
  - This is SAML 2.0 Entity ID from PingFederate.

To export your IdP metadata:

Steps

1. Go to the **Server Configuration > Metadata Export** screen.
2. On the **Metadata Mode** screen, choose **Select information to include in metadata manually** and click **Next**.
3. On the **Protocol** screen, click **Next**.
4. On the **Attribute Contract** screen, click **Next**.
5. On the **Signing Key** screen, select the PingFederate signing certificate for use and click **Next**.

Copyright © 2025 Ping Identity Corporation

6. On the **Metadata Signing** screen, click **Next**.
7. On the **XML Encryption Certificate** screen, click **Next**.
8. On the **Export & Summary** screen, click **Export** to and save the metadata XML file for later use.

Obtain PingFederate signing certificate

About this task

The following are the steps to obtain the signing certificate used for the digital signature verification of the PingFederate IdP. When asked during Configure GitHub for provisioning and SSO, provide the certificate that you have downloaded from PingFederate. Learn more in Managing digital signing certificates and decryption keys in the PingFederate documentation.

To export an existing certificate:

Steps

1. Sign onto the **PingFederate administrative console** as an administrative user for your organization.
2. Go to **Server Configuration > Certificate Management > Signing & Decryption Keys & Certificates**.
3. Click **Export** under **Action** for the certificate you want to export.
4. On the **Export Certificate** page, select **Certificate Only** export and then click **Next**.
5. On the **Export & Summary** page, click **Export** to save the certificate file on your system.
6. When finished, click **Done**.
7. Click **Cancel** to go back to the **Server Configuration** screen.

Install the provisioner

Install and set up the PingFederate GitHub Provisioner.

Before you begin

To download the PingFederate connector, go to https://www.pingidentity.com/en/resources/downloads/pingfederate.html, click the **Add-ons** tab, then click **GitHub Connector 1.0**.

> **Important**
>
> Make sure to select **GitHub Connector 1.0**, not **GitHub EMU Connector 1.0**.

About this task

To install the PingFederate GitHub Provisioner:

Steps

1. If the PingFederate server is running, stop the server.
2. Extract the GitHub Provisioner distribution `.zip` file.

Copyright © 2025 Ping Identity Corporation

3. Copy the contents of the `dist` directory into the `<pf_install>/pingfederate/server/default/deploy` directory.
4. **Optional:** To use the provisioner for outbound provisioning, in the `<pf_install>/pingfederate/bin/run.properties` file, set the `pf.provisioner.mode` property to `STANDALONE`.

   Example:

   ```
   pf.provisioner.mode=STANDALONE
   ```

   > **Note**
   >
   > For information about using the `FAILOVER` setting for runtime deployment, see Deploying provisioning failover.
5. Start the PingFederate server.

Enable outbound provisioning

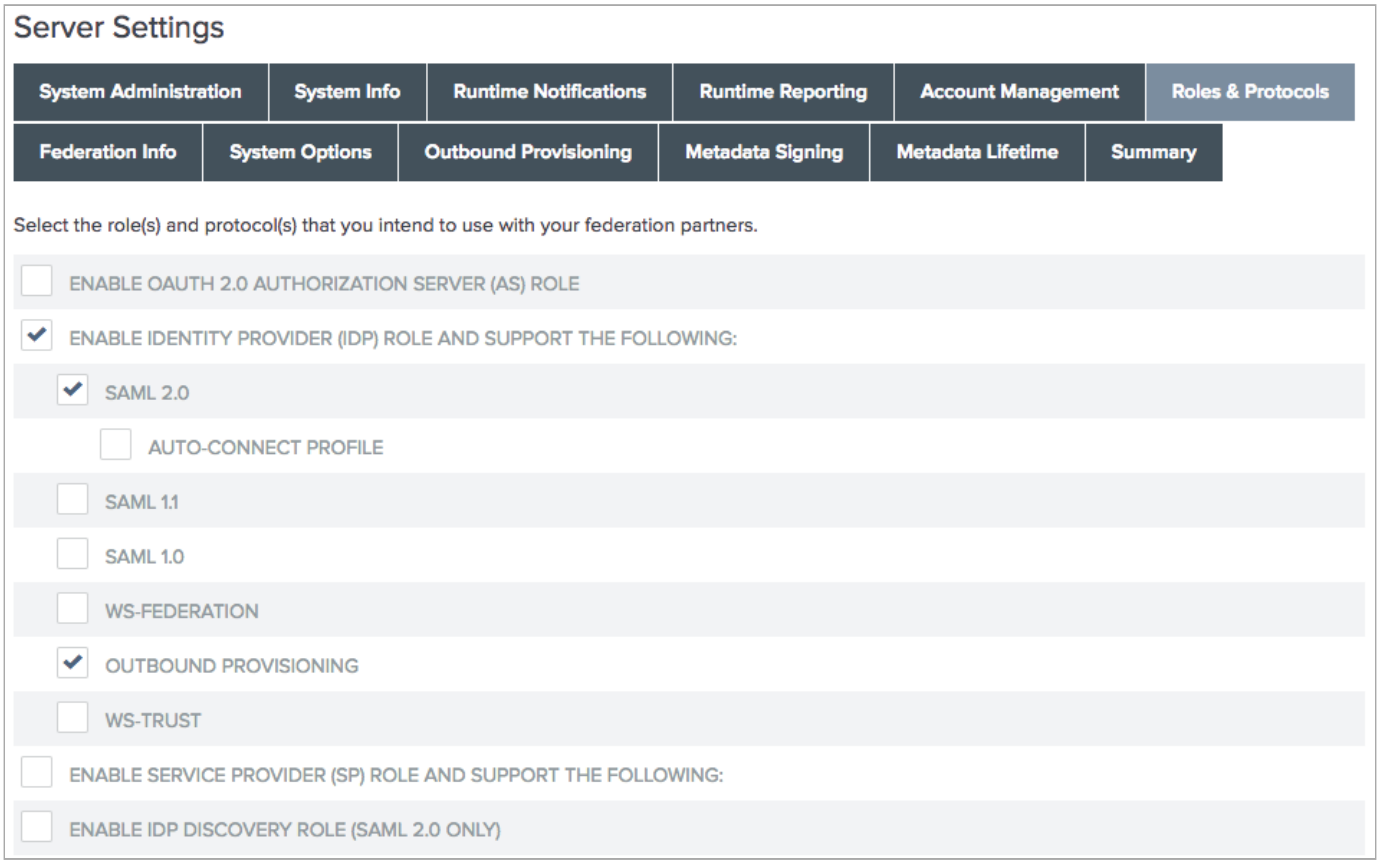
About this task

After enabling outbound provisioning in the `<pf_install>/pingfederate/bin/run.properties` file, you must also activate the outbound provisioning role in the administrative console.

Steps

1. Go to **Server Configuration > Server Settings > Roles & Protocol**.
2. Select the **Outbound Provisioning** checkbox.

Copyright © 2025 Ping Identity Corporation



> **Note**
>
> Enabling outbound provisioning adds the outbound provisioning screen and requires the selection of a database to facilitate provisioning. Learn more in Configuring outbound provisioning settings or click Help on the configuration screen.

Configure PingFederate for provisioning and SSO

About this task

To configure a connection for outbound provisioning and SSO to GitHub, follow the instructions in this section. Outbound provisioning details are managed within an SP connection and may be added to an existing SP connection.

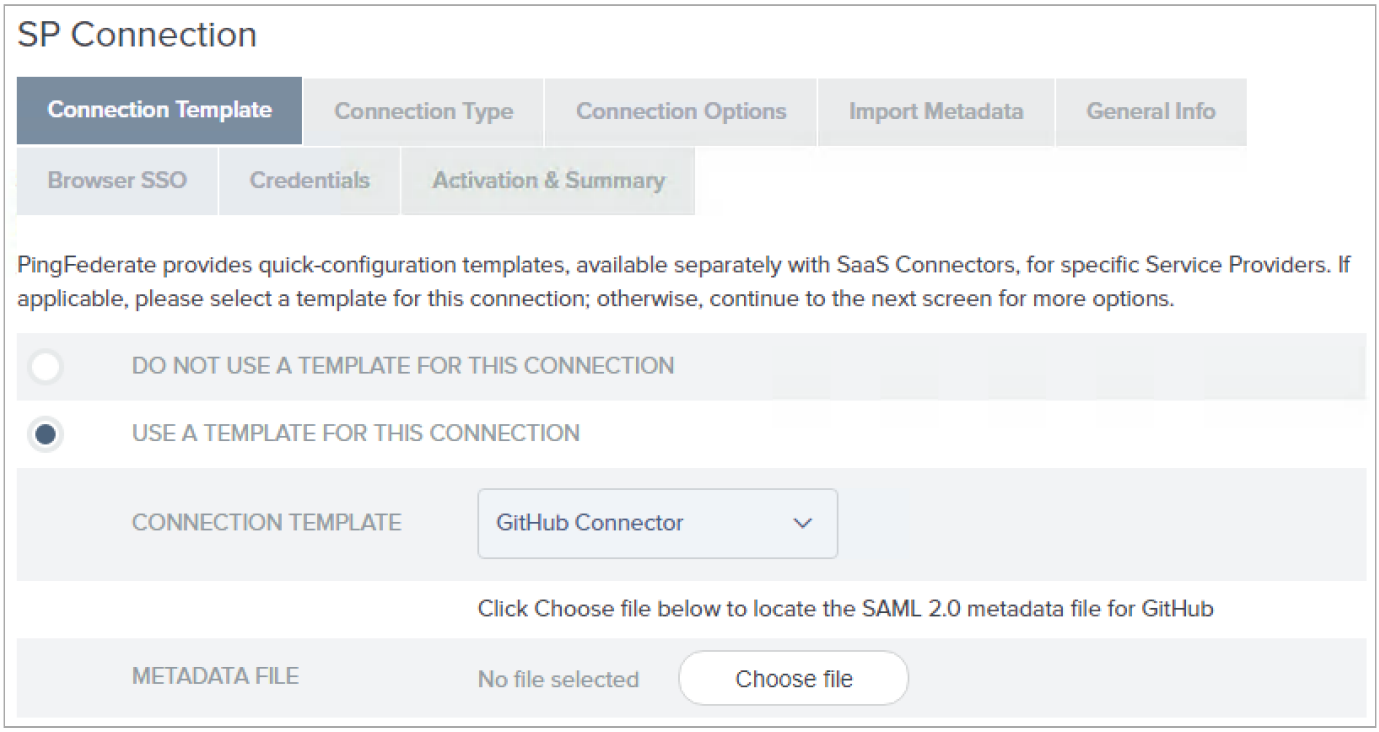
> **Note**
>
> The SCIM API requires GitHub Enterprise Cloud with SAML SSO enabled for the organization. See About SCIM in the GitHub documentation.

Steps

1. In the PingFederate administrator console, configure the datastore that PingFederate will use as the source of user data. You can find instructions in Datastores in the PingFederate documentation.
    - When targeting users and groups for provisioning, exclude the user account that you will use to administer users in your connection to GitHub. This prevents the PingFederate provisioning engine from interfering with the account that provisions users and groups.

Copyright © 2025 Ping Identity Corporation

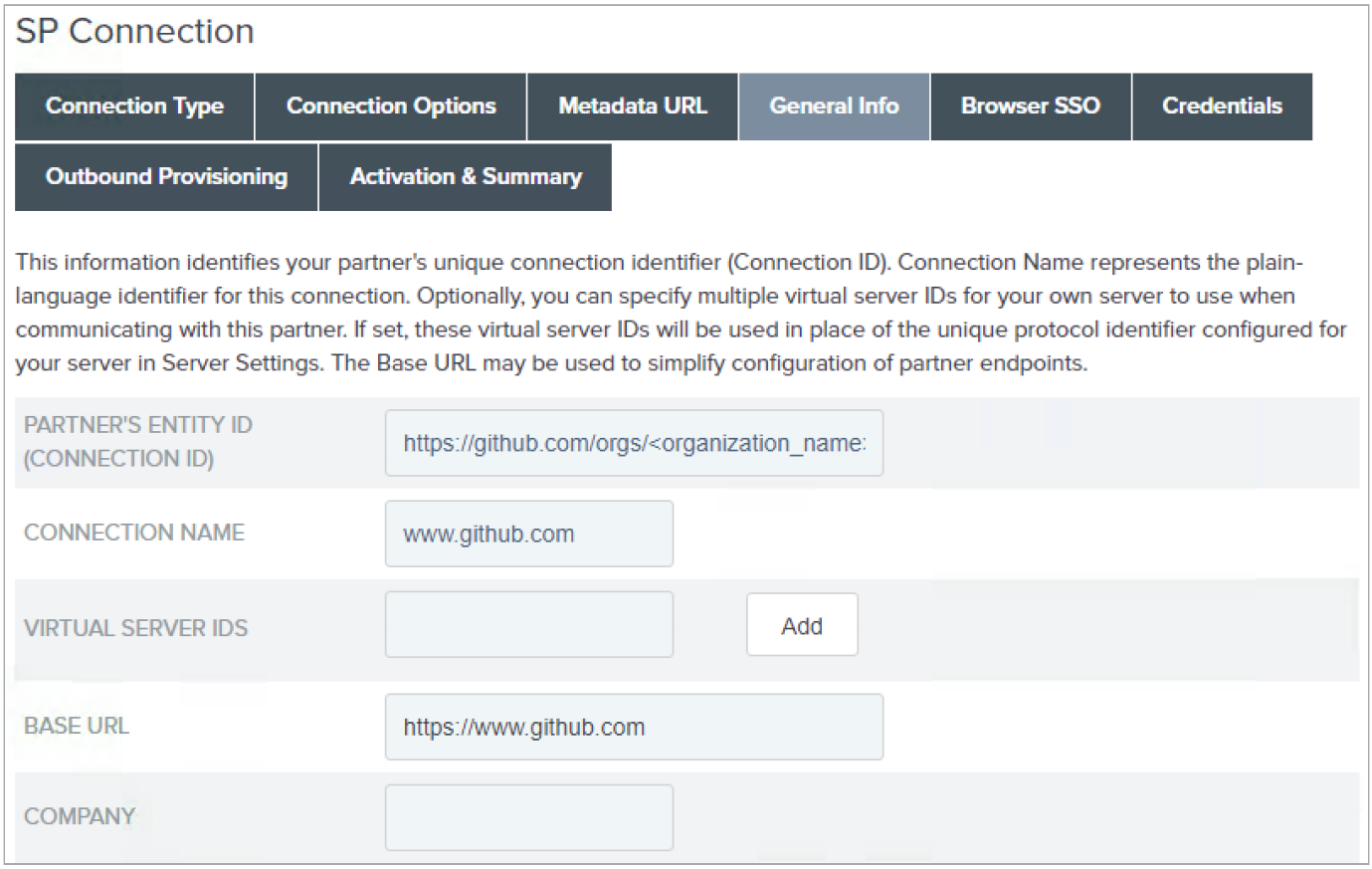
2. Create a new SP connection or select an existing SP connection from the SP Configuration menu.
3. On the Connection Template page, select Use a template for this connection and select GitHub Connector in the Connection Template list. When asked during the connection configuration steps, import the `github-saml-metadata.xml` packaged with this connector.



> **Tip**
>
> If this selection is not available, verify the connector installation and restart PingFederate.

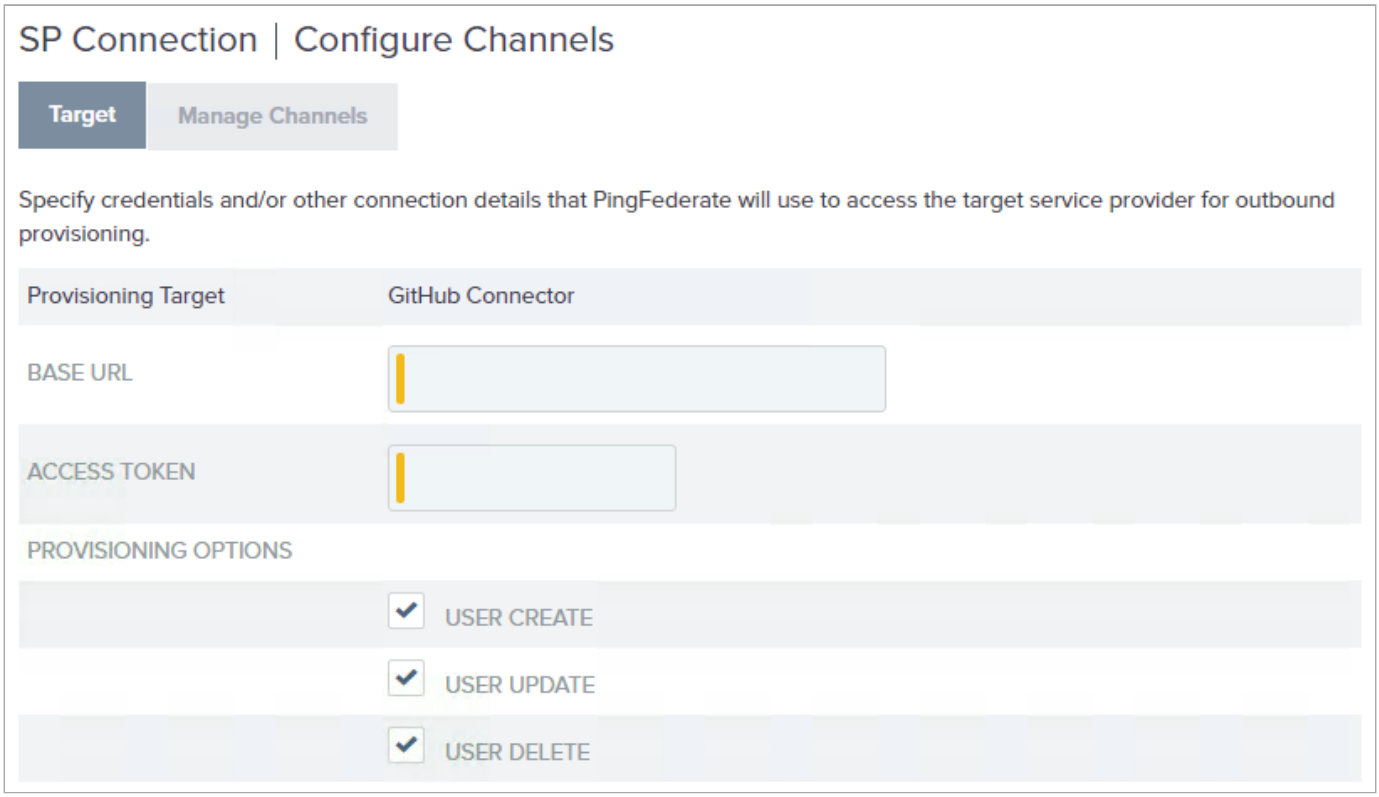
4. On the Connection Type page, ensure both the Outbound Provisioning and Browser SSO Profiles checkboxes are selected.
5. On the General Info page, the default values are taken from the metadata file you selected in step 2. In the Partner's Entity ID (Connection ID) field and update with your corresponding organization name.

Copyright © 2025 Ping Identity Corporation



6. Click Next to continue the Browser SSO configuration. You can find more information in the following sections under Identity provider SSO configuration:
    - Managing IdP adapters
    - Configure IdP Browser SSO
7. On the Assertion Creation page, click Next.
8. On the Protocol Settings page, click Configure Protocol Settings.
9. On the Summary page, navigate to Assertion Consumer Service URL.
10. On the Assertion Consumer Service URL screen, edit the existing entry. Enter the Endpoint URL corresponding to your organization name. For example, `https://github.com/orgs/<organization_name>/saml/consume`
11. Click Update and Done to proceed.
12. On the **Credentials > Digital Signature Settings** page, select the signing certificate.
13. On the Outbound Provisioning page, click Configure Provisioning.
14. On the Target page, enter the values for each field as required by the GitHub Connector.

Copyright © 2025 Ping Identity Corporation



Target screen options

Field Name	Description
Base URL	The base URL for GitHub. For example: `https://api.github.com/scim/v2/organizations/<organization_name>` **Note** To determine your organization name, see Accessing an organization in the GitHub documentation.
Access Token	The access token used by the connector to make authenticated API calls to GitHub. You can find more information about obtaining the access token in Obtain client ID and secret from GitHub and Generate OAuth access tokens.
Provisioning Options	

Copyright © 2025 Ping Identity Corporation

Field Name	Description
User Create	***True (default)*** Users will be created in GitHub. ***False*** Users will not be created in GitHub. **Note** The `provisioner.log` will display a warning within the create user workflow that the user was not created in GitHub.
User Update	***True (default)*** Users will be updated in GitHub. ***False*** Users will not be updated in GitHub. **Note** The `provisioner.log` will display a warning within the update user workflow that the user was not updated in GitHub. Enabling a previously deleted user in GitHub will trigger a create and as such, users can be enabled when User Update is set to false.
User Delete	***True (default)*** Users will be deleted in GitHub. ***False*** Users will not be deleted in GitHub. **Note** The `provisioner.log` will display a warning indicating that the user was not deleted in GitHub.

15. Click Next to continue the provisioning configuration. Learn more in the following sections under Configuring outbound provisioning in the PingFederate documentation:
    - Managing channels
    - Specifying channel information
    - Identifying the source datastore
    - Modifying source settings
    - Specifying a source location
    - Mapping attributes
    - Reviewing channel settings

Copyright © 2025 Ping Identity Corporation

> **Note**
>
> Credentials will be verified when the channel and SP connection is set to Active and provisioning is initiated.

> **Tip**
>
> If you are not ready to complete the provisioning configuration, you can click Save and return to the configuration page later.
> To return to the configuration page, select the connection from **Identity Provider** > **SP Connections** > **Manage All**.

Supported attributes reference

The following table consists of the attributes that can be mapped for user provisioning. For more information on available attributes, see SCIM in the GitHub documentation.

Attribute	Description
Username	GitHub's unique identifier for the user. This attribute is required.
Email	The email for the user (for example, `bjensen@example.com`). This attribute is required.
First Name	The given name of the user, or first name in most Western languages (for example, 'Barbara' given the full name 'Ms. Barbara Jane Jensen, III'). This attribute is required.
Last Name	The family name of the user, or last name in most Western languages (for example, 'Jensen' given the full name 'Ms. Barbara Jane Jensen, III'). This attribute is required.
External ID	A string that is an identifier for the resource as defined by the provisioning client.

Configure GitHub for provisioning and SSO

Before you begin

Below are the tasks to enable provisioning and single sign-on (SSO) on the GitHub Admin Portal.

> **Note**
>
> The SCIM API is only available on the GitHub "Enterprise" and "One" plans with SAML SSO enabled. For more information, see SCIM REST API v3.

To configure GitHub for provisioning and SSO you will require both metadata and signing certificate from your PingFederate Identity Provider (IdP) setup. For more information, see Obtaining PingFederate SAML 2.0 metadata and Obtain PingFederate signing certificate.

About this task

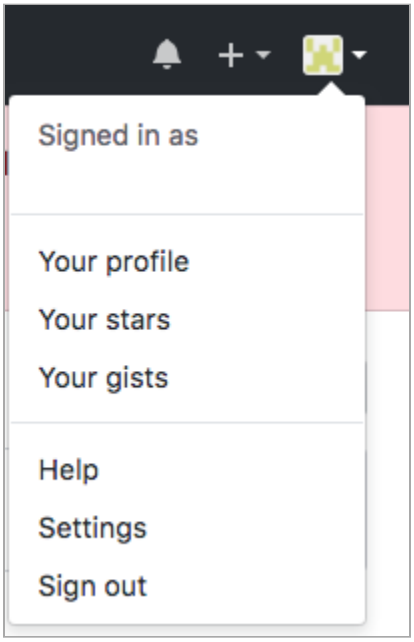
Copyright © 2025 Ping Identity Corporation

To configure GitHub for provisioning and SSO:

> **Note**
>
> For more information on how to set up SSO for GitHub, see Enabling and testing SAML single sign-on for your organization in the GitHub documentation.

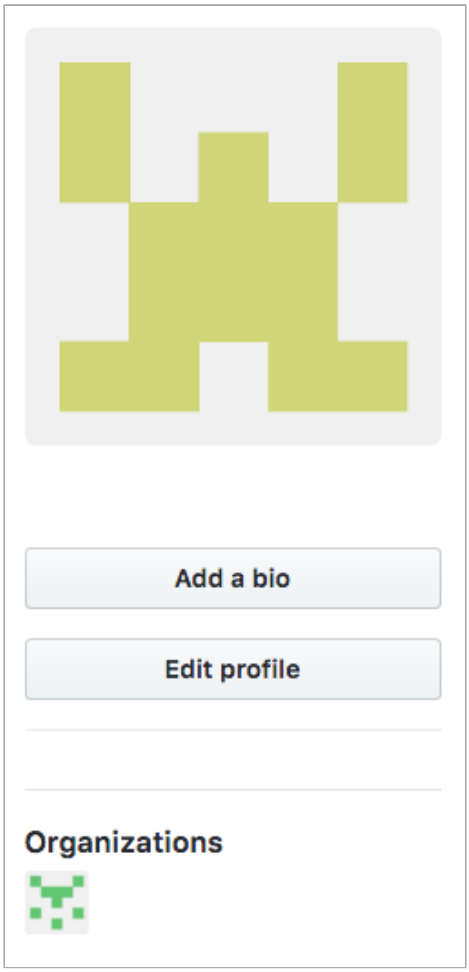
Steps

1. Log into your GitHub account as an administrative user for your organization.
2. In the top right corner of GitHub, click your profile photo, then click **Your profile**.

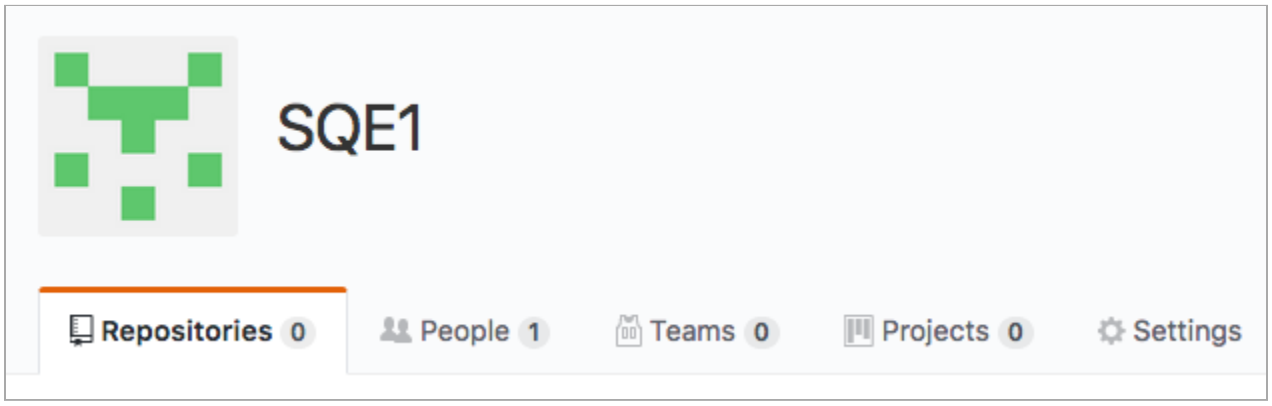


3. On the left side of your profile page, under **Organizations**, click the icon for your organization.

Copyright © 2025 Ping Identity Corporation

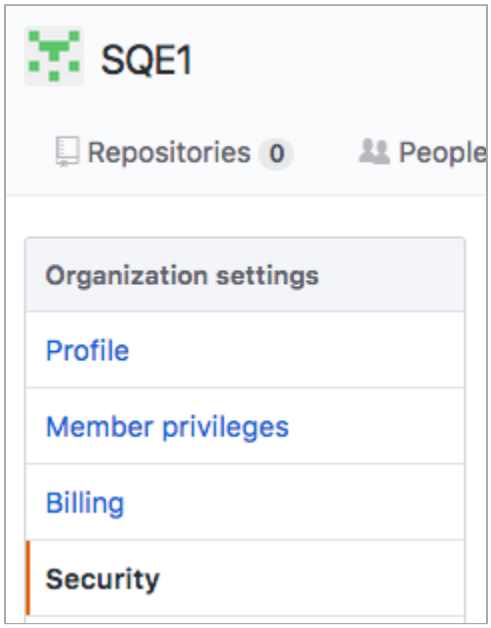


4. Under your organization name, click **Settings**.

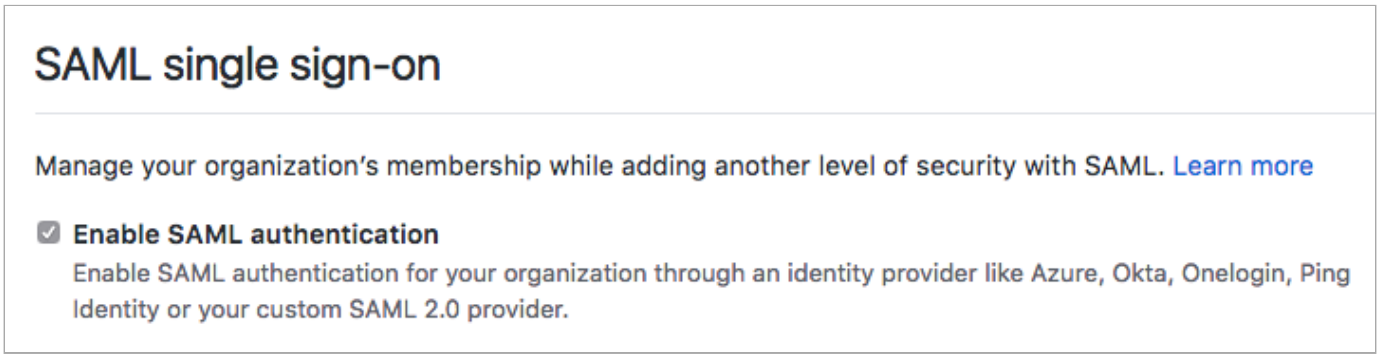


5. In the organization settings sidebar, click **Security**.

Copyright © 2025 Ping Identity Corporation



6. Under **SAML single sign-on**, select **Enable SAML authentication**.



> **Important**
>
> After enabling SAML SSO, you can download your single sign-on recovery codes so that you can access your organization even if your IdP is unavailable. For more information, see Downloading your organization's SAML single sign-on recovery codes in the GitHub documentation.

7. Open your PingFederate metadata XML file using a text editor.
8. In the **Sign on URL** field, type the HTTPS endpoint of your IdP for single sign-on requests. This value is the SingleSignOnService POST binding available in your PingFederate metadata file. For example, `https://<pf_hostname>:<pf_port>/idp/SSO.saml2`
9. **Optional:** In the **Issuer** field, type your SAML issuer's name. This verifies the authenticity of sent messages. This value is `entityID` available in your PingFederate metadata file.
10. Open your PingFederate signing certificate file using a text editor.
11. Under **Public Certificate**, paste your PingFederate signing certificate used to verify SAML responses.

Copyright © 2025 Ping Identity Corporation

> **Note**
>
> GitHub requires the public certificate to be a valid x509 formatted certificate enclosed between `-----BEGIN CERTIFICATE-----` and `-----END CERTIFICATE-----`.

12. Click the pencil icon to edit the **Signature Method** and **Digest Method** drop-downs, choose the hashing algorithm used by PingFederate issuer to verify the integrity of the requests.
13. Before enabling SAML SSO for your organization, click **Test SAML configuration** to ensure that the information you have entered is correct.

> **Note**
>
> To enforce SAML SSO and remove all organization members who haven't authenticated via your IdP, see Enforcing SAML single sign-on for your organization in the GitHub documentation.

14. Click **Save** to complete the SAML single sign-on configuration.

Release notes

Changelog

GitHub Connector 1.0 – July 2018 (current release)

- Initial release
- Added support for user life cycle management (including creates, updates and deletes)
- Added configuration options for workflow capabilities (for example, the ability to disable updates)

Known issues and limitations

- Due to a limitation with PingFederate 8.1 and earlier versions, when configuring two SP connections with the same provisioner, the second connection built may be pre-populated with the channel from the first connection. To avoid conflicts, delete this pre-populated channel and create a unique channel for each connection.
- When an LDAP user is deleted in a targeted group distinguished name (DN), the provisioning connector does not propagate the deletion until a new user is added to the group. This limitation is compounded when the **User Create** provisioning option is disabled. For solutions, see SaaS provisioner does not remove the user in the Knowledge Base.
- Due to PingFederate limitations, user attributes cannot be cleared once set.
- Enabling a previously deleted user in GitHub will trigger a create and as such, users can be enabled when User Update is set to false.
- When a user is deleted from GitHub they are removed from the organization. The user will still have a GitHub account but no access to the organization's resources.

Copyright © 2025 Ping Identity Corporation

Performance limitations

- Due to GitHub API limitations, provisioning with multiple threads or making more than 5000 requests per hour could trigger GitHub's abuse detection mechanism, rate limiting, or both. This prevents requests from being completed. Learn more in Rate limiting in the GitHub documentation.

Download manifest

The distribution `.zip` file for the connector contains the following:

- `ReadMeFirst.pdf` – contains links to this online documentation.
- `github-saml-metadata.xml` – SAML metadata file for use with a GitHub connection. For more information, see Configure PingFederate for provisioning and SSO.
- `/legal`:
  - `Legal.pdf` – copyright and license information.
- `/dist` – contains libraries needed for the connector:
  - `pf-github-quickconnection-[version].jar` – PingFederate GitHub Connector

GitHub EMU Provisioner

The PingFederate GitHub Enterprise Managed Users (EMU) Provisioner enables an enterprise to provision users and groups to GitHub. A quick connection template is also included to simplify the configuration of single sign-on (SSO).

Learn more about GitHub EMU in the GitHub website.

Features

- Outbound User Provisioning
- Outbound Group Provisioning
- Browser-based SSO initiated by the service provider (SP) or identity provider (IdP)
- Pre-populates some connection settings with the included quick connection template and Security Assertion Markup Language (SAML) metadata file
- Includes support for user life cycle management (including create, update, disable, and delete)
- Includes configuration options for workflow capabilities (for example, the ability to disable updates)

> **Note**
>
> GitHub EMU does not support user or group provisioning until you configure SSO and SAML.

Copyright © 2025 Ping Identity Corporation

Intended audience

This document is intended for PingFederate administrators.

If you need help during the setup process, see the following sections of the PingFederate documentation:

- SP connection management
- Identity provider SSO configuration
- Datastores
- Configuring outbound provisioning
- Configuring outbound provisioning settings

System requirements

- PingFederate 9.0 or later with Java 8.
- Could require the following endpoints to be allowlisted on the firewall to allow outbound connections:
  - https://api.github.com/scim/v2/enterprises/<*enterprise slug*>
  - https://api.<*enterprise_slug*>.ghe.com/scim/v2/enterprises/<*enterprise_slug*> for EMU data residency
- A GitHub Enterprise Cloud account with Enterprise Managed Users and SAML SSO enabled.

User and group management

The GitHub EMU Provisioner synchronizes users and groups from your datastore to GitHub.

SCIM API provisioning calls sent from the PingFederate Enterprise Managed User (EMU) Provisioner create the users, groups, and group memberships in a GitHub EMU enterprise. If an EMU enterprise account hasn't been created via SCIM provisioning, PingFederate users cannot use a GitHub EMU account to login and access their GitHub EMU enterprise.

> **Note**
>
> The GitHub EMU Provisioner sends a SCIM API `GET` call with a `<SCIM User ID>` value of `0` to `/scim/v2/enterprises/<enterprise>/Users/<SCIM User ID>` as part of a connection check that's performed before every provisioning operation. This `GET` call causes `404` errors in the PingFederate and GitHub enterprise logs, because there isn't a `<SCIM User ID>` with a value of `0`. These error messages can be safely ignored. This is expected behavior and it does not indicate an issue.

> **Tip**
>
> You can configure the following capabilities and specify which users to provision during the Configure PingFederate for provisioning and SSO part of the setup process.

User provisioning

PingFederate provisions users if a user is added to the datastore group or filter that is targeted by the provisioner.

Copyright © 2025 Ping Identity Corporation

You can define which users PingFederate targets for provisioning on the **Source Location** tab of your provisioning connection configuration. You can find more information in Specifying a source location.

User updates

PingFederate updates users when a user attribute changes in your datastore.

You can define which attributes PingFederate monitors for changes on the **Attribute Mapping** tab of your provisioning connection configuration.

User deprovisioning

PingFederate deprovisions users if:

- A user is suspended from the user store.
- A user is disabled in the user store.
- A user is removed from the datastore group or filter targeted by the provisioner.

When the **Remove User Action** setting in the connection configuration is set to **Disable**, PingFederate sends a `PATCH` SCIM deprovisioning call to GitHub for an enterprise managed user, and the user account gets suspended in the GitHub enterprise. If the same user account in the datastore is re-provisioned, the enterprise managed user account becomes unsuspended.

> **Important**
>
> Make sure to configure this setting for **Disable**. If the **Remove User Action** setting is set to **Delete**, PingFederate sends a `DELETE` SCIM deprovisioning call to GitHub for an enterprise managed user, and the user account gets suspended in the GitHub enterprise. If the same user account in the datastore is re-provisioned, the enterprise managed user account does not become unsuspended.

Group provisioning

PingFederate provisions groups when you add a group to the datastore filter that is targeted by the provisioner. You can define which groups PingFederate targets for provisioning and monitors for changes on the **Source Location** tab in your provisioner configuration.

After you successfully provision a group to a GitHub EMU enterprise, an enterprise owner can see the group by following the steps in this GitHub article. When mapping a team in the GitHub EMU enterprise to a group, an EMU organization owner makes selections from the list of groups that are provisioned to the enterprise.

Group name updates

PingFederate renames groups when they are renamed in the datastore.

Group membership updates

PingFederate updates group memberships when memberships change in the datastore, regardless of whether the change is in the group's properties or a user's properties.

Group memberships in the datastore overwrite the group memberships in GitHub.

Copyright © 2025 Ping Identity Corporation

Group deletion

PingFederate deletes groups if:

- The group is deleted in the datastore.
- The group is removed from the datastore group or filter targeted by the provisioner.

> **Important**
>
> Group deletions are permanent and cannot be undone.

Setup

Creating a personal access token

To configure SCIM provisioning, you must create a personal access token under your GitHub enterprise setup user account.

Steps

- Sign on to GitHub using the setup user account for your EMU enterprise and follow the instructions in Creating a personal access token in the GitHub documentation.

Obtaining PingFederate SAML 2.0 metadata

About this task

To configure GitHub for provisioning and SSO, you will require the below information. These values can be obtained from your PingFederate Identity Provider (IdP) metadata. Learn more in Exporting connection-specific SAML metadata

- PingFederate SAML Endpoint
  - Example: `https://<pf_hostname>:<pf_port>/idp/SSO.saml2`
- Identity Provider Issuer
  - This is SAML 2.0 Entity ID from PingFederate.

To export your IdP metadata:

Steps

1. Go to the **Server Configuration > Metadata Export** screen.
2. On the **Metadata Mode** screen, choose **Select information to include in metadata manually** and click **Next**.
3. On the **Protocol** screen, click **Next**.
4. On the **Attribute Contract** screen, click **Next**.
5. On the **Signing Key** screen, select the PingFederate signing certificate for use and click **Next**.
6. On the **Metadata Signing** screen, click **Next**.

Copyright © 2025 Ping Identity Corporation

7. On the **XML Encryption Certificate** screen, click **Next**.
8. On the **Export & Summary** screen, click **Export** to and save the metadata XML file for later use.

Obtain PingFederate signing certificate

About this task

The following are the steps to obtain the signing certificate used for the digital signature verification of the PingFederate IdP. When asked during Configure GitHub for provisioning and SSO, provide the certificate that you have downloaded from PingFederate. Learn more in Managing digital signing certificates and decryption keys in the PingFederate documentation.

To export an existing certificate:

Steps

1. Sign onto the **PingFederate administrative console** as an administrative user for your organization.
2. Go to **Server Configuration > Certificate Management > Signing & Decryption Keys & Certificates**.
3. Click **Export** under **Action** for the certificate you want to export.
4. On the **Export Certificate** page, select **Certificate Only** export and then click **Next**.
5. On the **Export & Summary** page, click **Export** to save the certificate file on your system.
6. When finished, click **Done**.
7. Click **Cancel** to go back to the **Server Configuration** screen.

Install the provisioner

Install and set up the PingFederate GitHub EMU Provisioner.

Before you begin

To download the PingFederate connector, go to https://www.pingidentity.com/en/resources/downloads/pingfederate.html, click the **Add-ons** tab, then click **GitHub EMU Connector 1.0**.

> **Important**
>
> Make sure to select **GitHub EMU Connector 1.0**, not **GitHub Connector 1.0**. This integration only supports the EMU connector.

About this task

To install the PingFederate GitHub EMU Provisioner:

Steps

1. If the PingFederate server is running, stop the server.
2. Extract the GitHub Provisioner distribution `.zip` file.

Copyright © 2025 Ping Identity Corporation

3. Copy the contents of the `dist` directory into the `<pf_install>/pingfederate/server/default/deploy` directory.
4. To use the provisioner for outbound provisioning, in the `<pf_install>/pingfederate/bin/run.properties` file, set the `pf.provisioner.mode` property to `STANDALONE`.

   Example:

   ```
   pf.provisioner.mode=STANDALONE
   ```

   > **Note**
   >
   > For information about using the `FAILOVER` setting for runtime deployment, see Deploying provisioning failover.

5. Start the PingFederate server.

Enabling provisioning and single sign-on in PingFederate

To use PingFederate for provisioning and single sign-on, configure an external datastore and set a SAML entity ID.

About this task

Your external datastore acts as the source of data for provisioning. PingFederate also uses an internal datastore to store the state of synchronization between the source datastore and the target datastore.

For more information, see Datastores and Configuring outbound provisioning settings in the PingFederate documentation.

PingFederate 10.1 or later

Enabling provisioning and single sign-on in PingFederate 10.1 or later
Steps

1. In the PingFederate administrative console, configure the datastore for PingFederate to use as the source of user data.

   For help, see Datastores in the PingFederate documentation.

2. Go to **System > Server > Protocol Settings > Federation Info**.
3. In the **SAML 2.0 Entity ID** field, enter a name for PingFederate to use when identifying itself to SAML partners.
4. On the **Outbound Provisioning** tab, in the **Provisioner Data Store** list, select the internal database that will store the synchronization state. Click **Save**.

   For more information, see Configuring outbound provisioning settings in the PingFederate documentation.

Copyright © 2025 Ping Identity Corporation

PingFederate 10.0 or earlier

Enabling provisioning and single sign-on in PingFederate 10.0 or earlier

Steps

1. In the PingFederate administrative console, configure the datastore for PingFederate to use as the source of user data.

   For help, see Datastores in the PingFederate documentation.

2. Go to **System > Protocol Settings > Roles & Protocols**.
3. Select the **Enable Identity Provider IdP Role and Support the Following** check box.
4. Select the **SAML 2.0** and **Outbound Provisioning** check boxes. Click **Next**.
5. Click the **Federation Info** tab.
6. In the **SAML 2.0 Entity ID** field, enter a name for PingFederate to use to identify itself to SAML partners.
7. On the **Outbound Provisioning** tab, in the **Provisioner Data Store** list, select the internal database that will store the synchronization state. Click **Save**.

   For more information, see Configuring outbound provisioning settings in the PingFederate documentation.

Configure PingFederate for provisioning and SSO

Configure a service provider (SP) connection in PingFederate to manage outbound provisioning and single sign-on (SSO) to GitHub.

About this task

Outbound provisioning details are managed within an SP connection and can be added to an existing SP connection.

> **Note**
>
> The SCIM API requires that you have GitHub Enterprise Cloud with SAML SSO enabled for the enterprise. Learn more in About SCIM in the GitHub documentation.

Steps

1. In the PingFederate administrative console, configure the datastore that PingFederate will use as the source of user data.

   You can find configuration instructions in Datastores in the PingFederate documentation.

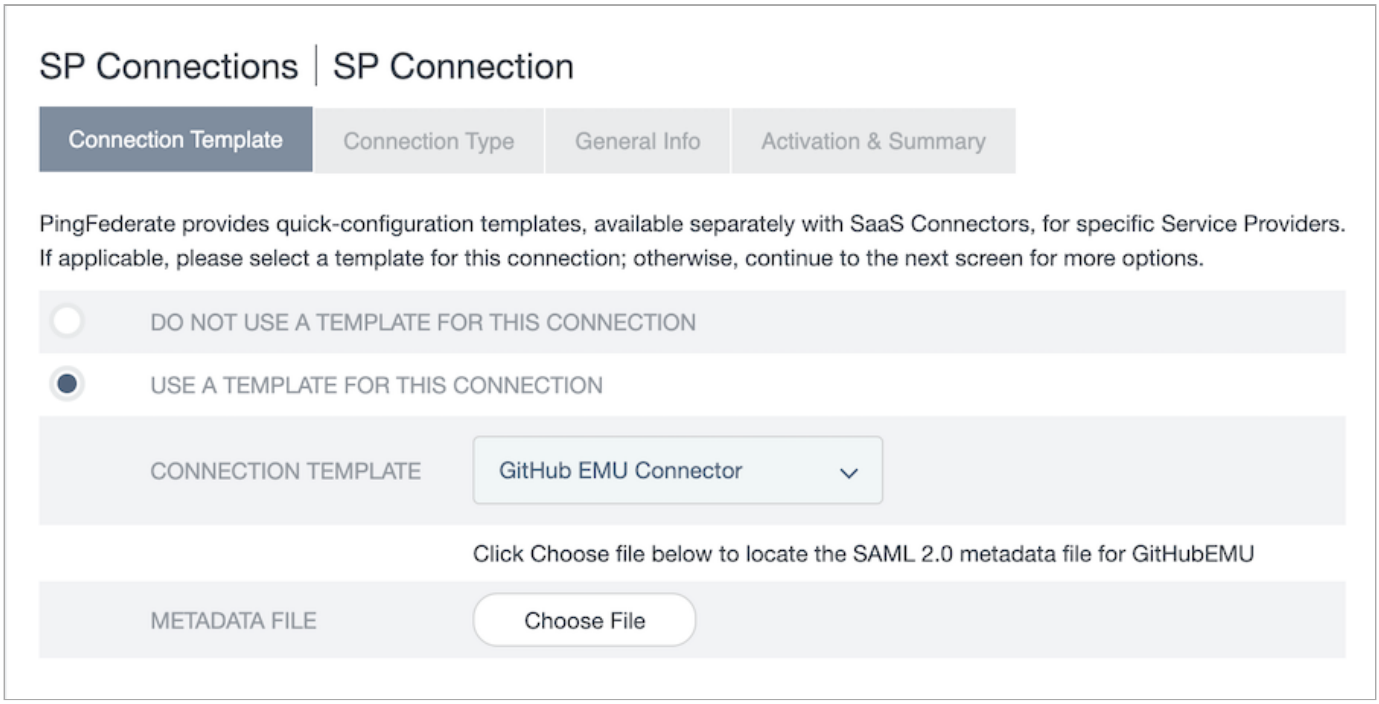
   > **Important**
   >
   > On the **Data Store Type** tab, you must select **Directory (LDAP)** as the **Type**. GitHub EMU enterprises require SCIM provisioning, and **Directory (LDAP)** is the only datastore type supported for PingFederate SCIM provisioning.

2. Create a new SP connection or select an existing SP connection from the **SP Connections** page.

Copyright © 2025 Ping Identity Corporation

3. On the **Connection Template** tab, select **Use a template for this connection**.
4. In the **Connection Template** list, select **GitHub EMU Connector**.

> **Important**
>
> You must select **GitHub EMU Connector**, not **GitHub Connector**. This integration only supports the EMU connector.
> If this selection isn't available, verify the connector installation and restart PingFederate.



5. In the **Metadata File** section, import one of the following metadata files.

   Choose from:

   - The enterprise's SAML metadata, which you can download at either of the following URLs:
     - https://github.com/enterprises/*<enterprise_slug>*/saml/metadata
     - https://*<enterprise_slug>*.ghe.com/enterprises/*<enterprise_slug>*/saml/metadata for EMU data residency
   - Either of the sample metadata files that are packaged with the GitHub EMU Connector:
     - `github-emu-saml-metadata.xml`
     - `github-emu-saml-enterprises-metadata.xml` for EMU data residency
6. On the **Connection Type** tab, ensure that both the **Outbound Provisioning** and **Browser SSO Profiles** check boxes are selected.
7. On the **General Info** tab, enter your corresponding enterprise name in the **Partner's Entity ID (Connection ID)** field, then click **Next**.

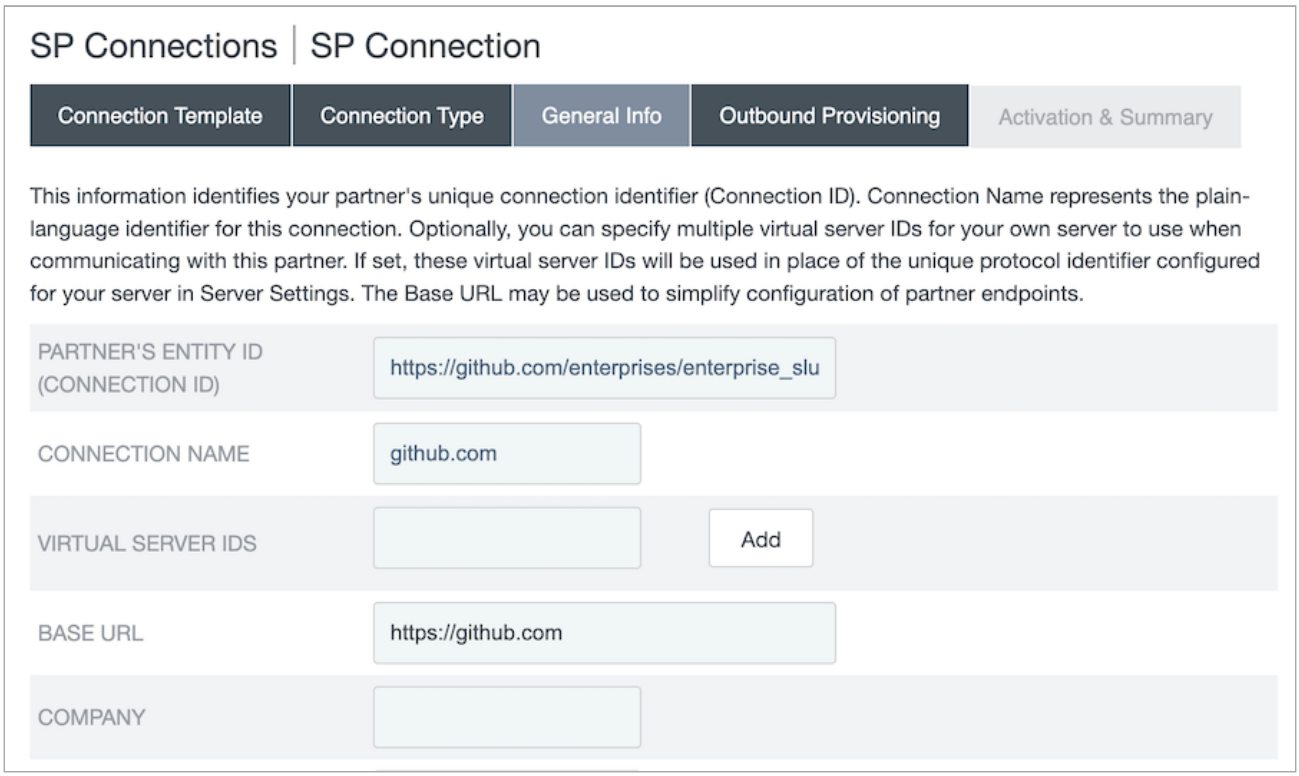
Copyright © 2025 Ping Identity Corporation

> **Note**
>
> The default values on the **General Info** tab are from the metadata file that you selected previously.

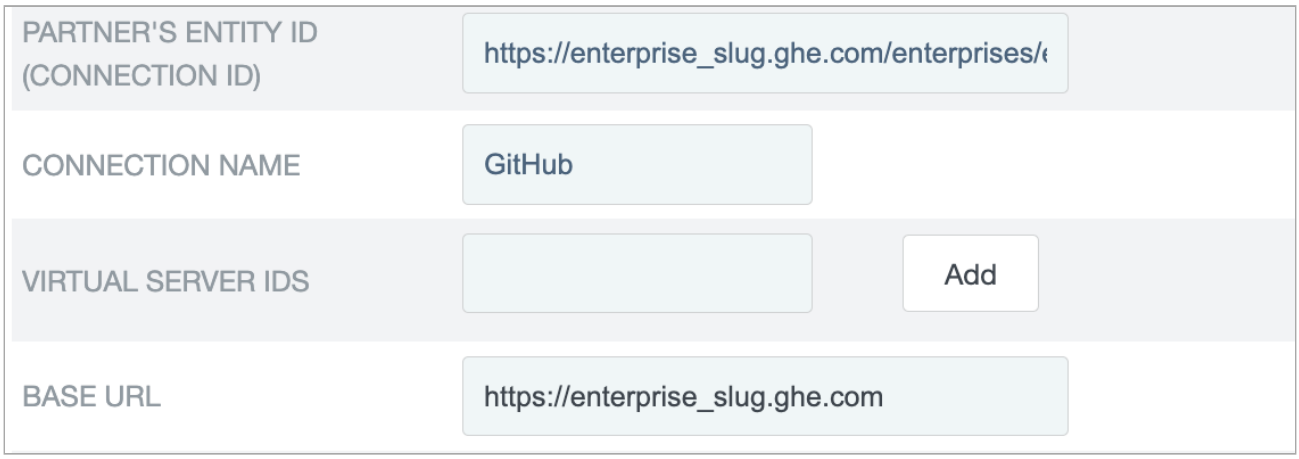
Use the following format:

Choose from:

- https://github.com/enterprises/<*enterprise_slug*>



- https://<*enterprise_slug*>.ghe.com/enterprises/<*enterprise_slug*> for EMU data residency



8. On the **Browser SSO** tab, configure your browser SSO settings.

Copyright © 2025 Ping Identity Corporation

Learn more about configuring browser SSO in the following sections of Identity provider SSO configuration:

- Managing IdP adapters
- Configuring IdP Browser SSO

    1. Click **Configure Browser SSO**.
    2. On the **Assertion Creation** tab, click **Next**.
    3. On the **Protocol Settings** tab, click **Configure Protocol Settings**.
    4. On the **Summary** tab, click **Assertion Consumer Service URL**, edit the existing entry on the **Assertion Consumer Service URL** page, and enter the **Endpoint URL** corresponding to your enterprise name.

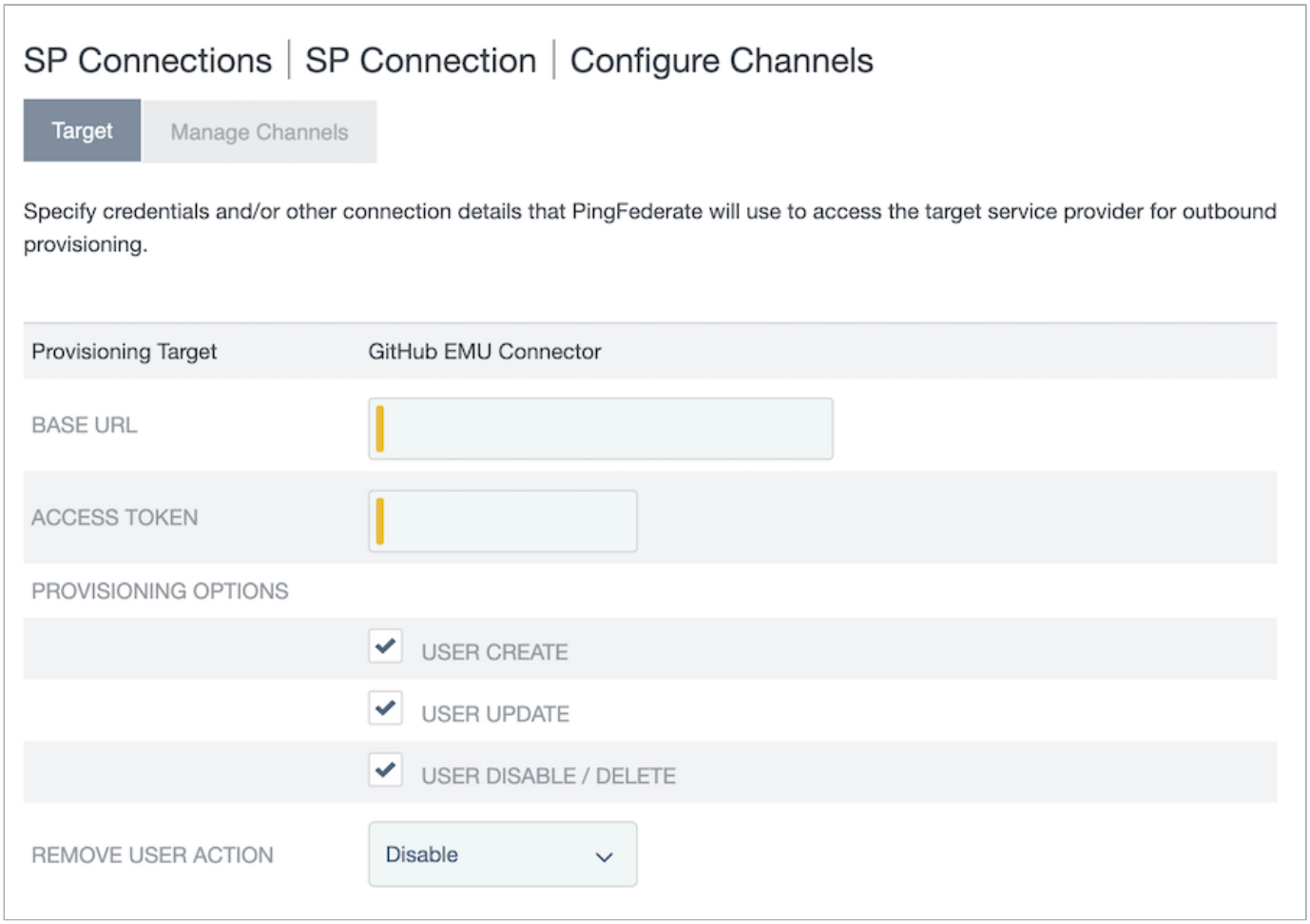
        Example:

        - `https://github.com/enterprises/<enterprise slug>/saml/consume`
        - `https://<enterprise_slug>.ghe.com/enterprises/<enterprise_slug>/saml/consume` for EMU data residency
    5. Click **Update** and **Done**, then click **Done** on the **Protocol Settings** tab.

9. On the **Credentials** tab, click **Configure Credentials**, then go to the **Digital Signature Settings** tab, select the signing certificate, and click **Done**, then **Next**.
10. On the **Outbound Provisioning** tab, click **Configure Provisioning**.
11. On the **Target** tab, enter the **Base URL** and **Access Token** values.

> **Important**
>
> Do not change any of the default **Provisioning Options** for this integration. Make sure that **User Create**, **User Update**, and **User Disable/Delete** are selected and that **Remove User Action** is set to **Disable**.

Copyright © 2025 Ping Identity Corporation



See the following table for instructions on how to configure the required values.

Field Name	Description
Base URL	The base URL for GitHub. For example: ◦ `https://api.github.com/scim/v2/enterprises/<enterprise slug>` ◦ `https://api.<enterprise_slug>.ghe.com/scim/v2/enterprises/<enterprise_slug>` for EMU data residency **Note** You can find information on how to determine your enterprise name in Accessing an enterprise in the GitHub documentation.
Access Token	The access token that the provisioner uses to make authenticated API calls to GitHub.
Provisioning Options	

Copyright © 2025 Ping Identity Corporation

Field Name	Description
User Create	***True (default)*** Users will be created in GitHub. **Important** Make sure that **User Create** is selected for this integration. ***False*** Users will not be created in GitHub. **Note** The `provisioner.log` will display a warning within the create user workflow that the user was not created in GitHub.
User Update	***True (default)*** Users will be updated in GitHub. **Important** Make sure that **User Update** is selected for this integration. ***False*** Users will not be updated in GitHub. **Note** The `provisioner.log` will display a warning within the update user workflow that the user was not updated in GitHub. Enabling a previously suspended user in GitHub will trigger a create and as such, users can be enabled when **User Update** is not selected.
User Disable/Delete	***True (default)*** Users will be suspended or disabled in GitHub. **Important** Make sure that **User Disable/Delete** is selected for this integration. ***False*** Users will not be suspended or disabled in GitHub. **Note** The `provisioner.log` will display a warning indicating that the user was not suspended in GitHub.

Copyright © 2025 Ping Identity Corporation

Field Name	Description
Remove User Action	***Disable (default)*** PingFederate disables the user in GitHub. **Important** Make sure that **Disable** is selected for this integration. ***Delete*** PingFederate suspends the user in GitHub. The **Remove User Action** applies when **User Disable/Delete** is selected, and either: ◦ A previously provisioned user no longer meets the condition set on the **Source Location** tab. ◦ A user has been disabled or deleted from the datastore.

12. Click **Next**.
13. Configure a channel and complete the provisioning configuration.

Tip

If you aren't ready to complete the provisioning configuration, you can click **Save** and return to the configuration page later. To return to the configuration page, select the connection from **Identity Provider > SP Connections > Manage All**.

Learn more in the following sections under Configuring outbound provisioning in the PingFederate documentation:

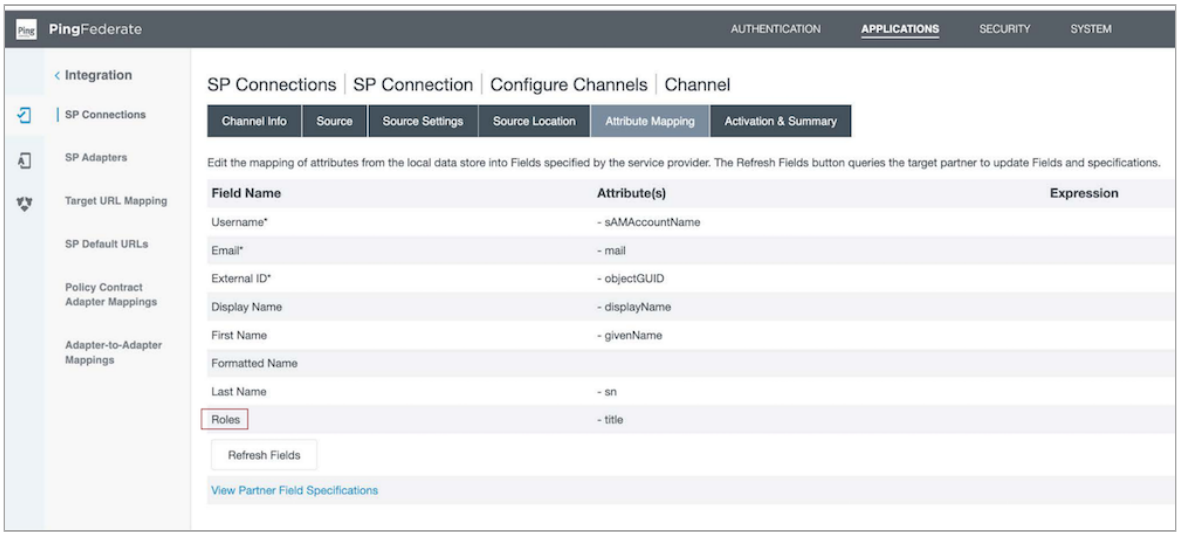
- Managing channels
- Specifying channel information
- Identifying the source datastore
- Modifying source settings
- Specifying a source location
- Mapping attributes
- Reviewing channel settings

Note

Credentials will be verified when the channel and SP connection is set to **Active** and provisioning is initiated.

1. Go to the **Manage Channels** tab and select the name of a channel to edit it.
2. On the **Attribute Mapping** tab, edit the **Roles** field by clicking **Edit** in the **Action** column.

Copyright © 2025 Ping Identity Corporation



3. After the **Attribute Mapping** window for the **Roles** field opens, map the **Roles** field to an LDAP attribute containing the value for the GitHub enterprise role that the user will have when they are provisioned.

   The LDAP attribute must contain one of the following string values:

   - `enterprise_owner`
   - `billing_manager`
   - `user`
   - `guest_collaborator`

4. Click **Done**.
5. After you finish mapping attributes, click **Next**.
6. In the **Channel Status** section of the **Activation and Summary** tab, click **Active**.
7. Click **Done**.

14. Link users authenticating through SAML to their provisioned SCIM identities in the GitHub enterprise.

> **Important**
>
> The SAML `NameID` and SCIM `userName` values must match for each user. Otherwise, a PingFederate user attempting to access their EMU account through SAML authentication might get a `404` error and an error message such as:
>
> ```
> Enterprise Managed Users must be provisioned via SCIM
> ```

To make sure that these values match in PingFederate:

1. On the **Attribute Contract Fulfillment** tab, make sure that the **Value** of the `SAML_SUBJECT` contract attribute is `sAMAccountName` if you are using Active Directory as the LDAP datastore, or `username` if you are using another LDAP datastore like PingDirectory.

Copyright © 2025 Ping Identity Corporation

To find the **Attribute Contract Fulfillment** tab:

1. On the **Browser SSO** tab, click **Configure Browser SSO**.
2. Go to the **Assertion Creation** tab and click **Configure Assertion Creation**.
3. Go to the **Authentication Source Mapping** tab, click **Map New Adapter Instance**, and then go to the **Attribute Contract Fulfillment** tab.

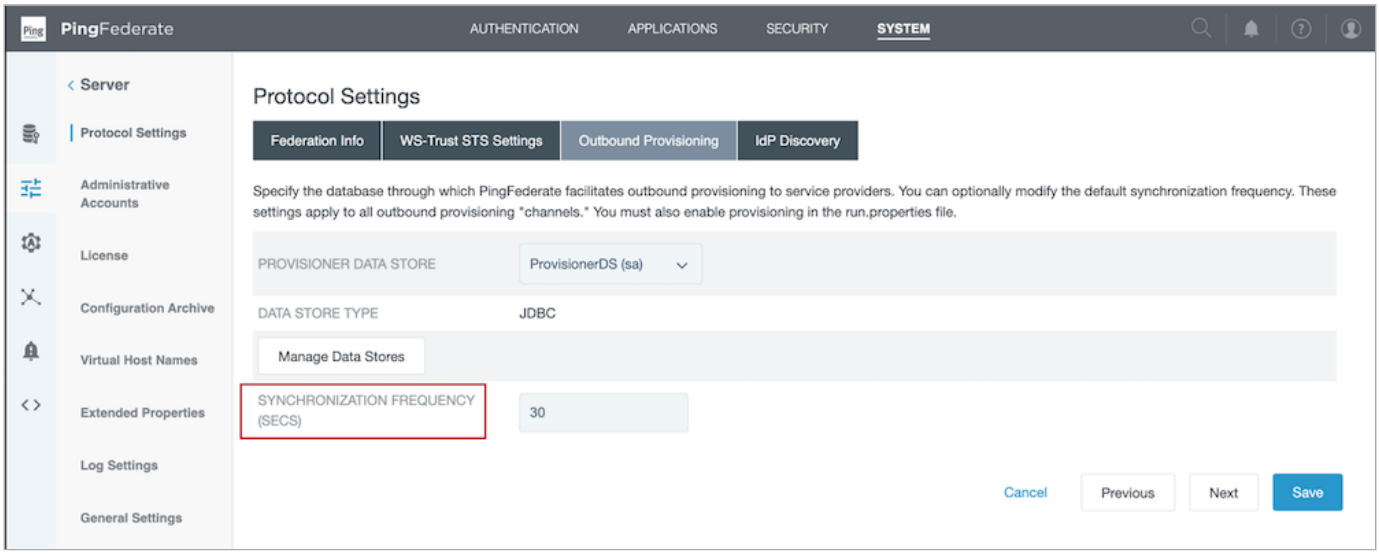
2. On the **Attribute Mapping** tab, make sure that the **Username** field is mapped to the same attribute value that the `SAML_SUBJECT` contract attribute is mapped to on the **Attribute Contract Fulfillment tab**. For example, this value might be `sAMAccountName` if you are using Active Directory, or `username` if you are using PingDirectory.

15. (Optional) Configure the **Synchronization Frequency** of your outbound provisioning channels.

    By default, PingFederate attempts to process user, group, and group member updates and send these updates to GitHub every 60 seconds. This interval is controlled by the **Synchronization Frequency** value, which affects all outbound provisioning channels.

    To update this value:

    1. Go to **System > Server > Protocol Settings** and select the **Outbound Provisioning** tab.
    2. Enter a new **Synchronization Frequency** value, then click **Save**.



Supported attributes reference

The following table consists of the attributes that can be mapped for user provisioning. Find more information about available attributes in SCIM in the GitHub documentation.

Copyright © 2025 Ping Identity Corporation

Attribute	Description
Username	GitHub's unique identifier for the user. This attribute is required. **Note** GitHub usernames can't exceed 39 characters. This character limit includes the underscore and short code that GitHub attaches to a normalized username. Find more information about username components and normalization in Username considerations for external authentication in the GitHub documentation.
Email	The email for the user (for example, "bjensen@example.com"). This attribute is required.
External ID	A string that is an identifier for the resource as defined by the provisioning client. **Note** External ID is mapped to a binary attribute ObjectGUID in AD, and needs to be added to PingFederate's binary attribute list in the Data Store.
First Name	The given name of the user, or first name in most Western languages (for example, 'Barbara' given the full name 'Ms. Barbara Jane Jensen, III').
Last Name	The family name of the user, or last name in most Western languages (for example, 'Jensen' given the full name 'Ms. Barbara Jane Jensen, III').
Display Name	A human-readable name for a user.
Formatted Name	The user's full name, including all middle names, titles and suffixes, formatted for display.
Roles	A list of the user's roles. For more information, see Provision a SCIM enterprise user in the GitHub documentation.

Configure GitHub for provisioning and SSO

Complete the following task to enable provisioning and single sign-on (SSO) on the GitHub Admin Portal.

Before you begin

To get the values that you need to configure a SAML integration in GitHub:

- Obtaining PingFederate SAML 2.0 metadata
- Obtain PingFederate signing certificate

About this task

You must configure the SAML integration before you configure the SCIM provisioning integration in GitHub. SCIM provisioning calls fail with a `404` error if SAML authentication is not configured.

Copyright © 2025 Ping Identity Corporation

Steps

- Sign on to GitHub and follow the Configuring your enterprise instructions in the GitHub documentation:
    1. For steps 7 and 8, enter the values that you collected in the Obtaining PingFederate SAML 2.0 metadata procedure.
    2. For step 9, paste the certificate that you exported in the Obtain PingFederate signing certificate procedure.

Release notes

Changelog

GitHub EMU Provisoner 1.0.0.7 – November 2024

- Added a second SAML metadata file to handle the new GitHub EMU **Entity ID** and URL that point to `https://<enterprise_slug>.ghe.com`. These changes were introduced to handle data residency requirements.

> **Note**
>
> The SAML metadata file that you select during setup pre-populates part of the adapter configuration. Learn more in steps 5 and 7 of Configure PingFederate for provisioning and SSO.

GitHub EMU Provisoner 1.0 – April 2023

- Initial release.

Known issues and limitations

Known issues

- There are no known issues.

Known limitations

- Attributes
    - Because of limitations in PingFederate, user attributes cannot be cleared after they are set.
- Provisioning
    - GitHub might enforce rate limiting for large numbers of users, such as more than 7,000 users, being provisioned at one time. Learn more in Rate limiting in the GitHub documentation.
    - GitHub doesn't support provisioning a user with an initial disabled state.

Copyright © 2025 Ping Identity Corporation

- Deprovisioning
    - When an LDAP user is deleted in a targeted group distinguished name (DN), the provisioning connector does not propagate the deletion until a new user is added to the group. This limitation is compounded when the **User Create** provisioning option is disabled. For solutions, see SaaS provisioner does not remove the user in the Knowledge Base.

Download manifest

The distribution `.zip` file for the connector contains the following:

- `github-emu-saml-metadata.xml` : SAML metadata file for use with a GitHub connection. Learn more in Configure PingFederate for provisioning and SSO.
- `github-emu-saml-enterprises-metadata.xml` : SAML metadata file for use with an EMU data residency connection. Learn more in Configure PingFederate for provisioning and SSO.
- `/legal` :
    - `Legal.pdf` : Copyright and license information.
- `/dist` : Contains libraries needed for the connector:
    - `pf-github-emu-quickconnection-<version>.jar` : PingFederate GitHub EMU Provisioner.

Copyright © 2025 Ping Identity Corporation

Overview of Google integrations

PingFederate Google Cloud Identity Connector

The Google Cloud Identity Connector allows a Service Provider (SP) to use Google as an Identity Provider (IdP) for access to applications in the SP domain.

PingFederate Google Apps Connector

The Google Apps Connector extends PingFederate capabilities, enabling enterprises to provision its users to Google Apps.

Google Login Integration Kit

The PingFederate Google Login Integration Kit allows PingFederate to use Google as an identity provider (IdP). This allows users to access service provider (SP) applications by signing on with Google or G Suite credentials.

Components

Google IdP Adapter

- This adapter allows PingFederate to communicate with the Google API to process sign-on requests and get user email addresses or "Basic Profile" information.
- For G Suite environments, the adapter uses the Directory API from the G Suite Admin SDK to get "Extended Profile" user information.

Intended audience

This document is intended for PingFederate administrators.

If you need help during the setup process, see the following resources:

- The following sections of the PingFederate documentation:
  - Managing IdP adapters
  - Authentication policies
- Google Sign-in in the Google documentation

System requirements

- PingFederate 11.3 or later
- To access the "Extended Profile" attributes for G Suite users:
  - A G Suite account:
    - In the same organization as the users
    - With access to the Google Developer Console

Copyright © 2025 Ping Identity Corporation

- With permission to view user profiles

- API access must be enabled for the organization in Google Admin console.

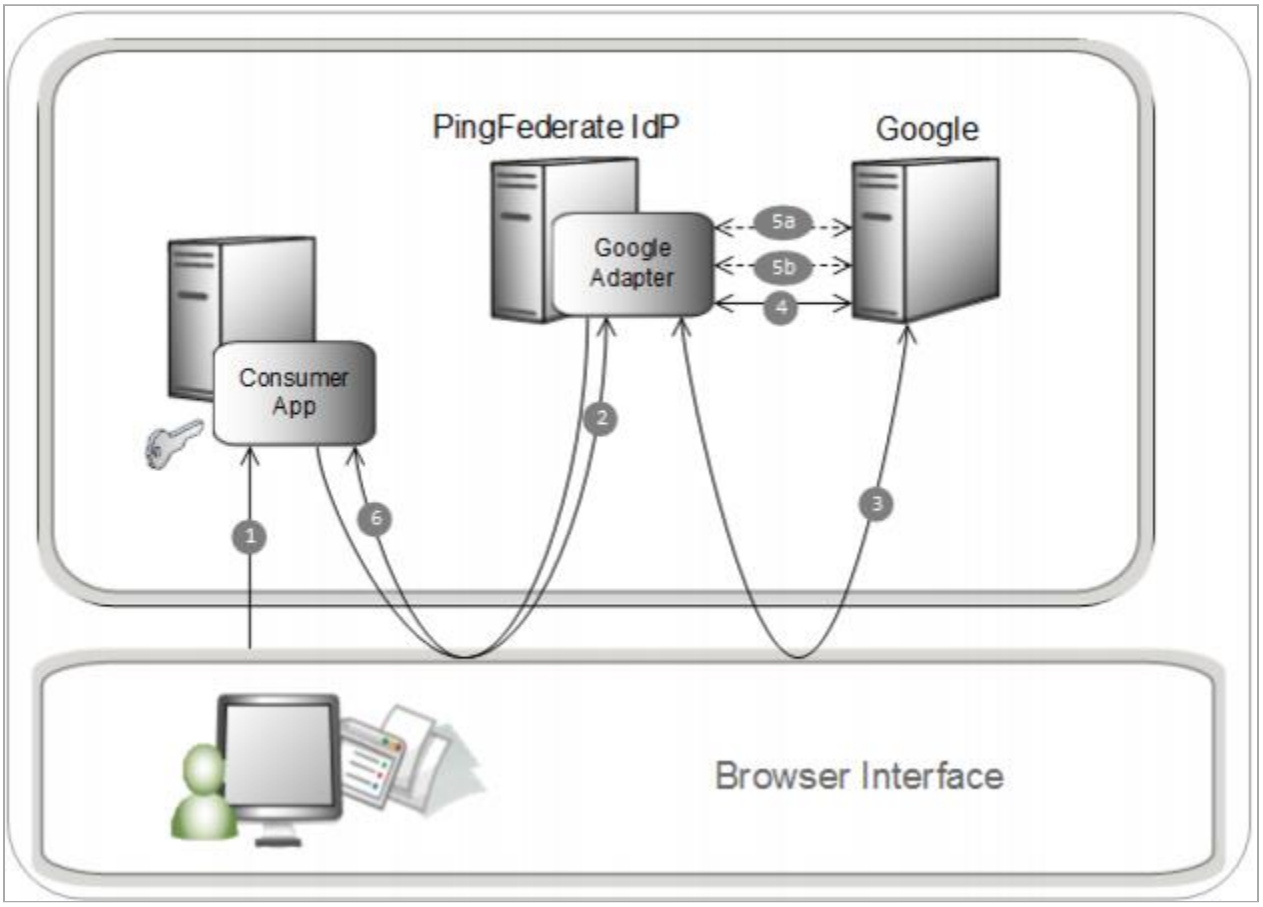
You can find more information in the following sections of the Google documentation:

- Additional Google services
- Pre-built administrator roles
- Control which third-party & internal apps access G Suite data

Overview of the SSO flow

With the Google Login Integration Kit, PingFederate includes the Google API in the sign-on flow.

The following figure illustrates a service provider (SP)-initiated single sign-on (SSO) scenario in which PingFederate authenticates users to an SP application using the Google IdP Adapter.



Description

1. The user opens a web application and chooses the **Google Sign-In** option.
2. The sign-on link points to the Google IdP Adapter, which redirects the browser...
3. ...to Google for authentication with a list of requested permissions.

Copyright © 2025 Ping Identity Corporation

The user authenticates their identity and then authorizes the requested permissions. Google redirects the browser to the PingFederate Google IdP Adapter authorization callback endpoint with an authorization code.

If the user fails to authenticate or does not authorize the request, the response includes an error code instead.

4. The Google IdP Adapter makes an HTTP request to the Google API to obtain an OAuth 2.0 access token. It provides the client ID and secret, and the authorization code. The Google API returns an access token and ID token that includes the user's email address.
5. (a) The Google IdP Adapter provides the access token and requests the email address or "Basic Profile" user attributes from the Google user info endpoint. Google returns a set of attributes in OpenID Connect format.

   (b) The Google IdP Adapter provides uses a refresh token to get a new access token. It provides the access token and requests the "Extended Profile" user attributes from the Directory API (part of the G Suite Admin SDK). Google returns a set of attributes in JSON format.
6. PingFederate redirects the user to the web application with the user information from Google.

Setup

Upgrading an existing deployment

If you're upgrading from a previous version of the Google Login Integration Kit, back up your PingFederate configuration and update the **Certificate URL** field.

Steps

1. Back up your current PingFederate configuration as shown in Exporting an archive in the PingFederate documentation.
2. Complete the steps in Deploying the integration files.
3. In the PingFederate administrative console, open your existing IdP adapter instance.

   Choose from:

   - For PingFederate 10.1 or later: go to **Authentication > Integration > IdP Adapters**. Select your IdP adapter instance.
   - For PingFederate 10.0 or earlier: go to **Identity Provider > Adapters**. Select your IdP adapter instance.
4. If you are upgrading from Google IdP Adapter 1.0.x, update the **Certificate URL** field.
   1. On the **IdP Adapter** screen, click **Show Advanced Fields**.
   2. In the **Certificate URL** field, type the following URL:

      ```
      https://www.googleapis.com/oauth2/v3/certs
      ```
5. Click **Save**.

Copyright © 2025 Ping Identity Corporation

Deploying the integration files

To get started with the integration, deploy the Google Login Integration Kit files to your PingFederate directory.

About this task

> **Note**
>
> If you operate PingFederate in a cluster, the following steps refer to the console node.

Steps

1. Download the Google Login Integration Kit `.zip` archive from the Ping Identity Integration Directory.
2. Stop PingFederate.
3. If you're upgrading an existing deployment, back up your customizations and delete earlier versions of the integration files:
    1. Back up any Google Login Integration Kit files that you customized in the `<pf_install>/pingfederate/server/default/conf/` directory.
    2. Delete the `pf-google-adapter-<version>.jar` file from your `<pf_install>/pingfederate/server/default/deploy` directory.
4. Extract the `.zip` archive and merge the contents of the `dist` directory with your `<pf_install>/pingfederate` directory.
5. If there is more than one version of the `pf-authn-api-sdk-<version>.jar` file in your `<pf_install>/pingfederate/server/default/lib` directory, delete all but the latest version of the file.
6. If you backed up any customized files, modify the new files with your customizations.
7. Start PingFederate.
8. If you operate PingFederate in a cluster, repeat steps 2-8 for each engine node.

Registering PingFederate in Google

To allow PingFederate to process sign-on requests using Google, add PingFederate as a sign-on provider in Google.

Steps

1. Sign on to the Google Developers Console with a Google account.

    > **Important**
    >
    > For G Suite environments, sign on using the G Suite account that meets the requirements described in Google Login Integration Kit.

2. Open an existing project, or create a new project as shown in Create, shut down, and restore projects in the Google documentation.
3. If you want to access the "Extended Profile" attributes for G Suite users, enable the **Admin SDK** API for your project as shown in Enable and disable APIs in the Google documentation.

Copyright © 2025 Ping Identity Corporation

4. Configure the OAuth consent screen as shown in the **User consent** section of Setting up OAuth 2.0 in the Google documentation.

> **Note**
>
> The consent screen is shown to users whenever the Google IdP Adapter requests access to their Google profile information.

5. Add information about your PingFederate server.

> **Tip**
>
> If you want to create multiple Google IdP Adapter instances, you can create a new OAuth client for each instance.

   1. On the **Credentials** tab, from the **Create Credentials** list, select **OAuth client ID**.
   2. On the **Create OAuth client ID** window, from the **Application type** list, select **Web application**.
   3. In the **Name** field, enter a name of your choosing.
   4. Under **Authorized JavaScript Origins**, click **Add URI**. Enter your PingFederate base URL based on the following:

```
https://pf_host:pf_port
```

   5. Under **Authorized redirect URIs**, click **Add URI**. Enter the Google IdP Adapter endpoint based on the following:

```
https://pf_host:pf_port/ext/google-authn
```

   6. If you want to retrieve "Extended Profile" attributes for G Suite users, click **Add URI** and enter the following URL:

```
https://oauth.pingone.com/ocs/ppm/rest/v1/oauth/oastempcredresponse/
```

6. Click **Create**.
7. On the **OAuth client created** modal, note the **Client ID** and **Client Secret**. You will use these in Configuring an adapter instance. Click **OK**.

Get a refresh token

To allow the Google IdP Adapter to retrieve "Extended Profile" attributes for G Suite users, use your application credentials to get an OAuth 2.0 refresh token.

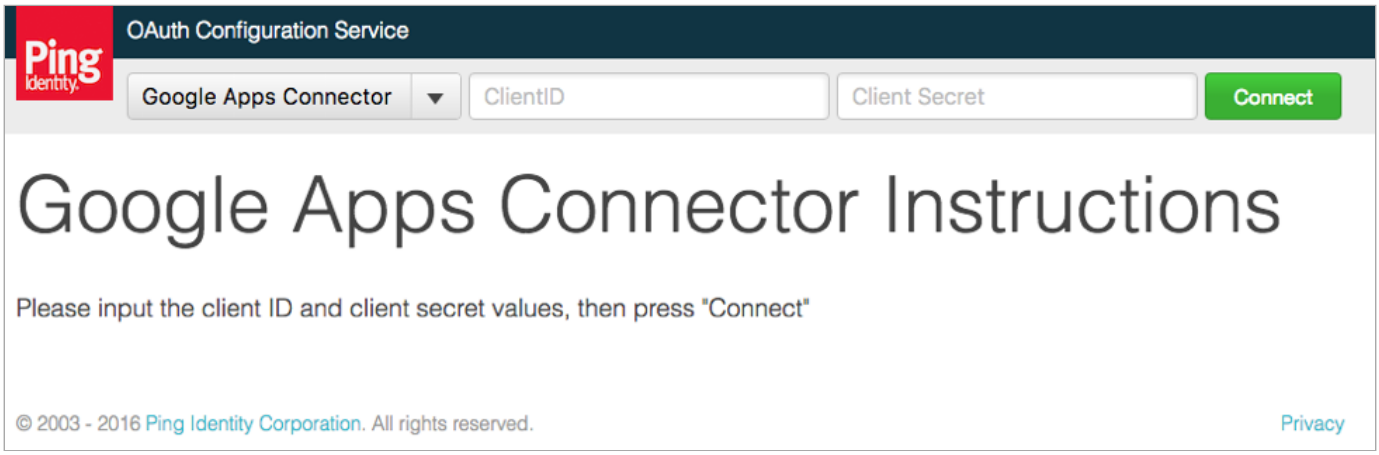
About this task

> **Note**
>
> These steps are not required if you only want the email address or "Basic Profile" attributes for your users. For a list of attributes available under each profile type, see Available user attributes reference.

Copyright © 2025 Ping Identity Corporation

Steps

1. Go to the Ping Identity OAuth Configuration Service.
2. Select **Google Apps Connector** from the drop-down menu.

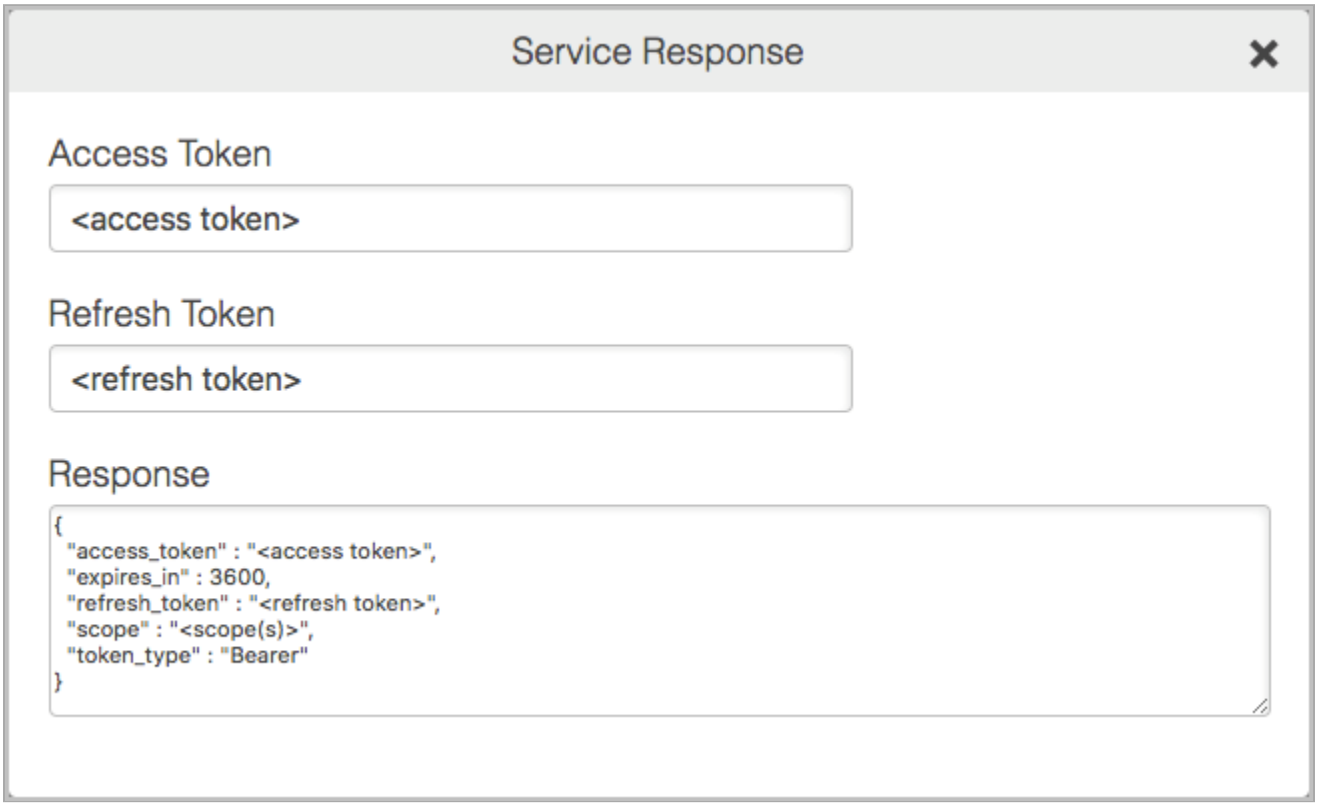


3. In the **Client ID** field, enter the value that you noted in Registering PingFederate in Google.
4. In the **Client Secret** field, enter the value that you noted in Registering PingFederate in Google.
5. Click **Connect**.
6. Sign on using the G Suite developer account that meets the requirements described in Google Login Integration Kit.

> **Note**
>
> If you have already signed on, Google might not prompt you to sign on again.

7. Grant access to the requested permissions by clicking **Allow**.
8. On the **Service Response** modal, note the **Refresh Token**. You will use this in Configuring an adapter instance.

Copyright © 2025 Ping Identity Corporation



Configuring an adapter instance

Configure the Google IdP Adapter to determine how PingFederate communicates with the Google API.

Steps

1. In the PingFederate administrative console, create a new IdP adapter instance:

   Choose from:

   - For PingFederate 10.1 or later: go to **Authentication > Integration > IdP Adapters**. Click **Create New Instance**.
   - For PingFederate 10.0 or earlier: go to **Identity Provider > Adapters**. Click **Create New Instance**.

2. On the **Type** tab, set the basic adapter instance attributes.
   1. In the **Instance Name** field, enter a name for the adapter instance.
   2. In the **Instance ID** field, enter a unique identifier for the adapter instance.
   3. In the **Type** list, select **Google IdP Adapter**. Click **Next**.
3. On the **IdP Adapter** tab, configure the adapter instance by referring to Google IdP Adapter settings reference. Click **Next**.
4. On the **Extended Contract** tab, add any attributes that you want to include in the contract. Click **Next**.
   1. Add each attribute associated with the **Attribute Retrieval** setting you selected. For a list of attributes associated with each **Attribute Retrieval** setting, see Available user attributes reference.
   2. If you changed the **Authentication URL** to expose the `openid_id` parameter, add `openid_id`.

Copyright © 2025 Ping Identity Corporation

5. On the **Adapter Attributes** tab, set pseudonym and masking options as shown in Set pseudonym and masking options in the PingFederate documentation. Click **Next**.
6. On the **Adapter Contract Mapping** tab, configure the contract fulfillment details for the adapter as shown in Define the IdP adapter contract in the PingFederate documentation. Click **Next**.
7. On the **Summary** tab, check and save your configuration:

   Choose from:

   - For PingFederate 10.1 or later: click **Save**.
   - For PingFederate 10.0 or earlier: click **Done**. On the **Manage IdP Adapter Instances** tab, click **Save**.

Google IdP Adapter settings reference

Field descriptions for the Google IdP Adapter configuration screen.

Standard fields

Field	Description
Attribute Retrieval	Determines the user attributes that the Google IdP Adapter gets from Google. Your selection determines the set of attributes that you can add on the **Extended Contract** tab. For the list of attributes associated with each option, see Available user attributes reference. ***Email*** Returns the user's email address and identifier. ***Basic Profile*** Returns the user's full basic using the Google userinfo endpoint. ***Extended Profile*** Returns the user's full profile using Directory API from the G Suite Admin SDK. For details, see Method:users.get in the Google documentation. **Note** The Extended Profile is only available when the refresh token is authorized for the user's G Suite organization. **Note** The `externalIds` attribute is returned as structured JSON attributes. Make sure the service provider application can parse attributes in that format. The default selection is **Email**.
Client ID	The **App ID** that you noted in Registering PingFederate in Google. This field is blank by default.
Client Secret	The **App Secret** that you noted in Registering PingFederate in Google. This field is blank by default.

Copyright © 2025 Ping Identity Corporation

Field	Description
Refresh Token	The refresh token that you noted in Get a refresh token. This field is blank by default.
Error Redirect URL	When an error occurs in the adapter, PingFederate redirects the browser to this URL instead of the default error page. This field is blank by default.
Unauthorized redirect URL	When the user does not authorize Google to share their user information, PingFederate redirects the browser to this URL instead of the default error page. This field is blank by default.
Callback Endpoint	The end of the callback endpoint that you authorized in Registering PingFederate in Google. The default value is `/google-authn`.

Advanced fields

Field	Description
PingFederate Base URL	Overrides the PingFederate base URL, such as `https://pf_host:pf_port/`. This field is blank by default.
Authentication URL	The Google endpoint used for authentication. If Google changes this endpoint, enter the new URL. If you want the adapter to expose the `openid_id` parameter, enter `https://accounts.google.com/o/oauth2/auth?openid.realm=https://pf_host:pf_port`. The default value is `https://accounts.google.com/o/oauth2/auth`.
Access Token URL	The Google endpoint used to retrieve an OAuth access token. If Google changes this endpoint, enter the new URL. The default value is `https://accounts.google.com/o/oauth2/token`.
Logout URL	The Google logout URL. The adapter redirects users to this address to sign them out. If Google changes this endpoint, enter the new URL. The default value is `https://accounts.google.com/Logout?hl=en`.
User Info URL	The Google URL used to retrieve user data. If Google changes this endpoint, enter the new URL. The default value is `https://openidconnect.googleapis.com/v1/userinfo`.
Extended Profile Data URL	The Google URL used to retrieve user data from the Directory API for G Suite organizations. If Google changes this endpoint, enter the new URL. The default value is `https://www.googleapis.com/admin/directory/v1/users`.
Group Request	Determines whether the adapter retrieves user group information from the Google API. This check box is cleared by default.

Copyright © 2025 Ping Identity Corporation

Field	Description
Extended Profile Group URL	The Google URL used to retrieve group data from the Directory API for G Suite organizations. If Google changes this endpoint, enter the new URL. The default value is `https://www.googleapis.com/admin/directory/v1/groups?userKey=`.
Certificate URL	The Google URL used to retrieve the latest certificate that Google uses to sign the ID token. If Google changes this endpoint, enter the new URL. The default value is `https://www.googleapis.com/oauth2/v3/certs`.
Google Sign-On Presentation	Determines how the user is directed to Google for authentication. If automatic redirects are blocked in your environment, select **Pop-up window**. If you are not using PingFederate in authentication API mode, this alternative presents a template file that requires user interaction. The default value is **Redirect**.
Google Pop-Up Template	The template file that presents the Google sign-on form. Applies only when **Google sign-on presentation** is set to **Pop-up window**. The default value is `google-pop-up-template.html`.
Google Post Auth Template	The template that adapter presents after the user signs on. This template signals to the main template or authentication widget to continue the authentication flow. Applies only when **Google sign-on presentation** is set to **Pop-up window**. The default value is `google-post-auth-template.html`.
Google Messages File	The language-pack file associated with the Google Pop-Up Template. The default value is `pingfederate-google-adapter-messages`.
Retry Request	Determines whether the adapter retries requests when the Google API responds with an error code listed in the **Retry Error Codes** field.
Maximum Retries Limit	Determines the number of times the adapter retries a request. The default value is `5`.
Retry Error Codes	The list of error codes that cause the adapter to retry a request. Separate multiple error codes with a comma. For a list of error codes, see Error Handling in the Google documentation. The default value is `403,503`.
API Request Timeout	The amount of time in milliseconds that PingFederate waits for the Google API to respond to requests. A value of `0` disables the timeout. The default value is `10000`.
Connection Timeout	The amount of time in milliseconds that PingFederate allows to establish a connection with the Google API. A value of 0 disables the timeout. The default value is `10000`.
Proxy Settings	Defines proxy settings for outbound HTTP requests. The default value is **System Defaults**.

Copyright © 2025 Ping Identity Corporation

Field	Description
Custom Proxy Host	The proxy server host name to use when **Proxy Settings** is set to **Custom**. This field is blank by default.
Custom Proxy Port	The proxy server port to use when **Proxy Settings** is set to **Custom**. This field is blank by default.

Available user attributes reference

The Google IdP Adapter can retrieve the following user attributes from Google.

To retrieve a specific attribute, configure the Google IdP Adapter instance in two places:

- On the **IdP Adapter** screen, select the **Attribute Retrieval** option for the attribute.
- On the **Extended Contract** screen, add the attribute.

The `email` attribute is always available.

For more information about these attributes, see Users in the Google documentation.

Attribute Retrieval: Email

Attribute Name	Description
`sub`	The subject identifier of the authenticated user.
`email`	The email address of the authenticated user. This value is retrieved from the ID token.

Attribute Retrieval: Basic Profile

Attribute Name	Description
`sub`	The subject identifier of the authenticated user.
`email`	The email address of the authenticated user. This value is retrieved from the ID token.
`name`	The user's full name.
`given_name`	The user's first name.
`family_name`	The user's last name.
`locale`	The user's preferred locale.
`hd`	The hosted domain name of the user's G Suite account, such as `example.com`.

Copyright © 2025 Ping Identity Corporation

Attribute Name	Description
`email_verified`	A boolean flag that indicates the verification status of the user's email address. A value of `true` means the email address is verified.

Attribute Retrieval: Extended Profile

Attribute Name	Description
`email`	The email address of the authenticated user. This value is retrieved from the ID token.
`kind`	Identifies this resource as a person in OpenID Connect format.
`id`	The user's unique ID.
`givenName`	The user's first name.
`familyName`	The user's last name.
`fullName`	The user's full name.
`externalIds`	The raw JSON string.
`groups`	A multi-value list of groups that the user is a member of.

Configuring SSO to an application within the PingFederate domain

To enable SSO to an application within the PingFederate domain, configure an adapter-to-adapter mapping.

Steps

1. On the PingFederate administrative console, create an SP adapter instance to integrate with your web application.

   For an overview of SP adapters, see Managing SP adapters in the PingFederate documentation. For most use cases, we recommend the Reference ID Adapter included in the Agentless Integration Kit.

2. Map the attributes from your Google IdP Adapter instance to the SP adapter instance by creating an adapter-to-adapter mapping. For instructions, see Adapter-to-adapter mappings in the PingFederate documentation.

3. In your web application, add sign-on hyperlink based on the following URL:

   ```
   https://pf_host:pf_port/pf/adapter2adapter.ping?
   IdpAdapterId=IdpAdapterId&SpSessionAuthnAdapterId=SpAdapterId
   ```

pf_host	The host name or IP address of the PingFederate server.
pf_port	The port number of the PingFederate server.

Copyright © 2025 Ping Identity Corporation

IdpAdapterId	The instance ID of the IdP adapter instance.
SpAdapterId	The instance ID of the SP adapter instance.

Configuring SSO to an SP outside the PingFederate domain

To enable SSO to a service provider (SP) outside the PingFederate domain, configure an SP connection and add a sign-on link to the SP application.

Steps

1. Create an SP connection and select your Google IdP Adapter instance on the **Authentication Source Mapping** tab.

   For detailed instructions, see SP connection management in the PingFederate documentation.

2. In your web application, add sign-on hyperlink based on the following URL:

```
https://pf_host:pf_port/idp/startSSO.ping?PartnerSpId=ConnectionId
```

pf_host	The host name or IP address of the PingFederate server.
pf_port	The port number of the PingFederate server.
ConnectionId	The ID of the SP connection.

Troubleshooting

Enabling debug logging

To help with troubleshooting or monitoring, you can turn on activity logging for the adapter.

About this task

These steps are optional. For general information about logging, see Enabling debug messages and console logging in the PingFederate documentation.

Steps

1. Open the `<pf_install>/pingfederate/server/default/conf/log4j2.xml` file for editing.
2. If you want to log activity for PingFederate and all adapters, do the following.
    1. Find the following section.

Copyright © 2025 Ping Identity Corporation

```
<AsyncRoot level="INFO" includeLocation="false">
        <!-- <AppenderRef ref="CONSOLE" /> -->
        <AppenderRef ref="FILE" />
</AsyncRoot>
```

2. Change `INFO` to `DEBUG`.

```
<AsyncRoot level="DEBUG" includeLocation="false">
        <!-- <AppenderRef ref="CONSOLE" /> -->
        <AppenderRef ref="FILE" />
</AsyncRoot>
```

3. If you want to see the adapter activity in the console, remove the comment tags.

```
<AsyncRoot level="INFO" includeLocation="false">
        <AppenderRef ref="CONSOLE" />
        <AppenderRef ref="FILE" />
</AsyncRoot>
```

3. If you want to log activity just for the Google IdP Adapter, add the following line.

```
<Logger name="com.pingidentity.adapters.idp.google" level="DEBUG"/>
```

4. If you want to log all HTTP requests and responses with the Google API, add the following line.

```
<Logger name="org.shaded.googlecic.apache.wire" level="DEBUG"/>
```

5. Save the file.

Release notes

Changelog

Google Login Integration Kit 1.5.2 - March 2025

- Fixed a vulnerability where unconsumed non-`200` HTTP responses could impact Jetty thread availability, potentially leading to service disruption or Denial of Service. Users should upgrade to mitigate any potential risks.

Google Login Integration Kit 1.5.1 - November 2022

- Fixed an issue that caused the Google IdP adapter to not encode spaces in scopes.
- Fixed an issues that caused an error when configuring the Google login adapter with custom proxy and host settings.

Copyright © 2025 Ping Identity Corporation

Google Login Integration Kit 1.5 - September 2020

- Added the ability to show the Google sign-on prompt in a pop-up window for environments that block the automatic redirect.
- Added support for the JavaScript Widget for the PingFederate Authentication API.
- Added support for the PingFederate authentication API.
- Added settings for retrying failed requests.
- Added settings for API connection and request timeouts.
- Added settings to override the PingFederate system-default proxy settings.

Google Login Integration Kit 1.4.1 - June 2019

- Fixed an issue that was causing an error in upgrade scenarios.

Google Login Integration Kit 1.4 - May 2019

- Added support for retrieving group memberships in the adapter configuration and extended contract.
- Added the ability to override the system-default proxy settings in the adapter configuration.
- Added the ability to configure timeout durations in the adapter configuration.
- Added support for virtual host names. In the adapter configuration, **PingFederate Base URL** is now an optional override field. See Virtual host names in the PingFederate documentation.

Google Login Integration Kit 1.3 - February 2019

- Updated the adapter to use a different user data source in preparation for the shutdown of the Google+ API.

Google Login Integration Kit 1.2 - August 2018

- Resolved an iframe issue with SLO.
- Added support to periodically retry a failed request with increasing delays between retries.
- Improved logging for error scenarios.
- Removed the bundled OAuth CIC helper as Ping Identity's OAuth Configuration Service replaces the functionality.

Google Login Integration Kit 1.1.1 - June 2016

- Resolved an issue with using the Google IdP Adapter within a Composite Adapter.

Google Login Integration Kit 1.1 - April 2015

- Upgraded the adapter interface to V2 of the PingFederate SDK (IdpAuthenticationAdapterV2).
- Added username attribute to the Basic and Extended Profile schemas.

Copyright © 2025 Ping Identity Corporation

- Added support for additional attributes to the Extended Profile schema.
- Added mitigation for Google API throttling through the use of the quotaUser parameter.
- Minor bug fixes.

Google Login Integration Kit 1.0.1 - May 2015

- Added support for openid_id parameter.
- Security enhancements.

Google Login Integration Kit 1.0 - May 2014

- Initial release.

Known issues and limitations

The following are known issues or limitations for the Google Login Integration Kit.

Known issues

There are no known issues.

Known limitations

- Sign-on can fail when attempted immediately after the user logs out. Refresh or close the browser window, then try again.

Download manifest

The following files are included in the Google Login Integration Kit `.zip` archive:

- `ReadMeFirst.pdf` – contains links to this online documentation
- `Legal.pdf` – copyright and license information
- `dist` – contains the integration files
  - `deploy` – contains the Java libraries
    - `pf-google-adapter-<version>.jar` – JAR file that contains the Google IdP Adapter.
  - `conf` – contains the HTML template that presents the Google sign-on form.
    - `language-packs` – contains files with customizable user-facing messages
      - `authn-api-messages.properties` – a version of the common PingFederate authentication API variable file that includes adapter-specific messages
      - `google-messages.properties` – a variable file that customizes the messages that appear on the template file.

Copyright © 2025 Ping Identity Corporation

- `template` – contains user-facing HTML template files
  - `google-pop-up-template.html` – a page that presents the Google sign-on window in the **Pop-up window** sign-on presentation mode.
  - `google-post-auth-template.html` – a page that signals the main template or authentication widget to continue the authentication flow in the **Pop-up window** sign-on presentation mode.
  - `assets` – contains functional scripts and files used by the template
    - `css/google.css` – a CSS file that customizes the appearance of the template file.
    - `fonts/end-user` – contains template fonts and icons
      - `icons` – contains template icons
      - `proxima-nova` – contains template fonts
    - `images` – contains template image files
      - `ping-logo.svg` – an image file with company branding
- `lib/pf-authn-api-sdk-<version>.jar` – a JAR file that contains the PingFederate Authentication API SDK

Google Workspace Provisioner

The Google Workspace Provisioner extends PingFederate capabilities, enabling enterprises to provision its users to Google Apps. It includes a quick-connection template to easily set up a Single Sign-On (SSO) connection and provisioning to Google Apps.

The Google Workspace Provisioner makes use of Google's Admin SDK. The Google Workspace Provisioner also includes a separate Password Manager application that can be used in cases where SSO users might need to reset Google Apps passwords. For more information, see The Password Manager.

Features

- Outbound User Provisioning
- Outbound Group Provisioning
- Ability to Add Users to Groups
- Browser-based SP-initiated SSO

Intended audience

This document is intended for PingFederate administrators.

If you need help during the setup process, see the following sections of the PingFederate documentation:

- SP connection management
- Identity provider SSO configuration
- Datastores

Copyright © 2025 Ping Identity Corporation

- Configuring outbound provisioning
- Configuring outbound provisioning settings

System requirements

- PingFederate 9.0 or later
- You might need to allow `https://www.googleapis.com` in your firewall to allow outbound connections.

User and group management

The Google Workspace Provisioner synchronizes users and groups from your datastore to Google Workspace. The following describes the behavior of each provisioning capability.

> **Tip**
>
> You can configure the following capabilities and specify which users to provision when you get to the Creating a connection part of the setup process.

Synchronizing existing users

PingFederate synchronizes users based on the `Email` attribute in Google Workspace. If a user already exists in your datastore and Google Workspace, mapping this attribute correctly links the two records together.

For example:

- In Google Workspace, Janet's `Email` is `jsmith@domain.com`.
- In your datastore, Janet's `mail` is `jsmith@domain.com`.
- On the **Attribute Mapping** tab of your provisioning connection configuration, map the `Email` attribute to `mail`.
- When the provisioning connector runs, the datastore user is provisioned with a `Email` of `jsmith@domain.com`. That matches Janet's existing `Email` in Google Workspace, so her information in the datastore is synchronized to her Google Workspace account.

User provisioning

PingFederate provisions users when any of the following happens:

- A user is added to the datastore group or filter that is targeted by the provisioning connector.
- A user with `disabled` status is added to the datastore group or filter that is targeted by the provisioning connector, and the **Provision disabled users** provisioning option is enabled. This feature is not available in all provisioning connector versions.

You can define which users PingFederate targets for provisioning on the **Source Location** tab of your provisioning connection configuration.

User updates

PingFederate updates users when a user attribute changes in your datastore.

Copyright © 2025 Ping Identity Corporation

You can define which attributes PingFederate monitors for changes on the **Attribute Mapping** tab of your provisioning connection configuration.

User deprovisioning

PingFederate deprovisions users when any of the following happens:

- A user is deleted from the user store.
- A user is disabled in the user store.
- A user is removed from the datastore group or filter that is targeted by the provisioning connector.

The **Remove User Action** setting in the connection configuration determines whether the deprovisioning action disables or deletes the user.

Synchronizing existing groups

PingFederate synchronizes groups from the datastore to the target service based on the group name.

For example:

- In Google Workspace, there is a group is named `Accounting`.
- In your datastore, there is a group with a `CN` of `Accounting`.
- When the provisioning connector runs, the two groups are synchronized.

Group provisioning

PingFederate provisions groups when a group is added to the datastore filter that is targeted by the provisioning connector.

You can define which groups PingFederate targets for provisioning and monitors for changes on the **Source Location** tab in your provisioning connection configuration.

Group name updates

PingFederate renames groups when they are renamed in the datastore.

Group membership updates

PingFederate updates group memberships when memberships change in the datastore, whether the change is in the group's properties or a user's properties.

Group memberships in the datastore overwrite the group memberships in Google Workspace.

Group deletion

PingFederate deletes groups when any of the following happens:

- The group is deleted in the datastore.
- The group is removed from the datastore group or filter that is targeted by the provisioning connector.

Copyright © 2025 Ping Identity Corporation

> **Important**
>
> Group deletions are permanent and cannot be undone.

Setup

Obtain an application name, client ID, and secret

About this task

The Google Workspace Connector's Outbound Provisioning functionality is built using Google's Admin SDK, which requires an OAuth 2.0 access token for authentication. To obtain the access token, you will need to first obtain your Application Name, Client ID and Secret from Google Apps.

> **Important**
>
> The Google Workspace Provisioner uses APIs provided by Google, which are subject the Terms of Service in the Google Admin SDK documentation.

> **Note**
>
> API Access will need to be enabled on the Google domain in order to use the Google Workspace Provisioner for Outbound Provisioning. For information on how to enable API Access for a Google domain, see Control which third-party & internal apps access Google Workspace data in the Google documentation.

Steps

1. Sign on to the Google Developers Console with a Super Administrator Google account.

   For information, see Google Developers Console in the Google documentation.

2. Create a new project (if you don't already have one).
3. Set the APIs for your project by clicking **Enable APIs and Services**. Enable the **Admin SDK** for your project.
4. Configure the Consent Screen for your project.

   > **Tip**
   >
   > The Consent Screen is the screen admins will see when obtaining their Access and Refresh tokens in another step.

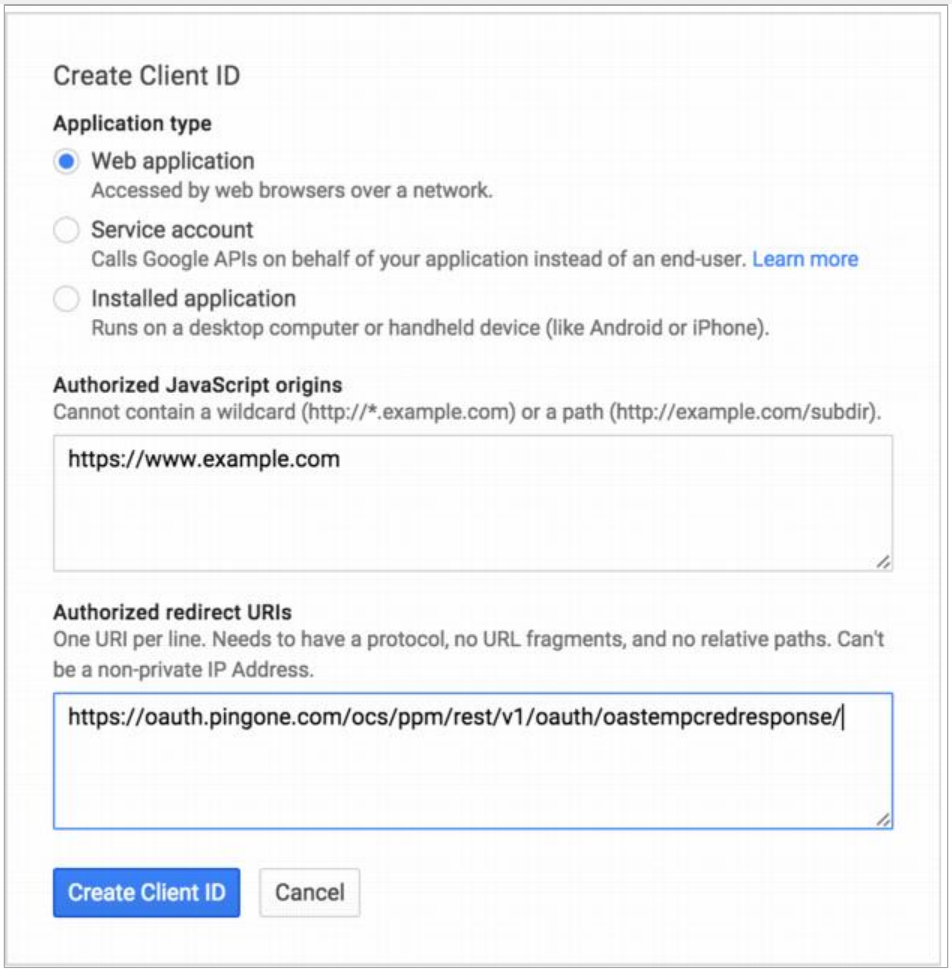
   1. Set the Product Name and make note of this value as it is required for the APPLICATION_NAME.
   2. Set the other fields as required by your organization.

5. Generate Credentials for your project by doing the following:

   1. Create a new OAuth 2.0 Client ID for a Web Application type application.
   2. Set the Authorized Redirect URI field to:

Copyright © 2025 Ping Identity Corporation

```
https://oauth.pingone.com/ocs/ppm/rest/v1/oauth/oastempcredresponse/
```



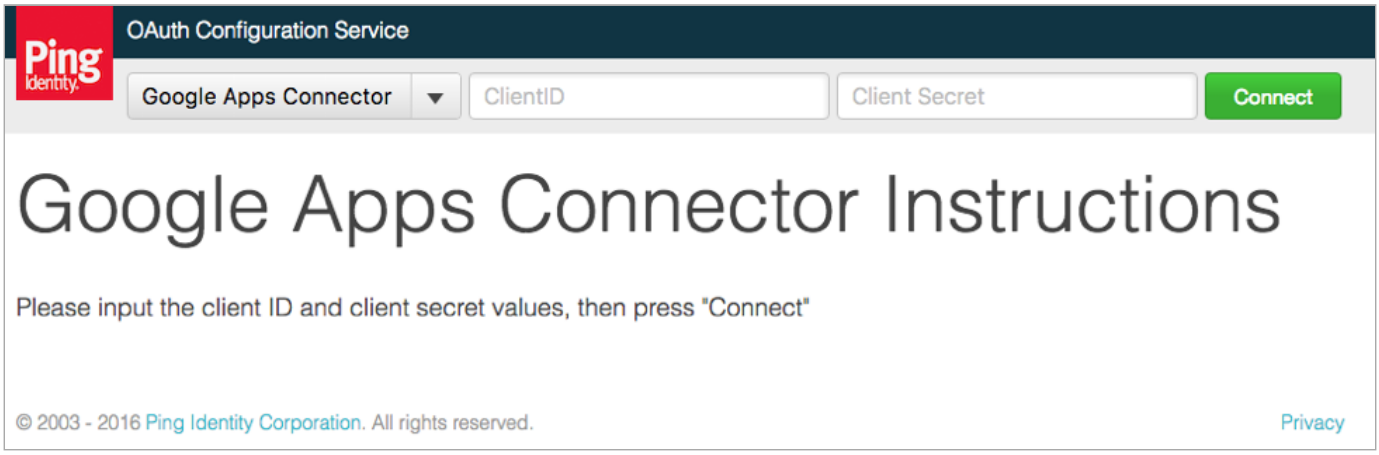
6. Finish by saving or creating the Client ID.
7. Make note of the Client ID and Client Secret that you just created.

Generate authorized OAuth 2.0 tokens

Steps

1. Go to the Ping Identity OAuth Configuration Service.
2. Select the **Google Workspace Connector** in the list.
3. Enter the **Client ID** and **Client Secret** that you generated in the Obtain an application name, client ID, and secret section of this guide.

Copyright © 2025 Ping Identity Corporation



4. Click **Connect** to proceed. This generates an OAuth 2.0 authorization token and redirects you to Google for authorization.
5. Log on to Google with a Super Administrator account.

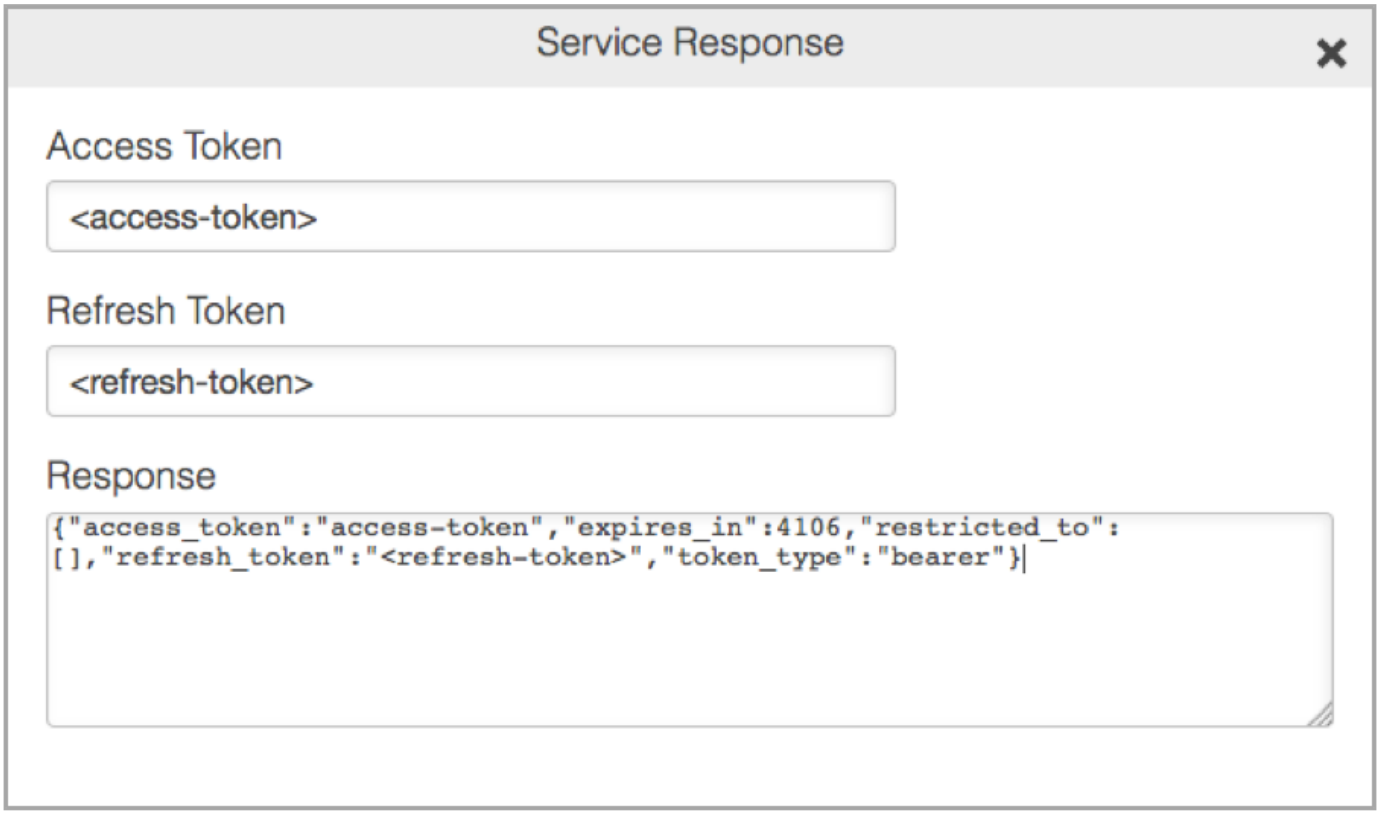
> **Note**
>
> If you are already signed on to Google Apps, you will not be asked to sign on again.
> Please be sure that the account you are signed in under is an administrative account.

6. On successful login, you are redirected to Google's OAuth authorization screen, where you'll be asked to grant access to the scopes that the Google Workspace Provisioner uses to make requests to the Google Admin SDK.
7. Once you grant access on the OAuth authorization screen, you will be redirected to the OCS and presented with an authorized Access Token and Refresh Token to use when configuring the Google Workspace Provisioner.

> **Warning**
>
> Only one Refresh Token will be generated per Client ID; so it is important to make note of the Refresh Token presented by the OCS in the final step.

> **Warning**
>
> If another Refresh Token is required, you will need to obtain a new Client ID and Secret and to use with the OCS again.

Copyright © 2025 Ping Identity Corporation



8. Copy the **Access Token** and **Refresh Token** to use when configuring the Google Workspace Provisioner.

Upgrade an existing deployment

Steps

1. Before stopping the PingFederate server to upgrade the Google Workspace Provisioner, access the **Attribute Mapping** screen for existing channel configurations and note the current configuration.

> ⚠ **Warning**
>
> The upgrade process removes existing mappings and defaults on the **Attribute Mapping** screen. These must be configured again before activating the channel configuration.

2. Disable the existing SP Connection where the Google Workspace Provisioner is configured.
3. Delete the existing Google Workspace Provisioner SP Connection and save.
4. Stop the PingFederate server if it is running.
5. Unzip the Google Workspace Provisioner distribution `.zip` archive into a holding directory.
6. Remove any versions of pf-google-quickconnection-x.x.jar from this directory:

```
<pf_install>/pingfederate/server/default/deploy
```

Copyright © 2025 Ping Identity Corporation

7. Also remove the following files from the same directory if they are present:
    - prov-google-x.x.x.jar
    - pf-google-oauth-helper.war

   as well as these Google Admin SDK libraries:
    - google-api-client-x.x.x-rc.jar
    - google-api-services-admin-directory-directory_v1-rev-x.x.x-rc.jar
    - google-http-client-x.x.x-rc.jar
    - google-http-client-jackson2-x.x.x-rc.jar
    - google-oauth-client-x.x.x-rc.jar
    - google-oauth-client-jetty-x.x.x-rc.jar
    - jackson-core-x.x.x.jar
8. From the dist directory of the new version of the connector, copy this file:

   ```
   pf-google-quickconnection-3.1.1.jar
   ```

   into this directory:

   ```
   <pf_install>/pingfederate/server/default/deploy
   ```
9. Start the PingFederate server.
10. Create a new SP Connection, using Google Apps as the Connection Template.
11. Follow the instructions in Creating a connection to configure Metadata, and OAuth
12. Access the Attribute Mapping for existing channel configurations and click **Refresh Fields**.
13. Ensure all new required fields (if any), are mapped appropriately or have a default value.
14. Once completed with the attribute configuration, click **Done**, **Done**, and **Save**.
15. Activate the SP Connection to resume Outbound Provisioning.

Deploying the integration files

About this task

To get started with the integration, deploy the Google Workspace Provisioner files to your PingFederate directory.

Steps

1. Stop the PingFederate server if it is running.

Copyright © 2025 Ping Identity Corporation

2. Remove any existing Google Workspace Provisioner files from the directory:

   `<PF_install>/pingfederate/server/default/deploy`

3. Unzip the distribution file and copy the contents of the `/dist` directory to the PingFederate directory:

   `<PF_install>/pingfederate/server/default/deploy`

4. Edit the `run.properties` file located in `<pf_install>/pingfederate/bin`, changing the property `pf.provisioner.mode` to the value shown here:

   `pf.provisioner.mode=STANDALONE`

   The property is located near the end of the file.

   For information about using the `FAILOVER` setting for runtime deployment, see the PingFederate Server Clustering Guide.

5. Start or restart the PingFederate server.

Enabling provisioning and single sign-on in PingFederate

To use PingFederate for provisioning and single sign-on, configure an external datastore and set a SAML entity ID.

About this task

Your external datastore acts as the source of data for provisioning. PingFederate also uses an internal datastore to store the state of synchronization between the source datastore and the target datastore.

For more information, see Datastores and Configuring outbound provisioning settings in the PingFederate documentation.

PingFederate 10.1 or later

Enabling provisioning and single sign-on in PingFederate 10.1 or later
Steps

1. In the PingFederate administrative console, configure the datastore for PingFederate to use as the source of user data.

   For help, see Datastores in the PingFederate documentation.

2. Go to **System > Server > Protocol Settings > Federation Info**.

3. In the **SAML 2.0 Entity ID** field, enter a name for PingFederate to use when identifying itself to SAML partners.

4. On the **Outbound Provisioning** tab, in the **Provisioner Data Store** list, select the internal database that will store the synchronization state. Click **Save**.

   For more information, see Configuring outbound provisioning settings in the PingFederate documentation.

Copyright © 2025 Ping Identity Corporation

PingFederate 10.0 or earlier

Enabling provisioning and single sign-on in PingFederate 10.0 or earlier
Steps

1. In the PingFederate administrative console, configure the datastore for PingFederate to use as the source of user data.

   For help, see Datastores in the PingFederate documentation.

2. Go to **System > Protocol Settings > Roles & Protocols**.
3. Select the **Enable Identity Provider IdP Role and Support the Following** check box.
4. Select the **SAML 2.0** and **Outbound Provisioning** check boxes. Click **Next**.
5. Click the **Federation Info** tab.
6. In the **SAML 2.0 Entity ID** field, enter a name for PingFederate to use to identify itself to SAML partners.
7. On the **Outbound Provisioning** tab, in the **Provisioner Data Store** list, select the internal database that will store the synchronization state. Click **Save**.

   For more information, see Configuring outbound provisioning settings in the PingFederate documentation.

Creating a connection

To allow PingFederate to act as an identity provider and manage users in Google Workspace, create a service provider (SP) connection.

Steps

1. In the PingFederate administrator console, create a new SP connection:

   Choose from:

   - For PingFederate 10.1 or later: go to **Applications > Integration > SP Connections**. Click **Create Connection**.
   - For PingFederate 10.0 or earlier: go to **Identity Provider > SP Connections**. Click **Create Connection**.

2. Configure the basic connection details with the Google Workspace quick connection template.
   1. On the **Connection Template** tab, select **Use a template for this connection**.
   2. From the **Connection Template** list, select **Google Workspace Provisioner**.
   3. In the **Google Domain** field, the Google Domain used by your organization for SSO access to Google Apps.

      Note

      If your Google Apps administrative implementation supports more than one domain, select the **USE A DOMAIN SPECIFIC ISSUER** checkbox under the Google Domain. Checking this box allows you to configure additional SP connections for other domains at your site registered with Google Apps.

Copyright © 2025 Ping Identity Corporation

4. On the **Connection Type** tab select **Browser SSO Profiles** and **Outbound Provisioning**. Click **Next**.
5. On the **Connection Options** tab, click **Next**.
6. On the **General Info** tab, in the **Connection Name** field, enter a name of your choosing. Click **Next**.

3. On the **Browser SSO** tab, configure browser SSO.

   For more information, see Configuring IdP Browser SSO in the PingFederate documentation.

   1. On the **Browser SSO > SAML Profiles** tab, select only **SP-Initiated SSO**.
   2. On the **Browser SSO > Protocol Settings > Allowable SAML Bindings** tab, select only **Redirect**.

4. On the **Browser SSO > Protocol Settings > Signature Policy** tab, select the **Always sign the SAML Assertion** check box. Click **Next**.
5. On the **Credentials** tab, configure the connection credentials as shown in Configuring credentials in the PingFederate documentation. Click **Next**.
6. On the **Outbound Provisioning** tab, configure provisioning with the following details.

   For help, see Configuring outbound provisioning in the PingFederate documentation.

   1. On the **Target** tab, complete the fields as follows.

Field Name	Description
Application Name	The Application Name for the application created in Google Apps. For more information on obtaining an application name, client Id and secret, see Obtain an application name, client ID, and secret.
Domain	The Domain for the Google Apps account.
OAuth Access Token	The OAuth Access Token generated by the OAuth Configuration Service. For more information on obtaining authorized OAuth tokens, see the Generate authorized OAuth 2.0 tokens.
Oauth Client ID	The Oauth client ID for the application created in Google Apps. For more information on obtaining an application name, client Id and secret, see Obtain an application name, client ID, and secret.
Oauth Client Secret	The Oauth client secret generated during application creation for Google Apps.
OAuth Refresh Token	The OAuth refresh token generated by the OAuth Configuration Service.
User Create	**Selected** (default) – PingFederate creates users in Google Apps. **Cleared** - PingFederate does not create users in Google Apps.
User Update	**Selected** (default) – PingFederate updates existing users in Google Apps. **Cleared** - PingFederate does not update existing users in Google Apps.

Copyright © 2025 Ping Identity Corporation

Field Name	Description
User Disable/Delete	**Selected** (default) – PingFederate disables or deletes users in Google Apps. **Note** PingFederate can only re-enable a user if **User Update** is selected. **Cleared** – PingFederate does not disable or delete users in Google Apps.
Provision Disabled Users	This option applies when: ■ the **User Create** option is selected, and ■ the provisioning engine targets a user in the data store that has a "disabled" status. **Selected** (default) – PingFederate creates the user in Google Apps with a "disabled" status. **Cleared** – PingFederate does not create the user in Google Apps.
Remove User Action	This option applies when: ■ **User Disable/Delete** is selected, and ■ a previously-provisioned user no longer meets the condition set on the **Source Location** screen, or ■ a user has been disabled or deleted from the data store. **Disable** (default) – PingFederate disables the user in Google Apps. **Delete** – PingFederate deletes the user from Google Apps.

7. On the **Activation and Summary** tab, above the **Summary** section, click the toggle to turn on the connection. Click **Save**.

Supported attributes reference

The following table consists of the list of User attributes that can be mapped from the Attribute Mapping screen when configuring Outbound Provisioning channels.

Note

All values are validated by Google and must meet their requirements. See Admin SDK: Directory API in the Google Workspace Admin SDK documentation.

Attribute	Description
`primaryEmail`	The user's primary email address. This field is required and unique; it cannot be an alias of another user or group.
`familyName`	The user's last name. This field is required.
`givenName`	The user's first name. This field is required.

Copyright © 2025 Ping Identity Corporation

Attribute	Description
`password`	The user's initial password. Works with the `passwordHashFunction` attribute to support plain text and hashed passwords. **Important** To force users to update their password the next time they sign on by setting their `changePasswordAtNextLogin` attribute to `true`. **Note** A user's password can only be set when provisioning the user for the first time. The provisioner cannot manage it on subsequent updates.
`aliases`	List of user's alias email addresses. **Note** The maximum number of aliases a user may have is 30.
`includeInGlobalAddressList`	Indicates if the user's profile is visible in the Google Apps global address list when the contact sharing feature is enabled for the domain. Valid values include: • **true** • **false**
`passwordHashFunction`	The hash format of the password attribute. Depending on the password type, do one of the following: • For plain-text passwords, do not set a value for the `passwordHashFunction` attribute. • For hashed passwords, set the `passwordHashFunction` attribute to the appropriate hash type. The supported values are: • SHA-1 • MD5 • CRYPT • None (blank) **Note** Google recommends sending the password property value as a base 16 bit encoded hash value.

Copyright © 2025 Ping Identity Corporation

Attribute	Description
`orgUnitPath`	The full path of the parent organizational unit associated with the user. To add a user to the root OU, the `orgUnitPath` should be set to a forward slash (/). An example value for placing a user under an organization unit one level below the root, which is the domain and indicated with the forward slash (/), the value would be as follows: `/example` An example value for placing a user under an organizational unit two levels below the root: `/first_level/second_level`
`changePasswordAtNextLogin`	Indicates if the user is forced to change their password at next login. Valid options include: • **true** • **false**
`ipWhitelisted`	Indicates if the user's IP address is whitelisted. Valid values include: • **true** • **false**
`isAdmin`	Indicates a user with super administrative privileges. Valid values include: • **true** • **false** **Note** This field can only be set if a Super Admin account was used when generating the OAuth Access and Refresh Tokens used during the Connection configuration.
`orgName`	The name of an organization. **Note** Versions 3.0.1 and later of the Google Workforce Connector supports a single organization on the user.
`orgDept`	Specifies the users department within the organization, such as sales or engineering.
`orgSymbol`	The text string value of the organization. For example, the text symbol for Google is GOOG.
`orgCostCenter`	The cost center of the user's organization.
`orgDescription`	The description of the organization.
`orgDomain`	The domain the organization belongs to.
`orgTitle`	The user's title within the organization such as member or engineer.
`orgLocation`	The physical location of the organization.

Copyright © 2025 Ping Identity Corporation

Attribute	Description
`addressStreet`	The user's street address, such as 1600 Amphitheatre Parkway. **Note** Versions 3.0.1 and later of the Google Workforce Connector supports a single address on the user.
`addressPostalCode`	The ZIP or postal code of the address.
`addressExtended`	The extended portion of an address, such as an address that includes a sub-region.
`addressCountryCode`	The country code of the address. Uses the ISO 3166-1 standard (http://www.iso.org/iso/iso-3166-1_decoding_table).
`addressLocality`	The town or city of the address.
`addressRegion`	The abbreviated province or state of the address.
`addressCountry`	The country of the address.
`addressPoBox`	The post office box of the address.
`addressFormatted`	A full and unstructured postal address. This is not synced with the structured address fields. For this single string attribute you can include any of the following values: • Street address • P.O. box • City • State/province • ZIP/postal code • Country/region **Note** This attribute is only supported in version 3.2.1 or later of the provisioner.
`workPhone`	A human-readable phone number. It may be in any telephone format. **Note** When a value is mapped to this field, a phone is created for the user of type `work`.
`workPager`	A human-readable phone number. It may be in any telephone format. **Note** When a value is mapped to this field, a phone is created for the user of type `work_pager`.

Copyright © 2025 Ping Identity Corporation

Attribute	Description
`workFax`	A human-readable phone number. It may be in any telephone format. **Note** When a value is mapped to this field, a phone is created for the user of type `work_fax`.
`workMobile`	A human-readable phone number. It might be in any telephone format. **Note** When a value is mapped to this field, a phone is created for the user of type `work_mobile`.
`primaryPhone`	Indicates which phone is the user's primary phone. Only one phone may be marked as the primary phone. Valid values include: • `work` • `work_pager` • `work_mobile` • `work_fax` When set, the corresponding phone (`workPhone`, `workPager`, `workMobile` or `workFax`) will have its primary field set to true.

Complete setup of SAML SSO to Google Apps

About this task

In order to setup your Google Apps account for SSO you will need to do the following.

Note

This section requires the exported certificate used to sign the SAML assertion configured in step 5 of Creating a connection.

Steps

1. Go to `https://admin.google.com/<domain_name>/` and sign in with your Administrator credentials.
2. Go to **Security** to view the **Set up single sign-on (SSO)** section.
3. Select **Setup SSO with third party identity provider**.
4. Enter the PingFederate SSO SAML endpoint in the **Sign-in page URL** field.

   ```
   https://<pf_host>:<pf_port>/idp/SSO.saml2
   ```

5. (Optional) Enter the PingFederate SLO SAML endpoint in the **Sign-out page URL** field.

Copyright © 2025 Ping Identity Corporation

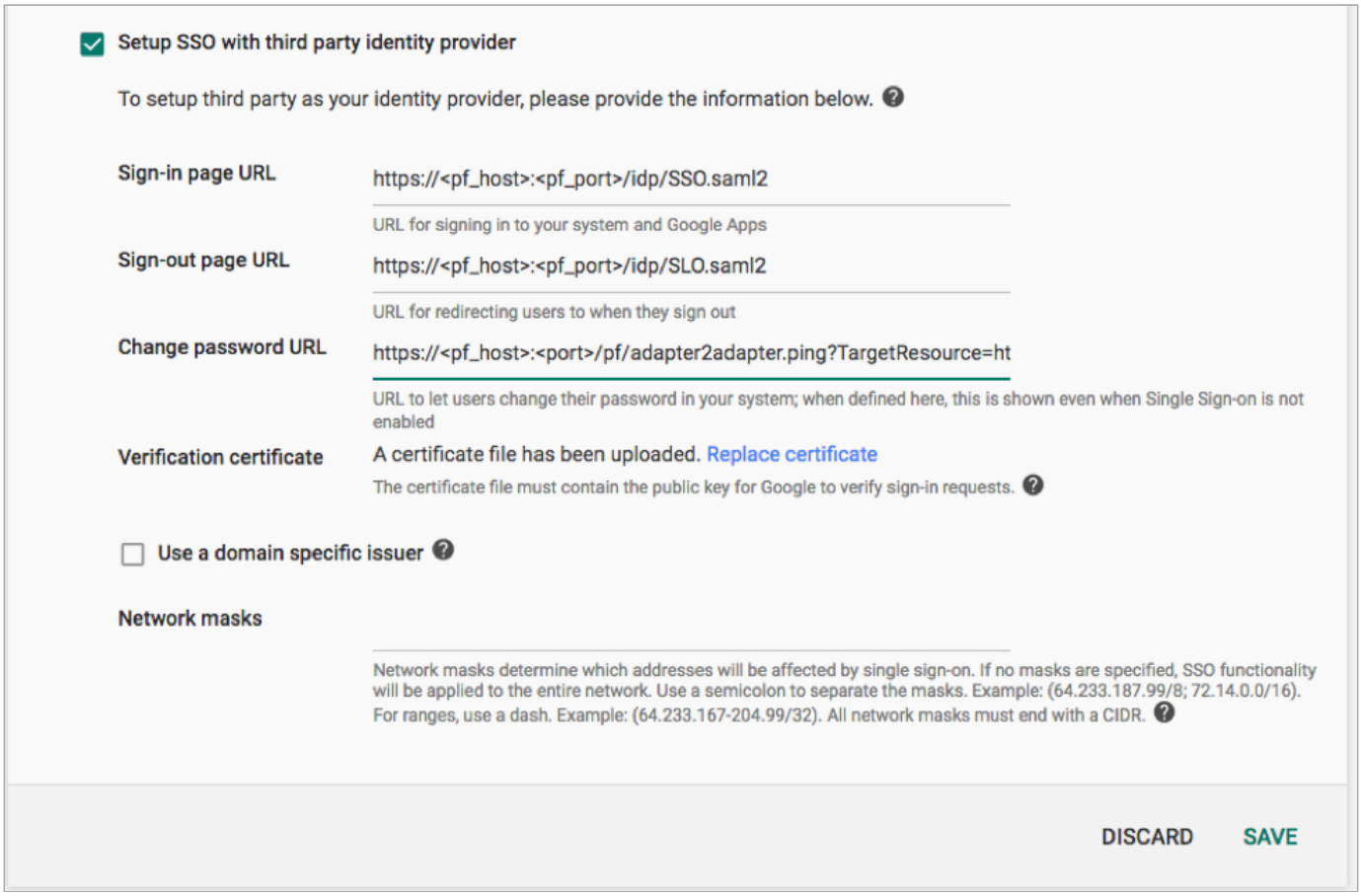
```
https://<pf_host>:<pf_port>/idp/SLO.saml2
```

6. (Optional) Enter the Password Manager URL in the **Change password URL** field.

```
http[s]://<pf_host>:<port>/pf/adapter2adapter.ping?TargetResource=http[s]://<g_apps_pm_host>:<port>/
gapps-password-manager/ResetPassword
```

7. Upload the signing certificate exported from PingFederate in the **Verification certificate** field.

8. Select **Use a domain specific issuer** if applicable.



9. Click **Save** to complete Google Apps SSO Setup.

> **Note**
>
> Learn more about Google Apps SSO setup in Set up SSO via a third-party Identity provider in the Google Workspace Admin Help.

Copyright © 2025 Ping Identity Corporation

The Password Manager

This Google Workspace Provisioner package includes a separate application, the Google Apps Password Manager that may be configured with PingFederate to allow end users to reset Google passwords. Because users who access Google Apps via SSO do not need Google credentials, they may forget (or never receive) individual passwords. The optional Password Manager is provided for situations in which enterprise users may need to obtain reset passwords to access Google Apps directly.

> **Note**
>
> As stated in Google's less secure apps policy change, starting May 30, 2022 the use of third-party apps or devices that ask a user to sign into their Google account using only a username and password will no longer be supported. For more information, see Less secure apps & your Google Account in the Google documentation.

The Password Manager is not dependent on the PingFederate SSO connection to Google Apps; it is deployed a stand-alone application either within PingFederate or in a separate Web container.

To use the Password Manager, PingFederate must be configured to act as both an IdP and an SP. Then define an instance of the PingFederate OpenToken SP Adapter (if one does not exist), which is used to identify the user via an encrypted security token, based on the user's ID attribute mapped directly to the SP adapter from the IdP authentication adapter.

The appendix below provides instructions for installing, configuring, and using the Password Manager.

- IdP Adapter Setup
- SP Adapter Setup
- IdP - SP Adapter Mapping
- Deploying the Application

IdP Adapter Setup

The IdP adapter instance can be the same one already used for Google Apps SSO, or you may use an instance of any IdP adapter (see Configuring an IdP adapter instance in the PingFederate documentation).

No special configuration is required for any IdP adapter to authenticate users for the Password Manager.

SP Adapter Setup

The Password Manager relies on a PingFederate security token, opentoken, to identify users and ensure that they are authenticated before resetting their Google passwords. To enable token creation, you must create an instance of the SP OpenToken Adapter and export a configuration file, which the Password Manager uses to retrieve and decrypt the token.

Refer to Configuring an OpenToken SP Adapter instance in the PingFederate documentation for setup instructions.

> **Note**
>
> In the Adapter Instance setup, no Extended Contract is required, and no changes are needed to the default Instance Configuration Advanced Fields—but be sure to click download on the Actions screen and Export the properties file to use later (see Deploying the Application).

Copyright © 2025 Ping Identity Corporation

> **Tip**
>
> On the Instance Configuration screen, under Show Advanced Fields, you may change the default Transport Mode of the opentoken from Query Parameter to Cookie (but not to POST).

IdP - SP Adapter Mapping

About this task

PingFederate provides an IdP-to-SP Adapter Mapping option on the Main Menu for special IdP use cases requiring PingFederate to act also as an SP on behalf of the actual SP partner. This mapping allows authentication credentials to be directly mapped to create an SP authenticated session or security context. In these cases, the special mapping eliminates the need to create complete SP and IdP connections in a loopback configuration for sending SAML messages back and forth to the same PingFederate server.

This section provides specific instructions for configuring this mapping to enable the Google Apps Password Manager. (For more information, see Adapter-to-adapter mappings in the PingFederate documentation.)

Steps

1. Ensure that PingFederate is configured to act as both an IdP and an SP, with applicable adapter instances defined on both sides (see the previous sections).
2. On the PingFederate Main Menu under System Settings, click **IdP-to-SP Adapter Mapping**.
3. On the Manage Mappings screen, select the Source and Target Instance for the IdP and SP Adapter Instances, respectively.

   > **Important**
   >
   > The Target Instance must be for the OpenToken Adapter (see SP Adapter Setup).

4. Click **Add Mapping**.
5. On the Data Store screen, click **Next**.

   Data-store lookup is not required for this application.
6. On the Adapter Contract Fulfillment screen, for subject, choose Adapter from the Source drop-down list and map the attribute to the subject ID coming from the IdP Adapter.
7. Click **Done** and then **Save** on the Manage Mappings screen.

Deploying the Application

The Password Manager is located in the Google Workspace Provisioner distribution package in the form of an extracted Web archive (WAR). The WAR directory can be installed and deployed either within PingFederate or inside a separate Web servlet container.

After the WAR is installed, one configuration file must be modified. An additional SP-adapter configuration file must be added before the application can be deployed. Learn more in Creating a connection.

- Configure and Deploy the Password Manager

Copyright © 2025 Ping Identity Corporation

- Access the Password Manager

Configure and Deploy the Password Manager

Steps

1. Copy `gapps-password-manager.war` from the `dist/gapps-password-manager` directory to either:

   `<pf_install>/pingfederate/server/default/deploy/`

   Or the application-deployment directory in a different Web-servlet container of your choice.

2. In the directory `gapps-password-manager.war/WEB-INF/classes`, edit the file `gapps-password-manager-config.props`, to provide valid client id, client secret, and oauth tokens for Google Apps.

   Follow the instructions in Obtain an application name, client ID, and secret section of this guide to obtain the client id and secret. Refer to Generate authorized OAuth 2.0 tokens for instructions on obtaining the token values.

   > **Tip**
   >
   > You can use the `obfuscate.bat|sh` utility to mask the client secret, access token and refresh token value in the configuration file (recommended). The utility is located in the `<pf-install>/pingfederate/bin` directory. Make sure to run the obfuscate utility with `-l` flag.
   > Example: `obfuscate.[bat|sh] -l <Value to be obfuscated>`

   As an option in this file, you may also change the default specifications (usable characters and length) for the randomly generated reset passwords that users will receive from the Password Manager.

3. Copy the `agent-config.txt` file, which was exported during the SP adapter, configuration, into the same directory. Learn more in SP Adapter Setup.

   `../gapps-password-manager.war/WEB-INF/classes/`

4. Start or restart PingFederate, or the servlet container in which the Manager is installed.

Access the Password Manager

After configuring PingFederate and deploying the application, users can access the Google Apps Password Manager using the URL below. You can find additional parameters in System-service endpoints in the PingFederate Administrator's Manual.

> **Note**
>
> If you have configured more than one IdP-to-SP adapter mapping, you must specify the SP-adapter instance ID as the value for the query parameter `SpSessionAuthnAdapterId`.

Copyright © 2025 Ping Identity Corporation

`http[s]://<pf_host>:<port>/pf/adapter2adapter.ping?TargetResource=http[s]://<g_apps_pm_host>:<port>/gapps-password-manager/ResetPassword`

where:

- `<pf_host>:<port>` is the PingFederate host server name or IP address and port number.
- `<g_apps_pm_host>:<port>` is the host server name or IP address and port number where the Password Manager is deployed (may be the same as for PingFederate).

Release notes

Changelog

Google Workspace Provisioner 3.2.1 – March 2023

- Added support for the `addressFormatted` user attribute.
- Fixed an issue that caused new users to not be provisioned with group membership.
- Fixed an issue where the disable provision action failed to disable users.

Google Workspace Provisioner 3.2 – March 2022

- Changed the product name to Google Workspace Connector.
- Added support for Google Admin SDK 1.32.1.
- Added the ability to disable and delete users.
- Added the ability to provision disabled users.
- Added the ability to remove user actions.

Google Workspace Provisioner 3.1.2 – December 2021

- Changed the product name to Google Workforce Connector
- No change to the product

Google Workspace Provisioner 3.1.1 – April 2017

- Fixed synchronization on update of users, that were previously created with "User Create Enabled" set to false in configurable options

Google Workspace Provisioner 3.1 – September 2016

- Added configuration options for CRUD capabilities
- Improved exception handling and reporting
- Added support for Google Admin SDK v1.22.0

Copyright © 2025 Ping Identity Corporation

Google Workspace Provisioner 3.0.4 – April 2015

- Fixed bug for orgUnitPath

Google Workspace Provisioner 3.0.3 – April 2015

- Updated Google Apps Password Manager to use latest API
- Removed Oauth Helper App in favor of using the Oauth Configuration Service (OCS)
- Bug Fixes

Google Workspace Provisioner 3.0.1 – October 2014

- Improved error handling when the SP Connection contains invalid target credentials

Google Workspace Provisioner 3.0 – August 2014

- Updated Connector to use Google Admin SDK
- Added support for new User fields
- Added support for Groups provisioning

Known issues and limitations

- When create, read, update, or delete operations are performed on an LDAP user, most changes propagate to Google within minutes. However, changes can sometimes take up to 24 hours to propagate. For more information, see How changes propagate to Google services.
- When the `onDisable` or `onUpdate` property's value is changed from false to true for an LDAP user, changes won't propagate to the target unless a new update is made to the user.

> **Note**
>
> The `onDisable` property doesn't disable LDAP users at the target when set to false. Neither does the `onUpdate` property update LDAP users at the target when set to false. This behavior is expected.

- You can only edit the `isAdmin` property through the `users.makeAdmin` method. Edits made to the `isAdmin` property through the `users.insert` or `users.update` methods are ignored by the Google API service. Learn more in Make a user an administrator in the Google developer documentation.
- User attributes cannot be cleared once set.
- When an LDAP user is deleted in a targeted group distinguished name (DN), the provisioning connector does not propagate the deletion until a new user is added to the group. This limitation is compounded when the **User Create** provisioning option is disabled. You can find solutions SaaS provisioner does not remove the user in the Ping Identity Knowledge Base.
- Google does not properly handle creating users with an invalid `addressCountry` value.

Copyright © 2025 Ping Identity Corporation

- The Connector sends the value of work for the Organization type but Google does not retain this value and as a result the Organization type has no value.
- Google treats certain user attributes as complex data sets: Address (address* attributes), organization (org* attributes), and phone (work* attributes). Any unmapped or empty fields within a complex data set will be cleared in the corresponding Google account.
- The Password Manager requires PingFederate 9.3 or later if deploying it within PingFederate.
- Nested group membership cannot be preserved. The Google Workspace Provisioner can provision all of the nested groups to Google Workspace when **Nested Search** is selected in the channel settings, but can't maintain the nested hierarchy. Learn more in Specifying a source location.

Download manifest

The distribution `.zip` archive for the Google Workspace Provisioner contains the following:

- `dist` – contains the integration files
  - `gapps-password-manager.war` – The Google Password Manager Application
  - `pf-google-quickconnection-<version>.jar` – PingFederate Google Workspace Provisioner
- `legal/Legal.pdf` – copyright and license information.

Google Chrome Enterprise Integration Kit

The Google Chrome Enterprise Integration Kit allows PingFederate to communicate with Google and retrieve Google Chrome Enterprise device posture information for use in authentication and authorization decisions.

Components

Google Chrome Enterprise Device Trust IdP Adapter

When a user signs on through PingFederate, the adapter validates if the request is from Google Chrome Enterprise and retrieves the device signal information for use in authentication policy decisions.

Intended Audience

This document is intended for PingFederate administrators.

If you need help during the setup process, see the following resources:

- Establish device trust with Chrome Enterprise and Ping Identity
- Manage Chrome Enterprise device trust connectors
- The following sections of the PingFederate documentation:
  - Managing IdP adapters

Copyright © 2025 Ping Identity Corporation

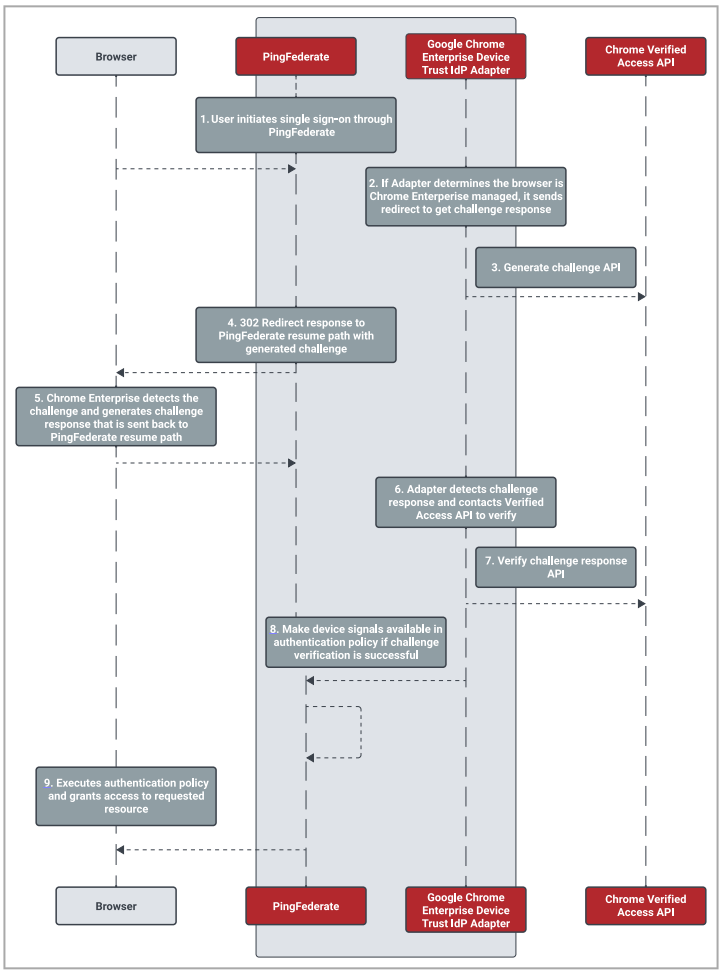
- Authentication policies

System requirements

- PingFederate 10.3 or later
- To allow PingFederate to make outbound HTTPS connections, you might need to allow the `https://verifiedaccess.googleapis.com` host name in your firewall.

Overview of the SSO flow

The following figure illustrates an example SSO process flow.



1. A user initiates the sign-on process by requesting access to a protected resource.
2. The Google Chrome Enterprise Device Trust IdP Adapter determines if the incoming request is from the Google Chrome Enterprise browser.
3. If the user's browser is Google Chrome Enterprise (Managed), the adapter makes a backend call to the Chrome Verified Access API endpoint to generate a challenge.
4. The adapter sends a 302 redirect to the PingFederate resume path with the challenge set in the response header.

Copyright © 2025 Ping Identity Corporation

5. The Google Chrome Enterprise browser processes the challenge and sets the response in the resume path request header.
6. The adapter finds the challenge response set by the browser and makes a backend call to the Chrome Verified Access API endpoint to verify the response.
7. After successful verification, the adapter has access to the device signals from the browser.
8. The adapter uses the decoded device signals to fulfill the core contract for the authentication policy for subsequent decision making.

Setup

Deploying the integration files

To get started with the integration, deploy the Google Chrome Enterprise Integration Kit files to your PingFederate directory.

Before you begin

Sign up for Chrome Browser Cloud Management and enroll cloud-managed Chrome browsers. For more information, see Chrome Browser Cloud Management.

Steps

1. Download the Google Chrome Enterprise Integration Kit `.zip` archive from the Ping Identity Integration Directory.
2. Stop PingFederate.
3. Extract the `.zip` archive and merge the contents of the `dist` directory with your `<pf_install>/pingfederate/server/default/deploy` directory.
4. Start PingFederate.
5. If you operate PingFederate in a cluster, repeat steps 2-4 for each engine node.

Configuring an adapter instance

Configure the Google Chrome Enterprise Device Trust IdP Adapter to determine how PingFederate communicates with Google Chrome Enterprise.

About this task

To get started with the integration, deploy the Google Chrome Enterprise Integration Kit files to your PingFederate directory.

Steps

1. In the PingFederate administrative console, go to **Authentication > Integration > IdP Adapters**. Click **Create New Instance**.
2. On the **Type** tab, set the basic adapter instance attributes:
    1. In the **Instance Name** field, enter a name for the adapter instance.

Copyright © 2025 Ping Identity Corporation

2. In the **Instance ID** field, enter a unique identifier for the adapter instance.
3. From the **Type** list, select **Google Chrome Enterprise Device Trust IdP Adapter**. Click **Next**.

3. On the **IdP Adapter** tab, in the **Google Verified Access API Response Mappings** section, map attributes from the Google Verified Access API response to the attribute contract:
    1. Click **Add a new row to 'Google Verified Access API Response Mappings'**.
    2. In the **Local Attribute** field, enter a name for an attribute.
    3. In the **Google Verified Access API Response Mapping** field, enter the JSON Pointer syntax for the source Google Verified Access API attributes as shown in JSON Pointer syntax reference.

        Example:

        The JSON pointer `/deviceSignals/displayName` returns the machine display name.
    4. In the **Action** column, click **Update**.
    5. To add more attributes, repeat steps a - d.

        Result:

        These attributes become available in your PingFederate authentication policy.
4. On the **IdP Adapter** tab, configure the adapter instance by referring to Google Chrome Enterprise Device Trust IdP Adapter settings reference. Click **Next**.
5. On the **Actions** tab, test your connection to Google Chrome Enterprise. Resolve any issues that are reported, and then click **Next**.
6. On the **Extended Contract** tab, add any attributes that you included in the **Google Verified Access API Response Mappings** section of the **IdP Adapter** tab. Click **Next**.
7. On the **Adapter Attributes** tab, set pseudonym and masking options as shown in Set pseudonym and masking options in the PingFederate documentation. Click **Next**.
8. On the **Adapter Contract Mapping** tab, configure the contract fulfillment details for the adapter as shown in Define the IdP adapter contract in the PingFederate documentation. Click **Next**.
9. On the **Summary** tab, review your configuration and then click **Save**.

JSON Pointer syntax reference

JavaScript Object Notation (JSON) Pointer defines a syntax for identifying a specific value within a JSON payload. Using the sample payload and JSON Pointer examples below, identify the attributes that you want to use to populate your attribute contract.

For a complete technical description of JSON Pointer syntax, see JavaScript Object Notation (JSON) Pointer on ietf.org.

Copyright © 2025 Ping Identity Corporation

Example Google Chrome Enterprise JSON payload


```
{
    "devicePermanentId": "bda2f4cc-4f6f-4c99-b2ce-5a35834a14d8",
    "deviceSignal": "{\"browserVersion\":\"119.0.6045.160\",\"builtInDnsClientEnabled\":true,
\"chromeRemoteDesktopAppBlocked\":false,\"deviceAffiliationIds\":[\"C0479s0n6\"],\"deviceEnrollmentDomain\":
\"pingtap.deviceadmin.goog\",\"deviceHostName\":\"EC2AMAZ-8FF6F2B.pfikteam.ping-eng.com\",\"deviceManufacturer\":
\"Amazon EC2\",\"deviceModel\":\"t3.xlarge\",\"diskEncrypted\":0,\"displayName\":\"EC2AMAZ-8FF6F2B\",
\"macAddresses\":[\"06-66-9C-B8-BF-89\"],\"os\":\"Windows\",\"osFirewall\":2,\"osVersion\":\"10.0.17763.1039\",
\"passwordProtectionWarningTrigger\":0,\"profileAffiliationIds\":[],\"realtimeUrlCheckMode\":
0,\"safeBrowsingProtectionLevel\":1,\"screenLockSecured\":2,\"secureBootEnabled\":0,\"serialNumber\":\"ec26134d-b166-
b3f9-1302-a2724eb7b6ab\",\"siteIsolationEnabled\":true,\"systemDnsServers\":[\"[::1]:53\",\"52.40.214.44:53\",
\"35.160.132.57:53\"],\"thirdPartyBlockingEnabled\":true,\"trigger\":1,\"windowsMachineDomain\":\"PFIKTEAM\",
\"windowsUserDomain\":\"PFIKTEAM\"}",
    "keyTrustLevel": "CHROME_BROWSER_OS_KEY",
    "virtualDeviceId": "95982e5e-5238-4a13-88a3-f0b09c6f182d",
    "customerId": "C0479s0n6",
    "deviceSignals": {
        "deviceManufacturer": "Amazon EC2",
        "deviceModel": "t3.xlarge",
        "operatingSystem": "WINDOWS",
        "osVersion": "10.0.17763.1039",
        "displayName": "EC2AMAZ-8FF6F2B",
        "diskEncryption": "DISK_ENCRYPTION_UNKNOWN",
        "serialNumber": "ec26134d-b166-b3f9-1302-a2724eb7b6ab",
        "osFirewall": "OS_FIREWALL_ENABLED",
        "systemDnsServers": [
            "[::1]:53",
            "52.40.214.44:53",
            "35.160.132.57:53"
        ],
        "hostname": "EC2AMAZ-8FF6F2B.pfikteam.ping-eng.com",
        "macAddresses": [
            "06-66-9C-B8-BF-89"
        ],
        "screenLockSecured": "SCREEN_LOCK_SECURED_ENABLED",
        "secureBootMode": "SECURE_BOOT_MODE_UNKNOWN",
        "windowsMachineDomain": "PFIKTEAM",
        "windowsUserDomain": "PFIKTEAM",
        "deviceEnrollmentDomain": "pingtap.deviceadmin.goog",
        "browserVersion": "119.0.6045.160",
        "deviceAffiliationIds": [
            "C0479s0n6"
        ],
        "builtInDnsClientEnabled": true,
        "chromeRemoteDesktopAppBlocked": false,
           "safeBrowsingProtectionLevel": "STANDARD",
        "siteIsolationEnabled": true,
        "passwordProtectionWarningTrigger": "POLICY_UNSET",
        "realtimeUrlCheckMode": "REALTIME_URL_CHECK_MODE_DISABLED",
        "thirdPartyBlockingEnabled": true,
        "trigger": "TRIGGER_BROWSER_NAVIGATION"
    }
}
```


Copyright © 2025 Ping Identity Corporation

JSON Pointer syntax

Description	JSON Pointer	Example value
Browser version	`/deviceSignals/browserVersion`	`19.0.6045.160`
MAC addresses	`/deviceSignals/macAddresses`	`["06-66-9C-B8-BF-89"]`

> **Note**
>
> To populate an attribute with the entire JSON response, leave the Google Verified Access API Response Mappings field blank.

Google Chrome Enterprise Device Trust IdP Adapter settings reference

Field descriptions for the Google Chrome Enterprise Device Trust IdP Adapter configuration screen.

Standard fields

Field	Description
API Key	The API key created in Google Cloud Platform. This field is required.
GCP Key File	The key file downloaded from GCP during the Service Account creation. This field is required to create an adapter instance.

Advanced fields

Field	Description
Google Verified Access Base URL	The endpoint used by Google Verified Access. This field is required and is set with default value https://verifiedaccess.googleapis.com/.
Generate Challenge Endpoint	The endpoint used to generate a new challenge. This is generated relative to **Google Verified Access Base URL** and **API Key** configured above. The default endpoint is `/v2/challenge:generate?key=%s`.
Verify Challenge Response Endpoint	The endpoint used to verify challenge response. This is generated relative to **Google Verified Access Base URL** and **API Key** configured above. The default endpoint is `/v2/challenge:verify?key=%s`.
API Request Timeout	The amount of time in milliseconds that PingFederate allows when establishing a connection with Google Verified Access API or waiting for a response to a request. A value of 0 disables the timeout. The default value is `5000`.

Copyright © 2025 Ping Identity Corporation

Field	Description
Proxy Settings	Defines proxy settings for outbound HTTP requests. The default value is **System Defaults**.
Custom Proxy Host	The proxy server host name to use when **Proxy Settings** is set to **Custom**. This field is blank by default.
Custom Proxy Port	The proxy server port to use when **Proxy Settings** is set to **Custom**. This field is blank by default.

Using Chrome device signals

The adapter looks for a specific request header to determine if the request is originating from a Google Chrome Enterprise managed browser to initiate its flow to retrieve device signals from the browser.

For requests originating from non-Chrome browsers, Chrome browsers that are not managed, or a Chrome incognito window, the adapter returns immediately with `SUCCESS` as the status with only one core contract attribute, **deviceTrustEnabled**, set to **false**.

The authentication policy should typically use the **deviceTrustEnabled** contract attribute to make decisions related to stepping authentication flow with multi-factor authentication (MFA).

For requests originating from managed Chrome browsers, the adapter initiates the flow as described in Overview of the SSO flow to retrieve device signals. The adapter completes the flow successfully if the request originates from a managed browser and the adapter can retrieve device signals for the account it is configured to work with Google Cloud Platform. Depending on the availability of the attribute in the device signals, the adapter fulfills several core contract attributes such as:

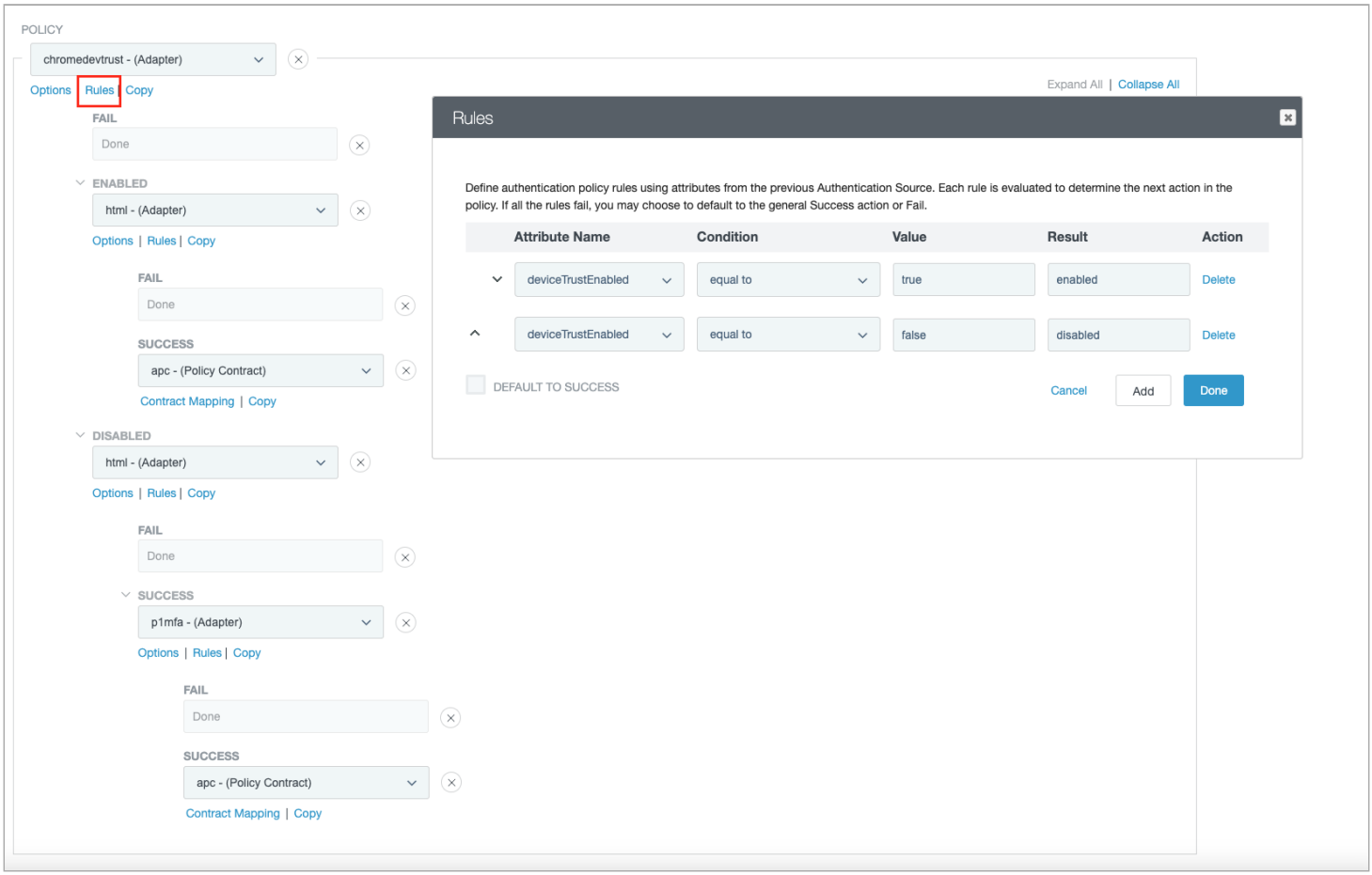
- **browserVersion**
- **displayName**
- **hostname**
- **macAddresses**
- **operatingSystem**

The core contract attribute **deviceTrustEnabled** is always set to **true** after a successful flow.

If a request originates from a managed browser that does not correspond to the account setup in the adapter, the adapter fails to retrieve device signals, and the adapter fails the authentication flow completely. Runtime errors, such as network errors invoking Google APIs resulting in timeouts, and other unexpected errors also cause the adapter to fail the authentication flow.

You can configure the authentication policy to handle these cases. For example, the following policy configuration shows a typical use case where a request coming from a trusted device undergoes only HTML form adapter authentication, and a request coming from an untrustworthy device undergoes HTML form adapter authentication as the first factor, followed by MFA authentication as the second factor.

Copyright © 2025 Ping Identity Corporation



Troubleshooting

Enabling debug logging

To help with troubleshooting or monitoring, you can optionally turn on activity logging for PingFederate, the Google Chrome Enterprise Device Trust IdP Adapter, or both.

About this task

You can use logging for troubleshooting or analytics.

For general information about logging, see Enabling debug messages and console logging in the PingFederate documentation.

Steps

1. Open the `<pf_install>/pingfederate/server/default/conf/log4j2.xml` file for editing.
2. To log activity for PingFederate and all adapters:
    1. Find the following section:

Copyright © 2025 Ping Identity Corporation

```
<AsyncRoot level="INFO" includeLocation="false">
        <!-- <AppenderRef ref="CONSOLE" /> -->
        <AppenderRef ref="FILE" />
</AsyncRoot>
```

2. Change `INFO` to `DEBUG`:

```
<AsyncRoot level="DEBUG" includeLocation="false">
        <!-- <AppenderRef ref="CONSOLE" /> -->
        <AppenderRef ref="FILE" />
</AsyncRoot>
```

3. **Optional:** To see the adapter activity in the console, remove the comment tags that surround the `CONSOLE` line:

```
<AsyncRoot level="INFO" includeLocation="false">
        <AppenderRef ref="CONSOLE" />
        <AppenderRef ref="FILE" />
</AsyncRoot>
```

3. To log activity relating to the Google Chrome Enterprise Device Trust IdP Adapter:

   Choose from:

   - To log activity for the Google Chrome Enterprise Device Trust IdP Adapter as well as its HTTPS and component activity, add the following line:

```
<Logger name="com.pingidentity.adapters.google.chrome.enterprise.device.trust" level="DEBUG"/>
```

   - To log activity for the adapter's HTTPS activity and other components but not for the adapter itself, add the following line:

```
<Logger name="com.pingidentity.adapters.google.chrome.enterprise.device.trust.shade"
level="DEBUG"/>
```

   - To log activity for the Google Chrome Enterprise Device Trust IdP Adapter but not its HTTPS or component activity, add the following lines:

```
<Logger name="com.pingidentity.adapters.google.chrome.enterprise.device.trust" level="DEBUG"/>
<Logger name="com.pingidentity.adapters.google.chrome.enterprise.device.trust.shade"
level="INFO"/>
```

4. Save the file.

Release Notes

Copyright © 2025 Ping Identity Corporation

Changelog

The following is the change history for the Google Chrome Enterprise Integration Kit.

Google Chrome Enterprise Integration Kit 1.0 – December 2023

- Initial release

Known issues and limitations

The following are known issues or limitations for the Google Chrome Enterprise Integration Kit.

Known limitations

- The adapter does not implement support for the PingFederate Authentication API, as it is not feasible to externally generate verified access challenge response values. The challenge response is currently built into the Google Chrome engine and works with only redirects.

Download manifest

The following files are included in the Google Chrome Enterprise Integration Kit `.zip` archive.

- `Legal.pdf` – copyright and license information
- `dist/pingfederate/server/default` – contains the integration files
    - `deploy` – contains the Java libraries
        - `pf-google-chrome-enterprise-adapter-<version>.jar` – The Google Chrome Enterprise Device Trust IdP Adapter

Copyright © 2025 Ping Identity Corporation

Heroku Integration Guide

This guide describes how to configure PingFederate for browser-based single sign-on (SSO) to Heroku, initiated by the identity provider (IdP) or service provider (SP). The PingFederate administrative console uses an SP connection along with imported metadata to configure most of the settings needed for Heroku SAML SSO.

Obtain the Heroku SAML 2.0 Metadata XML

About this task

This guide uses a metadata XML file to assist in configuring many settings in the service provider (SP) connection. When asked during the configuration steps, import the `saml-metadata.xml` built in the following steps.

Steps

1. Copy the following sample metadata into a text editor.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
     <md:EntityDescriptor
      xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata"

      entityID="https://sso.heroku.com/saml/<ORGANIZATION_NAME>" cacheDuration="PT1440M"

      ID="nYtE3pu8fofN4a5Z_ST5F8jDObh">
       <md:SPSSODescriptor
      AuthnRequestsSigned="false"  WantAssertionsSigned="true"
      protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">

      <md:NameIDFormat>urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress</md:NameIDFormat>
       <md:AssertionConsumerService
      isDefault="true"

      Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"

      Location="https://sso.heroku.com/saml/<ORGANIZATION_NAME>/finalize"
        index="0" />
      </md:SPSSODescriptor>

    </md:EntityDescriptor>
```

2. Replace the instances of *<ORGANIZATION_NAME>* with the organization name for the Heroku account.

   For example, if the URL you use to access your Heroku team account is

   ```
   https://dashboard.heroku.com/orgs/myOrgName/settings
   ```

   then your *<ORGANIZATION_NAME>* is `myOrgName`.

3. Once you have updated the file, save `saml-metadata.xml`.

Copyright © 2025 Ping Identity Corporation

Configure Server Settings

Steps

1. Sign on to the PingFederate administrative console.
2. On the **System > Protocol Settings > Roles & Protocols** page, ensure the IdP role is enabled and SAML 2.0 is selected for that role.

   > **Tip**
   >
   > Click **Server Settings** from the main menu to locate this screen after initial installation.

3. Click **Next** to continue to configure roles and protocols, or click **Save** for an existing configuration.

Configure a Connection

About this task

Use the following procedure to configure a quick connection for single sign-on (SSO) to Heroku.

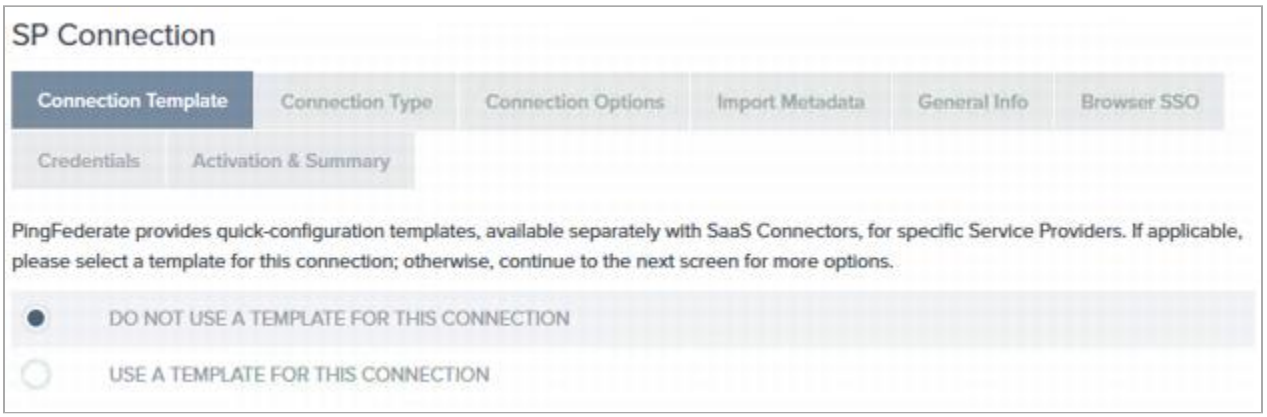
> **Tip**
>
> This procedure provides instructions for configuring minimum required connection settings. The instructions skip pages in which all necessary information is configured or defaults are used. You can add or change settings to suit your requirements.

Steps

1. If you have not already done so, use PingFederate to configure the identity provider (IdP) adapter you want to use.

   Learn more in Managing IdP adapters in the PingFederate documentation.

2. On the **Main Menu**, select **Create New** under SP Connections in the **IdP Configuration** section.
3. On the **Connection Template** page, select the **Do not use a template for this connection** and click **Next**.



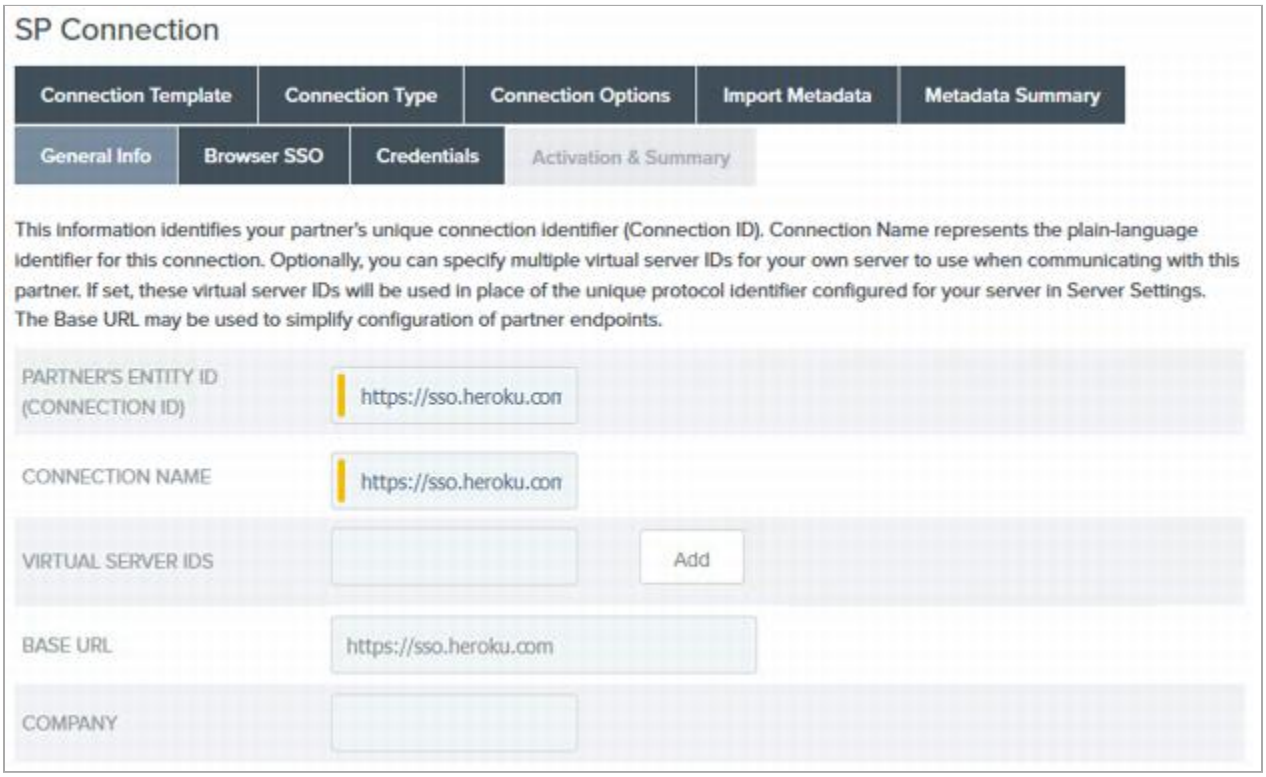
4. On the **Connection Type** page, ensure that the **Browser SSO** profile is selected and click **Next**.

Copyright © 2025 Ping Identity Corporation

5. On the **Connection Options** page, ensure **Browser SSO** is selected and click **Next**.
6. On the **Import Metadata** page, click **Choose file** and select the Heroku `saml-metadata.xml` file you created in Obtain the Heroku SAML 2.0 Metadata XML.

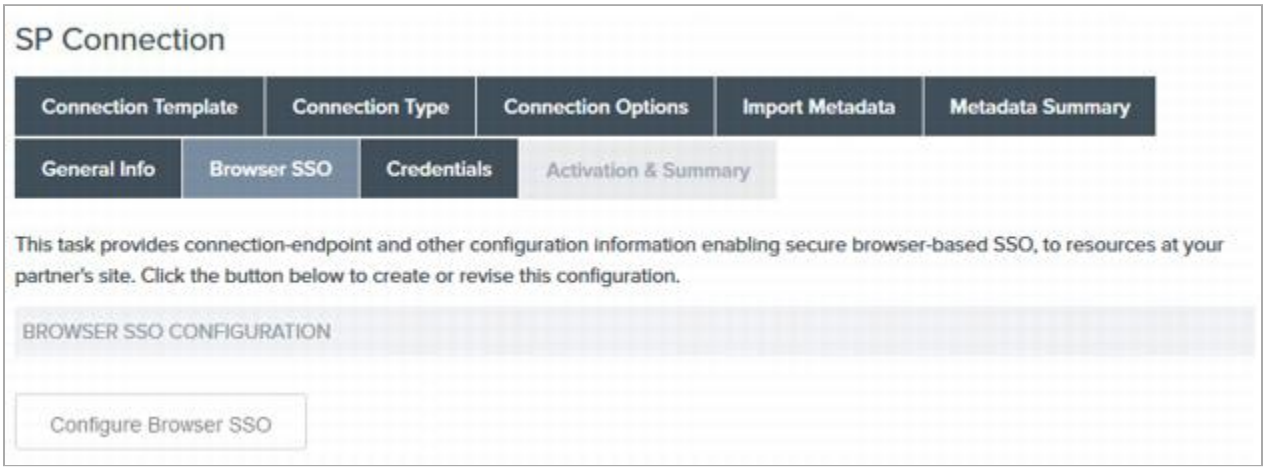


7. On the **Metadata Summary** page, click **Next**.
8. On the **General Info** page, ensure that the **Partner's Entity ID (Connection ID)**, **Connection Name**, and **Base URL** are accurate. Change details if required and click **Next**.

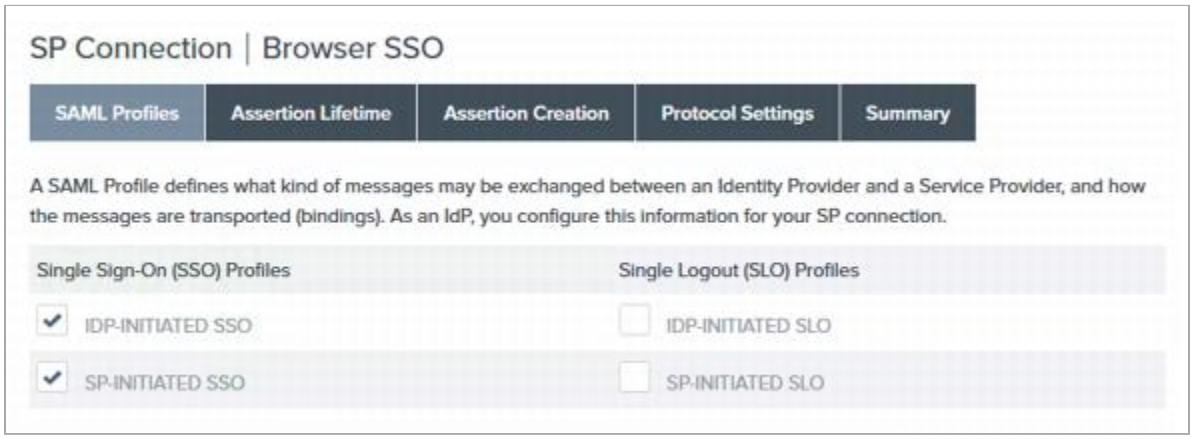


9. On the **Browser SSO** page, click **Configure Browser SSO**.

Copyright © 2025 Ping Identity Corporation



10. On the **SAML Profiles** page, ensure that the **IdP-Initiated SSO** and **SP-Initiated SSO** profiles are selected and click **Next**.



11. On the **Assertion Creation** page, click **Configure Assertion Creation**.



12. On the **Attribute Contract** page, ensure that the SAML_SUBJECT name format is set to:

Copyright © 2025 Ping Identity Corporation

```
following:urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress
```



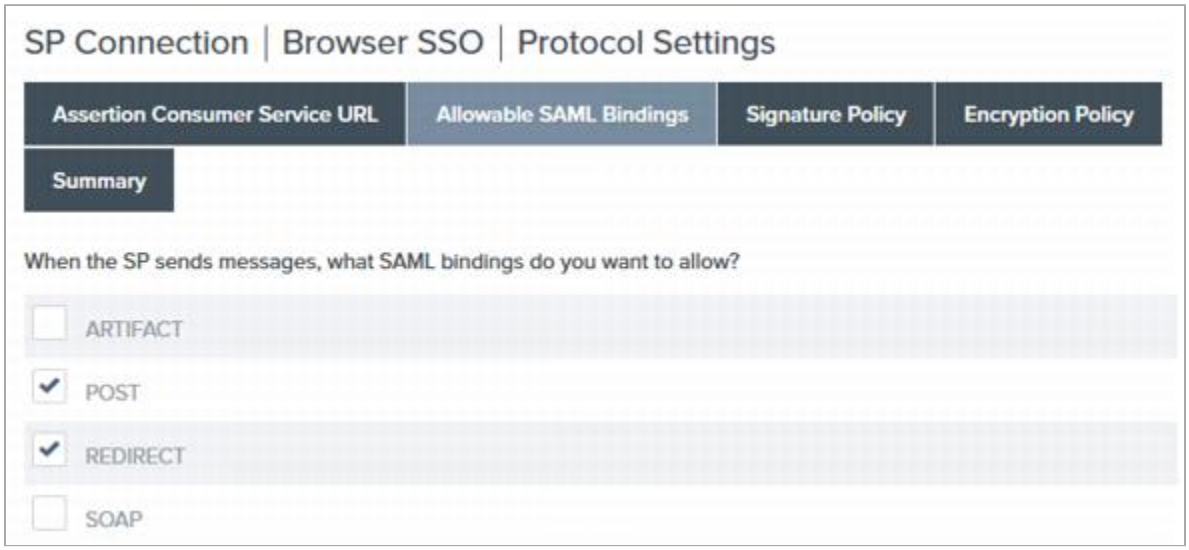
13. On the **Authentication Source Mapping** page, click **Map New Adapter Instance** and map the IdP Adapter Instance you defined earlier in this procedure. When you return to the Authentication Source Mapping page, click **Done**.

> **Note**
>
> This configuration is site-dependent and cannot be pre-configured. You can find detailed information and instructions in Managing authentication source mappings in the PingFederate documentation.



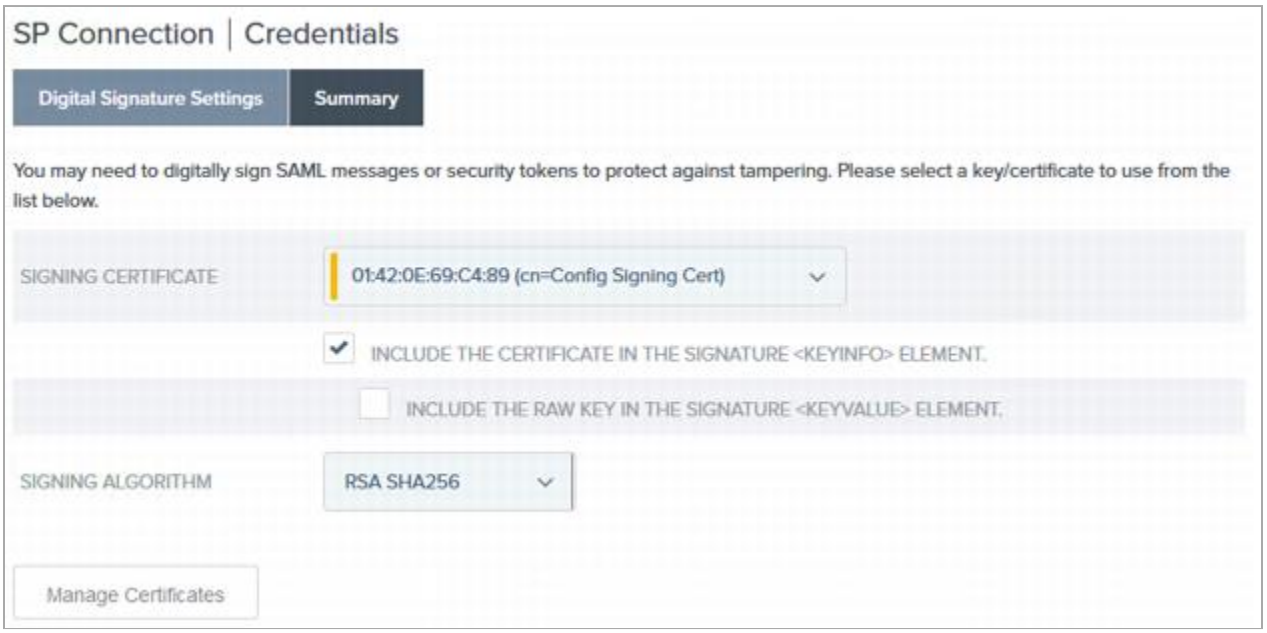
14. When you return to the **Assertion Creation** page, click **Next**
15. On the **Protocol Settings** page, click **Configure Protocol Settings**.
16. On the **Allowable SAML Bindings** page, ensure that only the **POST** and **Redirect** profiles are selected and click **Next**.

Copyright © 2025 Ping Identity Corporation



17. On the next page, ensure that the **Always sign the SAML Assertion** is selected and click **Next**.
18. On the **Browser SSO** page, click **Next** and on the **Credentials** page, click **Configure Credentials**.

    Learn more in Configuring digital signatures for service provider connections in the PingFederate documentation.
19. If you have not yet created or imported a signing certificate, click **Manage Certificates** and do so now. Learn more in Managing digital signing certificates and decryption keys in the PingFederate documentation.



20. Click **Next**.
21. On the **Summary** page, click **Done**.
22. On the **Credentials** page, click **Next**.
23. On the **Activation & Summary** page, **Activate** the SP Connection.
24. On the **Activation & Summary** page, click **Save**.

Copyright © 2025 Ping Identity Corporation

Configure Heroku for SSO

About this task

To complete single sign-on (SSO) setup for Heroku, configure these steps to enable SSO for users.

> **Note**
>
> Make sure you have the following information from PingFederate:
>
> - The SSO Application Endpoint, which can be found on the **Activation & Summary** page of the **SP Connection** for Heroku.
> - The exported certificate used to sign the SAML assertion that was configured in Configure a Connection.

Steps

1. Go to https://dashboard.heroku.com/orgs/<*ORGANIZATION_NAME*>/settings.
2. Sign on with your administrator credentials.
3. In the **Single Sign On** section, click **Add Metadata Manually**.



4. Enter the SSO Application endpoint into the **IdP Login Redirect URL** field.

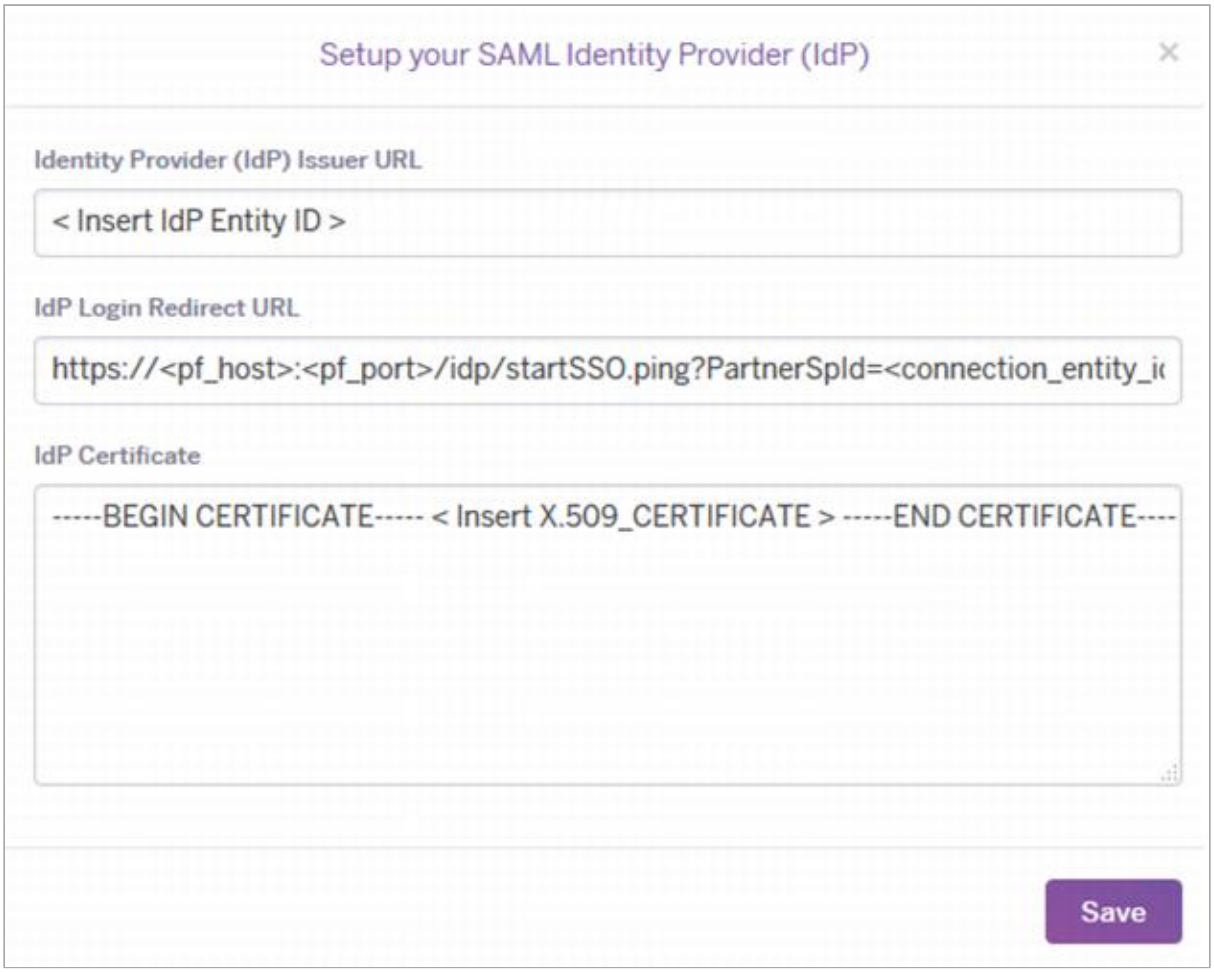
```
https://<pf_host>:<pf_port>/idp/startSSO.ping?PartnerSpId=<IdP_connection_entity_id>
```

> **Note**
>
> An email will be sent to new Heroku users instructing them on how to initiate SSO with the SSO Application endpoint.

5. Copy and paste the SAML 2.0 Entity ID and signing certificate into the **Identity Provider Issuer URL** and **Public Certificate** fields, respectively.

> **Tip**
>
> To override the SAML 2.0 Entity ID on the **Server Settings** page for your SP Connection, go to the **General Info** page and add a `Virtual Server ID`. This value sends as the SAML Issuer URL.

Copyright © 2025 Ping Identity Corporation



6. Click **Save**.

Copyright © 2025 Ping Identity Corporation

ID DataWeb Integration Kit

The ID DataWeb Integration Kit allows PingFederate to communicate with ID DataWeb for risk-based authentication using ThreatMetrix.

By sending a ThreatMetrix device profile and optional user attributes to ID DataWeb when a user signs on, PingFederate can get a security risk assessment for the sign-on event. Including the risk assessment in your PingFederate authentication policy allows you to adjust the user's authentication requirements dynamically each time they sign on.

Components

- Template and script files
  - When a user signs on through PingFederate, these files retrieve the JavaScript that collects a profile of the user's device. This allows you to use ThreatMetrix as a verification service in ID DataWeb.
- ID DataWeb IdP Adapter
  - When a user signs on through PingFederate, the adapter sends the device profile and user attributes to the ID DataWeb API, then retrieves a policy decision (`approve`, `obligation`, or `deny`) from ID DataWeb.

Intended audience

This document is intended for PingFederate administrators. Before starting, we recommend that you familiarize yourself with the following:

- The ID DataWeb documentation
- ThreatMetrix Overview in the ID DataWeb documentation
- Authentication policies in the PingFederate documentation

System requirements

- PingFederate 10.3 or later
- An ID DataWeb service administrator account

Overview of ID DataWeb

ID DataWeb evaluates the level of security risk for a user sign-on event based on user attributes and a device profile.

Device profile and user attributes

ID DataWeb uses a device profile to determine a "trust score" for each user sign-on event. The device profile is collected by a ThreatMetrix JavaScript script that runs during the sign-on flow.

You can opt to provide user attributes, such as name, address, and email, to ThreatMetrix or other verification services that are offered by ID DataWeb. You can populate these attributes from other sources in your PingFederate authentication policy. You can find a complete list of attributes that ID DataWeb can collect in Attribute List in the ID DataWeb documentation.

Copyright © 2025 Ping Identity Corporation

The ID DataWeb IdP Adapter is responsible for sending the device profile and user attributes to ID DataWeb as part of the sign-on flow.

Verification services

The ID DataWeb service supports a network of more than 70 identity verification services. ID DataWeb combines the results from all of these services into a unified trust score.

As an ID DataWeb administrator, you can choose the verification services that you want to involve in user sign on events as shown in Managing Verification Services in the ID DataWeb documentation.

The ID DataWeb Integration Kit is designed to use the ThreatMetrix verification service, which allows you to build risk-based authentication into your PingFederate authentication policy. Learn more in ThreatMetrix in the ID DataWeb documentation.

Policy decisions

ID DataWeb processes the sign-on event information through the rules and identity verification services that you configure in the ID DataWeb dashboard. It then matches the resulting trust score to one of three policy decisions: `approve`, `obligation`, or `deny`.

The ID DataWeb API provides the policy decision in a response to PingFederate. By including the ID DataWeb policy decision in your PingFederate authentication policy, you decide how each of the `approve`, `obligation`, and `deny` results affects a user's ability to sign on in your environment. For example, you can configure the `obligation` result to require a second authentication factor.

Attributes in the response from ID DataWeb

The response from ID DataWeb also contains user attributes and sign-on event data from the various ID DataWeb verification services.

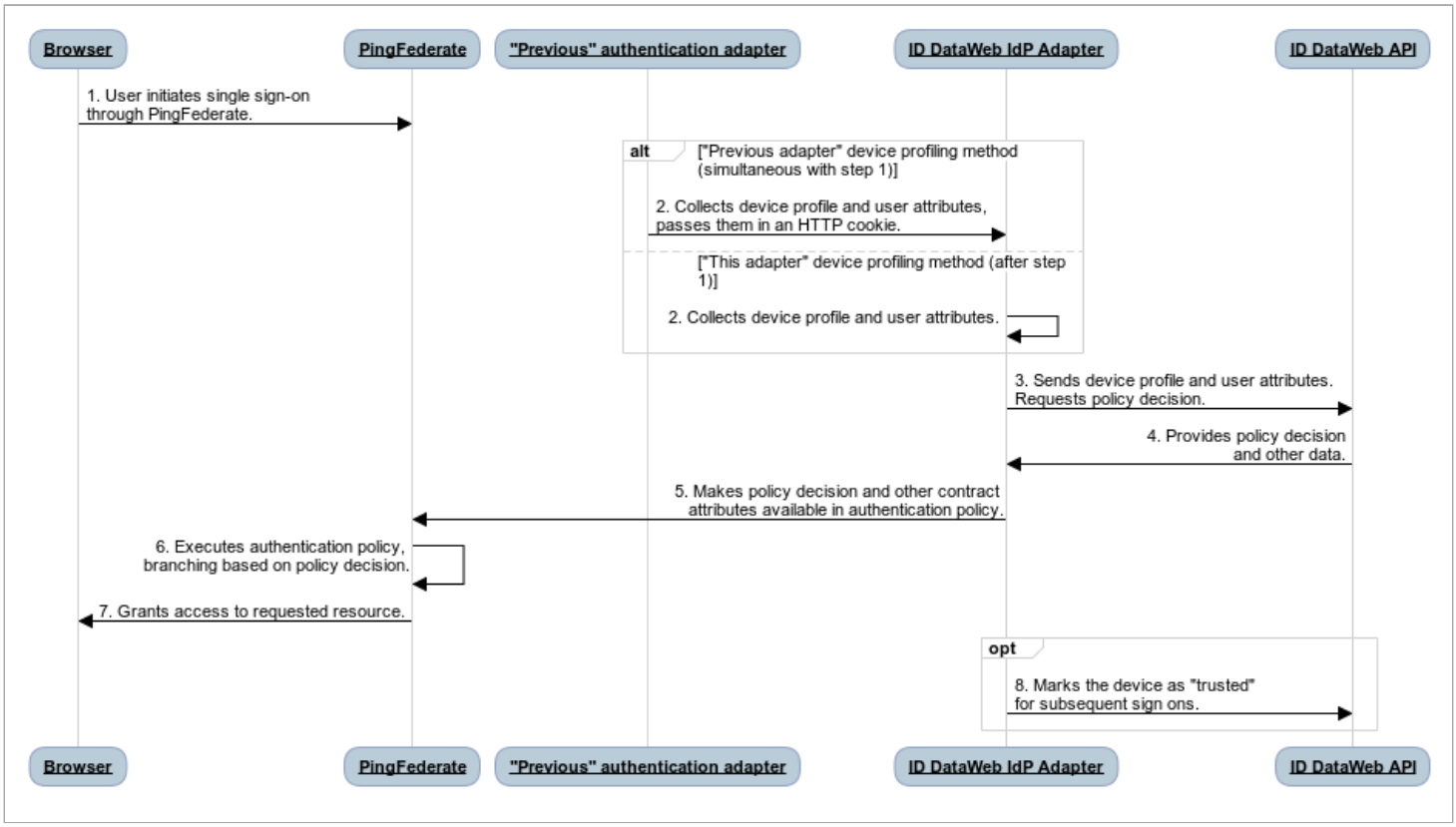
In your ID DataWeb IdP Adapter instance configuration, you can capture attributes from the response. This makes them available in other adapters and contracts in the PingFederate authentication policy.

Overview of the SSO flow

With the ID DataWeb Integration Kit, PingFederate includes the ID DataWeb API in the sign-on flow.

The following figure shows how the ID DataWeb API integrates into the sign-on process:

Copyright © 2025 Ping Identity Corporation



Description

1. A user initiates the sign-on process by requesting access to a protected resource.
2. Depending on the device profiling method, the ID DataWeb IdP Adapter or a previous authentication adapter retrieves the latest JavaScript from ThreatMetrix, which collects the device profile and sends it back to ThreatMetrix. The adapter can also collect user attributes. For the "previous adapter" method, this takes place at the same time as step 1.
3. The ID DataWeb IdP Adapter sends the device profile identifier and any user attributes to the ID DataWeb API and requests the policy decision (`approve` , `obligation` , or `deny`).
4. The ID DataWeb API returns a JSON payload with the policy decision and other attributes to the ID DataWeb IdP Adapter.
5. The ID DataWeb IdP Adapter makes the policy decision and contract attributes available in the PingFederate authentication policy.
6. PingFederate executes the authentication policy, which branches based on the policy decision provided by the ID DataWeb IdP Adapter.
7. PingFederate returns the resource that the user requested.
8. If **Update Device Trust** is enabled in the adapter instance configuration, the ID DataWeb IdP Adapter notifies ID DataWeb that the device is trustworthy. This gives the device a better trust score for subsequent sign on attempts.

   If **Update Device Trust Using User Consent** is enabled in the adapter configuration and the user selects **This is my device** in the HTML form adapter when authenticating, the ID DataWeb IdP Adapter notifies ID DataWeb that the device is trustworthy.

Copyright © 2025 Ping Identity Corporation

Device profiling methods

To include ThreatMetrix when producing a policy decision for a sign-on event, ID DataWeb requires information about the device as well as the user. There are different methods for collecting the device profile depending on your preferences and whether users are authenticating directly through PingFederate or through the PingFederate authentication API.

Direct authentication mode - Captured by the ID DataWeb IdP Adapter

When the ID DataWeb IdP Adapter is triggered in the sign-on flow, it inserts a page that runs the device profiling script.

This method does not require modifications to any other pages, but adds a wait time to the sign-on process. The length of the wait depends on your environment.

Sequence:

1. The user arrives at the first-factor sign-on page and enters their credentials.
2. The ID DataWeb IdP Adapter is triggered by the PingFederate authentication policy. It presents a spinner animation page that runs the device profiling script. The script sends the device profile to the ID DataWeb API with a unique session ID.
3. The ID DataWeb IdP Adapter requests a policy decision by sending the session ID and any user attributes to the ID DataWeb API.

Direct authentication mode - Captured by a previous adapter

To reduce perceived wait times for the user, you can run the device profiling script on a page that already requires user interaction.

Sequence:

1. The user arrives at a page presented by a PingFederate adapter, such as the HTML Form Adapter.
2. While the user interacts with the page (for example, entering their username and password), the device profiling script sends the device profile to the ID DataWeb API with a unique session ID. The script also stores the session ID in an HTTP cookie.
3. The ID DataWeb IdP Adapter is triggered by the PingFederate authentication policy. It gets the session ID from the HTTP cookie.
4. The ID DataWeb IdP Adapter requests a policy decision by sending the session ID and any user attributes to the ID DataWeb API.

You can integrate the device profiling script into any page that meets the following criteria:

- The page appears before the ID DataWeb IdP Adapter in your PingFederate authentication policy.
- The page is hosted in the same domain as your PingFederate server. This is required to accommodate the HTTP cookies. You might be able to work around this requirement by consolidating your domains with a reverse proxy server.

Copyright © 2025 Ping Identity Corporation

Authentication API mode - Captured by the ID DataWeb IdP Adapter

If you have a web application that uses the PingFederate authentication API, you can add the device profiling script to your existing sign-on page.

Sequence:

1. The user arrives at the sign-on page and enters their credentials.
2. The ID DataWeb IdP Adapter is triggered by the PingFederate authentication policy.
3. The web application gets a device profiling URL from the authentication API. The script creates the device profile and associated it with the session ID in the URL.
4. The web application tells PingFederate to continue the authentication flow.
5. The ID DataWeb IdP Adapter requests a policy decision by sending the session ID and any user attributes to the ID DataWeb API.

This method is recommended if you have a web application and cannot accommodate HTTP cookies.

Authentication API mode - Captured by your web application first-factor sign-on page

If your web application can accommodate HTTP cookies, you can reduce perceived wait times by running the device profiling script while the user interacts with your sign-on page.

Sequence:

1. The user arrives at a first-factor sign-on page presented by your web application.
2. While the user interacts with the page (for example, entering their username and password), the device profiling script sends the device profile to the ID DataWeb API with a unique session ID. The script also stores the session ID in an HTTP cookie.
3. The ID DataWeb IdP Adapter gets the session ID from the HTTP cookie.
4. The ID DataWeb IdP Adapter sends the session ID and any user attributes to the ID DataWeb API and requests the policy decision.

You can integrate the device profiling script into your web application if it meets the following criteria:

- The user sees your sign-on page before the ID DataWeb IdP Adapter is triggered in your PingFederate authentication policy.
- The page is hosted in the same domain as your PingFederate server. This is required to accommodate the HTTP cookies. You might be able to work around this requirement by consolidating your domains with a reverse proxy server.

Setup

To use the ID DataWeb Integration Kit with PingFederate, configure ID DataWeb, deploy the integration files, and then set up the integration on your PingFederate server.

Copyright © 2025 Ping Identity Corporation

Steps

1. In the ID DataWeb admin console, add the verification services that you want to use, such as ThreatMetrix.

   You can find instructions in Managing Verification Services in the ID DataWeb documentation.

2. In the ID DataWeb admin console, in the **Relying Parties** section, add PingFederate as a relying party.
3. Complete the steps in Deploying the integration files.
4. (Optional) Complete the steps in Adding device profiling to an authentication page.
5. Complete the steps in Configuring an adapter instance.
6. Complete the steps in Adding ID DataWeb policy decisions to your authentication policy.

Deploying the integration files

To get started with the integration, deploy the ID DataWeb Integration Kit files to your PingFederate directory.

About this task

> **Note**
>
> If you operate PingFederate in a cluster, the following steps refer to the console node.

Steps

1. Download the ID DataWeb Integration Kit `.zip` archive from the Ping Identity Integration Directory.
2. Stop PingFederate.
3. If you're upgrading an existing deployment, back up your customizations and delete earlier versions of the integration files:
   1. Back up any ID DataWeb Integration Kit files that you customized in the `<pf_install>/pingfederate/server/default/conf/` directory.
   2. Delete the `pf-id-dataweb-adapter-<version>.jar` file from your `<pf_install>/pingfederate/server/default/deploy` directory.
4. Extract the `.zip` archive and merge the contents of the `dist` directory with your `<pf_install>/pingfederate` directory.
5. If there is more than one version of the `pf-authn-api-sdk-<version>.jar` file in your `<pf_install>/pingfederate/server/default/lib` directory, delete all but the latest version of the file.
6. If you backed up any customized files, modify the new files with your customizations.
7. Start PingFederate.
8. If you operate PingFederate in a cluster, repeat steps 2 - 8 for each engine node.

Integrating device profiling

Depending on your authentication mode and preferences, complete one of the following steps to capture the device profile.

Copyright © 2025 Ping Identity Corporation

About this task

You can find detailed descriptions of each method in Device profiling methods.

Steps

- Do one of the following:

Method	Steps
Direct authentication mode - Captured by the ID DataWeb IdP Adapter	No extra steps are necessary. Continue to Configuring an adapter instance.
Direct authentication mode - Captured by a previous adapter, such as the HTML Form Adapter	Modify the adapter template by completing the steps in Adding device profiling to an authentication page.
Authentication API mode - Captured by the ID DataWeb IdP Adapter	Modify your web application by completing the steps in Adding device profiling to a web application.
Authentication API mode - Captured by your web application first-factor sign-on page	Modify your web application by completing the steps in Adding device profiling to an authentication page.

Adding device profiling to an authentication page

Instead of using the ID DataWeb IdP Adapter to collect the device profile, you can capture the device profile using an existing sign-on page. This can reduce perceived wait times for the user.

About this task

You can adapt these instructions to add device profiling to any page, such as the HTML Form Adapter or your web application. The page must meet the criteria listed in Device profiling methods.

Steps

1. Embed the JavaScript files in the page.
    1. If you're modifying an external web application, copy the `id_dataweb_device_profiling.js` file from the integration `.zip` archive to a location that your page can access.
    2. Add the following to the sign-on page and adjust the path to the script file:

```
<script type="text/javascript" src="id_dataweb_device_profiling.js"></script>
```

> **Tip**
>
> If you're modifying a PingFederate template, the script path is `assets/scripts/id_dataweb_device_profiling.js`.

    3. Save the file.
2. (Optional) In the script file, customize the name prefix for the device profile cookie to suit your environment.

Copyright © 2025 Ping Identity Corporation

```
var cookieName = "idwUUID";
```

3. When you complete the steps in Configuring an adapter instance:
    1. Set the **Device profiling method** to **Captured by a previous adapter**.
    2. Update the **Cookie name** field if you customized it in step 2.

Adding device profiling to a web application

If you have a web application that uses the PingFederate authentication API and cannot accommodate HTTP cookies, modify your web application to run the device profiling script.

About this task

Learn more in PingFederate authentication API in the PingFederate documentation.

Steps

1. Add the following code to the sign-on page:

    ```
    <script type="text/javascript"
    src="<device profiling URL>"></script>
    ```

    For each sign-on event, insert the device profiling URL from PingFederate.

2. Configure your application to complete the following actions in sequence:
    1. Get the device profiling URL from PingFederate. This is an attribute of the `DEVICE_PROFILE_REQUIRED` state.
    2. Run the device profiling script by including it in the page.
    3. POST `continueAuthentication` to the authentication API.
3. When you complete the steps in Configuring an adapter instance, set the **Device profiling method** to **Captured by this adapter**.

Configuring an adapter instance

Configure the ID DataWeb IdP Adapter to determine how PingFederate communicates with the ID DataWeb API.

Steps

1. In the PingFederate administrative console, create a new IdP adapter instance:
    1. Go to **Authentication > Integration > IdP Adapters** and click **Create New Instance**.
2. On the **Type** tab, set the basic adapter instance attributes.
    1. In the **Instance Name** field, enter a name for the adapter instance.

Copyright © 2025 Ping Identity Corporation

2. In the **Instance ID** field, enter a unique identifier for the adapter instance.
3. In the **Type** list, select **ID DataWeb IdP Adapter**. Click **Next**.

3. (Optional) On the **IdP Adapter** tab, in the **Additional Verification Attributes** section, configure additional attributes to send to ID DataWeb with the user data:
    1. Click **Add a new row to 'Additional Verification Attributes'**.
    2. In the **Incoming Attribute Name** field, enter the name of an attribute from any authentication source that appears earlier in your PingFederate authentication policy than the ID DataWeb IdP Adapter.
    3. In the **ID DataWeb User Attribute** list, select the ID DataWeb attribute that you want to populate.

    > **Note**
    >
    > The ID DataWeb IdP Adapter allows you to pass custom attributes to ID DataWeb. The name and value of the attribute are determined by the **Local Attribute** column.
    > For example, if you enter `officeName` in the **Local Attribute** column, ID DataWeb receives an attribute with the following name and value format: `{…officeName: headquarters…}`.

    4. In the **Action** column, click **Update**.
    5. To add more attributes, repeat steps a - d.
4. (Optional) On the **IdP Adapter** tab, in the **ID DataWeb Response Mappings (optional)** section, map attributes from the ID DataWeb response to the attribute contract:
    1. Click **Add a new row to 'ID DataWeb Response Mappings (optional)'**.
    2. In the **Local Attribute** field, enter a name of your choosing for an attribute.
    3. In the **ID DataWeb API Attribute Mapping** field, enter the JSON Pointer syntax for the value of the matching ID DataWeb attributes as shown in JSON Pointer syntax reference.
    4. In the **Action** column, click **Update**.
    5. To add more attributes, repeat steps a-d.
5. On the **IdP Adapter** tab, configure the adapter instance by referring to ID DataWeb IdP Adapter settings reference. Click **Next**.
6. On the **Actions** tab, test your connection to the ID DataWeb API. Resolve any issues that are reported, and then click **Next**.
7. On the **Extended Contract** tab, add any attributes that you included in the **ID DataWeb Response Mappings (optional)** section of the **IdP Adapter** tab.
8. On the **Adapter Attributes** tab, set pseudonym and masking options as shown in Set pseudonym and masking options in the PingFederate documentation. Click **Next**.
9. On the **Adapter Contract Mapping** tab, configure the contract fulfillment details for the adapter as shown in Define the IdP adapter contract in the PingFederate documentation. Click **Next**.
10. On the **Summary** tab, check your configuration and then click **Save**.

Copyright © 2025 Ping Identity Corporation

JSON Pointer syntax reference

JavaScript Object Notation (JSON) Pointer defines a syntax for identifying a specific value within a JSON payload. Using the sample payload and JSON Pointer examples below, identify the attributes that you want to use to populate your attribute contract.

You can find a complete technical description of JSON Pointer syntax in JavaScript Object Notation (JSON) Pointer on ietf.org.

You can find more information about the ID DataWeb attributes and payload structure in the Attribute List and Attribute Dictionary in the ID DataWeb documentation.

Copyright © 2025 Ping Identity Corporation

Example JSON payload

Copyright © 2025 Ping Identity Corporation

```
{
   "endpoint$transaction$serialVersionUID":"3437946115680492319",
   "endpoint$transaction$errorCode":"",
   "endpoint$transaction$errorDescription":"",
   "endpoint$transaction$mbun":"5d87fd39-af76-9732-3435-7fa1d0bc8148",
   "endpoint$transaction$transaction_id":"0375fc22-3321-4108-b0b9-69660ef19bf4",
   "endpoint$transaction$forwardApiKey":"e16774cfb79941b9",
   "endpoint$transaction$policyObligation":"true",
   "endpoint$transaction$policyDecision":"obligation",
   "userAttributes$APSessionID$apSessionId":"875efc0d-f6e3-4ca7-9c09-4b0c79446359",
   "userAttributes$FullName$fname":"Janet",
   "userAttributes$FullName$lname":"Smith",
   "userAttributes$FullName$mname":"middleName",
   "userAttributes$FullName$suffix":"null",
   "userAttributes$InternationalAddress$country":"US",
   "userAttributes$InternationalAddress$sublocality":"null",
   "userAttributes$InternationalAddress$locality":"Denver",
   "userAttributes$InternationalAddress$subpremise":"null",
   "userAttributes$InternationalAddress$sublocality_level_2":"null",
   "userAttributes$InternationalAddress$route":"Main Street",
   "userAttributes$InternationalAddress$administrative_area_level_2":"null",
   "userAttributes$InternationalAddress$premise":"null",
   "userAttributes$InternationalAddress$sublocality_level_5":"null",
   "userAttributes$InternationalAddress$administrative_area_level_3":"null",
   "userAttributes$InternationalAddress$sublocality_level_4":"null",
   "userAttributes$InternationalAddress$sublocality_level_3":"null",
   "userAttributes$InternationalAddress$administrative_area_level_1":"VA",
   "userAttributes$InternationalAddress$street_number":"18",
   "userAttributes$InternationalAddress$neighborhood":"null",
   "userAttributes$InternationalAddress$postal_code":"23225",
   "userAttributes$CustomAttribute1$name":"customAttribute1Val",
   "userAttributes$CustomAttribute1$value":"myCustomAttribute1Value",
   "acquiredAttributes$EquifaxScore$equifaxScore":"7",
   "acquiredAttributes$SmartIDBrowserstringPersonaAgeMonths$smartIDBrowserStringPersonaAge":"0",
   "acquiredAttributes$NEATPersonaAgeMonths$NEATPersonaAge":"0",
   "acquiredAttributes$DigitalID$digitalID":"0e12aa408fb114c3b0c7d780c3a69c91",
   "acquiredAttributes$DigitalIDConfidence$digitalIDConfidence":"6022",
   "acquiredAttributes$FuzzyDeviceIDConfidence$fuzzyDeviceIDConfidence":"100.00",
   "acquiredAttributes$TrueIP$trueIP":"127.0.0.1",
   "acquiredAttributes$TrueIPGeoCountry$trueIPGeoCountry":"CA",
   "acquiredAttributes$Platform$platform":"browser_computer",
   "acquiredAttributes$FuzzyDeviceID$fuzzyDeviceID":"7d9eeb9464784039bc427e97b3184da2",
   "acquiredAttributes$ExactIDIPPersonaAgeMonths$exactIDIPPersonaAge":"0",
   "acquiredAttributes$TMXScore$tmxScore":"1",
   "acquiredAttributes$RawAssertionScore$rawAssertionScore":"4",
   "acquiredAttributes$AssertionScore$assertionScore":"80",
   "acquiredAttributes$IDWScore$idwScore":"85",
   "userAssertionList$Equifax Identity Fraud Service$link.fullName_address":"pass",
   "userAssertionList$Equifax Identity Fraud Service$link.lastName_address":"pass",
   "userAssertionList$Threatmetrix Session Query Low Location Accuracy$blacklist.ofacIP":"pass",
   "userAssertionList$Threatmetrix Session Query Low Location Accuracy$blacklist.device":"pass",
   "userAssertionList$Threatmetrix Session Query Low Location Accuracy$detect.browserAnomaly":"pass",
   "userAssertionList$Threatmetrix Session Query Low Location Accuracy$test.lte3ProxyToday":"pass",
   "userAssertionList$Threatmetrix Session Query Low Location Accuracy$test.trustedDevice":"fail",
   "userAssertionList$Threatmetrix Session Query Low Location Accuracy$test.lte3CredentialsDevice7d":"pass",
   "userAssertionList$Threatmetrix Session Query Low Location Accuracy$link.timeZone_TrueGeo":"pass",
   "userAssertionList$Threatmetrix Session Query Low Location Accuracy$test.credentialLTE500mi1hr":"pass",
   "userAssertionList$Threatmetrix Session Query Low Location Accuracy$test.exactIDAgeGTE7d":"pass",
   "userAssertionList$Threatmetrix Session Query Low Location Accuracy$detect.vpn":"pass",
   "userAssertionList$Threatmetrix Session Query Low Location Accuracy$test.trueIPLTE500miInputIP":"pass",
```


Copyright © 2025 Ping Identity Corporation

```
    "userAssertionList$Threatmetrix Session Query Low Location Accuracy$detect.proxyAnonymous":"pass",
    "userAssertionList$Threatmetrix Session Query Low Location Accuracy$blacklist.ip":"pass",
    "userAssertionList$Threatmetrix Session Query Low Location Accuracy$test.expectedLanguage":"pass",
    "userAssertionList$Threatmetrix Session Query Low Location Accuracy$link.proxyOrg_TrueOrg":"pass",
    "userAssertionList$Threatmetrix Session Query Low Location Accuracy$detect.malware":"pass",
    "userAssertionList$Threatmetrix Session Query Low Location Accuracy$detect.aggregator":"pass",
    "userAssertionList$Threatmetrix Session Query Low Location Accuracy$test.smartIDAgeGTE7d":"pass",
    "userAssertionList$Threatmetrix Session Query Low Location Accuracy$detect.proxyOpenTransparent":"pass",
    "userAssertionList$Threatmetrix Session Query Low Location Accuracy$detect.unusualActivity":"pass",
    "userAssertionList$Threatmetrix Session Query Low Location Accuracy$detect.proxyHidden":"pass",
    "userAssertionList$Threatmetrix Session Query Low Location Accuracy$link.proxyISP_TrueISP":"pass",
    "userAssertionList$Threatmetrix Session Query Low Location Accuracy$detect.tor":"pass",
    "userAssertionList$Threatmetrix Session Query Low Location Accuracy$detect.torNode":"pass",
    "userAssertionList$Threatmetrix Session Query Low Location Accuracy$link.proxyGeo_TrueGeo":"pass"
}
```


Table 1. JSON Pointer syntax

Description	JSON Pointer	Resulting value
Result of the risk assessment	`/endpoint$transaction$policyDecision`	`obligation`
The user's first name	`/userAttributes$FullName$fname`	`Janet`

ID DataWeb IdP Adapter settings reference

Field descriptions for the ID DataWeb IdP Adapter configuration screen.

Standard fields

Field	Description
Client ID	The client ID provided to you by ID DataWeb.
Client Secret	The client secret provided to you by ID DataWeb.
ID DataWeb API Base URL	The URL of the ID DataWeb API. • Production: `https://api.iddataweb.com/` • Pre-production: `https://api.preprod.iddataweb.com/`
Update Device Trust	When enabled, users can receive a boost to their trust score for subsequent sign on attempts with a given device. When a user signs on successfully after being challenged with an "obligation" result from ID DataWeb, PingFederate contacts the ID DataWeb API to mark the device as "trusted." You can configure the amount and duration of the trust score boost in the ID DataWeb admin console. The boost affects the device, not the user account. This checkbox is cleared by default.

Copyright © 2025 Ping Identity Corporation

Field	Description
Update Device Trust Using User Consent	When the user selects **This is my device** in a previous adapter, such as the HTML Form Adapter, PingFederate contacts the ID DataWeb API after the user signs on to mark the device as "trusted" for subsequent sign on attempts. This setting works in conjunction with the **Update Device Trust** setting. If you do not select **Update Device Trust**, **Update Device Trust Using User Consent** will not mark the device as trusted. This checkbox is cleared by default.
Update Device Trust API Key	The API key that PingFederate uses to communicate with the ID DataWeb API when marking a device as "trusted". Applies only when **Update Device Trust** is enabled. This field is blank by default.
Device Profiling Method	Determines how the adapter handles session IDs and device profiling: ***Captured by this adapter*** The ID DataWeb IdP Adapter creates a session ID. In authentication API mode, it provides the device profiling URL (including session ID) to the external web application. In direct authentication mode, it runs the device profiling script. ***Captured by a previous adapter*** The ID DataWeb IdP Adapter looks for an existing session ID in an HTTP cookie. Learn more in Device profiling methods. **Note** If you completed the steps in Adding device profiling to an authentication page, select **Captured by a previous adapter**. Otherwise, select **Captured by this adapter**. The default value is **Captured by this adapter**.

Advanced fields

Field	Description
Device Profiling Script URL	The URL of the ID DataWeb script that collects the device profile during sign on. Applies only when **Device Profiling Method** is set to **Captured by this adapter**. Applies only with the "Direct authentication mode - Captured by this adapter" device profiling method. The default value is: `https://content.maxconnector.com/fp/tags.js?org_id=716kkpe1&api_key=bvrbl1ev61nw7zq7&pageid=verify&session_id=${sessionId}`

Copyright © 2025 Ping Identity Corporation

Field	Description
Device Profiling Timeout	The amount of time in milliseconds that PingFederate waits for the device profiling script to collect device details. Applies only when **Device Profiling Method** is set to **Captured by this adapter**. The minimum value is `3000`. The default value is `5000`.
Cookie Name	The name of the cookie that contains the identifier for the ThreatMetrix device profile. Applies only when **Device Profiling Method** is set to **Captured by a previous adapter**. If you customized the name for the cookie in the optional Adding device profiling to an authentication page steps, enter the same name in this field. The default value is `idwUUID`.
Failure Mode	When ID DataWeb is unavailable or an error occurs, this setting determines whether the user's sign-on attempt should fail or continue with a pre-determined policy decision. For cases where the ID DataWeb API is unavailable or returns an error, we recommend that you allow users to continue to sign on by satisfying stricter authentication requirements. You can do this in your adapter configuration by setting the **Failure mode** to return the `obligation` result. Alternately, you can do this in your authentication policy by setting the **Fail** outcome of the ID DataWeb IdP Adapter instance as shown in Adding ID DataWeb policy decisions to your authentication policy.
Fallback Policy Decision Value	The risk result (for example, `obligation`, `deny`, or `unknown`) to use in the authentication policy when ID DataWeb is unavailable or an error occurs, and **Failure Mode** is set to **Continue with fallback risk result**. The default value is `deny`.
Token API Endpoint	The ID DataWeb endpoint that issues access tokens. The default value is `/v1/token`.
Verify API Endpoint	The ID DataWeb endpoint that performs one-time user verification. The default value is `/v1/flat/slverify`.
Trust Device API Endpoint	The ID DataWeb endpoint that PingFederate contacts after a user successfully signs on. Applies only when **Update Device Trust** is enabled. The default value is `/v1/trustdevice`.
API Request Timeout	The amount of time in milliseconds that PingFederate waits for the ID DataWeb API to respond to requests. A value of `0` disables the timeout. **Note** Some ID DataWeb verification services have longer response times than others. Test your specific configuration and adjust this value based on the range of response times that you receive. The default value is `2000`.

Copyright © 2025 Ping Identity Corporation

Field	Description
Connection Timeout	The amount of time in milliseconds that PingFederate allows to establish a connection with the ID DataWeb API. A value of `0` disables the timeout. The default value is `2000`.
Proxy Settings	Defines proxy settings for outbound HTTP requests. The default value is **System Defaults**.
Custom Proxy Host	The proxy server host name to use when **Proxy Settings** is set to **Custom**. This field is blank by default.
Custom Proxy Port	The proxy server port to use when **Proxy Settings** is set to **Custom**. This field is blank by default.
Verify HTTPS Hostname	When a connection is established with the ID DataWeb API, PingFederate matches the target host name against the names stored inside the server's X.509 certificate. This security measure ensures that PingFederate is connecting to the correct server. This checkbox is selected by default.

Adding ID DataWeb policy decisions to your authentication policy

By modifying your PingFederate authentication policy to include the policy decision ("allow", "obligation", and "deny") from the ID DataWeb API, you can change authentication requirements dynamically based on security risk level.

About this task

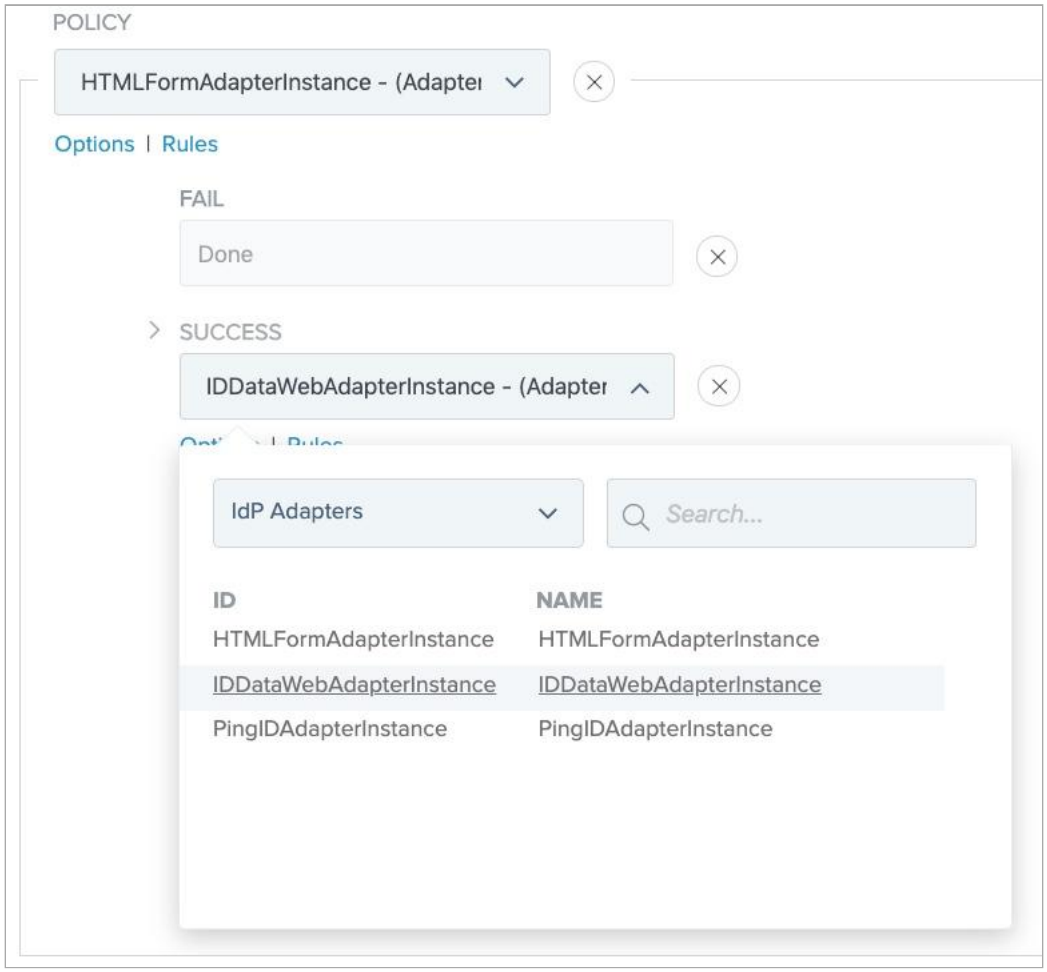
These steps are designed to help you add to an existing authentication policy. You can find general information about configuring authentication policies in Authentication policies in the PingFederate documentation.

Steps

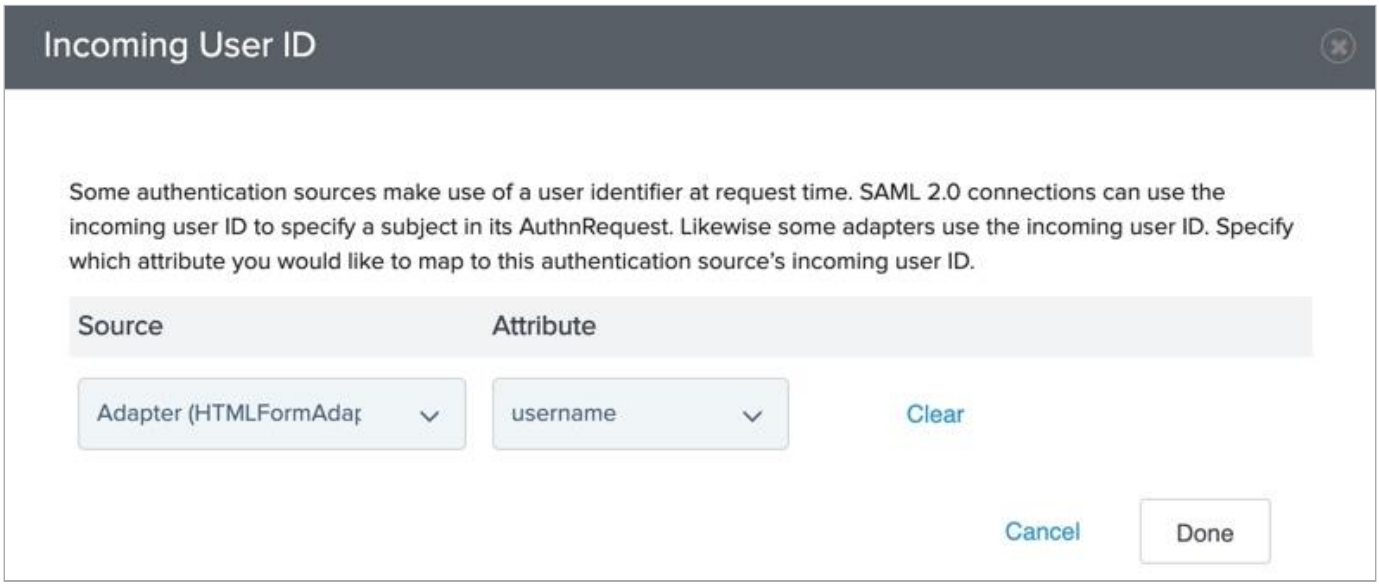
1. In the PingFederate administrative console, go to **Authentication > Policies > Policies**.
2. Select the **IdP Authentication Policies** checkbox.
3. Open an existing authentication policy, or click **Add Policy**.

   Learn more in Defining authentication policies in the PingFederate documentation.
4. In the **Policy** area, in the **Select** list, select an ID DataWeb IdP Adapter instance.

Copyright © 2025 Ping Identity Corporation



5. Map the user ID into the ID DataWeb IdP Adapter instance:

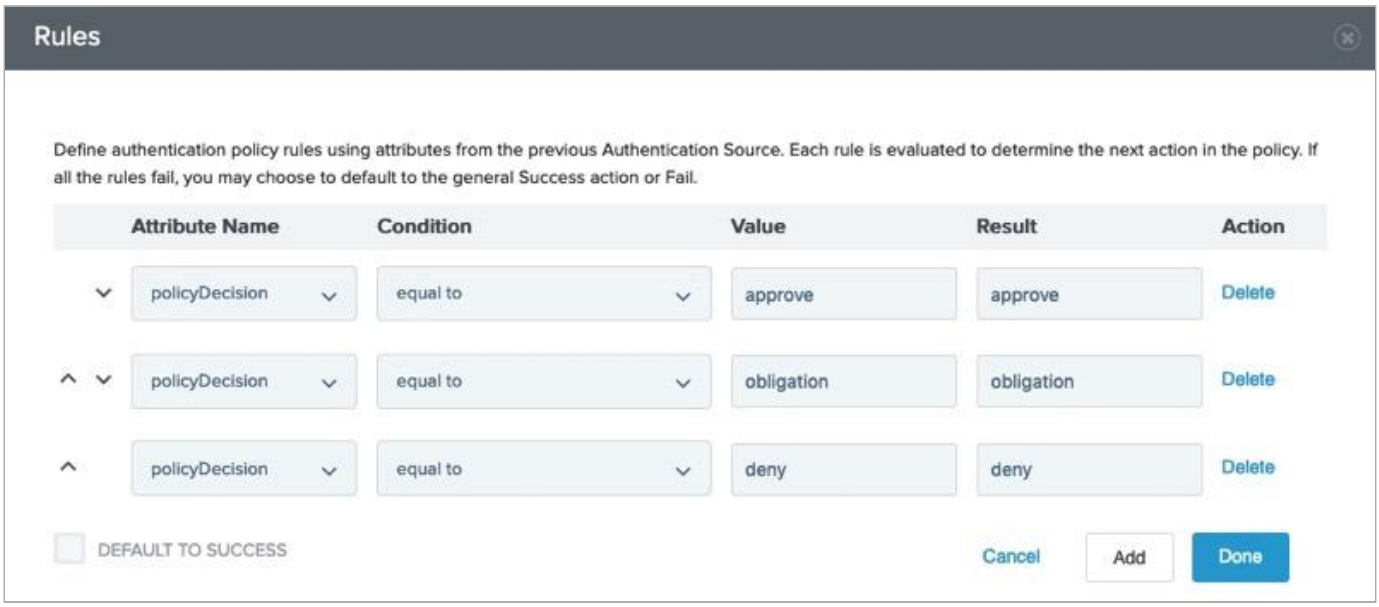


1. Under the ID DataWeb IdP Adapter instance, click **Options**.
2. On the **Options** dialog, in the **Source** list, select a previous authentication source that collects the user ID.

Copyright © 2025 Ping Identity Corporation

3. In the **Attribute** list, select the user ID. Click **Done**.

6. Define policy paths based on risk results:

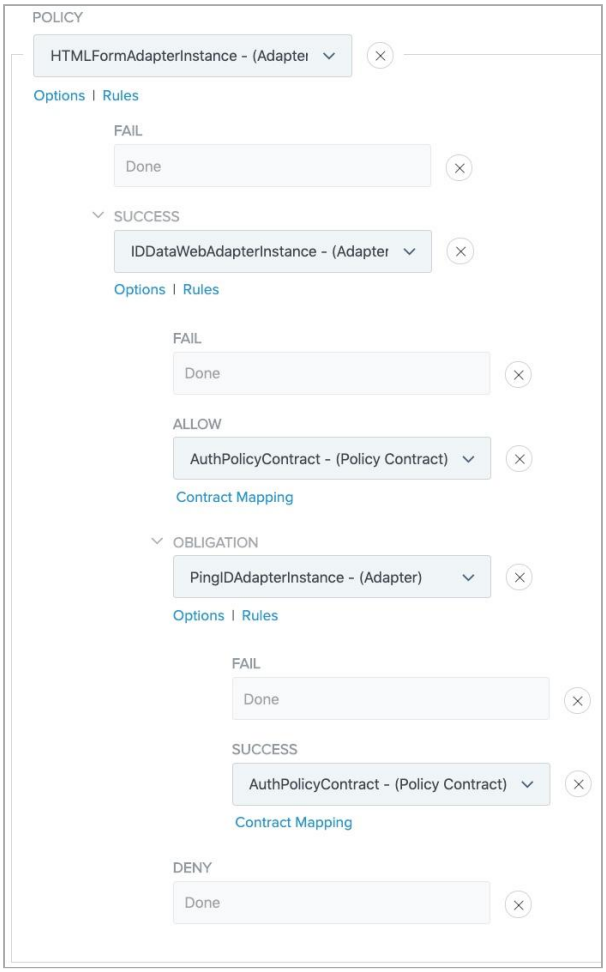


1. Under the ID DataWeb IdP Adapter instance, click **Rules**.
2. On the **Rules** dialog, in the **Attribute Name** list, select **policyDecision**.
3. In the **Condition** list, select **equal to**.
4. In the **Value** field, enter `approve`, `obligation`, or `deny`.
5. In the **Result** field, enter a name. This appears as a new policy path that branches from the authentication source.
6. If you want to add more authentication paths, click **Add** and repeat steps a-e.
7. Click **Done**.

7. Configure each of the authentication paths, including **Fail**, **Success**, and the paths that you defined in the **Rules** dialog.

> **Note**
>
> In case the ID DataWeb API is unreachable or returns an error, we recommend that you allow users to continue to sign on by satisfying stricter authentication requirements.
> You can do this in your authentication policy by setting the **Fail** outcome of the ID DataWeb IdP Adapter instance to point to a second authentication factor, as shown in the following example.
> Alternately, you can do this in your ID DataWeb IdP Adapter instance by setting the **Failure mode** as shown in Configuring an adapter instance.

Copyright © 2025 Ping Identity Corporation



8. Click **Done**.
9. In the **Policies** window, click **Save**.

Troubleshooting

Enabling debug logging

To help with troubleshooting or monitoring, you can turn on activity logging for the adapter.

About this task

These steps are optional.

Learn more about logging in Enabling debug messages and console logging in the PingFederate documentation.

Steps

1. Open the `<pf_install>/pingfederate/server/default/conf/log4j2.xml` file for editing.
2. If you want to log activity for PingFederate and all adapters, do the following:
    1. Find the following section:

Copyright © 2025 Ping Identity Corporation

```
<AsyncRoot level="INFO" includeLocation="false">
        <!-- <AppenderRef ref="CONSOLE" /> -->
        <AppenderRef ref="FILE" />
</AsyncRoot>
```

2. Change `INFO` to `DEBUG`.

```
<AsyncRoot level="DEBUG" includeLocation="false">
        <!-- <AppenderRef ref="CONSOLE" /> -->
        <AppenderRef ref="FILE" />
</AsyncRoot>
```

3. If you want to see the adapter activity in the console, remove the comment tags.

```
<AsyncRoot level="INFO" includeLocation="false">
        <AppenderRef ref="CONSOLE" />
        <AppenderRef ref="FILE" />
</AsyncRoot>
```

3. If you want to log activity just for the ID DataWeb IdP Adapter, add the following line.

```
<Logger name="com.pingidentity.adapters.id-dataweb" level="DEBUG"/>
```

4. If you want to log all HTTP requests and responses with ID DataWeb API, add the following line.

> **Note**
>
> You can use this information with a third-party log analysis tool to monitor for important events, such as when a sign-on event has a high-risk policy decision.

```
<Logger name="org.shaded.id-dataweb" level="INFO"/>
```

5. Save the file.

Troubleshooting information

The following information addresses technical situations that you might encounter after setting up the ID DataWeb Integration Kit.

Copyright © 2025 Ping Identity Corporation

Situation	Information
Using ID DataWeb with PingFederate 11.3 results in the following error: `Refused to load the script 'https://content.maxconnector.com/fp/tags.js?org_id=716kkpe1&api_key=bvrbl1ev61nw7zq7&pageid=verify&session_id=577fa681-da3d-45e0-b982-6fa0841cc2bd' because it violates the following Content Security Policy directive: "script-src 'self' 'nonce-sJyOa0AjeQgjsZDh'". Note that 'script-src-elem' was not explicitly set, so 'script-src' is used as a fallback.`	Update the HTML form template in `$PF_HOME/server/default/conf/templates`. HTML pages implementing ContentSecurityPolicy restrictions might require updating the `script-src` and `image-src` CSP settings when adding the `id_dataweb_device_profiling` JavaScript file to the page. Update `script-src` and `img-src` to include the `https://content.maxconnector.com` and `*.online-metrix.net` host names. For example, PingFederate 11.3 has updated default templates with strict CSP settings. To use the `id_dataweb_device_profiling` JavaScript file with PingFederate 11.3 default templates, update the following line in the template's CSP settings: `<meta http-equiv="Content-Security-Policy" content="script-src 'self' 'nonce-$CSPNonce'; style-src 'self'; img-src 'self'; font-src 'self';" />` Updated line: `<meta http-equiv="Content-Security-Policy" content="script-src 'self' 'unsafe-inline' https://content.maxconnector.com *.online-metrix.net; style-src 'self'; img-src 'self' https://content.maxconnector.com *.online-metrix.net; font-src 'self';" />`

Release notes

Changelog

The following is the change history for the ID DataWeb Integration Kit.

ID DataWeb Integration Kit 1.2 - November 2023

- Added the ability for API calls to send an App ID based on the **Application Name** from the **SP Connection** in PingFederate.
- Added a new configuration setting that enables end-users to mark their device as private to the trust device API.

ID DataWeb Integration Kit 1.1.2 - June 2022

- Updated the Authentication API `.JAR` file version.

Copyright © 2025 Ping Identity Corporation

ID DataWeb Integration Kit 1.1.1 - May 2022

- Removed `proxima-nova` fonts.
- Standardized the archive file structure.

ID DataWeb Integration Kit 1.1 - August 2020

- Improved the appearance of the device profiling template.
- Added support for the flat JSON payload structure from ID DataWeb.
- Added support for the JavaScript Widget for the PingFederate Authentication API.
- Added support for the PingFederate authentication API.

ID DataWeb Integration Kit 1.0 - January 2020

- Initial release.
- Added support for sending user attributes and ThreatMetrix device profiles to the ID DataWeb API.
- Added support for retrieving attributes and risk policy decisions from the ID DataWeb API.
- Added the ability to add device profiling into any browser-based authentication source.
- Added the **Base URL** setting to direct requests to the production or pre-production ID DataWeb API.
- Added the **Failure Mode** and **Fallback Risk Result Value** settings to handle failed risk result requests.
- Added the **API Request Timeout** and **Connection Timeout** settings.
- Added the **Proxy Settings**, **Custom Proxy Host**, and **Custom Proxy Port** settings to override PingFederate default proxy settings.
- Added the **Verify HTTPS Hostname** setting to skip hostname verification.

Known issues and limitations

The following are known issues or limitations for the ID DataWeb Integration Kit.

Known issues

There are no known issues.

Known limitations

There are no known limitations.

Copyright © 2025 Ping Identity Corporation

Download manifest

The following files are included in the ID DataWeb Integration Kit `.zip` archive:

- `Legal.pdf` : Copyright and license information
- `dist/pingfederate/server/default` : Contains the integration files
  - `deploy` : Contains the Java libraries
    - `pf-id-dataweb-adapter-<version>.jar` : `.jar` file that contains the ID DataWeb IdP Adapter.
  - `conf` : Contains the HTML template and script that send device data to the ID DataWeb API.
    - `language-packs/id-dataweb-messages.properties` : A variable file that customizes the messages that appear on the template file.
    - `template` : Contains user-facing HTML template files
      - `id-dataweb.adapter.template.html` : A sign-on redirect page used with the "captured by this adapter" device profiling method. Runs scripts to show a spinner animation and create the device profile.
      - `assets` : Contains functional scripts and files used by the template
        - `css/id-dataweb.css` : A CSS file that customizes the appearance of the template file.
        - `images` : Contains template image files
          - `ping-logo.svg` : An image file with company branding
          - `spinner.svg` : An image file used in a spinner animation
        - `scripts` : Contains script files used to collect and send information
          - `id_dataweb_device_profiling.js` : A JavaScript script that retrieves and runs the ThreatMetrix device profiling script and prepares a cookie with the device profile identifier. Used with the "captured by a previous adapter" device profiling method.
  - `lib/pf-authn-api-sdk-<version>.jar` : A `.jar` file that contains the PingFederate Authentication API SDK

Copyright © 2025 Ping Identity Corporation

Intune Integration Kit

The Intune Integration Kit allows PingFederate to integrate with the Microsoft Intune platform.

This integration allows you to get the security posture of the device that an employee is using to authenticate. Including the security posture in your PingFederate authentication policy allows you to dynamically allow or deny access to resources for each user that signs on. For example, you can:

- Deny access for users who have devices that are not managed by Intune.
- Allow access for users that only have devices that are in compliance.
- Deny access to a user whose current device has been compromised or rooted.

This mitigates the risk of corporate resources being accessed from employees' mobile and desktop devices.

> **Tip**
>
> You can learn how to use Microsoft Intune to manage the PingID app on mobile devices in Configuring Microsoft Intune for PingID in the PingID documentation.

Components

Intune IdP Adapter

- Allows you to retrieve the security posture for the authenticating user's current device, or all devices associated with that user.

Intended audience

This document is intended for PingFederate administrators.

If you need help during the setup process, you can find more information in the following:

- The following sections of the PingFederate documentation:
  - Managing IdP adapters
  - Authentication policies
- The X.509 Certificate Integration Kit documentation
- The following sections of the Microsoft documentation:
  - Enroll devices in Microsoft Intune
  - Configure the certification authority
  - Set up NDES
  - Configure infrastructure to support SCEP with Intune

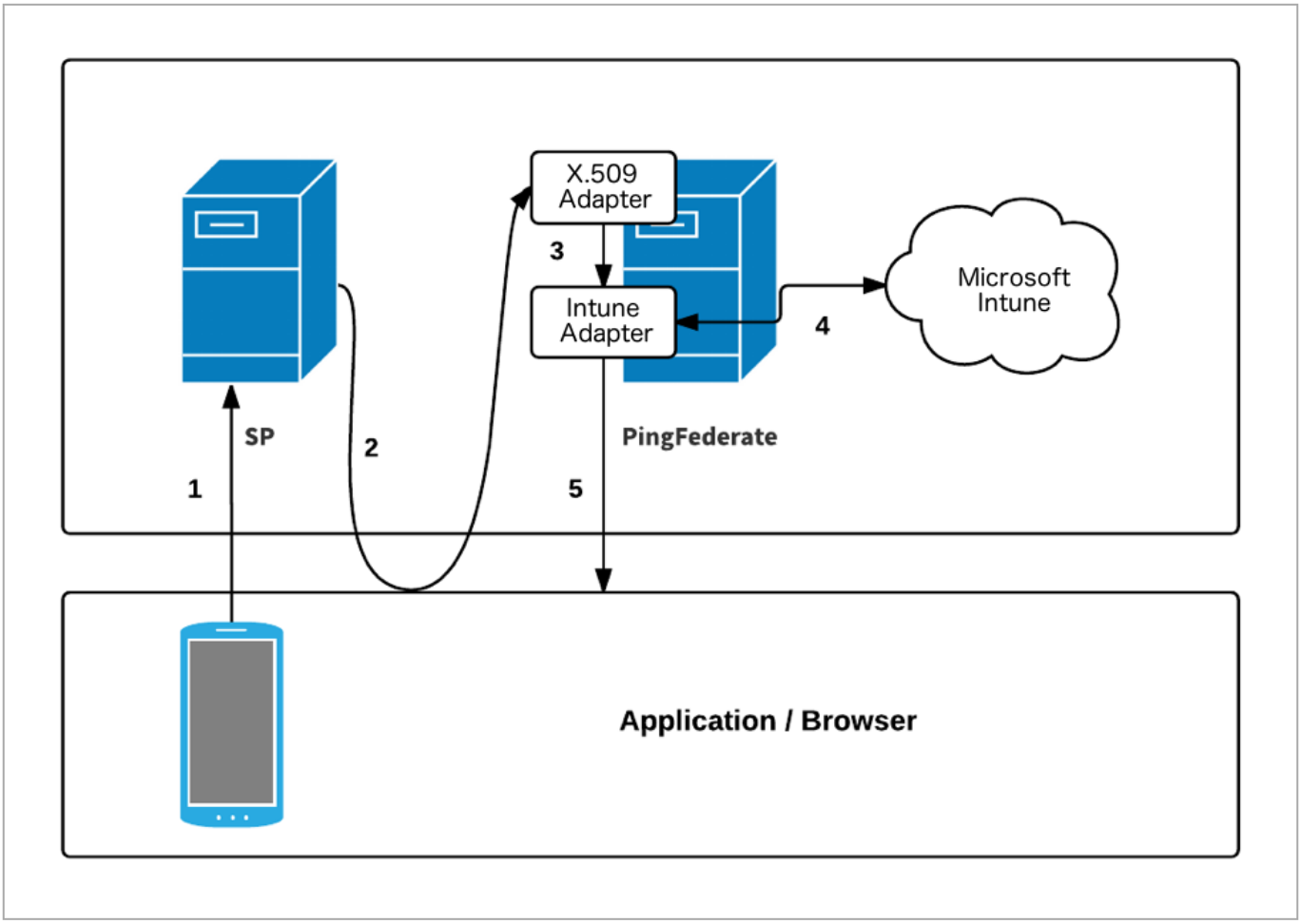
Copyright © 2025 Ping Identity Corporation

System requirements

- PingFederate 9.0 or later.
- A Microsoft Intune tenant.
- A Windows 2012 R2 (or later) server configured as a Certificate Authority (CA) server.
- A Windows 2012 R2 (or later) server configured with the Network Device Enrollment Service (NDES) server role and Simple Certificate Enrollment Protocol (SCEP).
- Mobile application support requires the following:
  - A Web Controller to allow the application to read the certificate from the device. For Android, Chrome 45+ & Chrome Custom Tabs. For Apple, iOS9+ & Safari View Controller.
  - AppAuth is required for shared sessions between supported mobile applications.

Overview of the SSO flow

With the Intune Integration Kit, PingFederate parses the user's `deviceId` or `userPrincipalName` attribute from an X.509 certificate and uses it to get the device's security posture from Microsoft Intune.



Copyright © 2025 Ping Identity Corporation

Description

1. A user requests access to a resource by using a device that is enrolled with Intune.
2. The service provider (SP) redirects the request to PingFederate. The browser requests the user's X.509 certificate.
3. The PingFederate X.509 Certificate Adapter validates the certificate against a specified list of issuers or the the server's list of trusted certificate authorities. Depending on your configuration, the X.509 Certificate Adapter passes the `deviceId` or `userPrincipalName` (UPN) attribute to the Intune IdP Adapter.
4. The Intune IdP Adapter contacts the Microsoft Graph API to look up the user's security posture information. Intune provides one of the following results depending on the Intune IdP Adapter instance configuration:
    - The security posture for the current device based on the `deviceId`.
    - An aggregate security posture for all of the current user's devices based on the `userPrincipalName`.
5. The PingFederate authentication policy uses the result from Intune to determine whether the user is redirected to the resource they requested.

Setup

Registering PingFederate as an application in Azure Active Directory

To allow PingFederate to access Intune through the Microsoft Graph API, add an application in Azure Active Directory with the necessary permissions.

Steps

1. In Azure Active Directory, follow the steps in Quickstart: Register an application in Microsoft Entra ID in the Microsoft documentation.
2. Add the `DeviceManagementManagedDevices.Read.All` application permission from the Microsoft Graft API to your application by following the steps in Configure app permissions for a web API in the Microsoft documentation.
3. Note your **Tenant ID**, **Application ID**, and **Application Secret**. You will use these in Configuring an adapter instance.

Deploying and configuring the X.509 Certificate Integration Kit

The Intune Integration Kit requires the X.509 Certificate Adapter to parse information out of the user's client certificate.

Steps

1. Follow the steps in Deploying the integration files in the X.509 Certificate Integration Kit documentation.
2. Follow the steps in Configuring an adapter instance in the X.509 Certificate Integration Kit documentation with one of the following modifications:

Copyright © 2025 Ping Identity Corporation

Choose from:

- If you want to base security posture results on the user's current device using the `deviceId` attribute, on the **Extended Contract** tab, add the CN attributes to parse from the SubjectDN certificate.
- If you want to base security posture results on all of a user's devices using the `userPrincipalName` attribute, do the following:
    1. On the **IdP Adapter** tab, in the **Advanced Fields** section, select the **Include Subject Alternative Name (SAN)** checkbox.
    2. On the **Extended Contract** tab, add the `deviceId` or `userPrincipalName` attribute. The attribute that you add determines the type of security posture result you get from Intune. Learn more in **Device Lookup Attribute** in the Intune IdP Adapter settings reference.

Deploying the integration files

To get started with the integration, deploy the Intune Integration Kit files to your PingFederate directory.

Steps

1. Download the integration kit `.zip` archive from the Ping Identity Integration Directory.
2. Stop PingFederate.
3. If you are upgrading from an earlier version of the adapter, delete `pf-intune-adapter-<version>.jar` from `<pf_install>/pingfederate/server/default/deploy`.
4. From the `.zip` archive, copy the contents of `dist` to `<pf_install>/pingfederate/server/default/deploy`.
5. Start PingFederate.
6. If you operate PingFederate in a cluster, repeat steps 2 - 5 for each instance of PingFederate.

Configuring an adapter instance

Configure the Intune IdP Adapter to allow PingFederate to get the security posture for sign-on devices.

Before you begin

Make sure that you've configured the X.509 Certificate Integration Kit. Learn more in Deploying and configuring the X.509 Certificate Integration Kit.

Steps

1. Sign on to the PingFederate administrative console.
2. On the **Identity Provider > Adapters** tab, click **Create New Instance**.
3. On the **Type** tab, set the basic adapter instance attributes.
    1. In the **Instance Name** field, enter a name for the adapter instance.
    2. In the **Instance ID** field, enter a unique identifier for the adapter instance.

Copyright © 2025 Ping Identity Corporation

3. In the **Type** list, select **Intune Adapter**, then click **Next**.

4. On the **IdP Adapter** tab, configure the adapter instance by referring to Intune IdP Adapter settings reference, then click **Next**.

5. On the **Extended Contract** tab, add any attributes that you expect to retrieve in addition to the core contract attributes, then click **Next**.

> **Important**
>
> Aside from the core contract attributes, the Intune IdP adapter can only provide the following two attributes:
> - `isCompliant`
> - `isManaged`

6. Complete the adapter configuration.

7. On the **Summary** tab, check that the configuration is correct, then click **Done**.

8. On the **Manage IdP Adapter Instances** tab, click **Save**.

Intune IdP Adapter settings reference

Field descriptions for the Intune IdP Adapter configuration page.

Standard fields

Field	Description
Tenant ID	The ID for your Intune tenant.
Application ID	The ID for your application in Intune.
Application Secret	The password for your application in Intune.
Device Lookup Attribute	**Device ID**: Gets the security posture for the current device by querying the `deviceId` attribute. The `deviceId` must be embedded in the Subject Alternative Name (SAN) in the client certificate. Learn more in Configure infrastructure to support SCEP with Intune in the Microsoft documentation. **User Principal Name**: Gets the security posture for all devices associated with the user by querying the `userPrincipalName` attribute.

Adding Intune security posture results to your authentication policy

By modifying your PingFederate authentication policy to include the `isManaged` and `isCompliant` results from Intune, you can control access to resources based on the device's security posture.

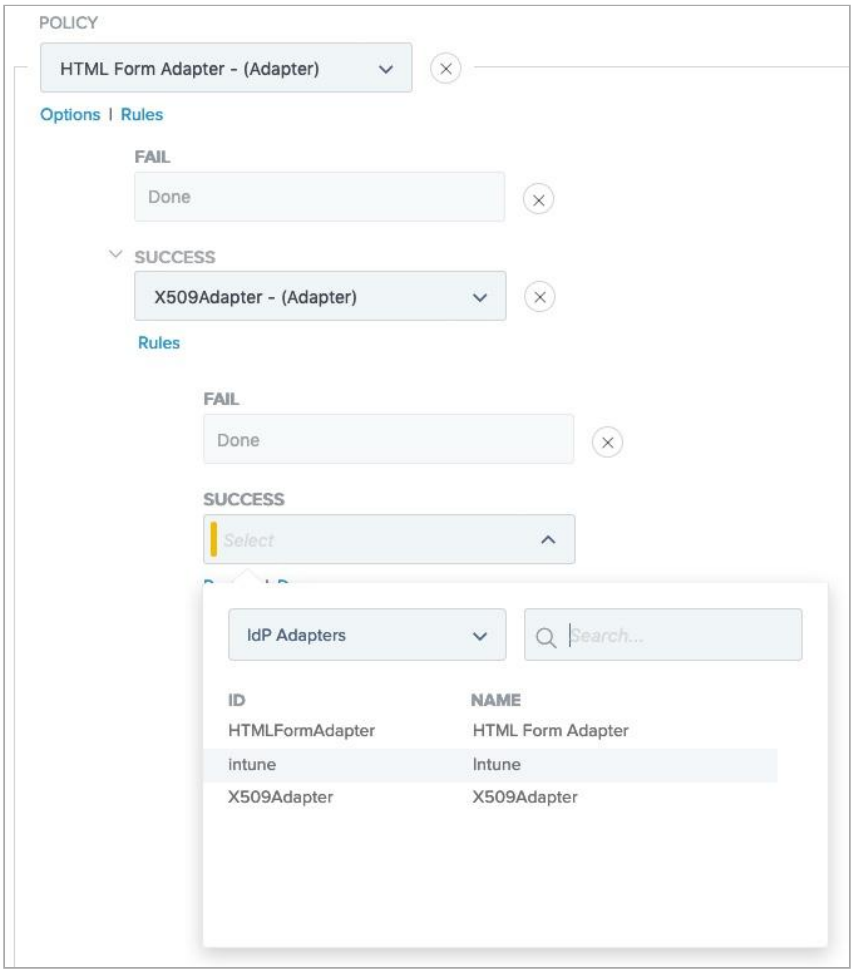
About this task

These steps are designed to help you add to an existing authentication policy. Learn more about configuring authentication policies in Policies in the PingFederate documentation.

Copyright © 2025 Ping Identity Corporation

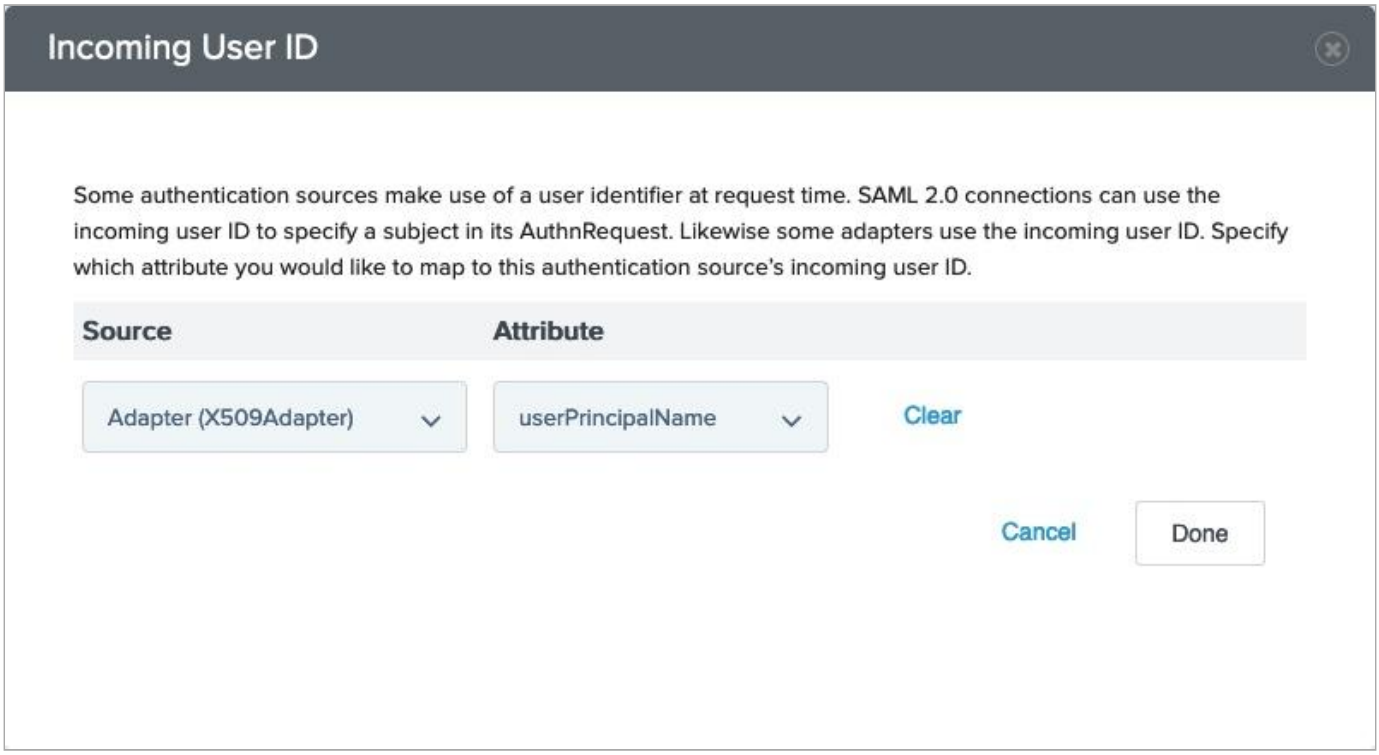
Steps

1. Sign on to the PingFederate administrative console.
2. On the **Identity Provider** page, under **Authentication Policies**, click **Policies**.
3. Open an existing authentication policy, or create a new one. Learn more in Defining authentication policies in the PingFederate documentation.
4. In the **Policy** area, in the **Select** list, select an Intune IdP Adapter instance.



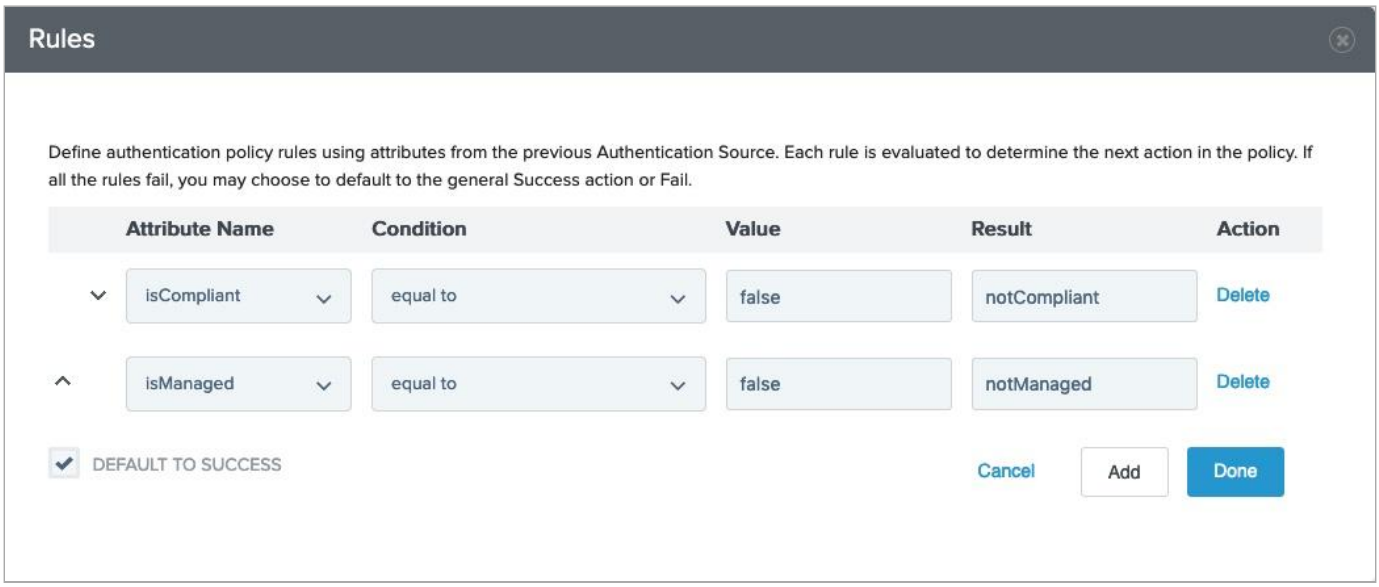
5. Map the `deviceId` (shown as **CN**) or `userPrincipalName` from your X.509 Adapter instance into the Intune IdP Adapter instance.

Copyright © 2025 Ping Identity Corporation



1. Under the Intune IdP Adapter instance, click **Options**.
2. On the **Options** dialog, from the **Source** list, select your X.509 Adapter instance.
3. From the **Attribute** list, select **CN** or **userPrincipalName**. Click **Done**.

6. Define policy paths based on the two security posture attributes, `isCompliant` and `isManaged`.

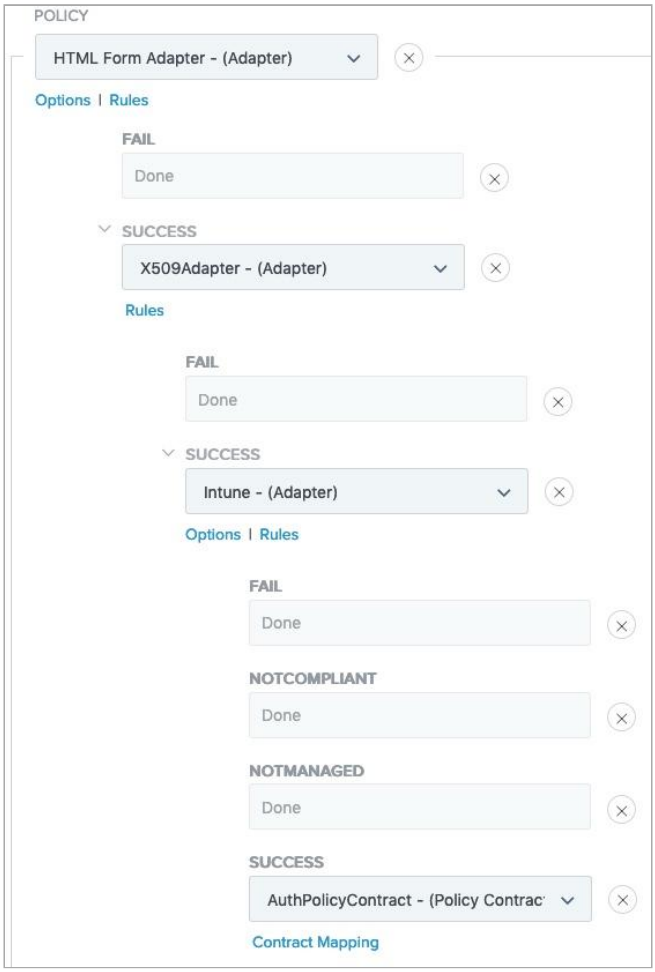


1. Under the Intune IdP Adapter instance, click **Rules**.
2. On the **Rules** dialog, in the **Attribute Name** list, select **isCompliant**.
3. In the **Condition** list, select **equal to**.

Copyright © 2025 Ping Identity Corporation

4. In the **Value** field, enter `true` or `false`.
5. In the **Result** field, enter a name. This appears as a new policy path that branches from the authentication source.
6. Repeat steps b - e for **isManaged**.
7. Click **Done**.

7. Configure each of the authentication paths, including **Fail**, **Success**, and the paths that you defined in the **Rules** dialog.



8. Click **Done**.
9. In the **Policies** window, click **Save**.

Creating certificate profiles in Intune

To complete the configuration, import the certificate authority (CA) root certificate into PingFederate, and configure trusted certificate and SCEP profiles in Intune.

About this task

Learn more about certificate profiles in Intune in the following sections of the Microsoft documentation:

- Use certificates for authentication in Microsoft Intune

Copyright © 2025 Ping Identity Corporation

- Create a device profile in Microsoft Intune

Steps

1. In PingFederate, import the root certificate from your CA into the global trust list. Learn more in Manage trusted certificate authorities in the PingFederate documentation.
2. Sign on to Microsoft Azure as an administrator.
3. Follow the steps in Create trusted certificate profiles in the Microsoft documentation.
4. Follow the steps in Create and assign SCEP certificate profiles in Intune in the Microsoft documentation, with the following details:
    1. On the **Configuration settings** tab, in the **Certificate type** list, select **User**.
    2. To get the security posture for all of the user's devices, from the **Subject name format** list, select **Common name**. From the **Subject alternative name** list, select **User principal name (UPN)**.
    3. To get the security posture for the user's current device only, from the **Subject name format** list, select **Custom**. In the **Custom** field, modify the value to include `CN={{AAD_Device_ID}}`.

Testing the integration

After you have completed the setup, you can test your Intune integration.

Steps

1. On a testing device, install the Intune Company Portal app. Follow the prompts to set up the device.
2. On the **Intune > Devices > All devices** page, select your test device. Click **Sync**.
3. Wait 3 - 5 minutes for the device to communicate with Microsoft Intune and for Microsoft Intune to communicate with the NDES server. You might need to follow prompts on the device to complete the process.
4. On the **Intune > Device configuration > Assignment status** page, check the profiles that you created to see if the device registration succeeded.
5. Test your configuration:
    1. In a browser on your test device, go to the PingFederate login URL for your adapter, such as `https://pf_host:pf_port/idp/startSSO.ping?PartnerSpId=YourConnectionID`.
    2. If you are prompted to select a certificate from the device's certificate store, select the certificate that is issued from your CA that shows either your device ID or username (depending on your configuration from Creating certificate profiles in Intune). After confirming your certificate selection, you should be redirected to your SP application.

Release notes

Changelog

The following is the change history for the Intune Integration Kit.

Copyright © 2025 Ping Identity Corporation

Intune Integration Kit 1.1 – May 2020

- Added the ability to get the security posture for all of the current user's devices or only the current device.

Intune Integration Kit 1.0 – December 2017

- Initial release

Known issues and limitations

The following are known issues or limitations for the Intune Integration Kit.

Known issues

There are no known issues.

Known limitations

There are no known limitations.

Download manifest

The following files are included in the Intune Integration Kit:

- `ReadMeFirst.pdf` : Contains links to this online documentation.
- `legal` : A directory that contains this document.
  - `Legal.pdf` : Contains copyright and license information.
- `dist` : A directory that contains the Java libraries needed to run the adapter.
  - `pf-intune-idp-adapter-<version>.jar` : A JAR file that contains the Intune IdP Adapter.

Copyright © 2025 Ping Identity Corporation

iovation Integration Kit

The iovation Integration Kit allows PingFederate to communicate with iovation Device Risk.

By sending a device profile to iovation when a user signs on, PingFederate can get a security risk assessment for the sign-on event. The risk assessment results allow you to adjust authentication requirements based on your authentication policies.

Components

- iovation IdP Adapter
  - When a user signs on through PingFederate, the adapter sends the device profile to the iovation API, and retrieves a risk result.
- Template and script files
  - When a user signs on through PingFederate, these files retrieve the latest JavaScript from iovation in order to build a profile of the device.

Intended audience

This document is intended for PingFederate administrators. Before starting, familiarize yourself with the following resources:

- PingFederate and iovation [PDF] solution brief
- Configuring Integration Points in the iovation Help Center
- Defining and Managing Business Rules in the iovation Help Center
- Authentication policies in the PingFederate documentation

System requirements

- PingFederate 9.0 or later
- An iovation cloud service account
- iovation Intelligence Center login credentials (provided by iovation):
  - Account
  - Login
  - Password
- iovation Device Risk credentials (provided by iovation):
  - Subscriber ID
  - Subscriber account
  - Passcode

Copyright © 2025 Ping Identity Corporation

Overview of iovation Device Risk

iovation Device Risk collects a device profile and other transaction data and uses a series of rules to evaluate the level of security risk for a transaction. The type of transaction is flexible, but in the context of PingFederate, it's usually a user sign-on event.

Device profile, blackbox, and transaction insight parameters

When a user signs on, iovation JavaScript collects hundreds of data elements associated with the device, including the device type, geolocation information, information about the browser, and system settings such as language settings. Together, this data is called the device profile.

The iovation JavaScript encrypts the device profile in a package called a *blackbox*.

In addition to the device profile data, you can take attributes from previous authentication sources and send them to iovation as transaction insight parameters.

The iovation IdP Adapter sends the blackbox and transaction insight parameters to iovation for analysis.

Integration points

Each iovation IdP Adapter instance communicates with the iovation API through a specific integration point. Each integration point typically represents one type of interaction, such as sign-on or password change.

You can find guidance on integration point design in Planning and Designing Integration Points in the iovation Help Center.

Rule sets and business rules

Each integration point is associated with one rule set, which is a collection of business rules and rule groups.

Each business rule has a numeric weight assigned to it. When the conditions of the rule are met by the device profile or transaction data, the weight affects the total risk score for a transaction.

You can find a list of business rule categories in About iovation Device Risk and an example business rule scenario in Business Rule Basics in the iovation Help Center.

Risk results

After processing the device profile through the rule set, Device Risk matches the resulting risk score to one of three risk results: `allow`, `review`, or `deny`. The rule set determines the numeric threshold associated with each of the results.

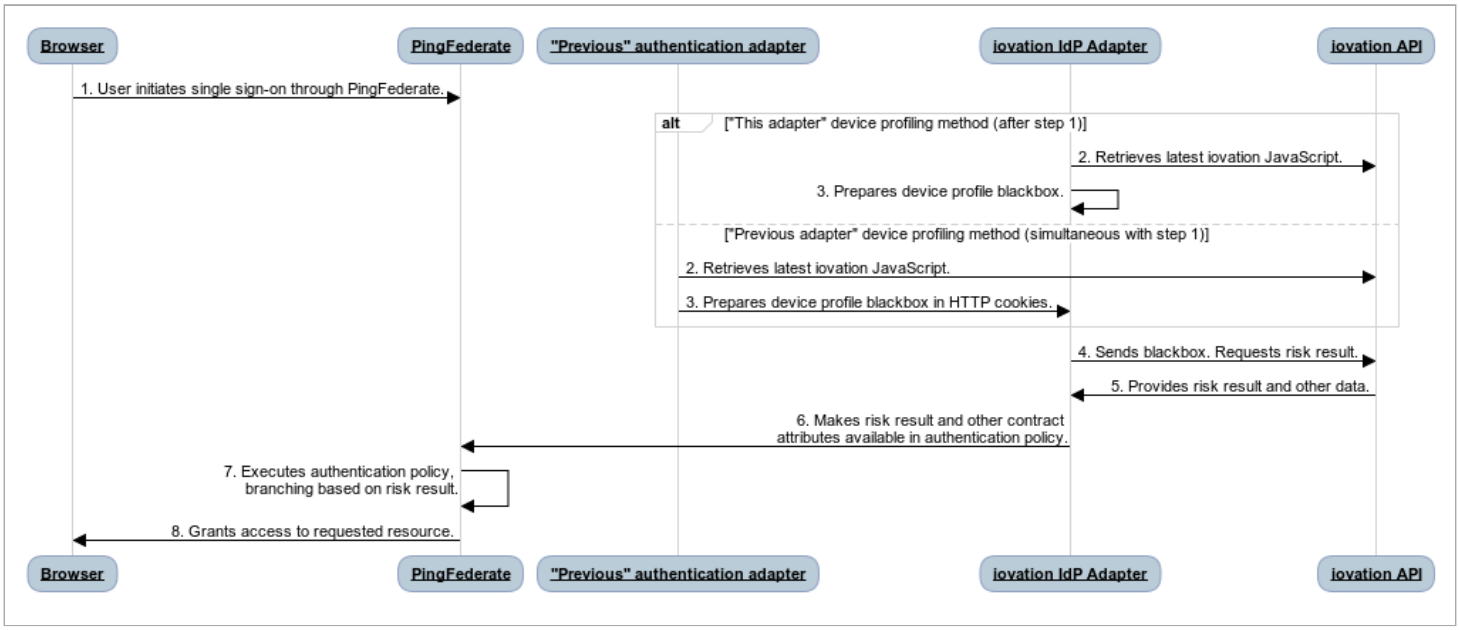
You can find more details about result thresholds in About Rule Weights and Thresholds in the iovation Help Center.

The iovation API provides the risk result and other data in a response to PingFederate. By including the risk result in your authentication policy, you decide how each of the `allow`, `review`, and `deny` results affects a user's ability to sign on in your environment.

Overview of the SSO flow

With the iovation Integration Kit, PingFederate includes the iovation API in the sign-on flow as follows.

Copyright © 2025 Ping Identity Corporation



Description

1. A user initiates the sign-on process by requesting access to a protected resource.
2. Depending on the device profiling method, the iovation IdP Adapter or a previous authentication adapter retrieves the latest JavaScript from iovation.

   > **Note**
   >
   > For the previous adapter method, this takes place at the same time as step 1.

3. Depending on the device profiling method, the iovation IdP Adapter or a previous authentication adapter runs the iovation JavaScript, which builds the device profile and packages it in an encrypted blackbox.
4. The iovation IdP Adapter sends the blackbox and transaction insight parameters to the iovation API and requests the risk result for the transaction.
5. The iovation API returns a JSON payload with the risk result and other attributes to the iovation IdP Adapter.
6. The iovation IdP Adapter makes the risk result and contract attributes available in the authentication policy.
7. PingFederate executes the authentication policy, which branches based on the risk result reported by the iovation IdP Adapter.
8. PingFederate returns the resource that the user requested.

Device profiling method

To produce a risk assessment for a sign-on event, FraudForce requires information about the user's device. You can collect this information using the iovation IdP Adapter or by attaching the scripts to another adapter in your sign-on flow.

Copyright © 2025 Ping Identity Corporation

Captured by the iovation IdP Adapter

The iovation IdP Adapter inserts a page into the sign-on flow that shows a spinner animation while the device profiling process runs. The page typically appears after the first-factor authentication page, as determined by your authentication policy.

> **Note**
>
> This method adds a short wait time to the sign-on process.

Captured by a previous adapter

Scripts are added to a page presented by another adapter in your authentication policy. This page must appear earlier in the sign-on flow than the iovation IdP Adapter, such as the first-factor authentication page.

This method reduces wait times by allowing the iovation JavaScript to prepare the device profile blackbox while the user types in their credentials.

The iovation IdP Adapter is still responsible for sending data back to the iovation API, so the script uses HTTP cookies to pass the device profile blackbox from the first-factor authentication page to the iovation IdP Adapter.

You can integrate the iovation script into any page that meets the following criteria:

- The page appears before the iovation IdP Adapter in the sign-on flow.
- The page is hosted in the same domain as your PingFederate server. This allows you to use HTTP cookies to pass the blackbox to the iovation IdP Adapter.

  > **Tip**
  >
  > You might be able to work around this requirement by consolidating your domains with a reverse proxy server.

- You have access to add JavaScript to the page's HTML code.

You can find instructions in Adding device profiling to an authentication page after you finish Deploying the integration files.

Setup

To use the iovation Integration Kit with PingFederate, set up a reverse proxy server, configure iovation, then set up the integration on your PingFederate server.

Steps

1. Set up a reverse proxy server as shown in Retrieving & Serving Dynamic iovation JavaScript in the iovation Help Center.

   By positioning PingFederate behind the reverse proxy server, you allow it to retrieve and run the latest iovation device-profiling JavaScript when a user signs on.

2. In the iovation Intelligence Center, create business rules and an integration point.

Copyright © 2025 Ping Identity Corporation

This determines how risk results are determined for sign-on transactions through your iovation IdP Adapter instance. For configuration guidance and instructions, explore the relevant topics in the iovation Help Center, such as Defining and Managing Business Rules and Configuring Integration Points. The iovation customer success team can help you plan business rules to suit your needs.

3. Deploy the integration files to your PingFederate server as shown in Deploying the integration files.
4. (Optional) Reduce user sign-on times by Adding device profiling to an authentication page.
5. Configure the iovation IdP Adapter as shown in Configuring an adapter instance.
6. Integrate iovation risk results into your authentication policy as shown in Adding risk results to your authentication policy.

Deploying the integration files

To get started with the integration, deploy the iovation Integration Kit files to your PingFederate directory.

Steps

1. Download the integration kit `.zip` archive from the **Add-ons** tab of the PingFederate downloads page or the Ping Identity Integration Directory.
2. Stop PingFederate.
3. If upgrading from a previous version of the adapter, delete the `pf-iovation-adapter-<version>.jar` file from the `<pf_install>/pingfederate/server/default/deploy` directory.
4. In the `.zip` archive, copy the integration kit files to the PingFederate directory.
    1. Copy the contents of `dist` to `<pf_install>/pingfederate/server/default/deploy`.
    2. Copy the contents of `conf` to `<pf_install>/pingfederate/server/default/conf`.
5. Start PingFederate.
6. If you operate PingFederate in a cluster, repeat steps 2 - 5 for each instance of PingFederate.

Adding device profiling to an authentication page

Instead of using the iovation IdP Adapter to gather the device profile, you can speed up the sign-on process by adding the device profiling scripts to an existing authentication adapter.

About this task

These steps describe how to add the device-profiling scripts to the HTML Form Adapter that's bundled with PingFederate. You can adapt these instructions for any page that meets the criteria listed in Device profiling method.

The authentication page you modify must appear earlier in the sign-on flow than the iovation IdP Adapter.

Copyright © 2025 Ping Identity Corporation

Steps

1. Embed the JavaScript files in the page.
    1. Open the `<pf_install>/pingfederate/server/default/conf/template/html.form.login.template.html` file for editing.
    2. At the end of the file, above the </body> tag, add the following external script references:

    ```
    <script language="javascript" src="../assets/scripts/iovation_adapter_custom.js"></script>
    <script language="javascript" src="../assets/scripts/iovation_device_profiling.js"></script>
    ```

    3. Save the file.
2. (Optional) Customize the name prefix for the blackbox cookie to suit your environment:
    1. Open the `<pf_install>/pingfederate/server/default/conf/assets/scripts/iovation_adapter_custom.js` file for editing.
    2. On the following line, change `iovation_bb` to a name prefix of your choosing:

    ```
    var bbCookieNamePrefix = "iovation_bb";
    ```

    > **Note**
    >
    > If you customize the name prefix, you must enter the new prefix value in the **Blackbox Cookie Name Prefix** field in Configuring an adapter instance. Additionally, you must set the **Device Profiling Method** to **Captured by a previous adapter**. Learn more in step 5.

    3. Save the file.
3. (Optional) Increase the client header buffer size setting on your proxy server.

    It must be able to accommodate the blackbox HTTP cookies (up to 8 KB) as well as any other cookies in your sign-on flow.
4. Configure your proxy server to pass the user's IP address to PingFederate through HTTP headers:
    1. In your reverse proxy server configuration, specify a header to store the IP address associated with the request, such as *<X-Forwarded-For>*.
    2. In the PingFederate admin console, go to **Security > System Integration > Incoming Proxy Settings**.
    3. In the **HTTP Header For Client IP Addresses** field, enter the header you specified in step 4a.

        You can find more details in Configure incoming proxy settings in the PingFederate documentation.
5. When Configuring an adapter instance, configure the **Device Profiling Method** and **Blackbox Cookie Name Prefix** fields as shown in iovation IdP Adapter settings reference.

Configuring an adapter instance

Configure the iovation IdP Adapter to determine how PingFederate communicates with the iovation API.

Copyright © 2025 Ping Identity Corporation

Steps

1. Sign on to the PingFederate administrative console.
2. Go to **Authentication > IdP Adapters** and click **Create New Instance**.
3. On the **Type** tab, set the basic adapter instance attributes:
    1. In the **Instance Name** field, enter a name for the adapter instance.
    2. In the **Instance ID** field, enter a unique identifier for the adapter instance.
    3. In the **Type** list, select **iovation IdP Adapter**. Click **Next**.
4. (Optional) On the **IdP Adapter** tab, in the **Optional Transaction Insight Parameters** section, configure additional transaction data to send to iovation with the device profile.

    You can find an overview in Transaction insight parameters.

    1. Click **Add a new row to 'Transaction Insight Parameters'**.
    2. In the **Incoming Attribute Name** field, enter the name of an attribute from any authentication source that appears earlier in your PingFederate authentication policy than the iovation IdP Adapter.
    3. In the **iovation API Attribute Parameter** field, enter the transaction insight parameter, such as `billingCountry` or `eventId`.
    4. In the **Action** column, click **Update**.
    5. To map another transaction insight parameter, repeat steps a - d.
5. (Optional) On the **IdP Adapter** tab, in the **Optional iovation Attributes** section, configure additional attributes to include in the attribute contract from the iovation payload.

    > **Note**
    >
    > The `transactionRiskResult` (`/result`) attribute is already included in the core contract.

    1. Click **Add a new row to 'Optional iovation Attributes'**.
    2. In the **Local Attribute** field, enter a name of your choosing for an attribute.
    3. In the **iovation API Attribute Mapping** field, enter the JSON Pointer syntax for the value of the matching iovation attributes as shown in JSON Pointer syntax reference.
    4. In the **Action** column, click **Update**.
    5. To add more attributes, repeat steps a - d.
6. On the **IdP Adapter** tab, configure the adapter instance by referring to iovation IdP Adapter settings reference. Click **Next**.
7. On the **Actions** tab, test your connection to the iovation API. Resolve any issues that are reported, and then click **Next**.
8. On the **Extended Contract** tab, add any attributes that you included in the **Optional iovation Attributes** section of the **IdP Adapter** tab.
9. Complete the adapter configuration.

Copyright © 2025 Ping Identity Corporation

10. On the **Summary** tab, check that the configuration is correct. Click **Done**.

Transaction insight parameters

When a user signs on, iovation can collect optional transaction data along with the device profile. In the iovation Intelligence Center, you can use these transaction insight parameters in business rules to affect risk results.

The iovation IdP Adapter allows you to take attributes from previous authentication sources in your PingFederate authentication policy and send them to iovation as transaction insight parameters.

For example, on the first-factor authentication page, you collect the user's security PIN type. Then, in your iovation IdP Adapter instance, you map the PIN type attribute to the iovation `securityPinType` parameter. This allows you to create a business rule that considers certain PIN types to be higher risk.

You can find a complete list of transaction insight parameters you can populate in Transaction Insight Fields in the iovation Help Center.

JSON Pointer syntax reference

JavaScript Object Notation (JSON) Pointer defines a syntax for identifying a specific value within a JSON payload. Using the sample payload and JSON Pointer examples below, identify the attributes that you want to use to populate your attribute contract.

You can find a complete technical description of JSON Pointer syntax in the IETF RFC 6901 - JavaScript Object Notation (JSON) Pointer.

You can find more information about the iovation payload structure in Transaction REST API Response Attributes in the iovation Help Center.

Copyright © 2025 Ping Identity Corporation

Example JSON payload

Copyright © 2025 Ping Identity Corporation

```
{
    "id": "4a5cc146-9a48-4663-b93e-cdc3fa4517ba",
    "result": "R",
    "reason": "Accounts Per Device- 3 Max",
    "statedIp": "127.0.0.1",
    "accountCode": "sampleAccountCode",
    "trackingNumber": 877301938420090395,
    "details": {
        "device": {
            "alias": 1139193327613838 48,
            "blackboxMetadata": {
                "age": 937853,
                "timestamp": "2019-10-10T23:42:03Z"
            },
            "browser": {
                "cookiesEnabled": true,
                "configuredLanguage": "EN-CA,EN-US;Q=0.7,EN;Q=0.3",
                "language": "EN-CA",
                "type": "FIREFOX",
                "timezone": "480",
                "version": "70.0"
            },
            "firstSeen": "2019-10-01T23:48:46.393Z",
            "isNew": false,
            "os": "INTEL MAC OS X 10.14",
            "screen": "1440X3440",
            "type": "MAC"
        },
        "statedIp": {
            "address": "127.0.0.1",
            "source": "subscriber"
        },
        "realIp": {
            "address": "8.8.8.8",
            "isp": "INTERNET NOW",
            "ipLocation": {
                "city": "MANASSAS",
                "country": "UNITED STATES",
                "countryCode": "US",
                "latitude": 38.74622,
                "longitude": -77.48911,
                "region": "VIRGINIA"
            },
            "parentOrganization": "INTERNET NOW",
            "source": "iovation"
        },
        "ruleResults": {
            "score": -1,
            "rulesMatched": 1,
            "rules": [
                {
                    "type": "Accounts Per Device",
                    "reason": "Accounts Per Device- 3 Max",
                    "score": -1
                }
            ]
        }
    }
}
```


Copyright © 2025 Ping Identity Corporation

JSON Pointer syntax

Description	JSON Pointer	Example value
Result of the transaction risk assessment. **A** `allow` **R** `review` **D** `deny`	`/result`	`R`
The numeric score that determines the risk result.	`/details/ruleResults/score`	`-1`
The country that the device is signing in from.	`/details/realIp/ipLocation/country`	`UNITED STATES`

iovation IdP Adapter settings reference

Field descriptions for the iovation IdP Adapter configuration page.

Field	Description
Subscriber ID	The subscriber ID iovation provides you.
Subscriber Account	The subscriber account iovation provides you. This is typically `OLTP`.
Subscriber Passcode	The subscriber passcode iovation provides you.
Iovation Integration Point	The name of the integration point you configured in the iovation Intelligence Center in Setup.
Iovation API Base URL	The URL of the iovation API. ***Production*** https://api.iovation.com ***Pre-production*** https://ci-api.iovation.com

Copyright © 2025 Ping Identity Corporation

Field	Description
Device Profiling Method	The stage of authentication that device details are collected in. You can find a more detailed description in Device profiling method. **Note** If you completed the optional steps in Adding device profiling to an authentication page, select **Captured by a previous adapter**. Otherwise, select **Captured by this adapter**. The default value is **Captured by this adapter**.

Field	Description
Device Profiling Timeout	The amount of time, in milliseconds, that PingFederate waits for the iovation scripts to collect device details. **Note** Applies only when **Device Profiling Method** is **Captured by this adapter**. The default value is `2000`.
Blackbox Cookie Name Prefix	The prefix of the cookies that contain the iovation blackbox captured by a previous adapter. **Note** Applies only when **Device Profiling Method** is **Captured by a previous adapter**. If you customized the name prefix for the blackbox cookie in Adding device profiling to an authentication page, enter the same name in this field. The default value is `iovation_bb`.
Fraud Check API Endpoint	The iovation REST Fraud Check API endpoint that assesses risk for transactions based on the device and transaction details. The default value is `/fraud/v1/subs/${subscriberId}/checks`.

Copyright © 2025 Ping Identity Corporation

Field	Description
Failure Mode	When iovation is unavailable or an error occurs, this setting determines whether the user's sign-on attempt should fail or continue with a pre-determined risk result. For cases where the iovation API is unavailable or returns an error, you can allow users to continue to sign on by satisfying stricter authentication requirements. You can configure this either: • In your adapter configuration by setting the **Failure Mode** to return the `review` result. • In your authentication policy by setting the **Fail** outcome of the iovation IdP Adapter instance as shown in Adding risk results to your authentication policy. The default value is **Continue with fallback risk result**.
Fallback Risk Result Value	The risk result (for example, `review`, `deny`, or `unknown`) to use in the authentication policy when iovation is unavailable or an error occurs, and **Failure Mode** is **Continue with fallback risk result**. The default value is `review`.
API Request Timeout	The amount of time, in milliseconds, that PingFederate waits for the API to respond to requests. A value of `0` disables the timeout. The default value is `2000`.
Connection Timeout	The amount of time, in milliseconds, that PingFederate allows to establish a connection with the API. A value of `0` disables the timeout. The default value is `2000`.
Proxy Settings	Defines proxy settings for outbound HTTP requests. The default value is **System Defaults**.
Custom Proxy Host	The proxy server hostname to use when **Proxy Settings** is set to **Custom**. This field is blank by default.
Custom Proxy Port	The proxy server port to use when **Proxy Settings** is **Custom**. This field is blank by default.
Verify HTTPS Hostname	When a connection is established with iovation, PingFederate matches the target hostname against the names stored inside the server's X.509 certificate. This security measure ensures that PingFederate is connecting to the correct server. This checkbox is selected by default.

Copyright © 2025 Ping Identity Corporation

Adding risk results to your authentication policy

By modifying your authentication policy to include risk results, you can dynamically change authentication requirements for higher-risk users.

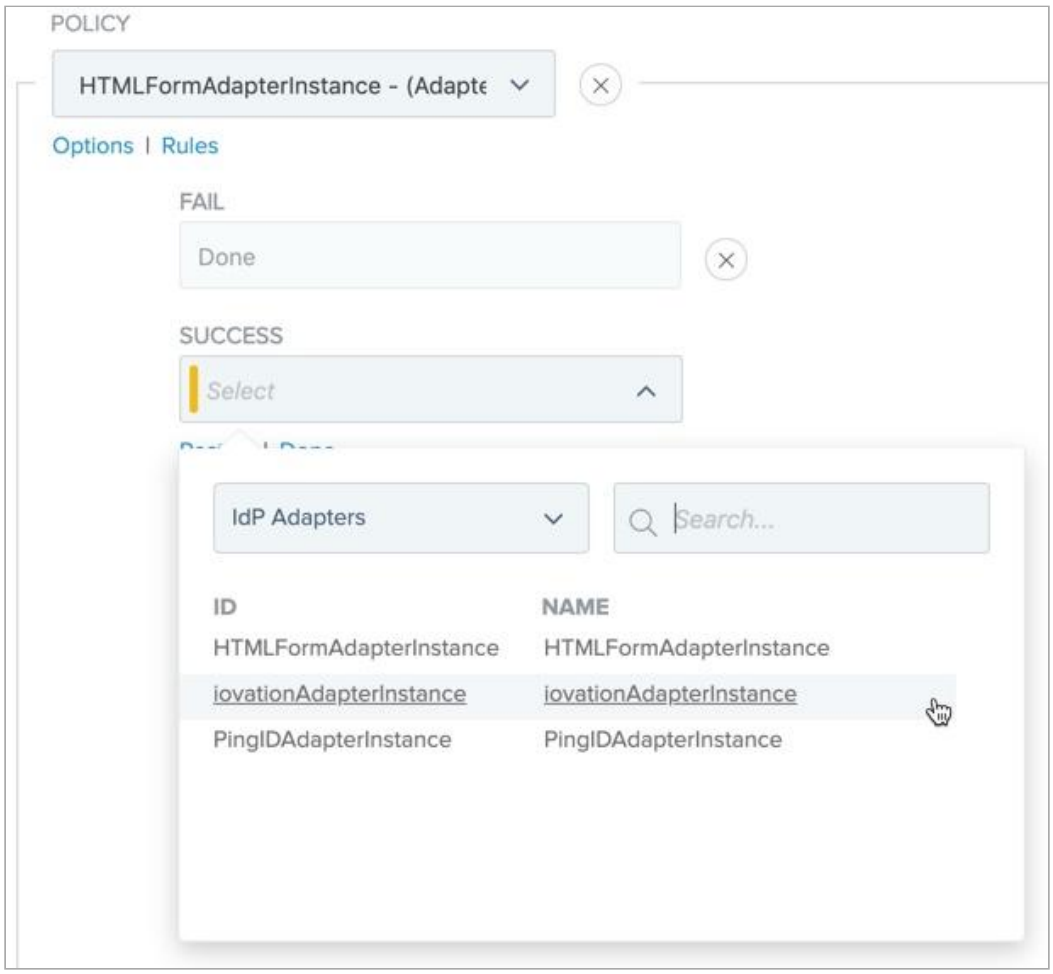
About this task

These steps are designed to help you add to an existing authentication policy. You can find general information about configuring authentication policies in Policies in the PingFederate documentation.

Steps

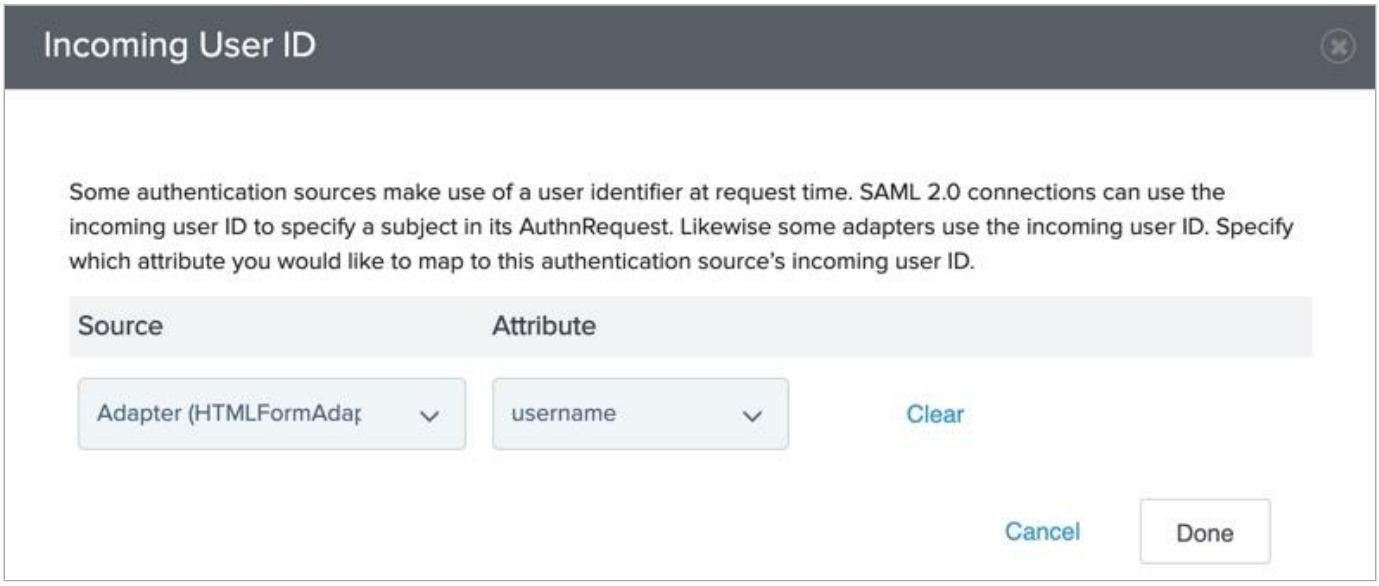
1. Sign on to the PingFederate administrative console.
2. Go to **Authentication > Policies** and either open an existing authentication policy, or create a new one.

   Learn more in Defining authentication policies in the PingFederate documentation.
3. In the **Policy** section, in the **Select** list, select an iovation IdP Adapter instance.



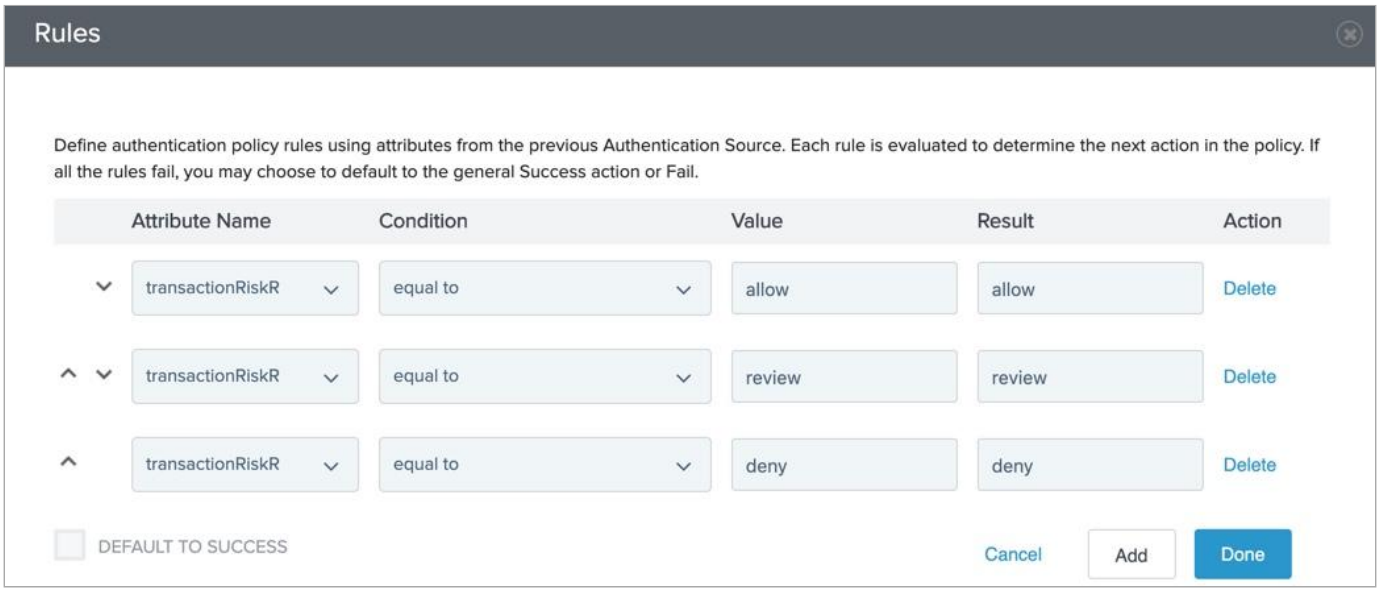
4. Map the user ID into the iovation IdP Adapter instance:

Copyright © 2025 Ping Identity Corporation



1. Under the iovation IdP Adapter instance, click **Options**.
2. On the **Options** modal, in the **Source** list, select a previous authentication source that collects the user ID.
3. In the **Attribute** list, select the user ID. Click **Done**.

5. Define policy paths based on risk results:



1. Under the iovation IdP Adapter instance, click **Rules**.
2. On the **Rules** modal, in the **Attribute Name** list, select **transactionRiskResult**.
3. In the **Condition** list, select **equal to**.
4. In the **Value** field, enter `allow`, `review`, or `deny`.
5. In the **Result** field, enter a name.

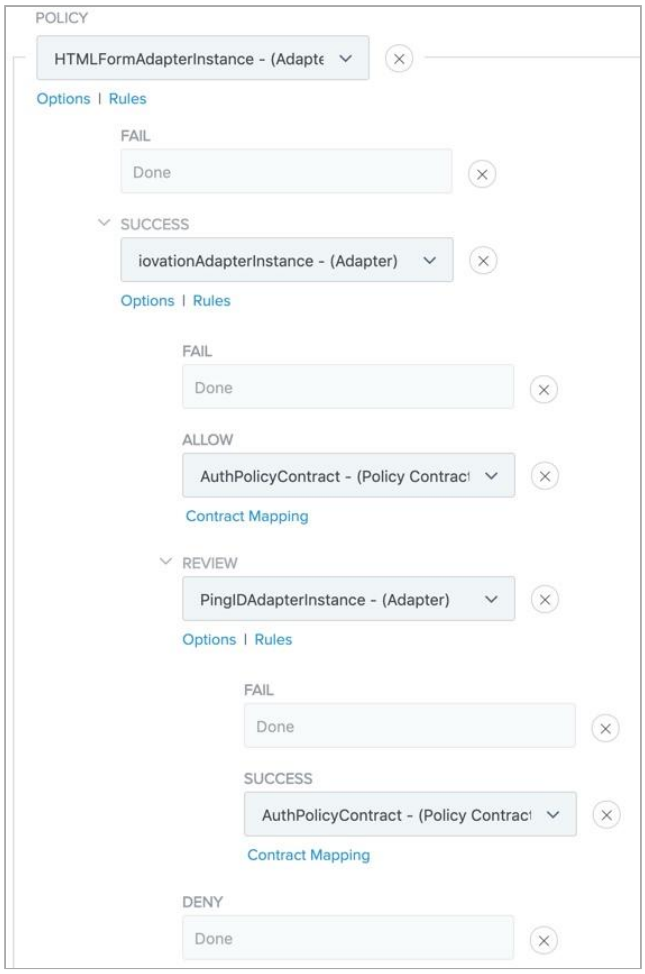
   This appears as a new policy path that branches from the authentication source.

Copyright © 2025 Ping Identity Corporation

6. To add more authentication paths, click **Add** and repeat steps a - d.
7. Clear the **Default to success** checkbox.
8. Click **Done**.

6. Configure each of the authentication paths, including **Fail**, **Success**, and the paths that you defined in the **Rules** modal.

> **Note**
>
> In case the iovation API is unreachable or returns an error, you can allow users to continue to sign on by satisfying stricter authentication requirements.
> You can do this in your authentication policy by setting the **Fail** outcome of the iovation IdP Adapter instance to point to a second authentication factor, as shown in the following example.
> Alternately, you can do this in your iovation IdP Adapter instance by setting the **Failure Mode** as shown in Configuring an adapter instance.



7. Click **Done**.
8. On the **Policies** page, click **Save**.

Release notes

Copyright © 2025 Ping Identity Corporation

Changelog

This list describes the changes made in each version of the iovation Integration Kit.

iovation Integration Kit 1.0.1 - May 2022

- Standardized the directory file structure.

iovation Integration Kit 1.0 - October 2019

- Initial release.
- Added support for iovation FraudForce.
- Added support for sending device profiles and transaction data to the iovation API.
- Added support for retrieving risk assessment results and transaction data from the iovation API.
- Added the ability to include device profiling in any browser-based authentication source.
- Added the **Iovation API Base URL** field to direct requests to the production or pre-production iovation API.
- Added the **Failure Mode** and **Fallback Risk Result Value** fields to handle failed risk result requests.
- Added the **API Request Timeout** and **Connection Timeout** fields.
- Added the **Proxy Settings**, **Custom Proxy Host**, and **Custom Proxy Port** fields to override PingFederate system-default proxy settings.
- Added the **Verify HTTPS Hostname** field to skip hostname verification.

Known issues and limitations

The following are known issues or limitations for the iovation Integration Kit.

Known issues

There are no known issues.

Known limitations

There are no known limitations.

Download manifest

The following files are included in the iovation Integration Kit `.zip` archive:

- `Legal.pdf`: Copyright and license information.

Copyright © 2025 Ping Identity Corporation

- `dist/pingfederate/server/default` : Contains the integration files.
  - `deploy` : Contains the Java libraries.
    - `pf-iovation-adapter-<version>.jar` : A JAR file that contains the iovation IdP Adapter.
  - `conf` : A directory that contains the HTML template and script that send device data to the iovation API.
    - `language-packs` : Contains files with customizable user-facing messages.
      - `iovation-messages.properties` : A variable file that customizes the messages that appear on the template file.
    - `template` : Contains user-facing HTML template files.
      - `iovation.adapter.template.html` : A sign-on redirect page that triggers the script file.
      - `assets` : Contains functional scripts and files used by the template.
        - `css` : Contains the template file style sheets.
          - `iovation.css` : A CSS file that customizes the appearance of the template file.
        - `scripts` : Contains script files used to collect and send information.
          - `iovation_device_profiling.js` : A JavaScript script that fetches the latest iovation JavaScript and collects device data.
          - `iovation_adapter_default.js` : A JavaScript script that configures the iovation device profiling script. Used with the default device profiling method.
          - `iovation_adapter_custom.js` : A JavaScript script that configures the iovation device profiling script and prepares blackbox cookies. Used with the custom device profiling method.
          - `iovation_spinner.js` : A JavaScript script that presents a spinner animation while user data is processed and sent to the iovation API.

Copyright © 2025 Ping Identity Corporation

Jamf Integration Kit

The Jamf Integration Kit allows PingFederate to integrate with Jamf Pro for secure device management. This allows you to make dynamic policy decisions based on the security posture of a user's Apple device.

For example, you can:

- Deny access for users who have devices that are not managed by Jamf Pro.
- Deny access to a user whose current device has been compromised or rooted.
- Allow access for users that only have devices that are in compliance.
- Branch your authentication policy based on device attributes, such as operating system.

Components

Jamf IdP Adapter

- Allows PingFederate to get the security posture for the authenticating user's current device.

Intended audience

This document is intended for PingFederate administrators.

If you need help during the setup process, see the following resources:

- The X.509 Certificate Integration Kit documentation
- The following sections of the Jamf documentation:
  - Managing Mobile Devices
  - Managing Computers
- The following sections of the PingFederate documentation:
  - Managing IdP adapters
  - Authentication policies

System requirements

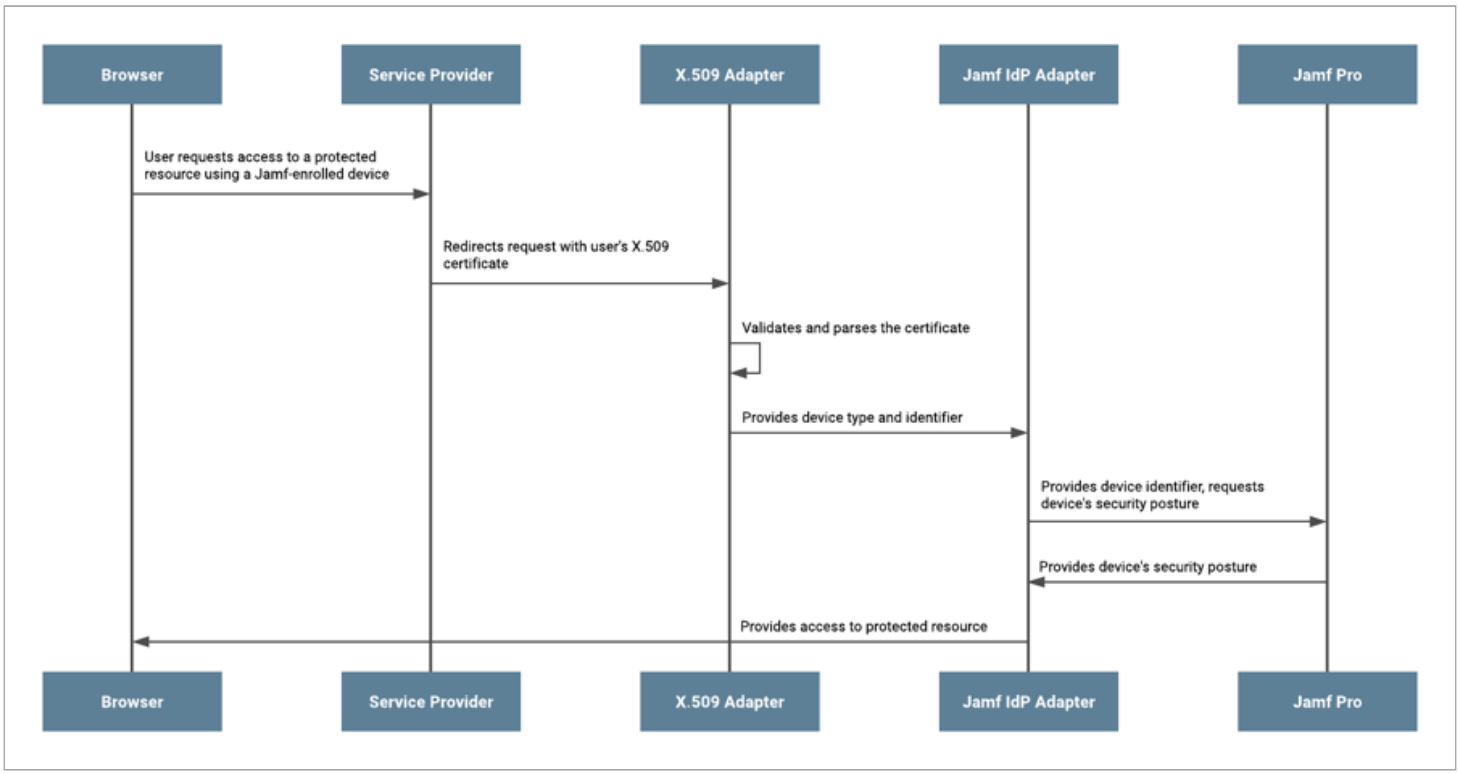
- PingFederate 9.3 or later
- A Jamf Pro account (cloud or on-premises)
- An Apple mobile device managed by Jamf Pro, such as an Apple computer, iPhone, iPad, or Apple TV

Overview of the SSO flow

With the Jamf Integration Kit, PingFederate parses identifying attributes from the X.509 certificate on the user's Apple device. The Jamf IdP Adapter uses these attributes to get the device's security posture from Jamf Pro.

Copyright © 2025 Ping Identity Corporation

The following figure illustrates a single sign-on scenario in which PingFederate retrieves the security posture of a user's device during authentication.



Description

1. The user initiates sign on with the service provider using a device that is enrolled with Jamf Pro.
2. The SP redirects the request to PingFederate. The browser provides the user's X.509 certificate.
3. The PingFederate X.509 Certificate identity provider Adapter validates the certificate against a specified list of issuers or the server's list of trusted certificate authorities, then parses the device information from the certificate.
4. The X.509 Certificate IdP Adapter provides the device type (mobile device or computer) and device identifier to the Jamf IdP Adapter.
5. The Jamf IdP Adapter provides the device identifier to Jamf Pro and requests the device's security posture.
6. Jamf Pro returns the device's security posture and a collection of other attributes.
7. PingFederate completes the sign-on flow or branches the authentication policy to a different result depending on the security posture result.

Setup

Creating API credentials in Jamf Pro

To allow PingFederate to communicate with Jamf Pro, create a user account with read privileges.

Copyright © 2025 Ping Identity Corporation

Steps

1. Sign on to Jamf Pro as an administrator.
2. Go to **Settings > System Settings > Jamf Pro user Accounts & Groups**.
3. Click **New**.
4. Select **Create Standard Account**. Click **Next**.
5. On the **Account** tab, enter a username and password.

> **Note**
>
> You will use these values in Configuring an adapter instance.

6. From the **Privilege Set** list, select **Custom**.
7. From the **Access Status** list, select **Enabled**.
8. On the **Privileges** tab, in the **Jamf Pro Server Objects** section, select **READ** for the following:
    - **Advanced Computer Searches**
    - **Advanced Mobile Device Searches**
    - **Computers**
    - **Mobile Devices**
9. Click **Save**.

Certificate requirements

For PingFederate to get a device's security posture, each device must provide a device identifier. One way to accomplish this is by configuring SSL certificates on each device.

> **Important**
>
> The Jamf IdP Adapter requires a device identifier attribute (such as `deviceId`). The adapter uses this attribute to request the device's security posture from Jamf Pro. If you can make the device identifier attribute available in the PingFederate authentication policy without using certificates, skip this topic and Setting up the X.509 Certificate Integration Kit.

As described in Overview of the SSO flow, the PingFederate X.509 Certificate IdP Adapter reads information from a user certificate provided through the browser.

Based on the specifics of your environment, you must determine a process for generating certificates and making them available on the enrolled devices.

The following describes the information that needs to be included in the certificate.

Copyright © 2025 Ping Identity Corporation

Device identifier and device type attributes

To use the Jamf Integration Kit, each device must have a certificate that includes a one of the following unique device identifiers:

- `deviceId`
- `serialnumber`
- `macaddress`
- `udid`

Optionally, you can also include a `device` attribute with a value of `computers` or `mobiledevices`. This identifies the type of device, and helps the Jamf IdP Adapter determine which Jamf Pro API to query. If the device type is not available, the adapter queries both APIs.

The X.509 Certificate IdP Adapter checks for the device identifier and device type attributes within Subject Alternative Name portion of the certificate. Specifically, the `otherName` part of `subjectAltName`.

Example certificate contents

As a source when generating a certificate, you might use a `.cnf` file similar to the following:

```
[ req ]
default_bits       = 2048
distinguished_name = req_distinguished_name
req_extensions     = req_ext
[ req_distinguished_name ]
countryName                = Country Name (2 letter code)
stateOrProvinceName        = State or Province Name (full name)
localityName               = Locality Name (eg, city)
organizationName           = Organization Name (eg, company)
commonName                 = Common Name (e.g. server FQDN or YOUR name)
[ req_ext ]
subjectAltName = @alt_names
[alt_names]
otherName.1=2.16.76.1.3.4;UTF8:deviceId=18
otherName.2=2.16.76.1.3.4;UTF8:device=computers
```

The last two lines define the device identifier and device type.

Certificate selection

When you finish setting up the Jamf Integration Kit, your users might be prompted to select the appropriate certificate during sign on. For the best user experience, we recommend that you configure automatic certificate selection. The approach you must use depends on your environment, devices, and browsers.

Setting up the X.509 Certificate Integration Kit

The Jamf Integration Kit requires the X.509 Certificate IdP Adapter to get device information from the device's client certificate.

Copyright © 2025 Ping Identity Corporation

Steps

1. Follow the steps in Deploying the integration files in the X.509 Certificate Integration Kit documentation.
2. Configure an adapter instance.

   Follow the general steps in Configuring an adapter instance in the X.509 Certificate Integration Kit, with the following details:

   1. On the **IdP Adapter** tab, in the **Advanced Fields** section, select the **Include Subject Alternative Name (SAN)** check box.
   2. On the **Extended Contract** tab, check that the device identifier and device type is available in the core or extended contract.

      In the Jamf IdP Adapter configuration, you can specify that the device identifier is contained in one of the following attributes:

      - `deviceId`
      - `macAddress`
      - `serialNumber`
      - `udid`

      Depending on your certificate configuration, you might need to add one of these attributes to the extended contract.

      Optionally, you can also include an attribute that contains the device type. The value must be `computers` or `mobiledevices`.

Deploying the integration files

To get started with the integration, deploy the Jamf Integration Kit files to your PingFederate directory.

About this task

> **Note**
>
> If you operate PingFederate in a cluster, the following steps refer to the console node.

Steps

1. Download the Jamf Integration Kit `.zip` archive from the Ping Identity Integration Directory.
2. Stop PingFederate.
3. If you are upgrading an existing deployment, delete `pf-jamf-adapter-<version>.jar` from `<pf_install>/pingfederate/server/default/deploy`.
4. Extract the `.zip` archive and merge the contents of the `dist` directory with your `<pf_install>/pingfederate` directory.
5. Start PingFederate.
6. If you operate PingFederate in a cluster, repeat steps 2-5 for each engine node.

Copyright © 2025 Ping Identity Corporation

Configuring an adapter instance

Configure the Jamf IdP Adapter to determine how PingFederate communicates with Jamf Pro.

Steps

1. In the PingFederate administrative console, create a new IdP adapter instance:

   Choose from:

   - For PingFederate 10.1 or later: go to **Authentication > Integration > IdP Adapters**. Click **Create New Instance**.
   - For PingFederate 10.0 or earlier: go to **Identity Provider > Adapters**. Click **Create New Instance**.

2. On the **Type** tab, set the basic adapter instance attributes.
   1. In the **Instance Name** field, enter a name for the adapter instance.
   2. In the **Instance ID** field, enter a unique identifier for the adapter instance.
   3. From the **Type** list, select **Jamf IdP Adapter**. Click **Next**.
3. **Optional:** On the **IdP Adapter** tab, in the **Jamf API Response Mappings** section, map attributes from the Jamf Pro response to the attribute contract. These attributes become available in your PingFederate authentication policy.

   > **Tip**
   >
   > If you use Computer Extension Attributes or Mobile Device Extension Attributes in Jamf Pro, use this table to include your extension attributes in the PingFederate attribute contract.

   > **Note**
   >
   > The `isManaged` and `isMDMCapable` attributes are already included in the core contract.

   1. Click **Add a new row to 'Jamf API Response Mappings'**.
   2. In the **Local Attribute** field, enter a name for the new attribute.
   3. In the **Jamf API Response Mapping** field, enter the JSON Pointer syntax for the value of the matching PingOne Verify attribute. For help, see JSON Pointer syntax reference.
   4. In the **Action** column, click **Update**.
   5. To add more attributes, repeat steps a-d.
4. On the **IdP Adapter** tab, configure the adapter instance by referring to Jamf IdP Adapter settings reference. Click **Next**.
5. On the **Actions** tab, test your connection to Jamf Pro. Resolve any issues that are reported, and then click **Next**.
6. On the **Extended Contract** tab, add any local attributes that mapped in the the **Jamf API Attribute Mappings** section. Click **Next**.
7. On the **Adapter Attributes** tab, set pseudonym and masking options as shown in Set pseudonym and masking options in the PingFederate documentation. Click **Next**.

Copyright © 2025 Ping Identity Corporation

8. On the **Adapter Contract Mapping** tab, configure the contract fulfillment details for the adapter as shown in Define the IdP adapter contract in the PingFederate documentation. Click **Next**.
9. On the **Summary** tab, check and save your configuration:

   Choose from:

   - For PingFederate 10.1 or later: click **Save**.
   - For PingFederate 10.0 or earlier: click **Done**. On the **Manage IdP Adapter Instances** tab, click **Save**.

Jamf IdP Adapter settings reference

Field descriptions for the Intune IdP Adapter configuration screen.

Table 1. Standard fields

Field	Description
API Username	The username that you created in Creating API credentials in Jamf Pro.
API Password	The password that you created in Creating API credentials in Jamf Pro.
API Hostname	The hostname for the Jamf Pro server. For example: • company-name.jamfcloud.com • jamf-pro.companyname.com:8443
Device Identifier	The attribute used to uniquely identify the user's authentication device. The adapter gets this attribute from the device's X.509 certificate and uses it to query the Jamf Pro for the device's security posture. This attribute must be available in the core or extended of your X.509 Certificate IdP Adapter instance. The default selection is **Serial Number**.
Device Type Attribute	The attribute that identifies whether the device is a computer or mobile device. The adapter gets this attribute from the device's X.509 certificate and uses it to determine which Jamf Pro API to query. The value of the attribute must be `computers` or `mobiledevices`. If the device type is provided in this attribute, the value overrides the **Possible Device Types** setting. The default value is `deviceType`.

Copyright © 2025 Ping Identity Corporation

Field	Description
Possible Device Types	Determines which Jamf Pro API the adapter queries for a device's security posture. Jamf Pro provides two separate APIs, one for computers and the other for mobile devices. This setting allows you to eliminate unecessary calls to one API if your users all use the same device type (either computers or mobile devices). If a valid **Device Type Attribute** is available, that value overrides this setting. The default selection is **Computers**.
Failure Mode	Determines whether the adapter blocks the user's sign-on attempt when it cannot find the device or the connection to Jamf Pro fails. The default selection is **Bypass authentication**.
API Request Timeout	The amount of time in milliseconds that PingFederate allows when establishing a connection with Jamf Pro or waiting for a response to a request. A value of 0 disables the timeout. The default value is `5000`.
Proxy Settings	Defines proxy settings for outbound HTTP requests. The default value is **System Defaults**.
Custom Proxy Host	The proxy server host name to use when **Proxy Settings** is set to **Custom**. This field is blank by default.
Custom Proxy Port	The proxy server port to use when **Proxy Settings** is set to **Custom**. This field is blank by default.

JSON Pointer syntax reference

JavaScript Object Notation (JSON) Pointer defines a syntax for identifying a specific value within a JSON payload. Using the sample payload and JSON Pointer examples below, identify the attributes that you want to use to populate your attribute contract.

For a complete technical description of JSON Pointer syntax, see JavaScript Object Notation (JSON) Pointer on ietf.org.

Example Jamf Pro JSON payload for computers

> **Note**
>
> Portions of the following payload have been removed for brevity. You can find a complete example in Find computers by ID in the Jamf Pro API documentation.

Copyright © 2025 Ping Identity Corporation

```
{
    "computer": {
        "general": {
            "id": 1,
            "name": "Admins iMac",
            "ip_address": "10.1.1.1",
            "last_reported_ip": "192.0.0.1",
            "udid": "55900BDC-347C-58B1-D249-F32244B11D30",
            "jamf_version": "9.99.0-t1494340586",
            "platform": "Mac",
            "remote_management": {
                "managed": true,
                "management_username": "casperadmin"
            },
            "mdm_capable": true,
            "mdm_capable_users": {
                "mdm_capable_user": "string"
            },
            "management_status": {
                "enrolled_via_dep": true,
                "user_approved_enrollment": true,
                "user_approved_mdm": true
            },
            "site": {
                "id": 0,
                "name": "None"
            }
        },
        "location": {
            "username": "JBetty",
            "realname": "Betty Jackson",
            "real_name": "Betty Jackson",
            "email_address": "jbetty@company.com",
            "position": "Systems Engineer",
            "phone": "123-555-6789",
            "phone_number": "123-555-6789",
            "department": "Sales Staff",
            "building": "New York Office",
            "room": 1159
        },
        "groups_accounts": {
            "computer_group_memberships": [
                {
                    "group": "All Managed Clients"
                }
            ],
            "local_accounts": [
                {
                    "user": {
                        "name": "_amavisd",
                        "realname": "AMaViS Daemon",
                        "uid": 83,
                        "home": "/var/virusmails",
                        "home_size": "-1MB",
                        "home_size_mb": -1,
                        "administrator": false,
                        "filevault_enabled": false
                    }
                }
            ]
```


Copyright © 2025 Ping Identity Corporation

```
        },
        "configuration_profiles": [
            {
                "size": 1,
                "configuration_profile": {
                    "id": 1,
                    "name": "string",
                    "uuid": "string",
                    "is_removable": false
                }
            }
        ]
    }
}
```

Table 1. JSON Pointer syntax

Description	JSON Pointer	Example value
Device IP address	`/computer/general/ip_address`	`10.1.1.1`
User's name	`/computer/location/realname`	`Betty Jackson`
User's location	`/computer/location/building`	`New York Office`
User's administrator status	`/computer/groups_accounts/local_accounts/0/user/administrator`	`false`

Note

To populate an attribute with the entire JSON response, leave the attribute mapping field blank.

Copyright © 2025 Ping Identity Corporation

Example Jamf Pro JSON payload for mobile devices

Copyright © 2025 Ping Identity Corporation

Note

Portions of the following payload have been removed for brevity. You can find a complete example in Find mobile devices by ID in the Jamf Pro API documentation.


```
{
    "mobile_device": {
        "general": {
            "id": 1,
            "display_name": "Tinas iPad",
            "device_name": "Tinas iPad",
            "name": "Tinas iPad",
            "os_type": "iOS",
            "udid": "270aae10800b6e61a2ee2bbc285eb967050b5984",
            "phone_number": "123-555-6789",
            "ip_address": "192.0.0.1",
            "managed": true,
            "supervised": true
        },
        "location": {
            "username": "JBetty",
            "realname": "Betty Jackson",
            "real_name": "Betty Jackson",
            "email_address": "jbetty@company.com",
            "position": "Systems Engineer",
            "phone": "123-555-6789",
            "phone_number": "123-555-6789",
            "department": "Sales Staff",
            "building": "New York Office",
            "room": 1159
        },
        "security": {
            "data_protection": true,
            "block_level_encryption_capable": true,
            "file_level_encryption_capable": true,
            "passcode_present": true,
            "passcode_compliant": true,
            "passcode_compliant_with_profile": true,
            "passcode_lock_grace_period_enforced": "Not Available",
            "hardware_encryption": "string",
            "activation_lock_enabled": true,
            "jailbreak_detected": "Normal"
        },
        "network": {
            "home_carrier_network": "Verizon",
            "phone_number": 5555555555
        },
        "mobile_device_groups": [
            {
                "size": 1,
                "mobile_device_group": {
                    "id": 1,
                    "name": "string"
                }
            }
        ]
    }
}
```


Copyright © 2025 Ping Identity Corporation

Table 2. JSON Pointer syntax

Description	JSON Pointer	Example value
Device IP address	`/mobile_device/general/ip_address`	`192.0.0.1`
User's name	`/mobile_device/location/realname`	`Betty Jackson`
User's location	`/mobile_device/location/building`	`New York Office`
Device jailbreak status	`/mobile_device/security/jailbreak_detected`	`Normal`

> **Note**
>
> To populate an attribute with the entire JSON response, leave the attribute mapping field blank.

Adding Jamf security posture results to your authentication policy

By modifying your PingFederate authentication policy to include the `isManaged` or `isMDMCapable` results from Jamf Pro, you can dynamically control access to resources based on the device's security posture.

About this task

These steps are designed to help you add to an existing authentication policy. For general information about configuring authentication policies, see PingFederate authentication API in the PingFederate documentation.

Steps

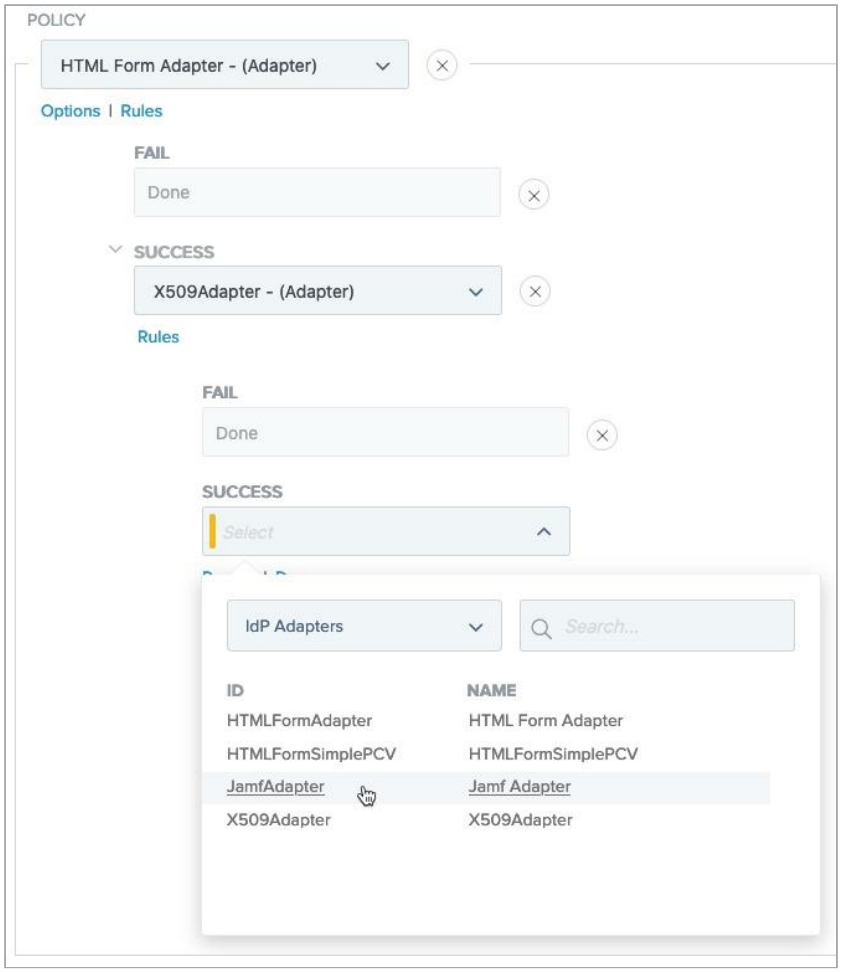
1. In the PingFederate administrative console, go to the **Policies** tab.

   Choose from:

   - For PingFederate 10.1 or later: go to **Authentication > Policies > Policies**.
   - For PingFederate 10.0 or earlier: go to **Identity Provider > Authentication Policies > Policies**.

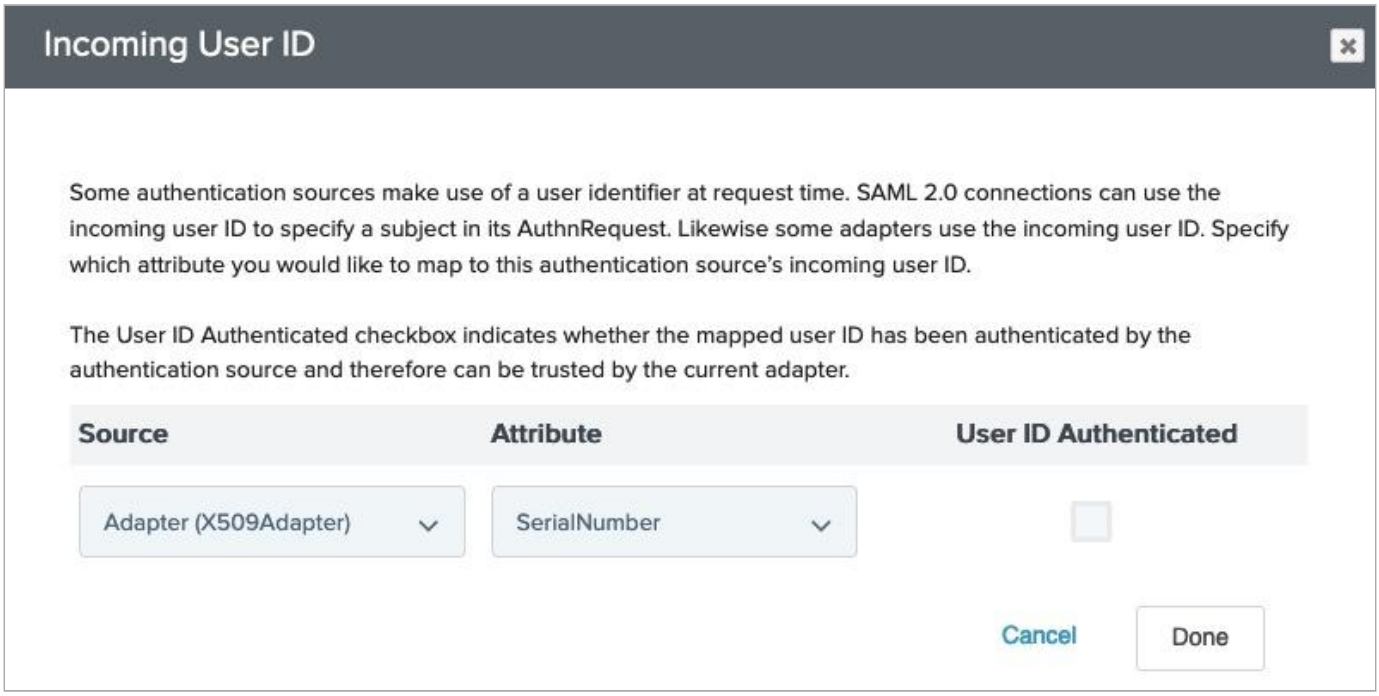
2. Select the **IdP Authentication Policies** check box.
3. Open an existing authentication policy, or click **Add Policy**. See Defining authentication policies in the PingFederate documentation.
4. In the **Policy** area, from the **Select** list, select a Jamf IdP Adapter instance.

Copyright © 2025 Ping Identity Corporation



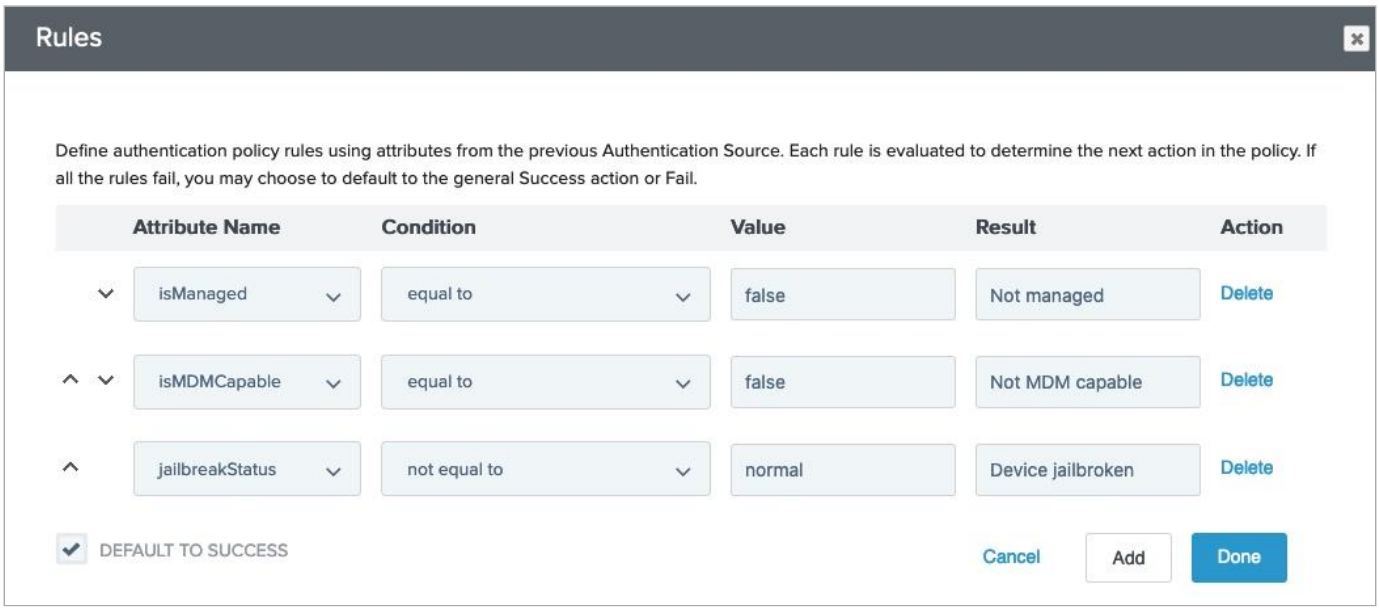
5. Map the attribute that contains the device identifier (shown here as **SerialNumber**) from your X.509 Certificate IdP Adapter instance into the Jamf IdP Adapter instance.

Copyright © 2025 Ping Identity Corporation



1. Under the Jamf IdP Adapter instance, click **Options**.
2. On the **Options** dialog, from the **Source** list, select your X.509 Certificate IdP Adapter instance.
3. From the **Attribute** list, select attribute that matches the **Device Identifier** that you selected in your adapter configuration. Click **Done**.

6. Define policy paths based on the security posture attribute `isCompliant`.

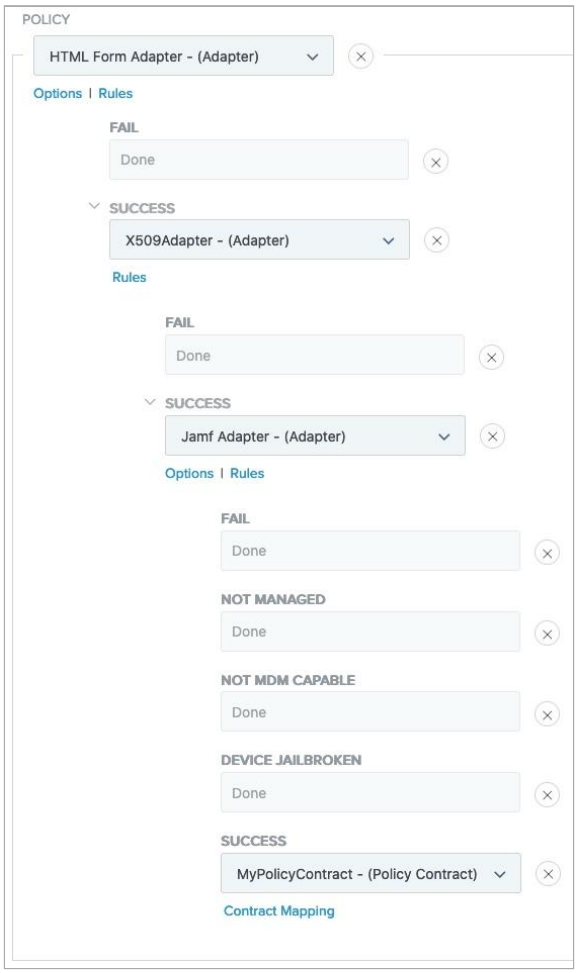


1. Under the Jamf IdP Adapter instance, click **Rules**.
2. On the **Rules** dialog, in the **Attribute Name** list, select **isCompliant**.
3. In the **Condition** list, select **equal to**.

Copyright © 2025 Ping Identity Corporation

4. In the **Value** field, enter `true` or `false`.
5. In the **Result** field, enter a name. This appears as a new policy path that branches from the authentication source.
6. **Optional:** Repeat steps a-e for `isMDMCapable` or any attributes that you mapped in the **Jamf API Attribute Mappings** table in the adapter configuration.
7. **Optional:** Clear the **Default to success** check box to ensure the authentication flow follows one of the paths that you defined.
8. Click **Done**.

7. Configure each of the authentication paths.



8. Click **Done**.
9. In the Policies window, click **Save**.

Authentication API Support

You can use the PingFederate authentication API to integrate the Jamf IdP Adapter into your application.

Copyright © 2025 Ping Identity Corporation

The PingFederate Authentication API provides access to the current state of the authentication flow as a user steps through the PingFederate authentication policy.

For more information, see PingFederate authentication API in the PingFederate documentation. To integrate the Jamf IdP Adapter into your authentication flow, configure your application based on the states, actions, and models available in the PingFederate Authentication API Explorer. For help, see Exploring the Authentication API in the PingFederate documentation.

Release notes

Changelog

The following is the change history for the Jamf Integration Kit.

Jamf Integration Kit 2.0.1 – June 2022

- Updated the Authentication API `.JAR` file version.

Jamf Integration Kit 2.0 – April 2022

- Added support for bearer tokens for authenticating API calls to Jamf. Removed support for Basic Authentication.
- Added the `User-Agent` header to all API requests.

Jamf Integration Kit 1.0 – March 2021

- Initial release.
- Added the ability to query Jamf Pro for a device's security posture.
- Added the ability to get the device identifier and device type from chained attributes in the PingFederate authentication policy.
- Added the ability to map any Jamf API response attribute to an attribute in the PingFederate authentication policy.
- Added the ability to control how the adapter handles sign-on attempts when errors occur.
- Added the ability to test the connection to Jamf Pro.
- Added support for the PingFederate authentication API.
- Added settings for API connection and request timeouts.
- Added settings to override the PingFederate system-default proxy settings.

Known issues and limitations

The following are known issues or limitations for the Jamf Integration Kit.

Known issues

There are no known issues.

Copyright © 2025 Ping Identity Corporation

Known limitations

- The Jamf IdP Adapter queries Jamf Pro with a unique device identifier (such as serial number) as described in Simple Mobile Device Searches and Simple Computer Searches in the Jamf Pro documentation. The adapter does not support advanced searches using criteria and operators.

Download manifest

The following files are included in the Jamf Integration Kit `.zip` archive:

- `Legal.pdf` – copyright and license information
- `dist` – contains the integration files
  - `pf-jamf-adapter-<version>.jar` – JAR file that contains the Jamf IdP Adapter.

Copyright © 2025 Ping Identity Corporation

Java Integration Kit

The Java Integration Kit allows PingFederate to integrate with identity provider (IdP) and service provider (SP) applications for single sign-on (SSO).

> **Tip**
>
> For new integrations, try the Agentless Integration Kit. It can integrate with a variety of platforms using a modern RESTful approach, and it doesn't require you to integrate agent software into your application.

Components

OpenToken adapter and agent

- Allows developers to integrate their Java applications with a PingFederate server acting as an IdP or an SP:
  - On the IdP side, the integration kit allows an IdP server to receive user attributes from a Java IdP application.
  - On the SP side, the integration kit allows a Java SP application to receive user attributes from the SP server.

Sample applications

- The Integration Kit distribution also contains sample IdP and SP applications. These applications can be installed quickly to test OpenToken processing and review a working demonstration of end-to-end SSO and single logout (SLO).

  The distribution includes source code and supporting files that application developers can modify or use as a reference.

Intended audience

This document is intended for PingFederate administrators.

If you need help during the setup process, you can find more information in the following sections of the PingFederate documentation:

- Managing IdP adapters
- Managing SP adapters
- Authentication policies
- OpenToken Adapter

System requirements

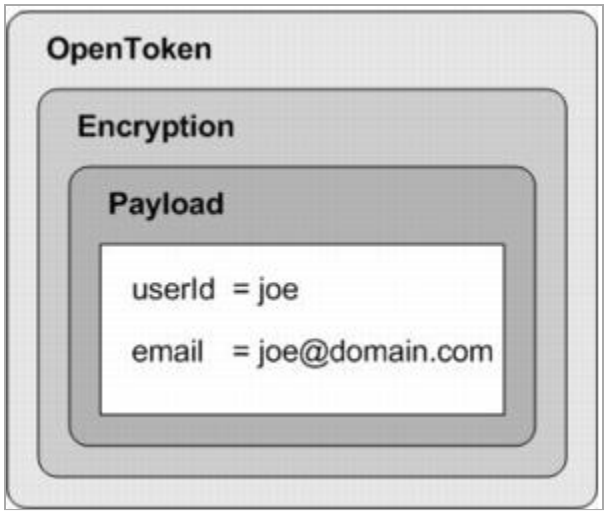
- PingFederate 11.3 or later.
- Java SE Runtime Environment 1.8 or later on the agent side
- To use strong Advanced Encryption Standard (AES) encryption with a key size of more than 128 bits, you must install the Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files in your Java Development Kit (JDK).

Copyright © 2025 Ping Identity Corporation

Overview of the OpenToken Adapter

The OpenToken Adapter is an open-standard secure token used to pass user information between an application and PingFederate. The OpenToken is passed through the user's browser as a URL query parameter, HTTP cookie, or form POST.

Data within the OpenToken is a set of key and value pairs. The data is encrypted using common encryption algorithms, as shown in the following illustration:



You can find more details in OpenToken Adapter in the PingFederate documentation.

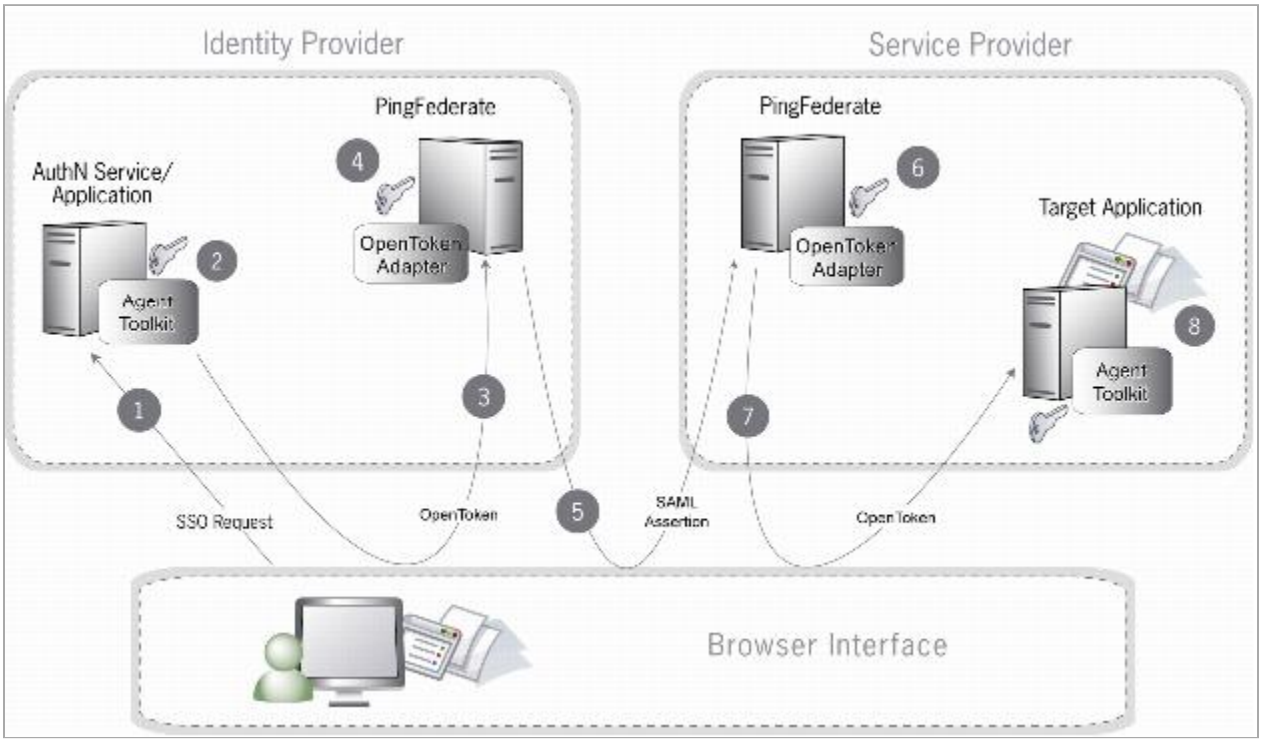
Overview of the SSO flow

The Java Integration Kit consists of two parts:

- The OpenToken Adapter that runs within the PingFederate server
- The Agent Toolkit for Java that's in the Java application

The following figure shows a basic IdP-initiated SSO scenario in which PingFederate federation servers using the Java Integration Kit exist on both sides of the identity federation:

Copyright © 2025 Ping Identity Corporation



Sequence

1. A user initiates an SSO transaction.
2. The IdP application inserts user attributes into the Agent Toolkit for Java, which encrypts the data internally and generates an `OpenToken`.
3. A request containing the `OpenToken` is redirected to the PingFederate IdP server.
4. The server invokes the OpenToken IdP Adapter that retrieves the `OpenToken`, then decrypts, parses, and passes the user attributes to the PingFederate IdP server. The PingFederate IdP server then generates a Security Assertion Markup Language (SAML) assertion.
5. The SAML assertion is sent to the service provider (SP) site.
6. The PingFederate SP server parses the SAML assertion and passes the user attributes to the OpenToken SP Adapter. The Adapter encrypts the data internally and generates an `OpenToken`.
7. A request containing the OpenToken is redirected to the SP application.
8. The Agent Toolkit for Java decrypts and parses the OpenToken and makes the user attributes available to the SP Application.

> **Note**
>
> You can configure PingFederate to look up additional attributes from an IdP or SP datastore. Learn more in Datastores in the PingFederate documentation.

Setup

Copyright © 2025 Ping Identity Corporation

Updating the OpenToken Adapter

The Java Integration Kit relies on the OpenToken Adapter that is distributed with the Java Integration Kit. Update the OpenToken Adapter to get the latest feature and security updates.

Steps

1. Download the Java Integration Kit `.zip` archive from the **Add-ons** tab of the PingFederate downloads page.
2. Stop PingFederate.
3. Delete the `opentoken-adapter-<version>.jar` file from your `<pf_install>/pingfederate/server/default/deploy` directory.
4. From the Java Integration Kit `.zip` archive, copy the contents of `dist/pingfederate` to your `<pf_install>/pingfederate` directory.

   > **Note**
   >
   > The `commons-collections`, `commons-beanutils`, and `commons-logging` libraries are provided as a convenience. Install them only if they're not already contained in the application `CLASSPATH`.

5. Start PingFederate.
6. If you operate PingFederate in a cluster, repeat steps 2 - 5 for each engine node.

Custom application setup

You can configure the Java Integration Kit to integrate PingFederate with your identity provider (IdP) or service provider (SP) application.

This section includes adapter configuration instructions for PingFederate administrators as well as application configuration instructions for Java developers.

If you want to see a working demonstration of the PingFederate before integrating your own applications, refer to Sample application setup.

Configuring an OpenToken SP Adapter instance

Configure the OpenToken Adapter to determine how PingFederate communicates with your service provider application.

Steps

1. In the PingFederate administrative console, go to **Authentication > Integration > IdP Adapters**. Click **Create new Instance**.
2. On the **Type** tab, set the basic adapter instance attributes.
   1. In the **Instance Name** field, enter a name for the adapter instance.
   2. In the **Instance ID** field, enter a unique identifier for the adapter instance.
   3. In the **Type** list, select **OpenToken Adapter**. Click **Next**.

Copyright © 2025 Ping Identity Corporation

3. On the **Instance Configuration** tab, configure the adapter instance by referring to Configuring an OpenToken SP Adapter instance in the PingFederate documentation. Click **Next**.

   > **Note**
   >
   > If you use the application to protect multiple sites on the same domain, in the OpenToken Adapter instance configuration, select **None** for SameSite Cookie and select the **Secure Cookie** checkbox.

4. Export the configuration file:

   1. On the **Actions** tab, click **Download**, and then click **Export**.
   2. Save the `agent-config.txt` file. Click **Next**.

5. On the **Extended Contract** tab, add any attributes that you expect to retrieve other than the SAML subject. Click **Next**.
6. On the **Target App Info** tab, enter the basic information about your SP application. Click **Next**.
7. On the **Summary** tab, check and save your configuration. Click **Save**.
8. Create or update an identity provider (IdP) connection to use the OpenToken Adapter instance as shown in Service provider SSO configuration in the PingFederate documentation.

Deploying the Java agent

About this task

If you are upgrading this integration, we strongly recommend that you reinstall the OpenToken agent in all existing applications (IdP or SP).

> **Note**
>
> As of Java Integration Kit 2.8, there are two versions of the OpenToken agent: the standard OpenToken agent and the Jakarta EE 9 OpenToken agent.

Steps

1. If you are upgrading this integration:
   1. Stop your web application.
   2. Remove the existing OpenToken agent file from your Java application server:

      Choose from:

      - If you're using the standard OpenToken agent, remove the `opentoken-agent-<version>.jar` from the `agent` folder.
      - If you're using the Jakarta EE 9 OpenToken agent, remove the `opentoken-agent-jakarta-<version>.jar` from the `agent/jakarta` folder

        > **Note**
        >
        > No code changes are required within applications when upgrading.

Copyright © 2025 Ping Identity Corporation

2. From the integration `.zip` archive, copy the version-specific `agent/opentoken-agent-<version>.jar` file to your Java application server and make it available in the `CLASSPATH` of the Java application:

   Choose from:

   - If you're using the OpenToken agent, copy the `agent/opentoken-agent-<version>.jar` file.
   - If you're using the Jakarta EE 9 OpenToken agent, copy the `agent/jakarta/opentoken-agent-jakarta-<version>.jar` file.

3. From the integration `.zip` archive, copy the necessary library file or files to your Java application server and make them available in the `CLASSPATH` of the Java application:

Option	Description
Standard OpenToken agent	- `agent/lib/commons-beanutils.jar` * - `agent/lib/commons-collections-3.2.2.jar` * - `agent/lib/commons-logging.jar`
Jakarta EE 9 OpenToken agent	- `agent/jakarta/lib/commons-beanutils.jar` * - `agent/jakarta/lib/commons-collections-3.2.2.jar` * - `agent/jakarta/lib/commons-logging.jar`

> **Note**
>
> Files marked with an asterisk (*) are only necessary if you are using the Apache commons multimap function to read and write OpenTokens.

4. If you are upgrading, restart your web application.
5. Repeat these steps as needed for additional custom applications.

   For a first-time installation, complete the integration as described in Reading and writing OpenTokens (next) and in either Identity provider applications or Service provider applications.

Reading and writing OpenTokens

You can use the Agent API to read an OpenToken from an HTTP request or and write an OpenToken to an HTTP response.

> **Note**
>
> As of Java Integration Kit 2.8, there are two versions of the OpenToken agent: the standard OpenToken agent, and the Jakarta EE 9 OpenToken agent.
> There are two new instances of sample code available for the Jakarta EE 9 OpenToken agent, which are documented below. These two instances of sample code contain new package locations.

When the agent object is instantiated, it reads the OpenToken Adapter configuration from the `agent-config.txt` file. This configuration data includes the name of the cookie that the agent object writes, as well as the key to use when encrypting a new OpenToken. If the agent object doesn't find the `agent-config.txt` file, the agent constructor will throw an exception.

To instantiate the agent object, invoke a constructor and load the configuration using one of the following methods.

Copyright © 2025 Ping Identity Corporation

Sample code - Java native multimap method (standard OpenToken agent)

```
import com.pingidentity.opentoken.TokenException;
import com.pingidentity.opentoken.generic.Agent;
import com.pingidentity.opentoken.generic.util.PingFederateMultiMap;
. . .
Agent agent = Agent.createUsingNativeMultiMap("<PATH_TO_FILE>/agent-config.txt");
```

Note

This method requires the following library to be available:

- `agent/lib/commons-logging.jar`

You can find help in Deploying the Java agent.

Sample code - Java native multimap method (Jakarta EE 9 OpenToken agent)

```
import com.pingidentity.opentoken.jakarta.TokenException;
import com.pingidentity.opentoken.jakarta.generic.Agent;
import com.pingidentity.opentoken.jakarta.generic.util.PingFederateMultiMap;
. . .
Agent agent = Agent.createUsingNativeMultiMap("<PATH_TO_FILE>/agent-config.txt");
```

Note

This method requires the following library to be available:

- `agent/jakarta/lib/commons-logging.jar`

You can find more information in Deploying the Java agent.

Sample code - Apache commons multimap method (standard OpenToken agent)

```
import java.io.IOException;
import java.io.File;
import java.io.InputStream;
import java.util.Map;
import java.util.HashMap;
import java.util.List;

import org.apache.commons.collections.MultiMap;
import org.apache.commons.collections.map.MultiValueMap;

import com.pingidentity.opentoken.Agent;
import com.pingidentity.opentoken.TokenException;
import com.pingidentity.opentoken.util.UrlHelper;
. . . .
Agent agent = new Agent("<PATH_TO_FILE>/agent-config.txt");
```


Copyright © 2025 Ping Identity Corporation

Note

This method requires the following libraries to be available:

- `agent/lib/commons-beanutils.jar`
- `agent/lib/commons-collections-3.2.2.jar`
- `agent/lib/commons-logging.jar`

You can find help in Deploying the Java agent.

Sample code - Apache commons multimap method (Jakarta EE 9 OpenToken agent)

```
import java.io.IOException;
import java.io.File;
import java.io.InputStream;
import java.util.Map;
import java.util.HashMap;
import java.util.List;

import org.apache.commons.collections.MultiMap;
import org.apache.commons.collections.map.MultiValueMap;

import com.pingidentity.opentoken.jakarta.Agent;
import com.pingidentity.opentoken.jakarta.TokenException;
import com.pingidentity.opentoken.jakarta.util.UrlHelper;
. . . .
Agent agent = new Agent("<PATH_TO_FILE>/agent-config.txt");
```

Note

This method requires the following libraries to be available:

- `agent/jakarta/lib/commons-beanutils.jar`
- `agent/jakarta/lib/commons-collections-3.2.2.jar`
- `agent/jakarta/lib/commons-logging.jar`

You can find help in Deploying the Java agent.

Identity provider applications

You can use the Java Integration Kit to integrate your identity provider (IdP) application with PingFederate for single sign-on (SSO) and single logout (SLO). This section provides implementation guidelines and code examples for Java developers.

IdP single sign-on (SSO)

When PingFederate is configured as an IdP, it needs to be able to identify a user before issuing a SAML assertion for that user. When using the OpenToken Adapter with PingFederate, this means that the PingFederate server attempts to read a cookie or query parameter containing an OpenToken and then use the values within to identify the user.

Copyright © 2025 Ping Identity Corporation

The application that starts the SSO must include an OpenToken so that PingFederate can identify the user. Use the Agent API to write an OpenToken. The API is a Java object that provides access to functionality for writing an OpenToken to a given HTTP response.

Encoding attributes

The `writeToken` method of the Agent class takes a `java.util.Map` collection of attributes and encodes them into an OpenToken, which is then written to the HTTP response.

> **Important**
>
> To generate a valid token, the collection of attributes must contain a key named `subject`.

If any errors are encountered while creating the token or writing the token out to the response, a `TokenException` is thrown.

The following example code shows how to use the writeToken method:

> **Note**
>
> This code is the same for both the Java native and Apache commons multimap methods described in Reading and writing OpenTokens.

Sample code

```
String username = (String)request.getSession().getAttribute("username");
Map<String, String> userInfo = new HashMap<String, String>();
userInfo.put(Agent.TOKEN_SUBJECT, username);
String returnUrl = "https://<PingFederate-DNS>:9031" + request.getParameter("resume");
. . . .
try {
   UrlHelper urlHelper = new UrlHelper(returnUrl);
   //See the "Reading and writing OpenTokens" topic for sample code
   //that instantiates and configures an Agent instance
   agent.writeToken(userInfo,response,urlHelper,false);
   returnUrl = (String)urlHelper.toString();
}
catch(TokenException e) {
// Handle exception
}
```

Sending multi-value attributes

The Agent Toolkit for Java allows your application to send multi-value attributes to PingFederate. Each attribute appears in its own `<AttributeValue>` element in the SAML 2.0 assertion.

Using the Agent Toolkit with Apache Commons MultiMap

Copyright © 2025 Ping Identity Corporation

Multi-value attributes are passed using the `org.apache.commons.collections.map.MultiValueMap` collection.

```
String username = (String)request.getSession().getAttribute("username");
MultiMap userInfo = new MultiValueMap();
userInfo.put(Agent.TOKEN_SUBJECT, username);
// Add an attribute GROUP with multiple values userInfo.put("GROUP", "Administrators");
userInfo.put("GROUP", "Users");
String returnUrl = "https://<PingFederate_DNS>:9031" + request.getParameter("resume");
. . . .
try {
   UrlHelper urlHelper = new UrlHelper(returnUrl);
   //see "Using the Agent API" section for sample code
   //that instantiates and configures an Agent instance
   agent.writeToken(userInfo,response,urlHelper,false);
   returnUrl = (String)urlHelper.toString();
}
catch(TokenException e) {
// Handle exception
}
```

Using the PingFederate MultiMap equivalent

```
String username = (String) request.getSession().getAttribute("username");
PingFederateMultiMap userInfo = new PingFederateMultiMap();
userInfo.put(Agent.TOKEN_SUBJECT, username);
// Add an attribute GROUP with multiple values userInfo.put("GROUP", "Administrators");
userInfo.put("GROUP", "Users");
String returnUrl = "https://<PingFederate_DNS>:9031" + request.getParameter("resume");
. . . .
try
{
    UrlHelper urlHelper = new UrlHelper(returnUrl);
    //see "Using the Agent API" section for sample code
    //that instantiates and configures an Agent instance
    agent.writeToken(userInfo, response, urlHelper, false);
    returnUrl = (String) urlHelper.toString();
}
catch (TokenException | IOException e)
{
    // Handle exception
}
```

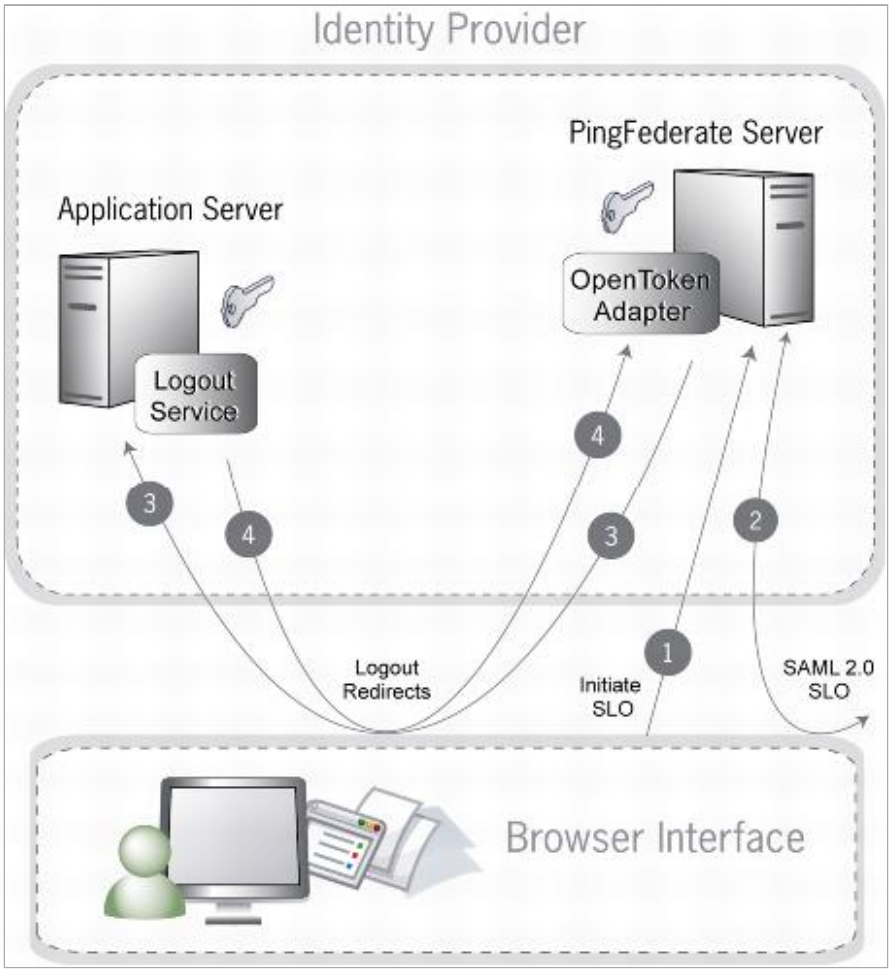
IdP single logout (SLO)

When an IdP PingFederate server receives a request for single logout (SLO), it redirects the user's browser to the logout service defined in the IdP OpenToken Adapter configuration.

The redirect URL includes an OpenToken containing the user attributes defined in the IdP OpenToken Adapter instance for the partner connection. The logout service should remove the user's session on the application server and redirect the user's browser back to the IdP PingFederate server.

The following diagram shows the flow of IdP-initiated SLO, but the architecture would also support SP-initiated SLO:

Copyright © 2025 Ping Identity Corporation



Sequence

1. User initiates a single logout request. The request targets the PingFederate server's `/idp/startSLO.ping` endpoint.
2. PingFederate sends a logout requests and receives responses for all SPs registered for the current SSO session.
3. If the application server has an SLO service configured, PingFederate redirects the request to the SLO service, which identifies and removes the user's session locally.
4. The application logout service redirects back to PingFederate to display a sign-off success page.

   If the web application doesn't have an SLO service configured, the adapter redirects back to PingFederate, which displays a sign-off success page.

Copyright © 2025 Ping Identity Corporation

Processing logout requests

An example code snippet for processing a logout request and sending it back to PingFederate through the user's browser.

```
request.getSession().invalidate();
String returnUrl = "https://<{pingfed} DNS>:9031" + request.getParameter("resume");
response.sendRedirect(returnUrl);
```

Service provider applications

You can use the Java Integration Kit to integrate your service provider (SP) application with PingFederate for single sign-on (SSO) and single logout (SLO). This section provides implementation guidelines and code examples for Java developers.

SP single sign-on (SSO)

When PingFederate is configured as an SP, it takes inbound SAML assertions and converts them to a local format (cookie or otherwise) that an application can use to create a user's session.

For an `OpenToken`, the PingFederate adapter takes the attributes and values from the SAML assertion and stores them in an `OpenToken` cookie or query parameter in the user's browser. The user is then redirected to the target application, which can identify the user from the `OpenToken` using the Agent API.

As with the IdP, you can use the Agent API to read tokens directly. The Agent API is a Java object that provides access to functionality for reading an `OpenToken` from a given HTTP request.

Decoding attributes

The `readToken` method inspects the cookie (or query parameters, depending on the configuration of the agent instance) and decodes the `OpenToken`, returning a `Collection` of attributes or `null` if no token is found or an error is encountered. In the case of an error, a `TokenException` is thrown.

The following example code shows how to use the `readToken` method:

> **Note**
>
> This code is the same for both the Java native and Apache commons multimap methods described in Reading and writing OpenTokens.

Sample code

Copyright © 2025 Ping Identity Corporation

```
try {
   //See the "Reading and writing OpenTokens" topic for sample code
   //that instantiates and configures an Agent instance

Map userInfo = agent.readToken(request);
if(userInfo != null) {
    String username = (String)userInfo.get(Agent.TOKEN_SUBJECT);
}
}
catch(TokenException e) {
   // Handle exception
}
```

Receiving multi-value attributes

The Agent Toolkit for Java supports receiving multi-value attributes in the SAML assertion from PingFederate.

Using the Agent Toolkit with Apache Commons MultiMap

Multi-value attributes are received as an `org.apache.commons.collections.MultiMap` collection of attributes.

```
try {
   //see "Using the Agent API" section for sample code
   //that instantiates and configures an Agent instance
   MultiMap userInfo = agent.readTokenToMultiMap(request);
   if(userInfo != null) {
     String username = (String)((List)userInfo.get(Agent.TOKEN_SUBJECT)).get(0);
     List groups = (List)userInfo.get("GROUP");
   }
}
catch(TokenException e) {
    // Handle exception
}
```

Using the PingFederate MultiMap equivalent

Copyright © 2025 Ping Identity Corporation

```
try
{
    //see “Using the Agent API” section for sample code
    //that instantiates and configures an Agent instance
    PingFederateMultiMap userInfo = agent.readTokenToNativeMultiMap(request);
    if (null != userInfo)
    {
        String username = (String) (userInfo.get(Agent.TOKEN_SUBJECT)).get(0);
        List groups = userInfo.get(“GROUP”);
    }
}
catch (TokenException e)
{
    // Handle exception
}
```

SP single sign-on (using account linking)

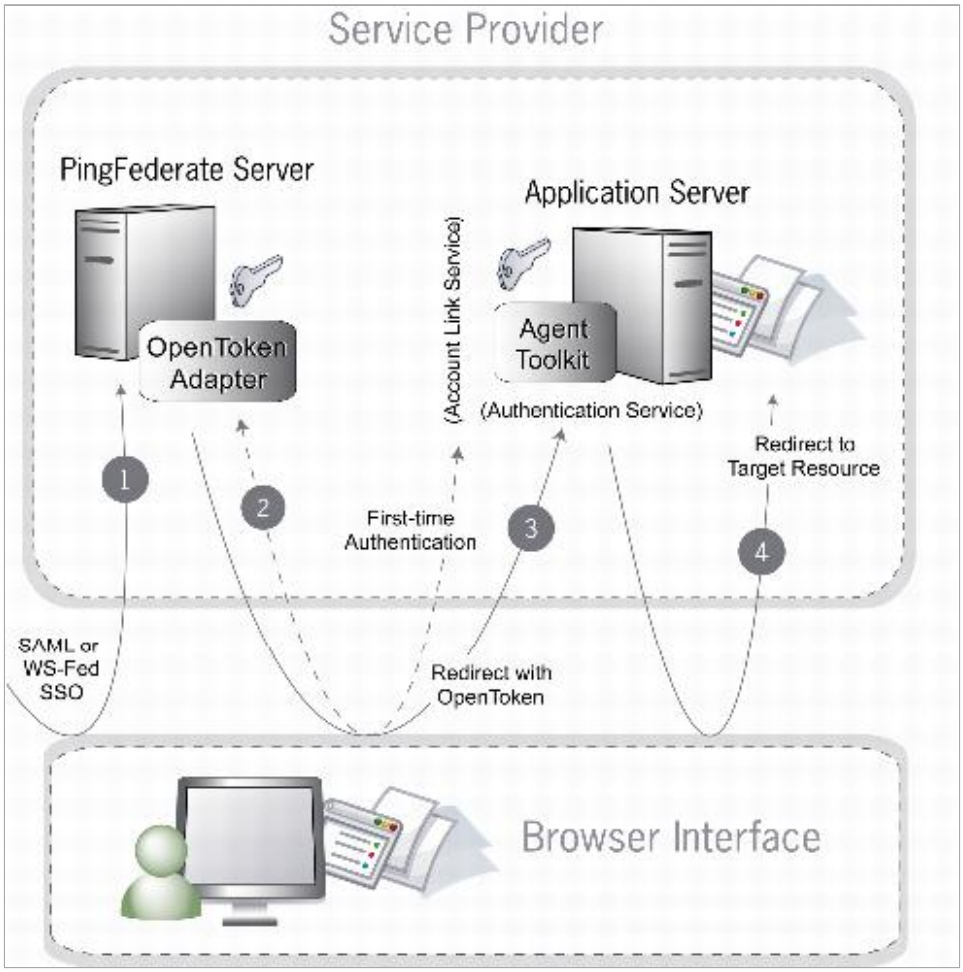
If an SP’s SSO implementation uses account linking, the flow of events is slightly different because a user must authenticate to the SP application the first time SSO is initiated. Learn more in Key concepts in the PingFederate documentation.

In this case, PingFederate and the OpenToken Adapter support an integration mechanism to redirect the user to an Account Link Service to which a user can initially authenticate. After the user authenticates successfully, the account link service must redirect the user back to PingFederate with an OpenToken that PingFederate uses to create an account link for the user.

For subsequent SSO requests, PingFederate uses the account link established in the first SSO request to identify the user. It then creates an OpenToken and sends it to the authentication service associated with the application.

The following diagram shows the SP SSO flow using account linking:

Copyright © 2025 Ping Identity Corporation



Sequence

1. PingFederate receives an assertion under the SAML 2.0 or WS-Federation protocol.
2. If this is the first time the user has initiated SSO to this SP, PingFederate redirects the browser to the application server's account link service, where the user must authenticate. After the user authenticates successfully, an `OpenToken` is returned to PingFederate, and an account link is established for this user within PingFederate. This account link is used on subsequent SSO transactions.
3. PingFederate retrieves the local user ID from its account link data store. PingFederate's OpenToken Adapter generates an `OpenToken` based on the assertion and account link. PingFederate then redirects the user's browser to the web application's SSO authentication service, passing the `OpenToken` in the redirect.
4. The authentication service extracts the contents of the `OpenToken`, establishes a session for the user, and redirects the user's browser to the target resource (the **resumePath** URL is sent as a query parameter).

Copyright © 2025 Ping Identity Corporation

Sample code

In an account linking event, the user's browser is redirected to the configured Account Linking Service in the SP OpenToken Adapter instance. The application should capture the resumePath upon a `GET` request made to this URL with something similar to the following:

```
String username = (String)request.getSession().getAttribute("username");
Map<String, String> userInfo = new HashMap<String, String>();
userInfo.put(Agent.TOKEN_SUBJECT, username);
String returnUrl = "https://<{pingfed} DNS>:9031" + request.getParameter("resume");
. . . .
try {
   UrlHelper urlHelper = new UrlHelper(returnUrl);
   //see "Using the Agent API" section for sample code
   //that instantiates and configures an Agent instance
   agent.writeToken(userInfo,response,urlHelper,false);
   returnUrl = (String)urlHelper.toString();
}
catch(TokenException e) {
   // Handle exception
}
response.sendRedirect(returnUrl);
```

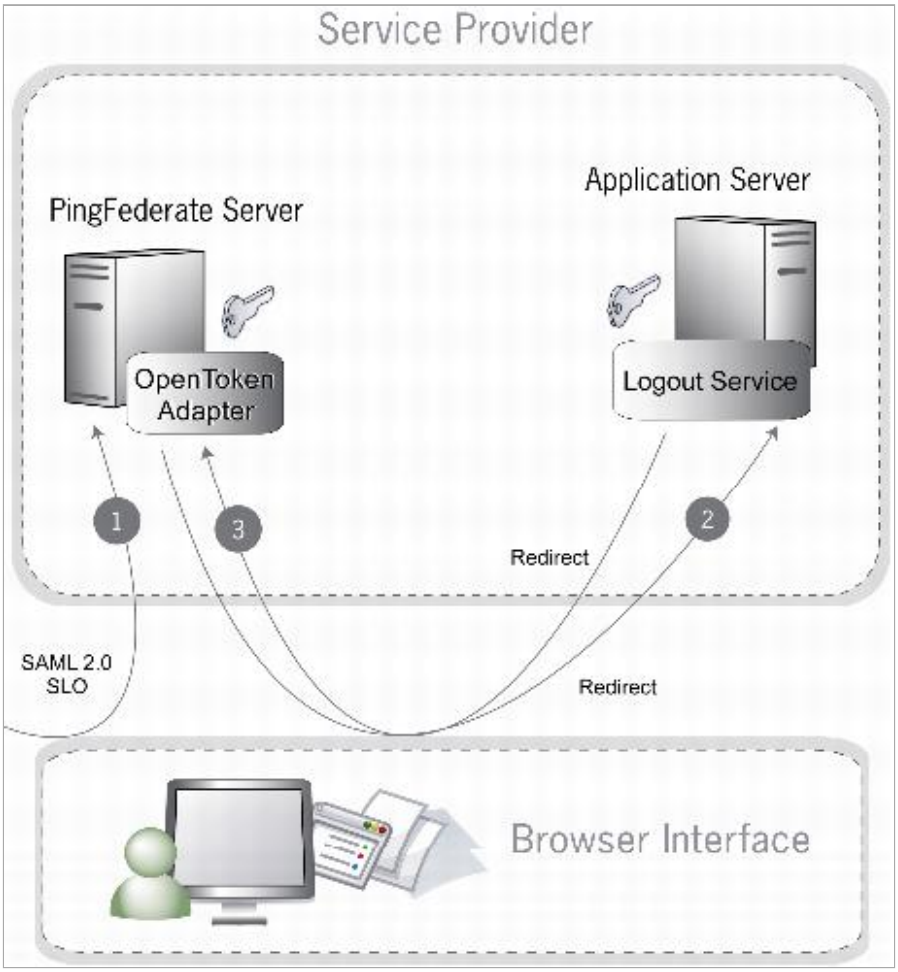
SP single logout (SLO)

When an service provider (SP) PingFederate server receives a request for single logout (SLO), it redirects the user's browser to the logout service as configured in the SP OpenToken Adapter instance. As part of the redirect, PingFederate and the OpenToken Adapter include both an OpenToken and a **resumePath** query parameter.

- The OpenToken includes attributes about the user.
- The **resumePath** query parameter provides the SP with the target URL where the user's browser must return after the application completes the local sign off.

A user can have multiple sessions. This sign-off sequence, as shown in the following diagram, happens for each of the user's sessions controlled by the SP PingFederate server.

Copyright © 2025 Ping Identity Corporation



Sequence

1. PingFederate receives an SLO request under the SAML 2.0 protocol.
2. If the application server has an SLO service configured, PingFederate redirects the user to the SLO service, which identifies and removes the user's session locally.
3. The application logout service redirects back to PingFederate to display a sign off success page.

   If the web application does not have an SLO service configured, the adapter redirects back to PingFederate, which displays a sign off success page.

The code needed to perform an SP-initiated SLO is identical to that required for an IdP-initiated SLO.

Testing

You can test the Java Integration Kit using one of the sample applications bundled with this distribution. Learn more in Sample application setup.

Copyright © 2025 Ping Identity Corporation

> **Caution**
>
> Do not install the `data.zip` archive as shown in Deploying the sample configuration archive. This will overwrite the PingFederate configuration that you just completed.

Sample application setup

You can configure the Java Integration Kit to work with the included sample applications.

The sample applications allow you to try a working demonstration of the Java Integration Kit with IdP-initiated SSO, SP-initiated SSO, and single logout (SLO).

Components

- Identity provider (IdP) sample application
  - Acts as an IdP application in your demonstration environment.
- Service provider (SP) sample application
  - Acts as an SP application in your demonstration environment.
- PingFederate configuration archive
  - This `data.zip` archive automatically configures PingFederate with OpenToken Adapter instances to work with the two sample applications.

Intended audience

The installation and basic usage portions of this guide are intended for PingFederate administrators and web-application architects and developers who want to test the deployment of the Java Integration Kit or validate end-to-end integration. Some knowledge of the PingFederate administrative console and identity federation using SAML is helpful but not required.

Sections describing the optional use of the Java source code and advanced alternative deployment scenarios are intended for experienced Java developers and administrators with some expertise using and deploying PingFederate in a production environment.

System requirements

- PingFederate 11.3 or later.
- The OpenToken Adapter (included).
- A JavaScript-enabled web browser.
- Apache Ant (if you want to rebuild the sample applications).

Deploying the sample configuration archive

The `data.zip` archive configures a PingFederate to work with the sample applications.

About this task

Copyright © 2025 Ping Identity Corporation

In this configuration, a single instance of PingFederate serves both the identity provider (IdP) and service provider (SP) roles by sending messages to and from itself. Although this is a valid scenario for demonstration and testing, it is not applicable to a real situation.

For a more realistic test, you can deploy the configurations into instances of PingFederate running on different hosts and then make necessary configuration updates manually as shown in Advanced installation and configuration.

Caution

Deploying `data.zip` overwrites your entire PingFederate configuration. If you have configured adapters, data stores, partner connections, or any other settings, back up your configuration in step 2.

Steps

1. Start PingFederate.
2. Back up your current PingFederate configuration as shown in Configuration archive in the PingFederate documentation.
3. From the Java Integration Kit `.zip` archive, copy the `sample/data.zip` archive to `<pf_install>/pingfederate/server/default/data/drop-in-deployer/`.

   Result:

   PingFederate adopts the new configuration and renames the `data.zip` file with a time stamp.

Deploy the sample applications

The Java Integration Kit includes two applications that act as identity provider (IdP) and service provider (SP) in the demonstration environment. Deploy the applications into the PingFederate servlet container as follows.

About this task

Important

As of Java Integration Kit 2.8, there are two versions of the OpenToken agent: the standard OpenToken agent and the Jakarta EE 9 OpenToken agent. The Jakarta EE 9 OpenToken agent has its own version of the sample applications located in the `/sample/jakarta` folder.

- You can only deploy the sample Jakarta applications on servers that support Jakarta EE 9. Currently, PingFederate does not support Jakarta EE 9.
- You shouldn't deploy both versions of the OpenToken agent on the same instance because they need to listen to the same endpoint.

Tip

To test a custom configuration, you can deploy the applications in a separate container on your network and then make the necessary configuration changes as described in Advanced installation and configuration.

Steps

1. Stop PingFederate.

Copyright © 2025 Ping Identity Corporation

2. From the Java Integration Kit `.zip` archive, copy the `sample/IdpSample.war` and `sample/SpSample.war` files to `<pf_install>/pingfederate/server/default/deploy`.
3. Start PingFederate.

Using the sample applications

The sample identity provider (IdP) and service provider (SP) applications demonstrate single sign-on (SSO) and single logout (SLO) processing to and from your PingFederate server.

Each application also:

- Displays attributes provided by the OpenToken Adapter on an IdP application web page and attributes received in SSO assertions on an SP application web page.

  > **Tip**
  >
  > You can also see artifacts of these transactions in the command window running PingFederate or in the server log file.

- Provides several configurable SSO and operational options that can be adjusted for testing alternative deployment and configuration scenarios. Learn more in Advanced installation and configuration.

You can initiate SSO from either the IdP application or the SP application.

> **Note**
>
> Developers can adjust the look and feel of the sample applications and modify the source code to change functionality for testing and demonstration.
> Developers can also use the source files to implement SSO/SLO functionality in their own web applications for enterprise deployment. Learn more in Modifying sample source files.

Using the IdP sample application

When accessed first, the identity provider (IdP) sample application simulates the IdP-initiated SSO and SLO scenario in which users authenticate to an IdP locally to access a remote service provider (SP) application. As an example scenario, users could be accessing a company portal that provides links to partner web resources such as HR or 401(k) information.

Steps

1. Start PingFederate.
2. In a web browser, open the sample application:

   https://*<pf_host>*:*<pf_port>*/IdpSample

   If you deployed the application elsewhere, modify the host name accordingly. Learn more in Advanced installation and configuration.
3. On the **IdP Application** home page, click **Local Login**.
4. On the **Local Login** page, sign on with the following credentials:

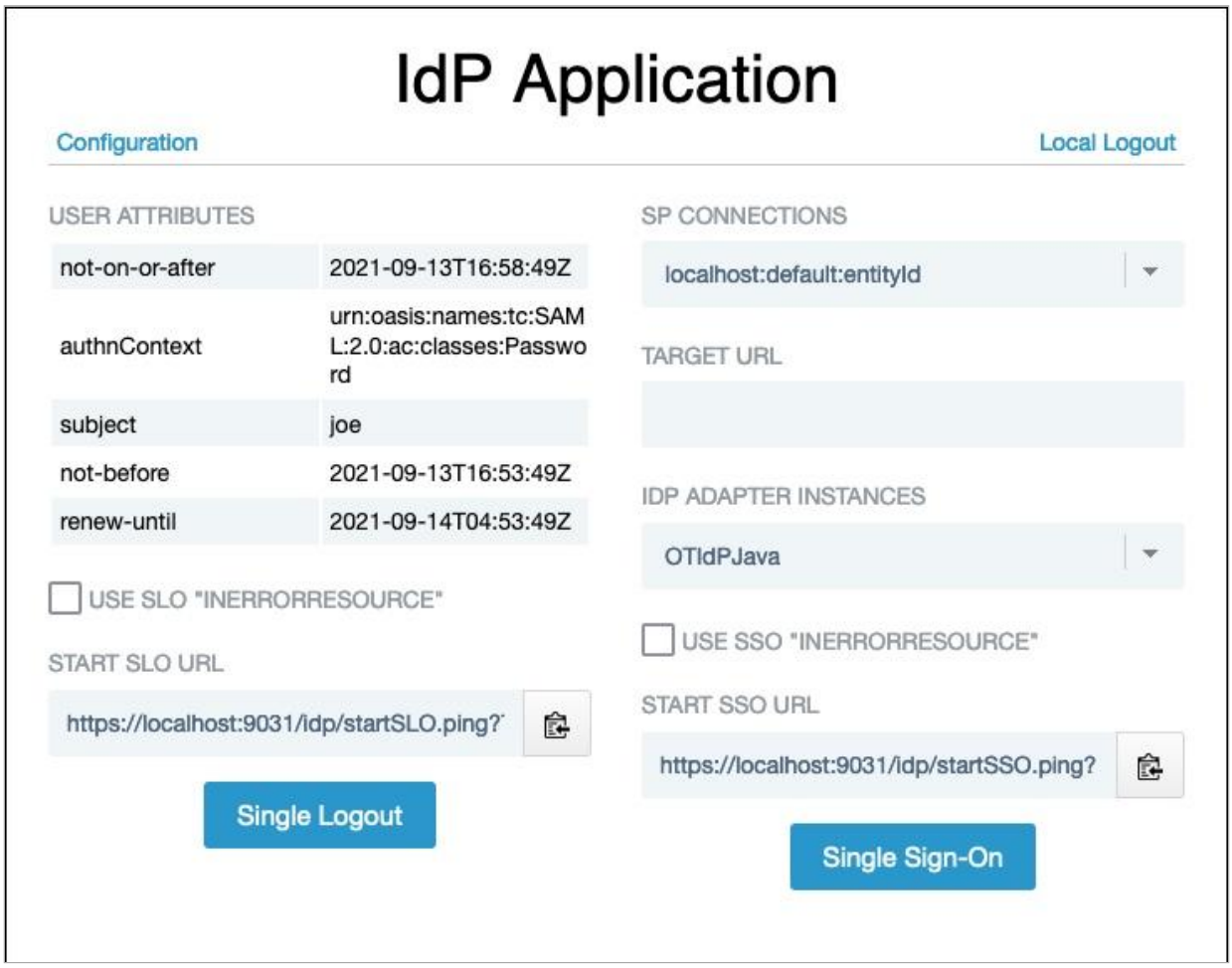
Copyright © 2025 Ping Identity Corporation

Username: `joe`

Password `test`

5. Click **Login**.

   When you authenticate locally to the IdP sample application, no communication occurs between the application and PingFederate; the user authenticates using the local user store. No SSO use cases are invoked until PingFederate is called on the home page via the **Single Sign-On** button.



6. If you customized your configuration by creating different SP connections or deploying the IdP application on a separate server, update the application configuration as shown in IdP sample application configuration reference.

7. After logging on to the IdP sample application, the **IdP Application** home page reappears. In addition to SSO/SLO information and controls, this page now shows attributes sent to the SP in the SAML assertion.

   The steps below describe how to test the basic SSO/SLO functionality of this page:

   1. Click **Single Sign-On** to begin IdP-initiated SSO to the SP sample application. A user session on the SP is created, and you are signed on to the SP sample application. You can find more information in Using the SP sample application.

Copyright © 2025 Ping Identity Corporation

2. To demonstrate IdP-initiated SLO, return to this identity provider page by changing the browser location back to https://*<pf_host>*:*<pf_port>*/IdpSample. Then click **Single Logout** to initiate an SLO request to the SP. When the user session on the remote SP is destroyed, the local session is destroyed as well (clicking **Single Sign-on** here or on the SP application page redirects to the IdP logon page).

> **Note**
>
> If you initiate SSO from the SP, the **Single Logout** button is operational and ends both sessions. If you click **Local Logout** on this screen, the SP session is still in force. You can access the SP application directly in the browser and see the attributes passed in from the original SAML assertion.

IdP sample application configuration reference

The sample applications automatically work with PingFederate configuration you deployed in Deploying the sample configuration archive. However, if you want to test your own deployment and sign-on scenarios, customize the sample application settings as follows:

Field	Description
PF Base URL	The fully qualified host name, port, and path of the PingFederate server. The default value is `https://localhost:9031/`.
SP Connections	The ID of the service provider (SP) connection to use. Separate multiple connection IDs with a pipe `|` character. The default value is `localhost:default:entityId`.
IdP Adapter Instances	The ID of the OpenToken Adapter instance to use. Separate multiple adapter IDs with a pipe `|` character. The default value is `OTPIdPJava`.
Attribute Names List	Single and multi-value attributes to include in the assertion. Separate multiple attribute names with a pipe `|` character. The default value is `password|authnContext`.
IdP OpenToken Config File	If you customized the OpenToken Adapter configuration provided by the sample deployment, export the `agent-config.txt` file from it, then upload the file here.

Using the SP sample application

When accessed first, the service provider (SP) sample application simulates SP-initiated SSO and SLO scenarios. The application also provides a simulated target resource for SSO/SLO transactions initiated from the identity provider (IdP).

Steps

1. Start PingFederate.

Copyright © 2025 Ping Identity Corporation

2. In a web browser, open the sample application:

   https://<*pf_host*>:<*pf_port*>/SpSample

   If you deployed the application elsewhere, modify the hostname accordingly. Learn more in Advanced installation and configuration.

3. If you customized your configuration by creating different IdP connections or deploying the SP application on a separate server, update the application configuration as shown in SP sample application configuration reference.

4. Click **Single Sign-On** to begin SP-initiated SSO.

   You are redirected to the IdP login page.

   > **Note**
   >
   > If you are already signed on at the IdP, this step simply redirects back to this page, unless **Force Authn** is selected. Click **Single Logout** to try again.

5. On the identity provider sign-on page, enter the following values:

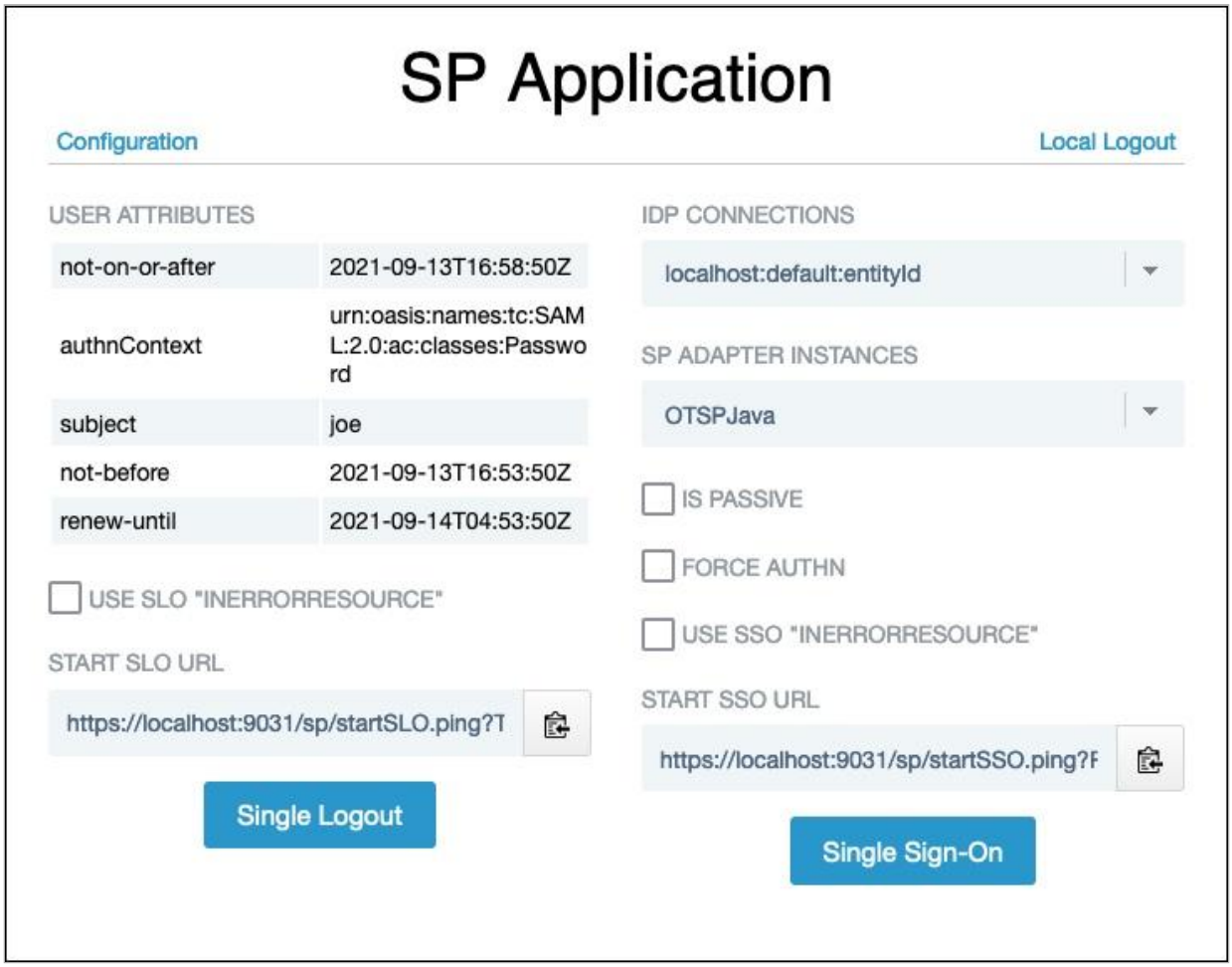
   Username: `joe`

   Password: `test`

6. Click **Login**.

   Having completed an SP-initiated SSO, you are redirected back to the Service Provider main page, which now displays attributes sent from the IdP in the SAML assertion.

Copyright © 2025 Ping Identity Corporation



7. Click **Single Logout** to begin an SP-initiated SLO transaction.

   Upon successful completion of this transaction, the service provider main page is shown again but without attributes listed.

> **Note**
>
> Clicking **Local Logout** in the top menu has the same effect. However, since you are still signed on at the IdP, if you click **Single Sign-On** again (either here or at the IdP), you are not required to reauthenticate. This page is redisplayed immediately with SSO attributes available from a new IdP-generated SAML assertion.

SP sample application configuration reference

The sample applications automatically work with the PingFederate configuration you deployed in Deploying the sample configuration archive. However, if you want to test your own deployment and sign-on scenarios, customize the sample application settings as follows:

Copyright © 2025 Ping Identity Corporation

Field	Description
PF Base URL	The fully qualified hostname, port, and path of the PingFederate server. The default value is `https://localhost:9031/`.
IdP Connections	The ID of the identity provider (IdP) connection to use. Separate multiple connection IDs with a pipe `|` character. The default value is `localhost:default:entityId`.
SP Adapter Instances	The ID of the OpenToken Adapter instance to use. Separate multiple adapter IDs with a pipe `|` character. The default value is `OTPSPJava`.
Attribute Names List	The single and multi-value attributes included in the assertion. Separate multiple attribute names with a pipe `|` character. The default value is `password|authnContext`.
SP OpenToken Config File	If you customized the OpenToken Adapter configuration provided by the sample deployment, export the `agent-config.txt` file from it, then upload the file here.

Modifying sample source files

The Java Integration Kit distribution provides the source files for the sample applications in the `sample_src` directory, including Java code and supporting JSPs, images, style sheets and other components.

Developers can use these files to change the appearance and behavior of the samples, create different samples for testing, or develop actual prototypes for production purposes. In addition, the commented Java code is engineered into discrete classes and methods that can be repurposed to work with existing web applications.

This section further identifies the source file directories and describes how to rebuild the sample applications, as a means of highlighting file locations and build scripts that developers could use for other customizing purposes.

JSPs and web page components

You can modify the JSPs, images, and cascading style sheets (CSSs) to customize the look and feel of the installed sample applications.

- The JSP files are located in the following directories in the Java Integration Kit distribution:

  `sample_src/<sample>/webapp/WEB-INF/jsp`

  where *<sample>* is either idp or sp

- Images and CSS files are located in:

  `sample_src/<sample>/webapp/images`

Rebuilding the sample applications

As a Java developer, you can make changes to the source code for any of the sample applications for your own needs.

Copyright © 2025 Ping Identity Corporation

About this task

The source code for the sample applications is located in `sample_src/<sample>/java`.

Important

As of Java Integration Kit 2.8, there are two versions of the OpenToken agent: the standard OpenToken agent and the Jakarta EE 9 OpenToken agent. The Jakarta EE 9 OpenToken agent has its own version of the sample applications. The source code for the Jakarta EE 9 sample applications is located in the `jakarta/idp` and `jakarta/sp` files.
You can only deploy the sample Jakarta applications on servers that support Jakarta EE 9, such as Tomcat 10.1. Currently, PingFederate does not support Jakarta EE 9.

After you modify the source code, you can build and repackage the application using `ant` commands executed from the `<sample>` directory for the `build.xml` file included in that directory. You can package either another WAR directory or a compressed WAR file (depending on deployment needs) using the following commands:

Note

For backward compatibility, the `build.xml` configuration file is set to use version 1.5 of the Java Development Kit (JDK). To recompile using the current JDK version, change the version number in the `build.xml` file.

Steps

1. Create a WAR directory. From the `<sample>` directory, enter:

```
ant
```

2. Create a WAR file. From the `<sample>` directory, enter:

```
ant war
```

Advanced installation and configuration

PingFederate system administrators might choose to override the automated PingFederate configuration and simplified deployment of the sample applications. Although the distribution is intended primarily for quick end-to-end testing and demonstration, it can also be used as a configuration exercise or to test more production-like deployments.

This section provides guidelines for alternative uses for the sample application distribution.

Important

Detailed instructions for configuring separate PingFederate servers and deploying applications in different servlet containers are beyond the scope of this document.
This document provides general information, but it might not be complete or applicable to every deployment configuration.

Copyright © 2025 Ping Identity Corporation

Manual configuration settings

Instead of using the sample `data.zip` archive from the Java Integration Kit package (see Deploying the sample configuration archive), you can configure PingFederate manually.

> **Tip**
>
> If you have a difficult time with the manual configuration, deploy the sample `data.zip` file. Then you can refer to the sample configuration as you create your own adapter instances and connections.

The following tables list the required adapter settings for the Java sample applications.

Identity provider adapter instance settings

Refer to the **IdP Adapter** tab of the identity provider adapter configuration.

Field Name	Value
Password	Must be six characters long and contain at least: • One uppercase letter • One lowercase letter • One number This password is included in the `agent-config.txt` file used by the application. It doesn't have to match the password you choose for the service provider adapter instance. **Note** The sample applications are preconfigured to use the following password: `Changeme1`
Authentication Service	https://*<pf_host>*:*<pf_port>*/IdpSample/MainPage?cmd=sso
Advanced Fields	
Logout Service	https://*<pf_host>*:*<pf_port>*/IdpSample/MainPage?cmd=slo

> **Note**
>
> The endpoints in these tables assume you deployed the sample applications in the same servlet container as PingFederate. If you deployed the applications elsewhere, change the hostname (`localhost`) and port accordingly. Learn more in Deploying the applications to separate servlet containers.

Copyright © 2025 Ping Identity Corporation

Service provider adapter instance settings

Refer to the **Instance Configuration** tab of the service provider adapter instance configuration.

Field Name	Value
Password	Must be six characters long and contain at least: • One uppercase letter • One lowercase letter • One number This password is included in the `agent-config.txt` file used by the application. It doesn't have to match the password you choose for the identity provider adapter instance. **Note** The sample applications are preconfigured to use the following password: `Changeme1`
Advanced Fields	
Account Link Service (Optional)	https://<*pf_host*>:<*pf_port*>/SpSample/MainPage?cmd=accountlink
Logout Service	https://<*pf_host*>:<*pf_port*>/SpSample/MainPage?cmd=slo

Service provider default URLs

See the **Default URLs** tab of the service provider configuration.

Field	Value
For SSO URL	https://<*pf_host*>:<*pf_port*>/SpSample/MainPage/
For SLO URL	https://<*pf_host*>:<*pf_port*>/SpSample/MainPage/

Using separate IdP and SP servers

Note

This configuration requires some familiarity with PingFederate and SSL certificate management.

Copyright © 2025 Ping Identity Corporation

You can also choose to deploy the `data.zip` archive on separate PingFederate servers functioning as in a specific identity-federation role. To do this, deploy the archive on each server, and then deactivate the identity provider (IdP) federation role on one server and the service provider (SP) role on the other.

You'll then need to change the **Base URL** on the **General Info** screens for each connection to the respective partner's host and port (to update SSO and SLO partner protocol endpoints).

Finally, you might need to install and exchange new SSL server certificates on each server to reestablish trust between them.

Deploying the applications to separate servlet containers

> **Note**
>
> This configuration requires some familiarity with web-application deployment and the target-container configuration, as well as SSL certificate management.

You can deploy either or both of the sample applications into their own servlet containers (such as Tomcat), rather than the container running PingFederate. If you do this, you'll need to update the URLs indicated under Manual configuration settings.

You might also need to update the Trusted CAs in PingFederate with the container's SSL server certificate, depending on the container's SSL configuration, and ensure the container server trusts the PingFederate certificate.

> **Note**
>
> If you deploy either application in a separate container, update the PingFederate server hostname in the associated sample application, as shown in IdP sample application configuration reference and SP sample application configuration reference.

Release notes

Changelog

The following is the change history for the Java Integration Kit.

Java Integration Kit 2.9.0 - February 2025

- Removed an advanced field in the adapter configuration that allowed administrators to skip password obfuscation for the exported OpenToken agent configuration file, `agent-config.txt`.

  You can find more details about this change in Configuring an OpenToken SP adapter instance and Configuring an OpenToken IdP adapter instance in the PingFederate documentation.

  > **Note**
  >
  > If you are using a third-party OpenToken integration and need to use a base64 password instead of an obfuscated password, refer to step 5 in Configuring an OpenToken SP adapter instance or Configuring an OpenToken IdP adapter instance.

Copyright © 2025 Ping Identity Corporation

Java Integration Kit 2.8.0 – October 2023

- Added a new OpenToken agent that uses Jakarta EE 9.
- Added new sample applications that use Jakarta EE 9.

Java Integration Kit 2.7.3 – December 2022

- Improved security by updating the bundled Apache Log4j2 component in the sample applications.

Java Integration Kit 2.7.2 – December 2021

- Improved security by updating the bundled Apache Log4j2 component. Learm more in security bulletin CVE-2021-44228.
- Added support for reading and writing OpenTokens with the Java native multimap method. Learn more in Reading and writing OpenTokens. There's no change to encoding or decoding attributes.

  The `commons-beanutils.jar` and `commons-collections-3.2.2.jar` libraries are not required for this method.

Java Integration Kit 2.7.1 – September 2021

- Improved the sample application configuration pages.
  - Removed the **PF SSO Directory Service ID** and **PF SSO Directory Service Shared Secret** fields, which relied on a feature that was deprecated in PingFederate 10.2.
  - You can now enter multiple or custom connection IDs. Learn more in IdP sample application configuration reference and SP sample application configuration reference.
- Improved the sample `data.zip` archive by enabling the redirection validation setting in PingFederate by default.

Java Integration Kit 2.7 – March 2021

- Added a new method for reading tokens as an alternate to the Apache Commons Multimap.

Java Integration Kit 2.6.2 – February 2020

- Fixed an issue that could cause cross-site single logout to stop working after upgrading the adapter.

Java Integration Kit 2.6.1 – February 2020

- Fixed an issue that caused session cookies to expire at the wrong time.

Java Integration Kit 2.6 – January 2020

- Added the **SameSite Cookie** field to support the SameSite cookie flag in web browsers.

Java Integration Kit 2.5.9 – January 2020

- Resolved an issue that caused an error in the sample application in PingFederate 9.1 and earlier.
- Resolved an issue in the sample application sign on button that could cause an error in an authentication policy tree.

Copyright © 2025 Ping Identity Corporation

Java Integration Kit 2.5.8 – August 2019

- Added support for RFC 6265 compliance to the OpenToken SP adapter when extended attributes are sent as cookies.
- Added support to do SLO without external logout service configuration.
- Fixed minor bugs in the sample applications.

Java Integration Kit 2.5.7 – November 2018

- Modernized the look and feel of the Java sample applications.
- Removed the preceding dot requirement in the **Cookie Domain** setting of the OpenToken Adapter **IdP Adapter** tab.

Java Integration Kit 2.5.6 – April 2017

- Expanded character support for entity id.

Java Integration Kit 2.5.5 – August 2016

- Update for compatibility with PingFederate 8.2.

Java Integration Kit 2.5.4 – May 2016

- Added an **HttpOnly** field.

Java Integration Kit 2.5.3 – June 2015

- Added support for log4j2.

Java Integration Kit 2.5.2 – June 2014

- Added logging to the OpenToken Agent to log the name of the token and its value at the debug level.
- Added support to validate the resume path for the IdP sample application, only enabling redirection along a relative path.
- Updated Ping Identity logos for the sample applications.
- Added support for configuring a context path for the sample applications.

Java Integration Kit 2.5.1 – November 2012

- Address a security issue in the previous release.
- Added support for OpenToken 2.5.1 Adapter and the OpenToken 2.5.1 Agent.

Java Integration Kit 2.5 – May 2012

- Fixed several minor defects in the IdP and SP sample applications.
- Corrected the spelling of `AgentConfiguration.setObfuscatePasword()` method name to `AgentConfiguration.setObfuscatePassword()` and deprecated the misspelled method.
- Improved Javadoc documentation for several areas of the API.

Copyright © 2025 Ping Identity Corporation

Java Integration Kit 2.4.2 – October 2010

- Removed extraneous configuration from the IdP and SP sample application data archive (data.zip).
- Added support for UTF-8 encoding in the IdP and SP sample applications for attributes.

Java Integration Kit 2.4.1 – August 2010

- Rewrote the IdP and SP sample applications to provide reference OpenToken integration for Java web applications.
- Added buildable source code for IdP and SP sample applications.

Java Integration Kit 2.4 – February 2010

- Added token Replay Prevention to the OpenToken IdP Adapter advanced settings.

Java Integration Kit 2.3 – November 2008

- Added POST transport method for OpenToken when used by an SP.
- Added an option to specify session vs. persistent cookie.
- Added an option to set the `Secure` attribute on an OpenToken when a cookie is used.
- Added ability to bypass password obfuscation and strength enforcement for backward compatibility with previous Java OpenToken agents.
- Improved handling of `null` parameters for single logout via the back-channel (SOAP).
- Empty query string (`?`) is not automatically appended to the URL when redirecting to the Target Resource.
- Target Resource URL is URL-encoded.

Java Integration Kit 2.2 – June 2008

- Added support for SAML 2.0 isPassive and ForceAuthn.
- Enforced UTF-8 encoding within OpenToken.
- Extended Force Sun JCE Provider option in the OpenToken Adapter to allow compatibility with SafeNet Luna HSM.
- Symmetric key in the OpenToken agent configuration file is encrypted.
- Combined the OpenToken Adapter and OpenToken Java library jar files into a single adapter file for easier deployment to PingFederate.

Java Integration Kit 2.1 – April 2008

- Added `AgentConfiguration` class to simplify Agent instantiation.
- Added Agent Toolkit API Javadoc.

Copyright © 2025 Ping Identity Corporation

Java Integration Kit 2.0 – December 2007

Modified to use an open-standard, secure token called OpenToken to pass user information between an application and PingFederate. The OpenToken is passed through the user's browser as a URL query parameter or an HTTP cookie. The data within the OpenToken is a set of key/value pairs, and the data is encrypted using common encryption algorithms.

Java Integration Kit 1.3 – November 2007

- Added option to force the use of the Sun Java Cryptography Extension (JCE) on the agent JDK (when necessary to support the SafeNet Luna SA Hardware Security Module) If this option is not selected, the default JCE provider is used, either SunJCE or IBMJCE.
- Modified salt-value generation to correct `PFTOKEN` creation delay on Linux operating systems
- Added support for setting and retrieving multi-value attributes in the SAML assertion
- Added Secure Cookie option which ensures the `PFTOKEN` cookie is sent only on a secure channel
- Added Session Lifetime option to specify the duration (in seconds) for which the token can be reissued without authentication (added for other purposes and has no effect on the Java Integration Kit)

Java Integration Kit 1.2.1 – August 2007

- Bundled Java sample application with distribution.
- Modified to allow backward compatibility of the Standard Adapter 1.2.1 with PingFederate 4.0.

Java Integration Kit 1.2 – May 2007

- Added option to encode `PFTOKEN` for handling special characters.
- Added additional constructor to allow `PFTOKEN` to use default properties for all configuration options except password, holder name, and max age.
- Added `PFTOKEN` time stamp information in the log file.

Known issues and limitations

The following are known issues or limitations with the Java Integration Kit.

Known issues

There are no known issues.

Known limitations

- The user interface doesn't enforce this, but the adapter instance's `Token Name` field must be unique within a federation.
- Multi-value attributes fail if the SP adapter sends extended attributes as cookies because multiple cookies with the same name aren't allowed.
- When running the sample applications in a separate container, the back-channel web-SSO directory service fails unless the certificate is trusted by the JDK. As a result, the sample applications won't list available partners to SSO in the list.

Copyright © 2025 Ping Identity Corporation

To get around this limitation, import the certificate into the JDK or web container's trusted certificate authority (CA) store or use HTTP instead.

- Support for UTF-8 encoding is limited to the sample applications and the attributes displayed. UTF-8 encoded usernames, passwords, and token names for the OpenToken configuration, SSO Directory Service configuration, and sample application configuration are not supported.
- The Java Integration Kit isn't compliant with FIPS-140 cryptographic standards. By forcing the Sun JCE to be used in the adapter, keys aren't stored in the hardware security module (HSM).

Download manifest

The following files are included in the Java Integration Kit:

- `Legal.pdf` : Copyright and license information.
- `agent` : Contains the OpenToken Agent and supporting libraries.
  - `opentoken-agent-<version>.jar` : A JAR file that contains the OpenToken Agent.
  - `lib` : Contains the libraries needed by the OpenToken Agent.
    - `commons-beanutils.jar` : The Apache Commons Bean Utility library.
    - `commons-collections-3.2.2.jar` : The Apache Commons Collections library.
    - `commons-logging.jar` : The Apache Commons Logging library.
  - `jakarta` : Contains the OpenToken Agent and supporting libraries using Jakarta EE 9.
    - `opentoken-agent-jakarta-<version>.jar` : A JAR file that contains the OpenToken Agent.
    - `lib` : Contains the libraries needed by the OpenToken Agent.
      - `commons-beanutils.jar` : The Apache Commons Bean Utility library.
      - `commons-collections-3.2.2.jar` : The Apache Commons Collections library.
      - `commons-logging.jar` : The Apache Commons Logging library.
- `dist/pingfederate/server/default` : Contains the integration files.
  - `deploy` : Contains the Java libraries.
    - `opentoken-adapter-<version>.jar` : JAR file that contains the OpenToken Adapter.
- `/docs/javadoc` : Contains the Agent Toolkit API Javadoc.
- `/sample` : Contains sample applications for testing the adapter.
  - `data.zip` : PingFederate configuration archive for the sample applications.
  - `/IdpSample.war` : Extracted WAR directory for the IdP sample application.
  - `/SpSample.war` : Extracted WAR directory for the SP sample application.

Copyright © 2025 Ping Identity Corporation

    - `/jakarta`: Contains the Jakarta EE 9 based sample applications for testing the adapter.
        - `/IdpSample.war`: Extracted WAR directory for the IdP sample application.
        - `/SpSample.war`: Extracted WAR directory for the SP sample application.
- `/sample_src`
    - `/idp`: Source code and supporting files for the IdP sample application.
    - `/sp`: Source code and supporting files for the SP sample application.
    - `/jakarta`: Contains the source code and supporting files for the IdP and SP sample applications for Jakarta EE 9.
        - `/idp`: Source code and supporting files for the IdP sample application for Jakarta EE 9.
        - `/sp`: Source code and supporting files for the SP sample application for Jakarta EE 9.

Copyright © 2025 Ping Identity Corporation

LinkedIn Login Integration Kit

The LinkedIn Login Integration Kit allows PingFederate to use LinkedIn as an identity provider (IdP). This allows users to access service provider (SP) applications by signing into LinkedIn.

Features

- Supports the PingFederate authentication API.
- Supports the JavaScript Widget for the PingFederate Authentication API.

Components

LinkedIn IdP Adapter

- Allows PingFederate to communicate with the LinkedIn API to process sign-on requests and get user information.

Templates

- Allows the adapter to prompt the user to sign on. The template can be presented with a browser redirect or as a pop-up window.
- Allows you to modify the appearance of the sign-on prompt.

Language packs

- Allows you to customize or localize the messages returned by the PingFederate authentication API and shown on the templates during authentication. Learn more in Localizing messages for end users in the PingFederate documentation.

Intended audience

This document is intended for PingFederate administrators.

If you need help during the setup process, you can find more information in the following:

- PingFederate documentation:
  - Identity provider SSO configuration
  - Managing IdP adapters
  - Authentication policies
  - SP connection management
- Sign in with LinkedIn in the Microsoft LinkedIn documentation.

System requirements

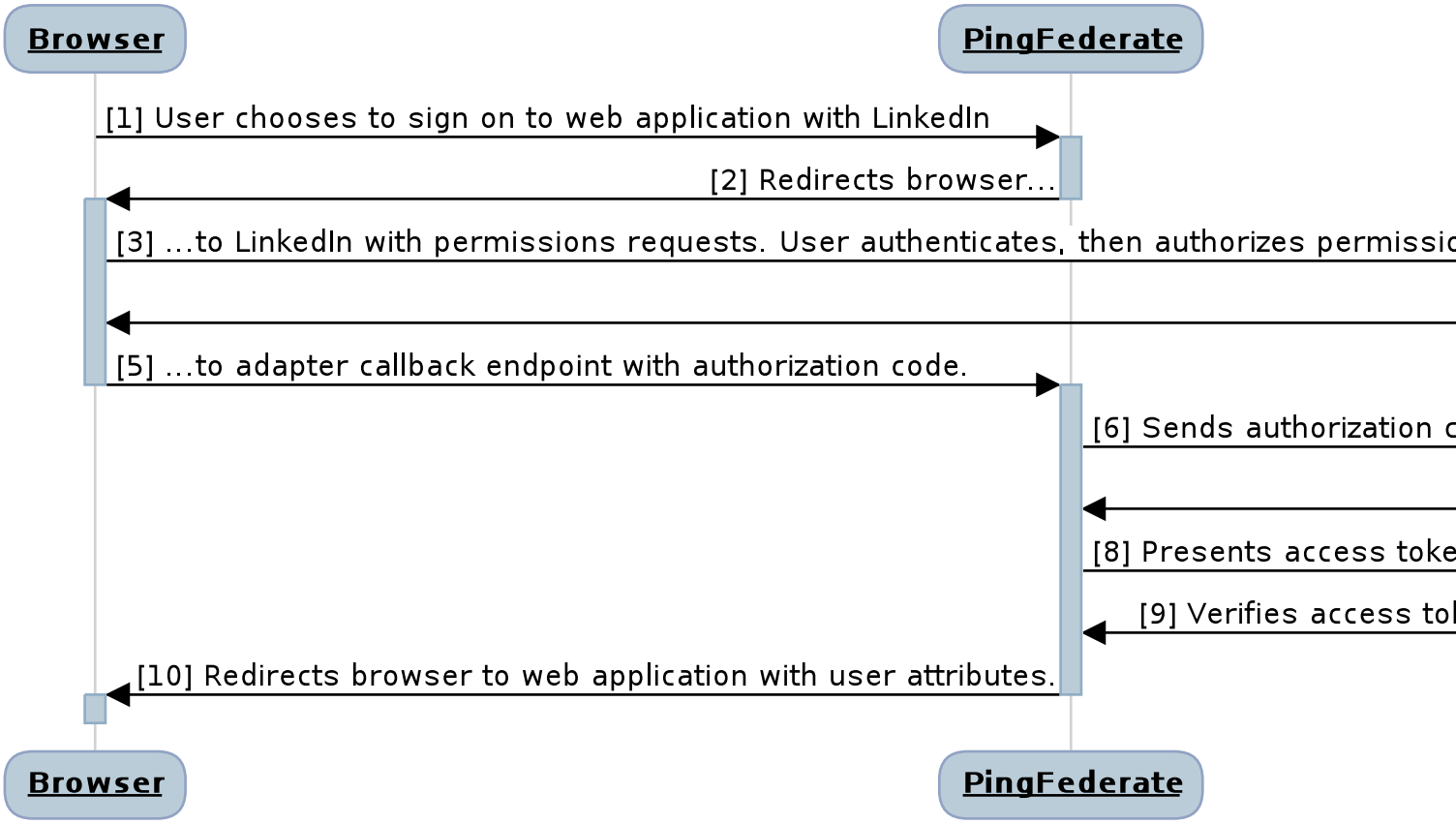
- PingFederate 9.0 or later, or 9.3 to use the authentication API
- A LinkedIn developer account

Copyright © 2025 Ping Identity Corporation

Overview of the SSO flow

With the LinkedIn Login Integration Kit, PingFederate includes the LinkedIn authentication API in the sign-on flow.

The following figure illustrates a single sign-on (SSO) scenario in which PingFederate authenticates users to an SP application using the LinkedIn IdP Adapter.



Description

1. The user opens a web application and chooses the LinkedIn sign-on option.
2. The sign-on link points to the LinkedIn IdP Adapter, which redirects the browser...
3. ..to LinkedIn with a list of requested permissions. On LinkedIn, the user authenticates their identity and then authorizes the requested permissions.
4. LinkedIn redirects the browser...
5. ...to the LinkedIn IdP Adapter authorization callback endpoint with an authorization code.

   If the user fails to authenticate or does not authorize the request, the response includes an error code instead.
6. PingFederate sends LinkedIn the authorization code.
7. LinkedIn returns an access token.
8. PingFederate sends LinkedIn a request for user attributes, and presents the access token.

Copyright © 2025 Ping Identity Corporation

9. LinkedIn verifies the access token, and provides the user information.
10. PingFederate redirects the user to the web application with the user attributes.

Authentication API Support

You can use the PingFederate authentication API to integrate the LinkedIn IdP Adapter into your application.

The PingFederate Authentication API provides access to the current state of the authentication flow as a user steps through the PingFederate authentication policy.

Learn more in PingFederate authentication API in the PingFederate documentation. To integrate the LinkedIn IdP Adapter into your authentication flow, configure your application based on the states, actions, and models available in the PingFederate Authentication API Explorer. For help, refer to Exploring the Authentication API in the PingFederate documentation.

Setup

Upgrading an existing deployment

If you are upgrading from a previous version of the LinkedIn Login Integration Kit, you might need to take extra steps.

Steps

1. Back up your current PingFederate configuration as shown in Configuration archive in the PingFederate documentation.
2. If you are upgrading from LinkedIn IdP Adapter 1.1.1 or earlier, note your adapter instance configuration, and then delete the instance.
    1. Note your adapter instance configuration.
    2. Delete the adapter instance.
    3. Complete the steps in Deploying the integration files.
    4. Complete the steps in Configuring an adapter instance. Configure the adapter instance settings based on your notes.

> **Note**
>
> LinkedIn IdP Adapter 1.1.1 and earlier is not compatible with the March 2019 changes to the LinkedIn API.

3. If you are upgrading from LinkedIn IdP Adapter 2.0.1 or earlier and used OGNL or JSON Pointer to get the `email` attribute:
    1. Complete the steps in Deploying the integration files.
    2. In your adapter instance configuration, remove the existing OGNL or JSON Pointer for the `email` attribute.
    3. In the **Permissions** field, include `r_emailaddress`.
    4. Select the **Retrieve Email** checkbox.
    5. On the **Extended Contract** tab, add `email`.

Copyright © 2025 Ping Identity Corporation

6. Save your changes.

Deploying the integration files

To get started with the integration, deploy the LinkedIn Login Integration Kit files to your PingFederate directory.

About this task

> **Note**
>
> If you operate PingFederate in a cluster, the following steps refer to the console node.

Steps

1. Download the LinkedIn Login Integration Kit `.zip` archive.
2. Stop PingFederate.
3. If you're upgrading an existing deployment, back up your customizations and delete earlier versions of the integration files:
    1. Back up any LinkedIn Login Integration Kit files that you customized in the `<pf_install>/pingfederate/server/default/conf/` directory.
    2. Delete the `pf-linkedin-adapter-<version>.jar` file from your `<pf_install>/pingfederate/server/default/deploy` directory.
4. Extract the `.zip` archive and merge the contents of the `dist` directory with your `<pf_install>/pingfederate` directory.
5. If there is more than one version of the `pf-authn-api-sdk-<version>.jar` file in your `<pf_install>/pingfederate/server/default/lib` directory, delete all but the latest version of the file.
6. If you backed up any customized files, modify the new files with your customizations.
7. Start PingFederate.
8. If you operate PingFederate in a cluster, repeat steps 2 - 8 for each engine node.

Register PingFederate as a Microsoft Application

To allow PingFederate to process social sign-on requests with LinkedIn, add PingFederate as an OAuth application in the LinkedIn administrative console.

Steps

1. Sign on to https://developer.linkedin.com/ using a LinkedIn developer account.
2. Click **Create App**.
3. Complete the **Create an app** form, and then click **Create app**.
4. Click the **Auth** tab.
5. Under **Application credentials**, note the **Client ID** and **Client Secret**.

Copyright © 2025 Ping Identity Corporation

You'll use these in Configuring an adapter instance.

6. Under **OAuth 2.0 settings**, click the **Edit** icon.
7. Enter the callback endpoint of the LinkedIn IdP Adapter, such as `https://pf.example.com:9031/ext/linkedin-authn`. Click **Update**.

> **Note**
>
> The default endpoint is `/linkedin-authn`. If you set a custom endpoint here, enter the matching value in the **Authorization Callback Endpoint** field of your adapter instance configuration in Configuring an adapter instance.

Configuring an adapter instance

Configure the LinkedIn IdP Adapter to determine how PingFederate communicates with LinkedIn.

Steps

1. In the PingFederate administrative console, create a new IdP adapter instance:

   Choose from:

   - For PingFederate 10.1 or later, go to **Authentication > Integration > IdP Adapters**. Click **Create New Instance**.
   - For PingFederate 10.0 or earlier, go to **Identity Provider > Adapters**. Click **Create New Instance**.

2. On the **Type** tab, set the basic adapter instance attributes.
   1. In the **Instance Name** field, enter a name for the adapter instance.
   2. In the **Instance ID** field, enter a unique identifier for the adapter instance.
   3. In the **Type** list, select **LinkedIn IdP Adapter**. Click **Next**.
3. (Optional) On the **IdP Adapter** screen, in the **Optional LinkedIn Attributes** section, define mappings between local attributes and the attributes from the core or extended contract.

   You can find a list of available attributes in Core contract attributes.
   1. Click **Add a new row to 'Attribute Selector'**.
   2. In the **Local Attribute** field, enter the name of a local attribute.
   3. In the **LinkedIn Attribute** field, enter the JSON Pointer syntax for the value of the matching LinkedIn attribute. Learn more in JSON Pointer syntax reference.

      > **Note**
      >
      > The **Permissions** field on the **IdP Adapter** page determines the available attributes.

   4. In the **Action** column, click **Update**.
   5. To add more attributes, repeat steps a - d.

Copyright © 2025 Ping Identity Corporation

4. On the **IdP Adapter** screen, configure the adapter instance by referring to LinkedIn IdP Adapter settings. Click **Next**.
5. On the **Actions** screen, click **Test Connection**. Resolve any issues that are reported, and then click **Next**.
6. On the **Extended Contract** screen, add attributes that you want to include in the contract.
    1. If you selected **Retrieve email** in the adapter configuration, add the `email` attribute.
    2. Add the list of local attributes that you mapped on the **IdP Adapter** screen.
    3. Add any other LinkedIn attributes that you want to include in the contract.

        > **Note**
        >
        > The values for unmapped LinkedIn attributes will be populated with raw JSON and could include arrays. You can find more information in Object Types in the LinkedIn API documentation.

    4. Click **Next**
7. On the **Adapter Attributes** tab, set pseudonym and masking options as shown in Set pseudonym and masking options in the PingFederate documentation. Click **Next**.
8. On the **Adapter Contract Mapping** tab, configure the contract fulfillment details for the adapter as shown in Define the IdP adapter contract in the PingFederate documentation. Click **Next**.
9. On the **Summary** tab, check and save your configuration:

    Choose from:

    - For PingFederate 10.1 or later, click **Save**.
    - For PingFederate 10.0 or earlier, click **Done**. On the **Manage IdP Adapter Instances** tab, click **Save**.
10. If you created this adapter instance to replace a previous one, update the adapter mappings in any related service provider connections.

Core contract attributes

Attribute	Details
id	A unique identifying value for the member. This value is linked to your specific application. This attribute is only available when the LinkedIn member has authorized a permission that includes the *id* attribute, such as *r_liteprofile*.
access_token	The access token that LinkedIn provides to your application.
access_token_expires_in	The expiry date for the access token.
refresh_token	The refresh token that LinkedIn provides to your application. **Note** LinkedIn only provides refresh tokens to select partners.

Copyright © 2025 Ping Identity Corporation

Attribute	Details
refresh_token_expires_in	The expiry date for the refresh token.

JSON Pointer syntax reference

JavaScript Object Notation (JSON) Pointer defines a syntax for identifying a specific value within a JSON payload. Using the sample payload and JSON Pointer examples below, identify the attributes that you want to use to populate your attribute contract.

You can find a complete technical description of JSON Pointer syntax in JavaScript Object Notation (JSON) Pointer on ietf.org.

Learn more about LinkedIn JSON payload structure in Object Types in the LinkedIn API documentation.

Example JSON payload - r_liteprofile

```
{
    "lastName": {
        "localized": {
            "en_US": "Smith"
        },
        "preferredLocale": {
            "country": "US",
            "language": "en"
        }
    },
    "firstName": {
        "localized": {
            "en_US": "Janet"
        },
        "preferredLocale": {
            "country": "US",
            "language": "en"
        }
    },
    "profilePicture": {
        "displayImage": "urn:li:digitalmediaAsset:C4E03GAGHGR09Bdl__Q"
    },
    "id": "lUNPTvH2At"
}
```

JSON Pointer syntax:

Description	JSON Pointer	Example value
First name	/firstName/localized/en_US	Janet
Last name	/lastName/localized/en_US	Smith
Profile picture	/profilePicture/displayImage	urn:li:digitalmediaAsset:C4E03GAGHGR09Bdl__Q

Copyright © 2025 Ping Identity Corporation

Description	JSON Pointer	Example value
User ID	/id	lUNPTvH2At

LinkedIn IdP Adapter settings

Standard fields

Field Name	Description
Client ID (API Key on PingFederate 8.x and earlier)	The client ID that you noted in Register PingFederate as a Microsoft Application. This field is required.
Client Secret (Secret Key on PingFederate 8.x and earlier)	The client secret that you noted in Register PingFederate as a Microsoft Application. This field is required.
Permissions	The combined scopes of authority that PingFederate and the application want to request from the LinkedIn member. These permissions determine which LinkedIn attributes can be added in the **Attributes** section of the **IdP Adapter** screen. Use a space to separate multiple permissions. Type one or more of the following: ***r_liteprofile*** The ability read the member's user ID, first name, last name, maiden name, and profile picture. If you include this permission, the **LinkedIn User Data v2 URL** field is required. Learn more in Lite Profile Fields in the LinkedIn API documentation. ***w_member_social*** The ability to create a LinkedIn post on a behalf of the member. If you include this permission, the **LinkedIn User Data v2 URL** field is required. Learn more in Share on LinkedIn in the LinkedIn API documentation. The default value is *r_liteprofile*.
Error Redirect URL	(Optional) The URL of a custom page that displays when PingFederate receives an error response from LinkedIn. This URL can contain query parameters. The URL has an `errorMessage` query parameter appended to it that contains a brief description of the error. If this field is blank, users will see a generic error page. Learn more about using custom pages in Customizable user-facing pages in the PingFederate documentation. This field is blank by default.

Copyright © 2025 Ping Identity Corporation

Field Name	Description
Unauthorized Redirect URL	(Optional) The URL of a custom page that displays when PingFederate receives a response from LinkedIn that says the user declined the authorization request. This URL can contain query parameters. If no URL is specified, users will see a default error page. Learn more about using custom pages in Customizable user-facing pages in the PingFederate documentation. This field is blank by default.
Callback Endpoint	The PingFederate endpoint that LinkedIn uses to respond to authorization requests. If you set a custom endpoint in Register PingFederate as a Microsoft Application, change this field to match. This default value is `/linkedin-authn`.

Advanced fields

Field Name	Description
PingFederate Base URL	(Optional) The fully-qualified host name, port, and path (if applicable) of the PingFederate server. For example, `https://sso.example.com:9031`. This field affects the redirect URL set in the Register PingFederate as a Microsoft Application step. If this override field is blank, the URL is determined automatically. This field is blank by default.
LinkedIn Authentication v2 URL	The LinkedIn API endpoint that PingFederate uses to get the verification codes. The default value is `https://www.linkedin.com/oauth/v2/authorization`.
LinkedIn Access Token v2 URL	The LinkedIn API endpoint that PingFederate uses to get access tokens. The default value is `https://www.linkedin.com/oauth/v2/accessToken`.
Retrieve Email	When selected, PingFederate makes a separate call to the **LinkedIn User Email V2 URL** to get the user's email address. If you included *r_emailaddress* in the **Permissions** field, select this checkbox, and add `email` to your extended contract. This checkbox is cleared by default.
LinkedIn User Email V2 URL	The LinkedIn API endpoint that PingFederate uses to get user email addresses. This field is required if you have selected **Retrieve email**. The default value is `https://api.linkedin.com/v2/emailAddress?q=members&projection=(elements*(handle~))`.
LinkedIn User Data v2 URL	The LinkedIn API endpoint that PingFederate uses to get user attributes for the *r_liteprofile* and *r_fullprofile* permissions. This field is optional if you have selected **Retrieve email**. The default value is `https://api.linkedin.com/v2/me`.

Copyright © 2025 Ping Identity Corporation

Field Name	Description
Retry Request	Determines whether PingFederate will retry requests after it receives a response with a failure code. This checkbox is selected by default.
Maximum Retries Limit	Determines how many times PingFederate will retry a request. The default value is `5`.
Retry Error Codes	Determines which response codes are considered failures. The default value is `429`.
API Request Timeout	The amount of time in milliseconds that PingFederate allows when establishing a connection with LinkedIn or waiting for a response to a request. A value of `0` disables the timeout. The default value is `5000`.
Proxy Settings	Defines proxy settings for outbound HTTP requests. The default value is **System Defaults**.
Custom Proxy Host	The proxy server host name to use when **Proxy Settings** is set to **Custom**. This field is blank by default.
Custom Proxy Port	The proxy server port to use when **Proxy Settings** is set to **Custom**. This field is blank by default.

Integrating social sign-on into your application

To complete your LinkedIn sign-on integration, add a sign-on hyperlink to your application.

Steps

1. Create a service provider (SP) connection to your application as shown in SP connection management in the PingFederate documentation. Use your IdP adapter instance and Mapping an adapter instance in the PingFederate documentation.

   > **Note**
   >
   > Use the LinkedIn IdP Adapter instance as an authentication source. Learn more in Mapping an adapter instance in the PingFederate documentation.

2. If your application is outside the PingFederate domain, configure an SP connection.

   1. Create an SP connection that uses your IdP adapter instance as shown in SP connection management and Mapping an adapter instance in the PingFederate documentation.

   2. In your web application, create a hyperlink to allow users to sign on to the SP application. Use the following URL and replace the variables based on the descriptions in the following table.

      ```
      https://<pf_host>:<pf_port>/idp/startSSO.ping?PartnerSpId=<ConnectionId>
      ```

Copyright © 2025 Ping Identity Corporation

Variable	Description
<pf_host>	The host name or IP address of the PingFederate server.
<pf_port>	The port number for PingFederate.
<ConnectionId>	The federation identifier of the SP for the connection that uses the LinkedIn IdP Adapter instance.

3. If your application is inside the PingFederate domain, configure an adapter-to-adapter mapping.
    1. On the **System > Protocol Settings > Roles & Protocols** page, select **Enable Identity Provider (IdP) role and support for the following** and **Enable Service Provider (SP) role and support for the following**.
    2. In both the **Enable Identity Provider** and **Enable Service Provider** sections, select any protocol, such as **SAML 2.0**. Click **Save**.

        > **Note**
        >
        > PingFederate requires a protocol selection to activate the roles. The protocol that you select is not used for this integration.

    3. On the **Service Provider > Adapters** page, create or select an adapter instance that's integrated with the application as shown in SP application integration settings in the PingFederate documentation.
    4. On the **Identity Provider > Adapter-to-Adapter Mappings** page, configure the IdP-to-SP adapter mapping as shown in Adapter-to-adapter mappings in the PingFederate documentation.
    5. In your web application, create a hyperlink to allow users to sign on to the SP application. Use the following URL and replace the variables based on the descriptions in the table below:

        ```
        https://<pf_host>:<pf_port>/pf/adapter2adapter.ping?IdpAdapterId=<IdpAdapterId>&SpSessionAuthnAdapterId=<SpAdapterId>
        ```

Variable	Description
<pf_host>	The host name or IP address of the PingFederate server.
<pf_port>	The port number for PingFederate.
<IdpAdapterId>	The instance ID of the LinkedIn IdP Adapter instance.
<SpAdapterId>	The instance ID of the SP adapter instance that has been integrated with the application.

Troubleshooting

Copyright © 2025 Ping Identity Corporation

Troubleshooting

PingFederate can’t connect to the LinkedIn API

Description:

- When you configure an adapter instance, you receive an error when you click **Test Connection**.

Recommendation:

- Check the **Client ID** and **Client Secret** values for the PingFederate application that you registered on https://developer.linkedin.com/.
- Copy and paste the values directly into the adapter instance configuration, and run the test again.

LinkedIn presents the user with an "unauthorized" page

Description:

- LinkedIn presents the user with an "unauthorized" page that contains a link to an error page provided by the LinkedIn IdP Adapter instance.

Explanation:

- During authentication, the user declined to give authorization to LinkedIn. Ask the user to repeat the process and authorize LinkedIn.

> **Note**
>
> You can customize this error page. Refer to the **Unauthorized Redirect URL** field in LinkedIn IdP Adapter settings.

All SSO attempts fail

Description:

- For all users, attempting to sign on to an application through the LinkedIn IdP Adapter fails. The PingFederate server log includes the following error: "Attribute retrieval from LinkedIn failed".

Recommendation:

- Check whether you are using the correct LinkedIn API endpoint for the permissions that you have requested. In your adapter instance configuration, check that the following settings and fields are in agreement:
  - Permissions
  - Retrieve email
  - LinkedIn User Email V2 URL
  - LinkedIn User Data V2 URL

Learn more in LinkedIn IdP Adapter settings.

Copyright © 2025 Ping Identity Corporation

LinkedIn does not return values for some requested attributes

Description:

- After successful authentication and authorization, LinkedIn does not return values for some of the requested attributes.

Explanation:

- Some attributes require specific LinkedIn partner programs. Learn more in https://developer.linkedin.com/support/faq.

Enabling debug logging

To help with troubleshooting or monitoring, you can turn on activity logging for the adapter.

About this task

These steps are optional. You can find general information about logging in Enabling debug messages and console logging in the PingFederate documentation.

Steps

1. Open the `<pf_install>/pingfederate/server/default/conf/log4j2.xml` file for editing.
2. If you want to log activity for PingFederate and all adapters, do the following:
    1. Find the following section.

```
<AsyncRoot level="INFO" includeLocation="false">
        <!-- <AppenderRef ref="CONSOLE" /> -->
        <AppenderRef ref="FILE" />
</AsyncRoot>
```

    2. Change `INFO` to `DEBUG`.

```
<AsyncRoot level="DEBUG" includeLocation="false">
        <!-- <AppenderRef ref="CONSOLE" /> -->
        <AppenderRef ref="FILE" />
</AsyncRoot>
```

    3. If you want to see the adapter activity in the console, remove the comment tags.

```
<AsyncRoot level="INFO" includeLocation="false">
        <AppenderRef ref="CONSOLE" />
        <AppenderRef ref="FILE" />
</AsyncRoot>
```

3. To log activity just for the LinkedIn IdP Adapter, add the following line.

Copyright © 2025 Ping Identity Corporation

```
<Logger name="com.pingidentity.adapter.idp.linkedin" level="DEBUG"/>
```

4. Save the file.

Release notes

Changelog

The following is the change history for the LinkedIn Login Integration Kit.

LinkedIn Login Integration Kit 2.2 – December 2020

- Added a setting to use browser redirect or pop-up window for the sign-on presentation.
- Added customizable sign-on templates for the pop-up window presentation.
- Added customizable user-facing language-pack messages.
- Added support for the JavaScript Widget for the PingFederate Authentication API.
- Added support for the PingFederate authentication API.

LinkedIn Login Integration Kit 2.1 – February 2020

- Added the ability to retrieve user attributes and email addresses with the same adapter instance.

LinkedIn Login Integration Kit 2.0.1 – February 2019

- Fixed an issue that prevented the LinkedIn adapter from working in some OAuth scenarios.
- Changed some fields on the adapter configuration screen to match LinkedIn terminology.

LinkedIn Login Integration Kit 2.0 – January 2019

- Updated the adapter to conform to OAuth 2.0 standards and version 2.0 of LinkedIn's APIs. For a description of the API changes, see the Self-Serve v1 to v2 API Migration Frequently Asked Questions in the LinkedIn documentation.

LinkedIn Login Integration Kit 1.1.1 – August 2018

- Simplified logout handling within the adapter by removing Perform Logout check box and unnecessary iframe based redirects.

LinkedIn Login Integration Kit 1.1 – October 2017

- Compatibility updates for the latest LinkedIn API.

LinkedIn Login Integration Kit 1.0.1 – July 2013

- Added masking to the application ID and application Secret fields in the adapter instance configuration screen.
- Added support for Scribe 1.3.3.

Copyright © 2025 Ping Identity Corporation

LinkedIn Login Integration Kit 1.0 – February 2011

- Initial release.

Known issues and limitations

Known issues

There are no known issues.

Known limitations

- LinkedIn does not support single logout (SLO).

Download manifest

The following files are included in the LinkedIn Login Integration Kit `.zip` archive.

- `ReadMeFirst.pdf` : Contains links to this online documentation
- `legal/Legal.pdf` : Copyright and license information
- `dist` : Contains the integration files
  - `deploy` : Contains the Java libraries
    - `pf-linkedin-adapter-<version>.jar` : JAR file that contains the LinkedIn IdP Adapter
  - `conf` : Contains the HTML template that presents the LinkedIn sign-on form
    - `language-packs` : Contains files with customizable user-facing messages
      - `authn-api-messages.properties` : A version of the common PingFederate authentication API variable file that includes adapter-specific messages
      - `pingfederate-linkedin-adapter-messages.properties` : A variable file that customizes the messages that appear on the template files
    - `template` : Contains user-facing HTML template files
      - `linkedin-pop-up-template.html` : Template that opens a pop-up window to prompt the user to sign on
      - `linkedin-post-auth-template.html` : Template that returns the user to the PingFederate sign-on flow after they sign on with LinkedIn
      - `assets` : Contains functional scripts and files used by the template
        - `css/linkedin.css` : CSS file that customizes the appearance of the template files
        - `fonts/end-user` : Contains template fonts and icons
          - `icons` : Contains template icons

Copyright © 2025 Ping Identity Corporation

        - `proxima-nova`: Contains template fonts
    - `images`: Contains template image files
        - `ping-logo.svg`: Image file with company branding
- `lib/pf-authn-api-sdk-<version>.jar`: JAR file that contains the PingFederate Authentication API SDK

Copyright © 2025 Ping Identity Corporation

Lucidchart Provisioner

The PingFederate Lucidchart Provisioner enables an enterprise to provision users to Lucidchart.

A quick connection template is also included to simplify the configuration of single sign-on (SSO). Additional details on Lucidchart can be found on the Lucidchart website.

Features

- Browser-based SP and IdP-initiated SSO
- Includes support for user life cycle management (including creates, updates, disables and deletes)
- Includes configuration options for workflow capabilities (for example, the ability to disable updates)

Intended audience

This document is intended for PingFederate administrators.

If you need help during the setup process, see the following sections of the PingFederate documentation:

- SP connection management
- Identity provider SSO configuration
- Datastores
- Configuring outbound provisioning
- Configuring outbound provisioning settings

System Requirements

- PingFederate 7.3 or later with Java 8
- Might require the following endpoints to be whitelisted on the firewall to allow outbound connections:
  - https://www.lucidchart.com/scim/v2/chart
- An Enterprise account is required for SCIM and SAML single sign-on with Lucidchart. To upgrade to Enterprise, visit Lucidchart's pricing page or contact sales.

User management

The Lucidchart Provisioner synchronizes users from your datastore to Lucidchart. The following describes the behavior of each provisioning capability.

> **Tip**
>
> You can configure the following capabilities and specify which users to provision when you get to the Configure provisioning part of the setup process.

Copyright © 2025 Ping Identity Corporation

Synchronizing existing users

PingFederate synchronizes users based on the `Username` attribute in Lucidchart. If a user already exists in your datastore and Lucidchart, mapping this attribute correctly links the two records together.

For example:

- In Lucidchart, Janet's `Username` is `bjensen`.
- In your datastore, Janet's `sAMAccountName` is `bjensen`.
- On the **Attribute Mapping** tab of your provisioning connection configuration, map the `Username` attribute to `sAMAccountName`.
- When the provisioning connector runs, the datastore user is provisioned with a `Username` of `bjensen`. That matches Janet's existing `Username` in Lucidchart, so her information in the datastore is synchronized to her Lucidchart account.

User provisioning

PingFederate provisions users when any of the following happens:

- A user is added to the datastore group or filter that is targeted by the provisioning connector.
- A user with `disabled` status is added to the datastore group or filter that is targeted by the provisioning connector, and the **Provision disabled users** provisioning option is enabled. This feature is not available in all provisioning connectors.

> **Tip**
>
> You can define which users PingFederate targets for provisioning on the **Source Location** tab of your provisioning connection configuration.

User updates

PingFederate updates users when a user attribute changes in your datastore.

> **Tip**
>
> You can define which attributes PingFederate monitors for changes on the **Attribute Mapping** tab of your provisioning connection configuration.

User deprovisioning

PingFederate deprovisions users when any of the following happens:

- A user is deleted from the user store.
- A user is disabled in the user store.
- A user is removed from the datastore group or filter that is targeted by the provisioning connector.

Copyright © 2025 Ping Identity Corporation

Setup

Obtain base URL and bearer token

About this task

Below are the tasks to enable SCIM provisioning on the Lucidchart Admin Portal.

Steps

1. Log into your Lucidchart account as an administrative user for your organization.
2. Go to **Team > App Integration**.
3. Under **App Integration** select **SCIM**.
4. Click **Generate Token**.
5. On the **SCIM** screen, note the displayed values for later use.
    - **Base URL**: This is the base URL for Lucidchart.
    - **Bearer Token**: This token is used by the connector to make authenticated API calls to Lucidchart.

Install the connector

About this task

This section describes the common steps required to install the PingFederate Lucidchart Connector.

Steps

1. Stop the PingFederate server if it is running.
2. Unzip the Lucidchart Connector distribution `.zip` file.
3. Copy the contents of the `dist` directory into the directory:

    `<pf_install>/pingfederate/server/default/deploy`
4. **Optional:** If you plan to use the connector for outbound provisioning, edit the `run.properties` file located in `<pf_install>/pingfederate/bin`, changing the property `pf.provisioner.mode` to `STANDALONE`. For example:

    `pf.provisioner.mode=STANDALONE`

    > **Note**
    >
    > For information about using the FAILOVER setting for runtime deployment, see Deploying provisioning failover.
5. Start the PingFederate server.

Copyright © 2025 Ping Identity Corporation

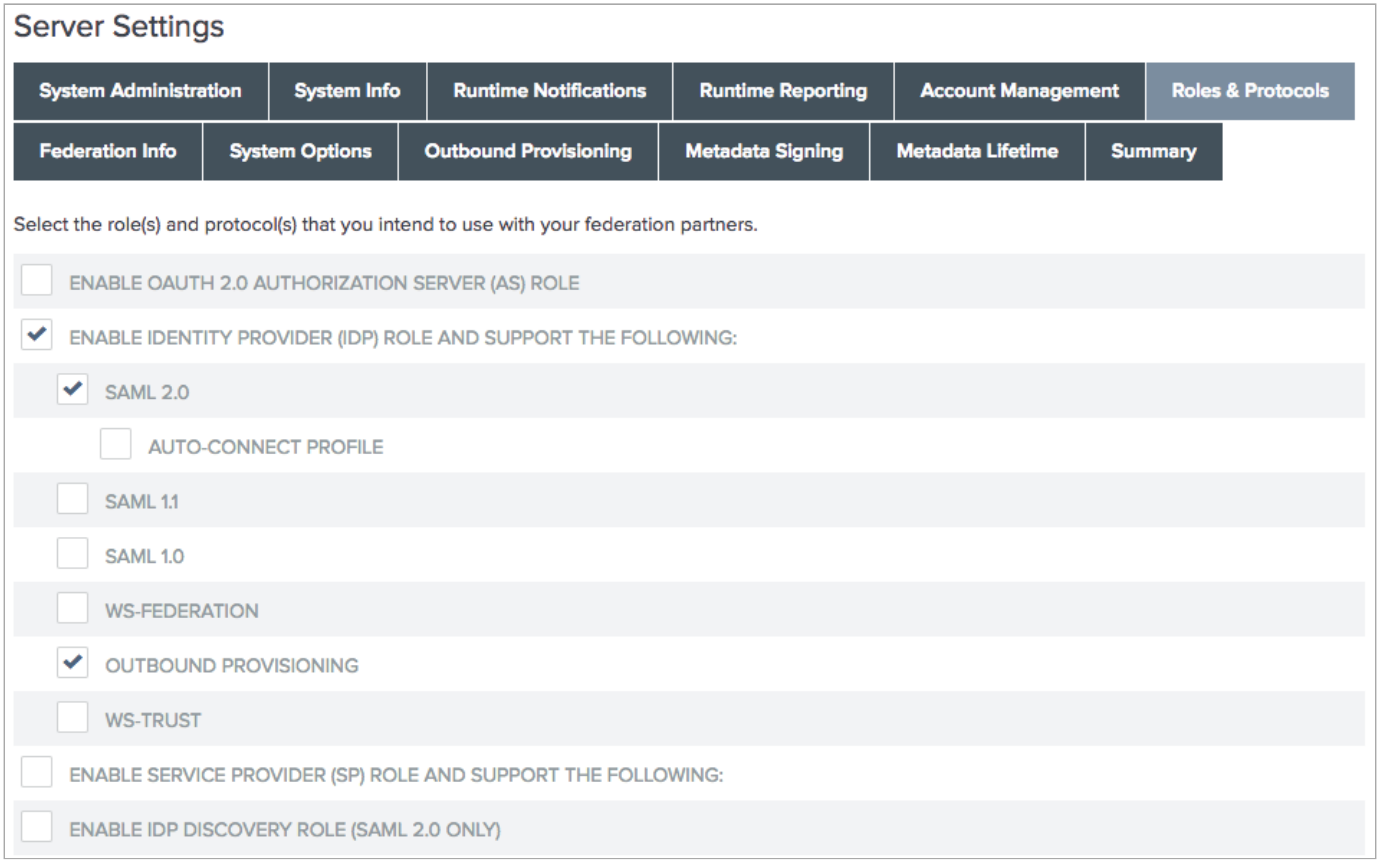
Enable outbound provisioning

About this task

After enabling outbound provisioning in the `<pf_install>/pingfederate/bin/run.properties` file, you must also activate the outbound provisioning role in the administrative console.

Steps

1. Go to the **Server Configuration > Server Settings > Roles & Protocol** screen.
2. Select the **Outbound Provisioning** check box.



> **Note**
>
> Enabling outbound provisioning adds the outbound provisioning screen and requires the selection of a database to facilitate provisioning. For more information, see Configuring outbound provisioning settings.

Configure provisioning

About this task

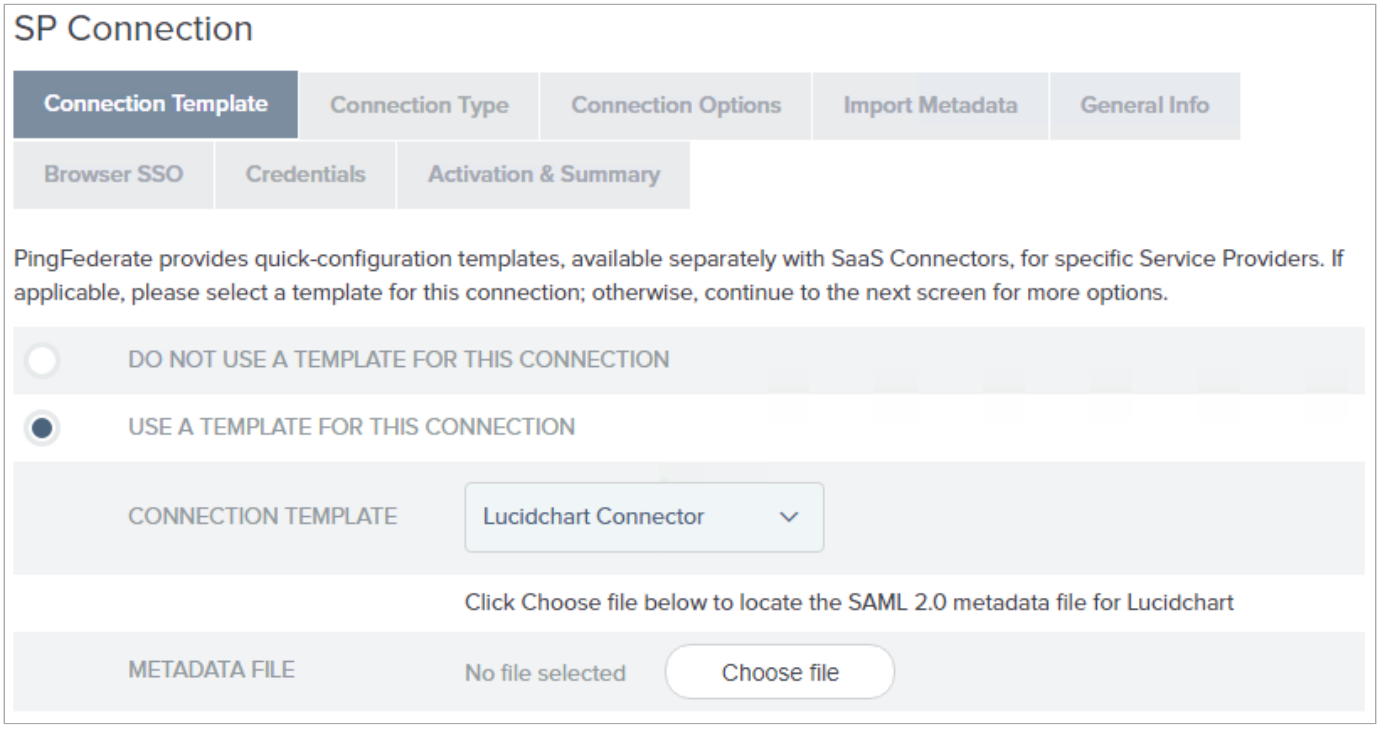
To configure a connection for outbound provisioning to Lucidchart, follow the instructions in this section.

Copyright © 2025 Ping Identity Corporation

Outbound provisioning details are managed within an SP connection. You can configure outbound provisioning with or without Browser SSO, WS-Trust STS, or both when you create a new SP connection. You can also add outbound provisioning to an existing SP connection.

Steps

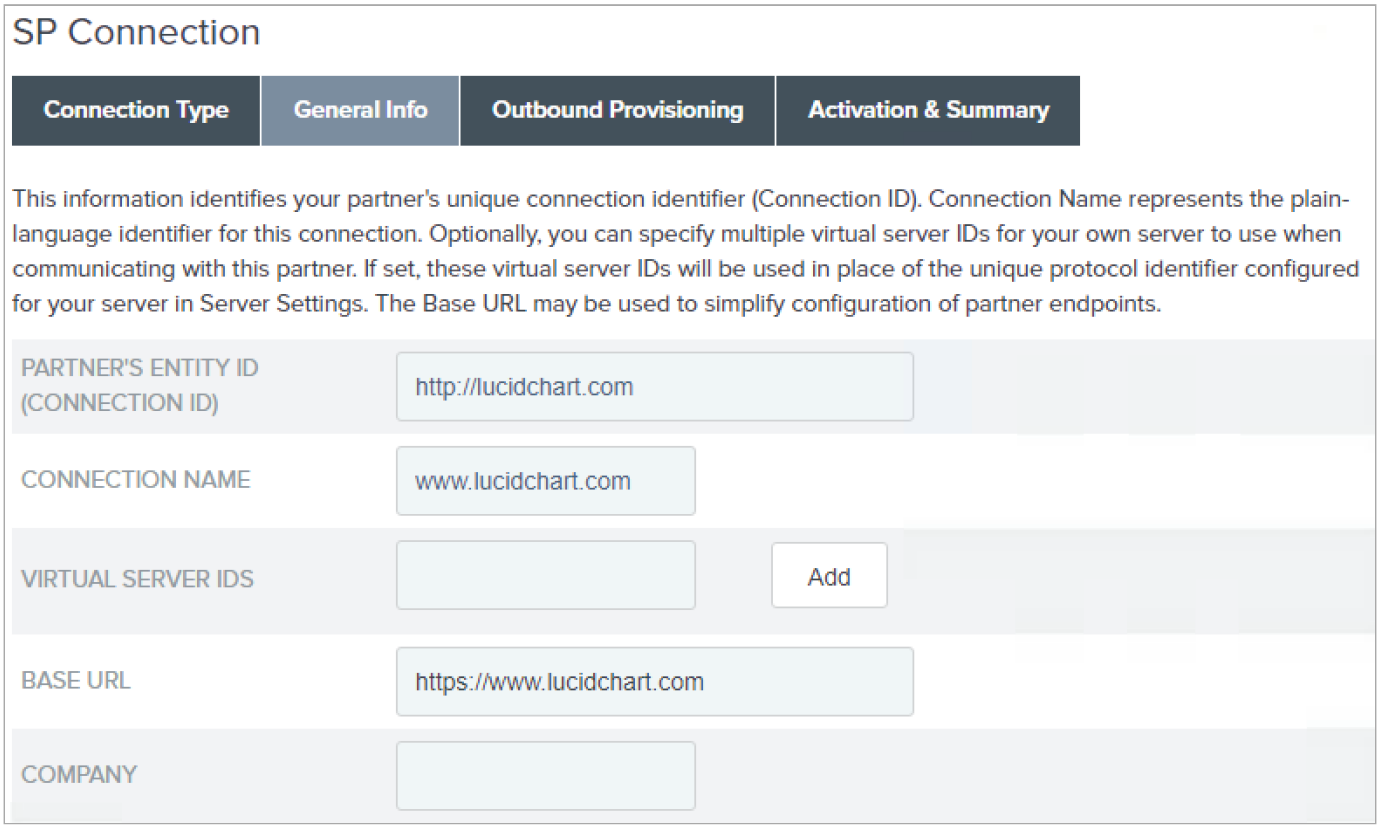
1. In the PingFederate administrator console, configure the data store that PingFederate will use as the source of user data. For instructions, see Datastores in the PingFederate documentation.
    - When targeting users and groups for provisioning, exclude the user account that you will use to administer users in your connection to Lucidchart. This prevents the PingFederate provisioning engine from interfering with the account that provisions users and groups.
2. Create a new SP connection or select an existing SP connection from the **SP Configuration** menu.
3. On the **Connection Template** screen, select **Use a template for this connection** and choose **Lucidchart Connector** from the **Connection Template** drop-down list. When asked during the connection configuration steps, import the `lucidchart-saml-metadata.xml` packaged with this connector.



> **Tip**
>
> If this selection is not available, verify the connector installation and restart PingFederate.

4. On the **Connection Type** screen, ensure the **Outbound Provisioning** checkbox is selected, and the **Browser SSO Profiles** checkbox is cleared (if appropriate).
5. On the **General Info** screen, the default values are taken from the metadata file you selected in step 2. We recommend using the metadata default values.

Copyright © 2025 Ping Identity Corporation



6. Follow the connection wizard to configure the connection.
7. On the **Outbound Provisioning** screen, click **Configure Provisioning**.
8. On the **Target** screen, enter the values for each field as required by the Lucidchart Connector.

Copyright © 2025 Ping Identity Corporation

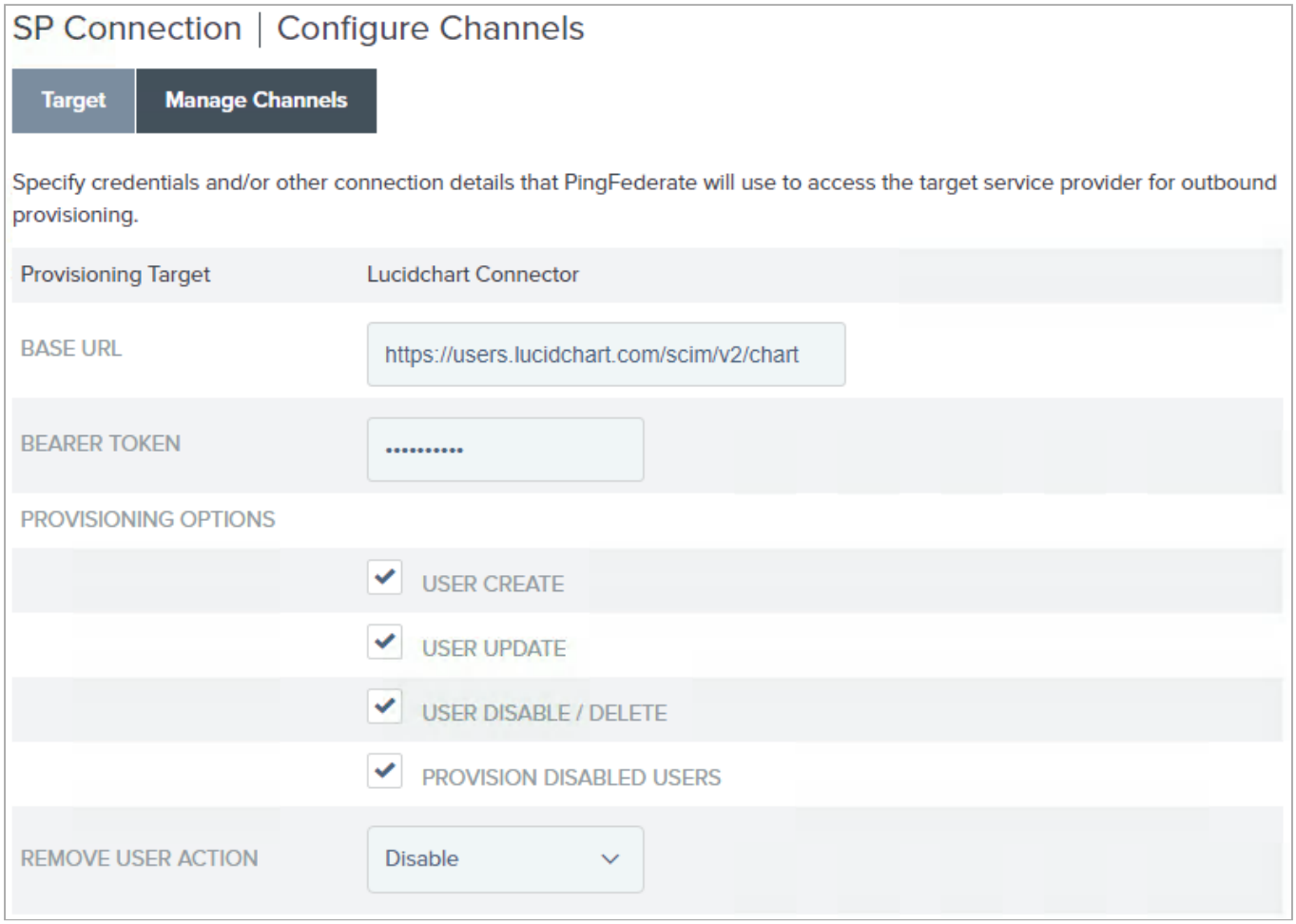


Table 1. Target screen options

Field Name	Description
Base URL	The base URL for Lucidchart. For more information on obtaining the base URL, see Obtain base URL and bearer token.
Bearer Token	The bearer token used by the connector to make authenticated API calls to Lucidchart. For more information on obtaining the bearer token, see Obtain base URL and bearer token.
Provisioning Options	
User Create	**True** (default) – Users will be created in Lucidchart. **False** – Users will not be created in Lucidchart. **Note** The `provisioner.log` will display a warning within the create user workflow that the user was not created in Lucidchart.

Copyright © 2025 Ping Identity Corporation

Field Name	Description
User Update	**True** (default) – Users will be updated in Lucidchart. **False** – Users will not be updated in Lucidchart. **Note** The `provisioner.log` will display a warning within the update user workflow that the user was not updated in Lucidchart.
User Disable / Delete	**True** (default) – Users will be disabled or deleted in Lucidchart. **Note** A disabled user can only be re-enabled if **User Update** is true. **False** – Users will not be disabled or deleted in Lucidchart. **Note** `The provisioner.log` will display a warning within the user workflow that the user was not disabled or deleted in Lucidchart.
Provision Disabled Users	This option is only relevant if you select **User Create**. **True** (default) – If a disabled user in the user store is targeted for provisioning, it will be created in a disabled state in Lucidchart. **False** – If a disabled user in the user store is targeted for provisioning, it will be not be created in Lucidchart. **Note** The `provisioner.log` will display a warning within the create user workflow indicating that the user was not created in Lucidchart.
Remove User Action	Select a deprovision method (**Disable** or **Delete**). Deprovisioning is triggered when previously provisioned users no longer meet the condition set in the **Source Location** screen, or when a user has been suspended or deleted from the data store. This option is only applicable if **User Disable / Delete** is set to True. **Disable** (default) – Deactivates the user account in Lucidchart (also known as a soft delete). **Delete** – Removes the user account in Lucidchart (also known as a hard delete).

9. Click **Next** to continue the provisioning configuration. For more information, see the following sections under Outbound provisioning for IdPs in the PingFederate documentation:
    - Managing channels
    - Specifying channel information
    - Identifying the source datastore
    - Modifying source settings

Copyright © 2025 Ping Identity Corporation

- Specifying a source location
- Mapping attributes
- Reviewing channel settings

> **Note**
>
> Credentials will be verified when the channel and SP connection is set to Active and provisioning is initiated.

> **Tip**
>
> If you are not ready to complete the provisioning configuration, you can click **Save** and return to the configuration page later. To return to the configuration page, select the connection from **Identity Provider > SP Connections > Manage All**.

Supported attributes reference

The following table consists of the attributes that can be mapped for user provisioning.

Attribute	Description
Username	Lucidchart's unique identifier for the user. A username cannot be updated. This attribute is required.
Email	The email for the user (for example, "bjensen@example.com"). This attribute is required.
First Name	The given name of the user, or first name in most Western languages (for example, 'Barbara' given the full name 'Ms. Barbara Jane Jensen, III'). This attribute is required.
Last Name	The family name of the user, or last name in most Western languages (for example, 'Jensen' given the full name 'Ms. Barbara Jane Jensen, III'). This attribute is required.
External ID	A string that is an identifier for the resource as defined by the provisioning client. For relevant Lucidchart API limitations, see Known issues and limitations.
Roles	The user's role or roles. The role used here must already exist in Lucidchart; the application does not create new roles. For relevant Lucidchart API limitations, see Known issues and limitations.

Configure SAML SSO

The following section describes the steps for configuring single sign-on (SSO) to Lucidchart.

Copyright © 2025 Ping Identity Corporation

> **Note**
>
> Configuring SSO is optional for outbound provisioning.

- Configure Lucidchart for SSO
- Configure PingFederate for SSO

Configure Lucidchart for SSO

Before you begin

To configure Lucidchart for SSO you will require metadata from your PingFederate Identity Provider (IdP) setup. For more information, see Exporting selected SAML metadata in the PingFederate documentation.

To export your IdP metadata:

1. Go to **Server Configuration > Metadata Export**.
2. On the **Metadata Mode** screen, choose **Select information to include in metadata manually** and click **Next**.
3. On the **Protocol** screen, click **Next**.
4. On the **Attribute Contract** screen, click **Next**.
5. On the **Signing Key** screen, select the PingFederate signing certificate for use and click **Next**.
6. Follow the rest of the workflow to export a metadata XML file, including selecting the certificate to sign the metadata XML file in the **Metadata Signing** screen (as needed).

To configure Lucidchart for SSO:

> **Note**
>
> For more information on how to set up SSO for Lucidchart, see SAML and SCIM: An Overview in the LucidChart documentation.

Steps

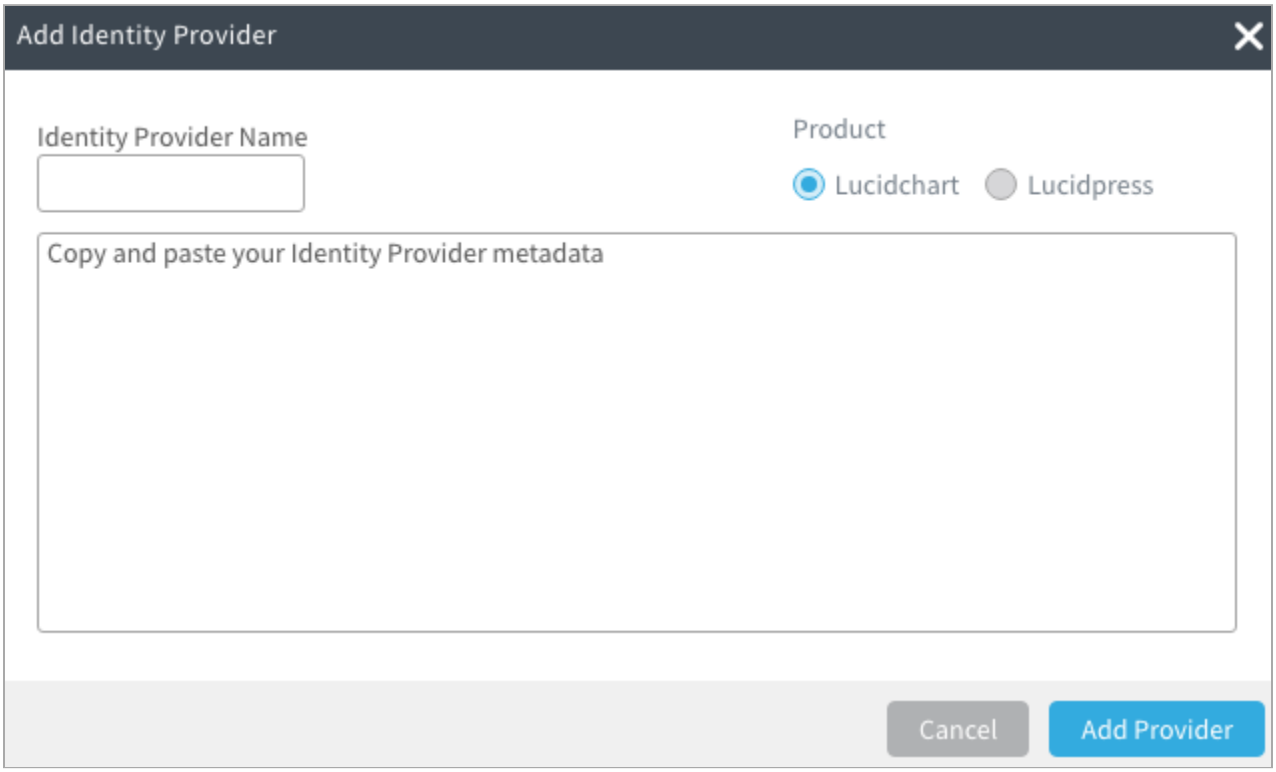
1. Log into your Lucidchart account as an administrative user for your organization.
2. Go to **Team > App Integration > SAML**.
3. Select the checkbox for **Enable SAML Integration**.
4. Enter your Lucidchart account domain in the **Domain** field under the section Lucidchart Sign in URL.

   > **Note**
   >
   > Enter the domain only, not a full URL. The SAML integration will use your domain to generate a Lucidchart sign-in URL that will be provided to PingFederate. For example, if you enter acme.com as your domain, the URL will be https://www.lucidchart/saml/sso/acme.com. A user may go directly to this URL to initiate SAML single sign on.

5. Under the Identity Providers section, select **Add Identity Provider**.

Copyright © 2025 Ping Identity Corporation

6. Open your IdP metadata `.xml` file using a text editor. Copy the text from the `.xml` file and paste it into the text box under **Identity Provider Metadata**.
7. Provide a description for **Identity Provider Name**, select the product **Lucidchart** and click **Add Provider**.



8. Under Download Service Provider Metadata, select **Download Metadata** and **Save changes**.



Configure PingFederate for SSO

About this task

The following section describes the steps for configuring single sign-on (SSO) to Lucidchart. Configuring SAML SSO involves both configuring PingFederate SP connection and the Lucidchart SSO screens.

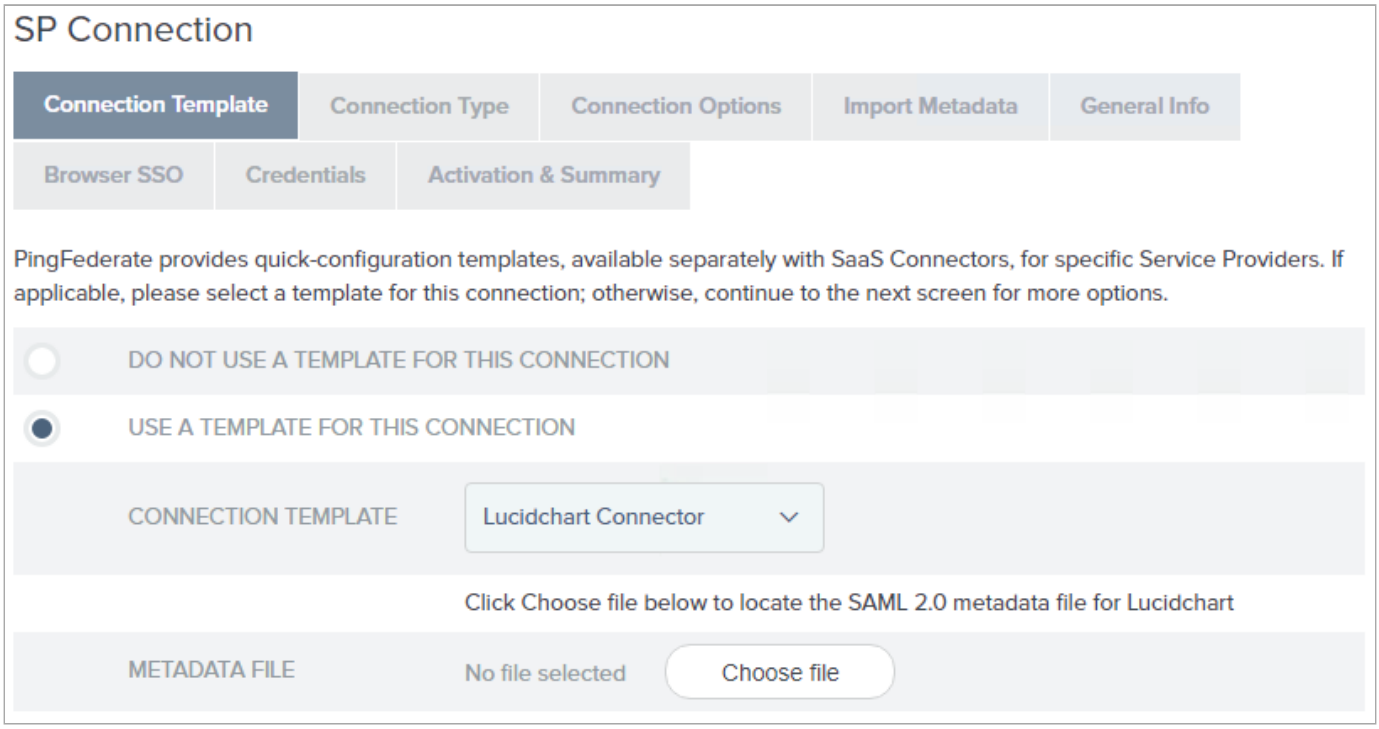
> **Note**
>
> Configuring SSO is optional for outbound provisioning.

To configure PingFederate for SSO:

Copyright © 2025 Ping Identity Corporation

Steps

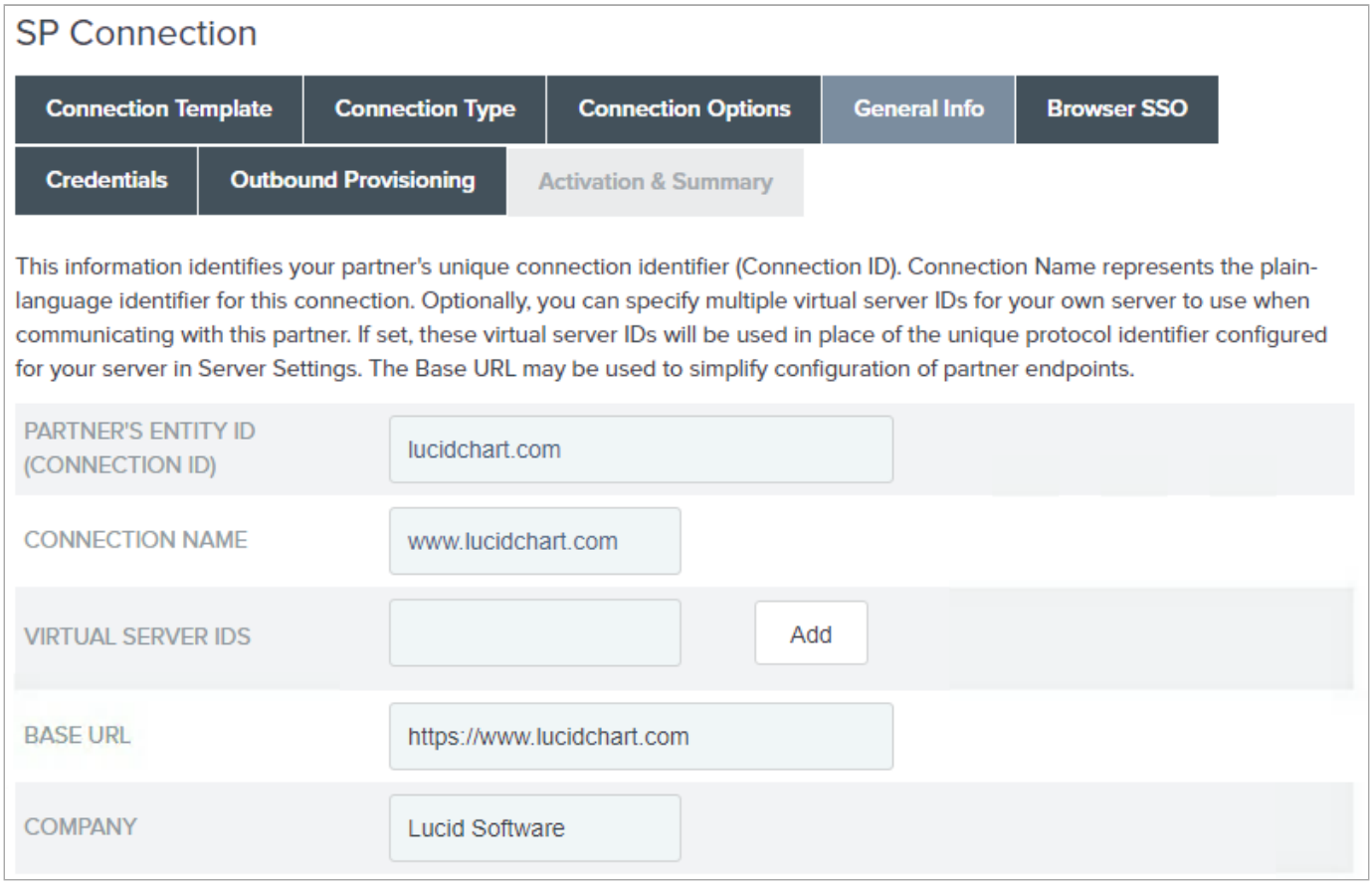
1. Create a new SP connection or select an existing SP connection from the **SP Configuration** menu.
2. On the **Connection Template** screen, select the **Use a template for this connection** option and choose **Lucidchart Connector** from the **Connection Template** drop-down list. You will be asked to provide the `metadata.xml` file you obtained earlier in Configure Lucidchart for SSO.



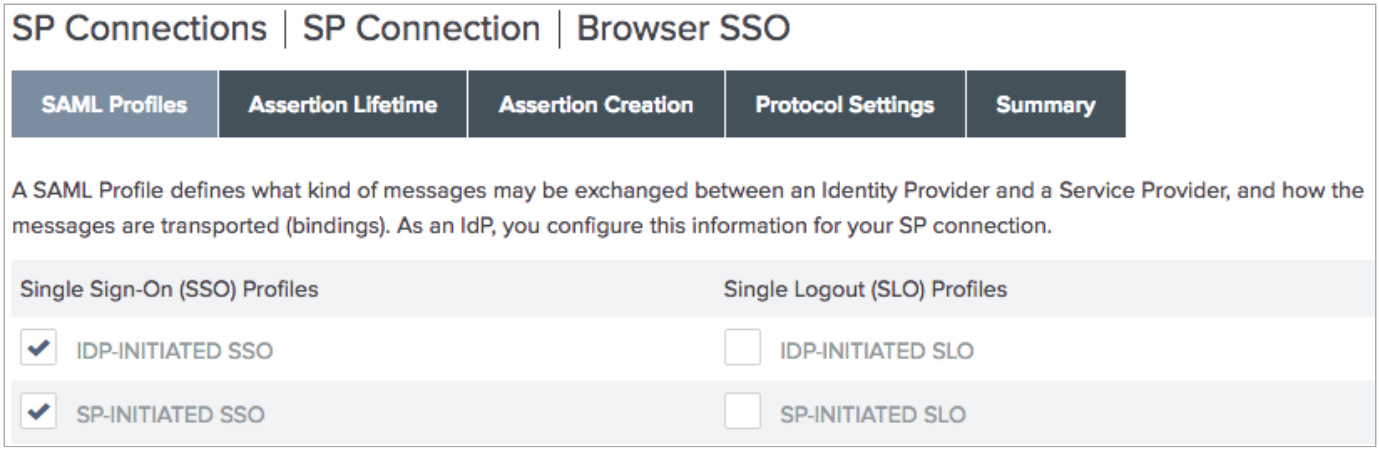
> **Tip**
>
> If this selection is not available, verify the connector installation and restart PingFederate.

3. On the **Connection Type** screen, ensure that the **Browser SSO Profiles** checkbox is selected.
4. On the **General Info** screen, the default values are taken from the metadata file you selected in an earlier step. We recommend using the metadata default values.

Copyright © 2025 Ping Identity Corporation

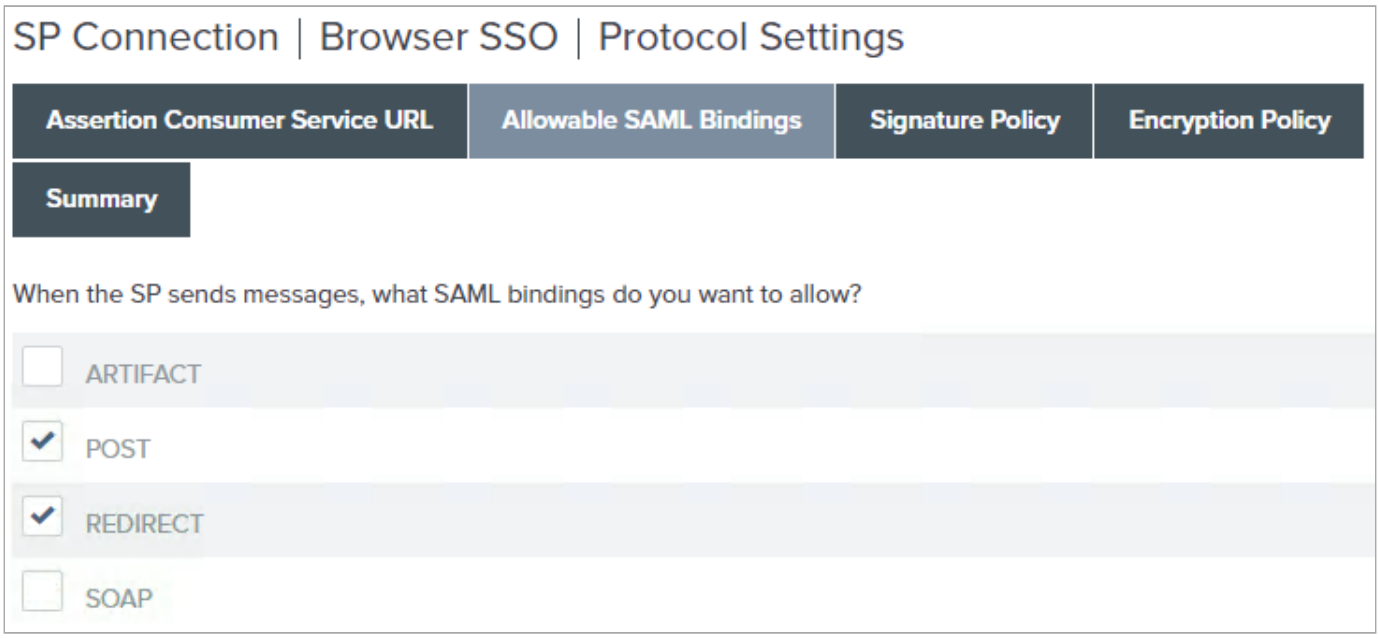


5. Click **Next** to continue the Browser SSO configuration. For more information, see the following sections under Identity provider SSO configuration:
    - Managing IdP adapters
    - Configure IdP Browser SSO
    - Configuring credentials
6. On the **Browser SSO > SAML Profiles** screen, ensure that the IdP-initiated SSO and SP-initiated SSO profiles are selected and click **Next**.



Copyright © 2025 Ping Identity Corporation

7. On the **Browser SSO > Protocol Settings > Allowable SAML Bindings** screen, ensure that the **POST** and **Redirect** profiles are selected (clear **Artifact** and **SOAP**). Click **Next**.



8. On the **Credentials > Digital Signature Settings** screen, select the signing certificate.
9. On the **Activation & Summary** screen, set **Connection Status** to **ACTIVE**, then click **Save**.

Release notes

Changelog

Lucidchart Connector 1.0 – June 2018 (current release)

- Initial release
- Added support for user life cycle management (including creates, updates, disables and deletes)
- Added configuration options for workflow capabilities (for example, the ability to disable updates)

Known issues and limitations

- Due to a limitation with PingFederate 8.1 and earlier versions, when configuring two SP connections with the same provisioner, the second connection built may be pre-populated with the channel from the first connection. To avoid conflicts, delete this pre-populated channel and create a unique channel for each connection.
- When an LDAP user is deleted in a targeted group distinguished name (DN), the provisioning connector does not propagate the deletion until a new user is added to the group. This limitation is compounded when the **User Create** provisioning option is disabled. For solutions, see SaaS provisioner does not remove the user in the Knowledge Base.
- Due to PingFederate limitations, user attributes cannot be cleared once set.

Copyright © 2025 Ping Identity Corporation

Performance limitations

- Due to Lucidchart API limitations, there will be a performance impact to creating users when mapping External ID or Roles. Both External ID and Roles may fail to be added to a user on the initial create. Should this happen, an exception will be thrown and an update to the user will be attempted on the next provisioning cycle.
- Due to Lucidchart API limitations, attempting to update a user immediately after creating them may result in user not found exceptions. This is due to a delay in Lucidchart between creating a user and being able to modify the user. Failed attempts to update the user will be reattempted on the next provisioning cycle.

Download manifest

The distribution `.zip` archive for the connector contains the following:

- `ReadMeFirst.pdf` – contains links to this online documentation.
- `lucidchart-saml-metadata.xml` – SAML metadata file for use with a Lucidchart connection. For more information, see Configure provisioning.
- `/legal`:
  - `Legal.pdf` – copyright and license information.
- `/dist` – contains libraries needed for the connector:
  - `pf-lucidchart-quickconnection-[version].jar` – PingFederate Lucidchart Connector

Copyright © 2025 Ping Identity Corporation

Microsoft EAM Integration Kit

The Microsoft EAM Integration Kit processes the **id_token_hint** and **claims** parameters sent by Microsoft Entra ID's external authentication mechanism (EAM). It also sets `acr` and `amr` values for downstream adapters in the PingFederate authentication policy.

Typically, a downstream adapter like PingID uses these values to perform multi-factor authentication (MFA).

Components

Microsoft EAM IdP Adapter

When PingFederate receives an OpenID Connect (OIDC) request from Microsoft Entra ID, the adapter validates the **id_token_hint** and extracts the `acr` and `amr` values from the **claims** parameter. Additional claims can be exported out of the **id_token_hint**.

Intended audience

This document is intended for PingFederate administrators.

You can find help during the setup process in the following resources:

- PingID Provisioner
- The following sections of the PingFederate documentation:
  - Authentication Policies
  - OIDC Policies
  - Configuring OAuth clients

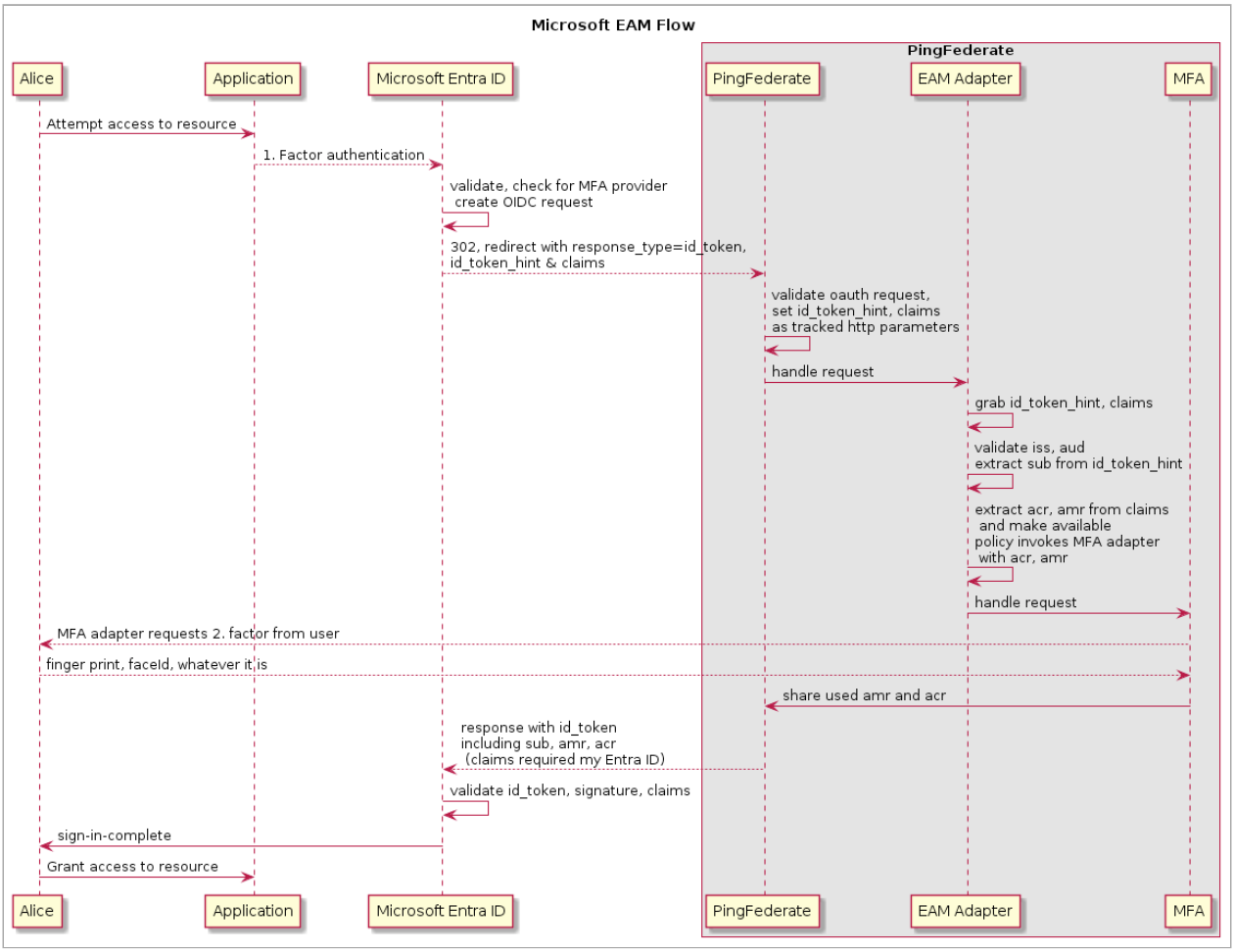
System requirements

- PingFederate 11.3 or later.
- A Microsoft Entra ID account with external authentication method enabled.

Overview of the SSO flow

The following figure demonstrates an example single sign-on (SSO) process flow.

Copyright © 2025 Ping Identity Corporation



In summary:

1. A user initiates the sign-on process by requesting access to an application that's protected by Microsoft Entra ID.
2. The user authenticates with Microsoft Entra ID using a primary authentication method.
3. Microsoft Entra ID makes an OpenID Connect (OIDC) request to PingFederate for secondary authentication.
4. PingFederate validates the OIDC request, then sets the **id_token_hint** and claims provided by Microsoft Entra ID as tracked parameters.
5. The Microsoft EAM IdP Adapter picks up the **id_token_hint** and claims; validates the **id_token_hint**; then sets the `sub`, `acr`, and `amr` values for downstream adapters like PingID.
6. The downstream adapter requests a second authentication factor from the user.
7. The user authenticates with the downstream adapter using a secondary authentication method.
8. The downstream adapter shares the used `amr` and `acr` values with PingFederate.
9. PingFederate sends a response to Microsoft Entra ID with the id_token, including the `sub`, `acr`, and `amr` claims required by Microsoft Entra ID.
10. Microsoft Entra ID validates the id_token, signature, and claims.
11. Microsoft Entra ID grants the user access to the protected application.

Copyright © 2025 Ping Identity Corporation

Setup

Deploying the integration files

To get started with the integration, deploy the Microsoft EAM Integration Kit files to your PingFederate directory.

Steps

1. Download the Microsoft EAM Integration Kit `.zip` archive from the **Add-ons** tab of the PingFederate Downloads page.
2. Stop PingFederate.
3. Extract the `.zip` archive and merge the contents of the `dist` directory with your `<pf_install>/pingfederate/server/default/deploy` directory.
4. Start PingFederate
5. If you operate PingFederate in a cluster, repeat steps 2 - 4 for each engine node.

Next steps

Configure an adapter instance.

Configuring an adapter instance

Configure the Microsoft EAM IdP Adapter to determine how PingFederate communicates with Microsoft Entra ID.

About this task

To begin the integration, deploy the Microsoft EAM Integration Kit files to your PingFederate directory.

Steps

1. In the PingFederate administrative console, go to **Authentication > Integration > IdP Adapters** and click **Create New Instance**.
2. On the **Type** tab, set the basic adapter instance attributes:
    1. In the **Instance Name** field, enter a name for the adapter instance.
    2. In the **Instance ID** field, enter a unique identifier for the adapter instance.
    3. From the **Type** list, select **Microsoft EAM IdP Adapter**. Click **Next**.
3. On the **IdP Adapter** tab, configure the adapter instance by referring to Microsoft EAM IdP Adapter settings reference. Click **Next**.
4. On the **Extended Contract** tab, add any attributes that should be extracted out of the **id_token_hint**. Click **Next**.
5. On the **Adapter Attributes** tab, set pseudonym and masking options as shown in Set pseudonym and masking options in the PingFederate documentation. Click **Next**.

Copyright © 2025 Ping Identity Corporation

6. On the **Adapter Contract Mapping** tab, configure the contract fulfillment details for the adapter as shown in Define the IdP adapter contract in the PingFederate documentation. Click **Next**.
7. On the **Summary** tab, review your configuration. Click **Save**.

Next steps

Track HTTP request parameters in the authentication policy.

Microsoft EAM IdP Adapter settings reference

Field descriptions for the Microsoft EAM IdP Adapter configuration screen.

Standard fields

Field	Description
Entra Tenant ID	The Azure (Microsoft Entra ID) Tenant ID.
Entra App ID	The application ID of the protected resource in Microsoft Entra ID.
Location of acr	The JSON Pointer location of the **acr** options within the claims parameter. By default, this does not need to be updated.
Location of amr	The JSON Pointer location of the **amr** options within the claims parameter. By default, this does not need to be updated.

Tracking HTTP request parameters

Configure tracked HTTP parameters in the PingFederate authentication policy for the Microsoft EAM IdP Adapter to reference.

Learn more about tracked HTTP parameters in defining authentication policies.

Steps

1. In the PingFederate administrative console, go to **Authentication > Policies** and click the **Tracked HTTP Parameters** tab.
2. In the **Parameter Name** field, enter **id_token_hint** and **claims**. Click **Add**.
3. Click **Save**.

Next steps

If you plan to use PingID as an MFA factor, export amr as an additional attribute for PingID.

Copyright © 2025 Ping Identity Corporation

Exporting amr as an additional attribute for PingID

If you plan to use PingID as an MFA factor, configure the PingID adapter to export `amr` as an additional attribute.

Steps

1. In the PingFederate administrative console, open the PingID adapter configuration and go to the **Extended Contract** tab.
2. In the **Extend the Contract** field, enter `amr`. Click **Add**.
3. Go to the **Adapter Contract Mapping** tab and click **Configure Adapter Contract**.
4. Convert PingID-specific `amr` values into Microsoft Entra ID-specific `amr` values:
    1. Go to the **Adapter Contract Fulfillment** tab.
    2. Find `amr` in the **Contract** column.
    3. In the corresponding **Source** list, select **Expression**.
    4. In the corresponding **Value** field, enter the following OGNL expression:

```
#xref = #{
"FIDO2":"fido",
"FIDO2_BIOMETRICS":"fido",
"BYPASS":"bypass",
"MOBILE_APP_BIOMETRICS":"bio",
"MOBILE_APP_SWIPE":"swk",
"NUMBER_MATCHING":"swk",
"MOBILE_APP_OTP":"otp",
"POLICY_APPROVE":"pop",
"EMAIL":"otp",
"SMS":"sms",
"VOICE":"tel",
"YUBIKEY":"hwk",
"OATH_TOKEN":"hwk",
"AUTHENTICATOR_APP":"otp",
"DESKTOP_OTP":"otp",
"SECURITY_KEY":"fido" },
#action = #this.get("pingid.authentication.type").toString(),
#result = (#action != null) && (#xref[#action] != null) ? #xref[#action] : "no_mapping",
#result
```

5. Click **Done**.

Configuring PingFederate to use the Microsoft EAM IdP adapter

Steps

1. Enable static signing keys in the OAuth and OpenID Connect (OIDC) key configuration:

Copyright © 2025 Ping Identity Corporation

Learn more in configuring static signing keys.

1. Go to **Security > Certificate & Key Management > OAuth & OpenID Connect keys** and select **Enable Static Keys**.
2. In the **Signing Keys** section, go to the RSA **Key Type**, select an **Active Signing Certificate**, and select the **Publish Certificate** checkbox.
3. Click **Save**.

2. Configure an access token manager (ATM).

   Learn more on the JSON token management tab.

3. Make the following adjustments to the OIDC policy configuration:

   Learn more in configuring policy and ID token settings. Use the following settings:

   1. Go to **Applications > OpenID Connect Policy Management** and open the policy configuration that you plan to use.
   2. On the **Manage Policy** tab, in the **Access Token Manager** list, select the ATM that you configured in the previous step.
   3. On the **Manage Policy** tab, select the **Include x.509 Thumbprint Header in ID token** checkbox.
   4. Configure the OIDC policy to expose the X5T header when PingFederate issues the id_token for Microsoft Entra ID.
   5. Click **Save**.

4. Go to **Applications > OAuth Clients** and register an OAuth Client for Microsoft Entra ID in PingFederate.

   Learn more in configuring OAuth clients. Use the following settings:

   - In the **Redirect URIs** section, in the **Redirection URIs** field, enter `https://login.microsoftonline.com/common/federation/externalauthprovider` and click **Add**.
   - Select the **Bypass Authorization Approval** checkbox.
   - In the **Allowed Grant Types** section, select `Implicit`.
   - Confirm that `id_token` is a valid response type.
   - In the **Default Access Token Manager** list, select the ATM that you configured in step 2.
   - In the **OpenID Connect** section, in the **Policy** list, select the OIDC policy that you configured in step 3.

5. If you are using PingID for MFA, make the following adjustments to the authentication policy configuration:

   1. If the PingID adapter follows the Microsoft EAM IdP Adapter in the authentication policy, set the Microsoft EAM IdP Adapter's `Sub` attribute as the `User ID Authenticated` for PingID.

      1. On the PingID adapter step, click **Options**.
      2. In the **Source** list, select the Microsoft EAM IdP Adapter.
      3. In the **Attribute list**, select `Sub`.
      4. Select the **User ID Authenticated** checkbox.
      5. Click **Done**.

Copyright © 2025 Ping Identity Corporation

2. If the PingID adapter follows the Microsoft EAM IdP Adapter in the authentication policy and the flow ends in a policy contract, select PingID's `amr` attribute as the **Source** for the **Contract Mapping**.
    - Learn more in configuring contract mapping.

Troubleshooting

Enabling debug logging

To help with troubleshooting or monitoring, you can turn on activity logging for PingFederate, the Microsoft EAM IdP Adapter, or both.

About this task

You can use logging for troubleshooting and analytics.

Learn more about logging in Enabling debug messages and console logging in the PingFederate documentation.

Steps

1. Open the `<pf_install>/pingfederate/server/default/conf/log4j2.xml` file for editing.
2. To log activity for PingFederate and all adapters:
    1. Find the following section:

```
<AsyncRoot level="INFO" includeLocation="false">
        <!-- <AppenderRef ref="CONSOLE" /> -->
        <AppenderRef ref="FILE" />
</AsyncRoot>
```

    2. Change `INFO` to `DEBUG`:

```
<AsyncRoot level="DEBUG" includeLocation="false">
        <!-- <AppenderRef ref="CONSOLE" /> -->
        <AppenderRef ref="FILE" />
</AsyncRoot>
```

    3. **Optional:** To see the adapter activity in the console, remove the comment tags that surround the `CONSOLE` line:

```
<AsyncRoot level="INFO" includeLocation="false">
        <AppenderRef ref="CONSOLE" />
        <AppenderRef ref="FILE" />
</AsyncRoot>
```

3. To log activity relating to the Microsoft EAM IdP Adapter:

Copyright © 2025 Ping Identity Corporation

Choose from:

- To log activity for the Microsoft EAM IdP Adapter as well as its HTTPS and component activity, add the following line:

```
<Logger name="com.pingidentity.adapters.microsoft.eam" level="DEBUG"/>
```

- To log activity for the adapter's HTTPS activity and other components but not for the adapter itself, add the following line:

```
<Logger name="com.pingidentity.adapters.microsoft.eam.shade" level="DEBUG"/>
```

- To log activity for the Microsoft EAM IdP Adapter but not its HTTPS or component activity, add the following lines:

```
<Logger name="com.pingidentity.adapters.microsoft.eam" level="DEBUG"/>
<Logger name="com.pingidentity.adapters.microsoft.eam.shade" level="INFO"/>
```

4. Save the file.

Release notes

Changelog

The following is the change history for the Microsoft EAM Integration Kit.

Microsoft EAM Integration Kit 1.0 – October 2024

- Initial release

Known issues and limitations

The following are known issues or limitations with the Microsoft EAM Integration Kit.

Known issues

There are no known issues.

Known limitations

The adapter does not support the PingFederate authentication API.

Copyright © 2025 Ping Identity Corporation

Download manifest

The following files are included in the Microsoft EAM Integration Kit `.zip` archive:

- `Legal.pdf` : copyright and license information.
- `dist/pingfederate/server/default` : contains the integration files.
  - `deploy` : contains the java libraries.
    - `pf-microsoft-eam-idp-adapter-<version>.jar` : The Microsoft EAM IdP Adapter.

Copyright © 2025 Ping Identity Corporation

Internet Information Services (IIS) Integration Kit

The Internet Information Services (IIS) Integration Kit allows PingFederate to coordinate user authentication and single sign-on (SSO) between an IIS web application and an identity provider (IdP).

> **Tip**
>
> For new integrations, try the Agentless Integration Kit. It can integrate with a variety of platforms using a modern RESTful approach, and it doesn't require you to integrate agent software into your application.

Components

OpenToken Adapter

Installed in PingFederate, this adapter uses the secure OpenToken standard to pass user attributes and session information from PingFederate to the OpenToken IIS Module on the IIS server.

OpenToken IIS Agent

Installed on the server running IIS, this program watches for protected resource requests and determines whether to grant access or redirect the user to PingFederate for authentication with an IdP.

OpenToken HTTP Module

Installed in IIS, this module reads OpenToken payloads from PingFederate that contain user attributes and session information.

Intended audience

This document is intended for PingFederate administrators and web application developers.

If you need help during the setup process, see the following sections of the PingFederate documentation:

- Managing SP adapters
- Configuring an OpenToken SP Adapter instance

System requirements

- PingFederate 9.0 or later
- IIS 7.0 or later using Integrated Mode
- ASP .NET application with .NET Framework 4.0

Overview of the OpenToken IIS Agent

The OpenToken IIS Agent acts as a filter in front of an application or protected resource.

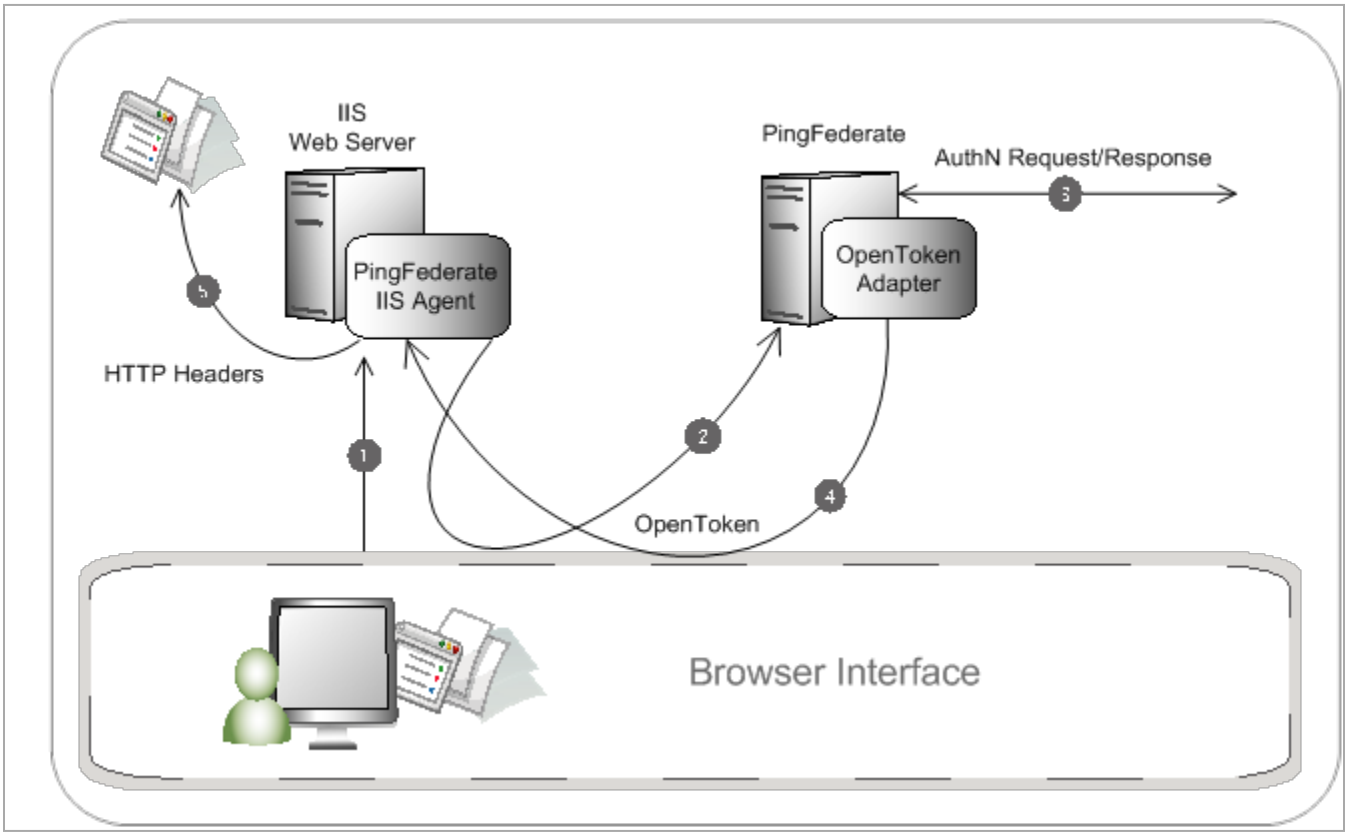
Copyright © 2025 Ping Identity Corporation

The basic responsibilities of the Agent are to filter requests to determine whether a request is for a protected resource:

- If the request is for an unprotected resource, the Agent passes the request to the application.
- If the request is for a protected resource, the Agent checks to see if there is a PingFederate session available and if it meets the policy for the session.
- If a session exists and the session meets the policy for the request, then the Agent passes the request back to the application.
- If a session does not exist, or if the existing session does not meet the session policy for that request, the Agent redirects the user's browser through the PingFederate server to an Identity Provider (IdP) for authentication. After authentication, PingFederate redirects the user back to the protected resource with a valid session.

Overview of the SSO flow

With the IIS Integration Kit, PingFederate handles authentication requests to an identity provider (IdP) on behalf of IIS.



The figure above illustrates an SP-initiated SSO scenario, showing the request flow and how the PingFederate OpenToken Adapter wraps attributes from an assertion into a secure OpenToken and passes it to IIS. For more detail, see OpenToken Adapter in the PingFederate documentation.

Description

1. A user attempts to access a resource on the IIS server protected by the OpenToken IIS Agent.

Copyright © 2025 Ping Identity Corporation

2. The user is redirected to the PingFederate server for authentication.
3. If an OpenToken session already exists, the user is granted immediate access.
4. The PingFederate server redirects the user's browser to an IdP for authentication using either the SAML or WS-Federation protocols. The IdP partner authenticates the user and returns a SAML assertion.
5. PingFederate validates the assertion and creates an OpenToken for the user including any configured attributes. PingFederate then redirects the browser, including the OpenToken, back to the OpenToken IIS Agent's OpenToken Exchange service, which converts the OpenToken into a cookie, and redirects to the original resource.
6. The OpenToken IIS Agent verifies the OpenToken and grants access to the protected resource. The User ID and any attributes from the OpenToken are exposed to the resource as HTTP Request Headers.

Session information and HTTP request headers

The OpenToken IIS Agent uses HTTP request headers to provide session information and user attributes from the OpenToken adapter to the protected application.

This allows the application to use the information to support various features, such as making authorization decisions or providing personalized content.

The application has access to the following session and attribute information:

- Attributes from the OpenToken Adapter contract
    - These include, by default, the subject (`SUBJECT`) and attributes specified on the **Extended Contract** screen of the adapter instance configuration. Only the attributes fulfilled at runtime are available to the application. Attributes with a NULL value are not included in the OpenToken.
- `NOT-ON-OR-AFTER`
    - The time that the token expires.
- `RENEW-UNTIL`
    - The time that the session expires. Tokens cannot be renewed past this time.
- `AUTH_NOT-BEFORE`
    - The time that the session began.
- `AUTHNCONTEXT`
    - Information from the SAML assertion that describes how the user was authenticated by the identity provider (IdP). For a complete description, see "Authentication context" in Terminology in the PingFederate documentation.

For security reasons, each HTTP request header is first prepended with a specific (configurable) prefix. The OpenToken IIS Agent always removes and rewrites these prefixed request headers for each request.

If applications protected by the OpenToken IIS Agent cannot be modified to accept headers with this prefix, you can Configuring the OpenToken IIS Agent to omit the HTTP header prefix.

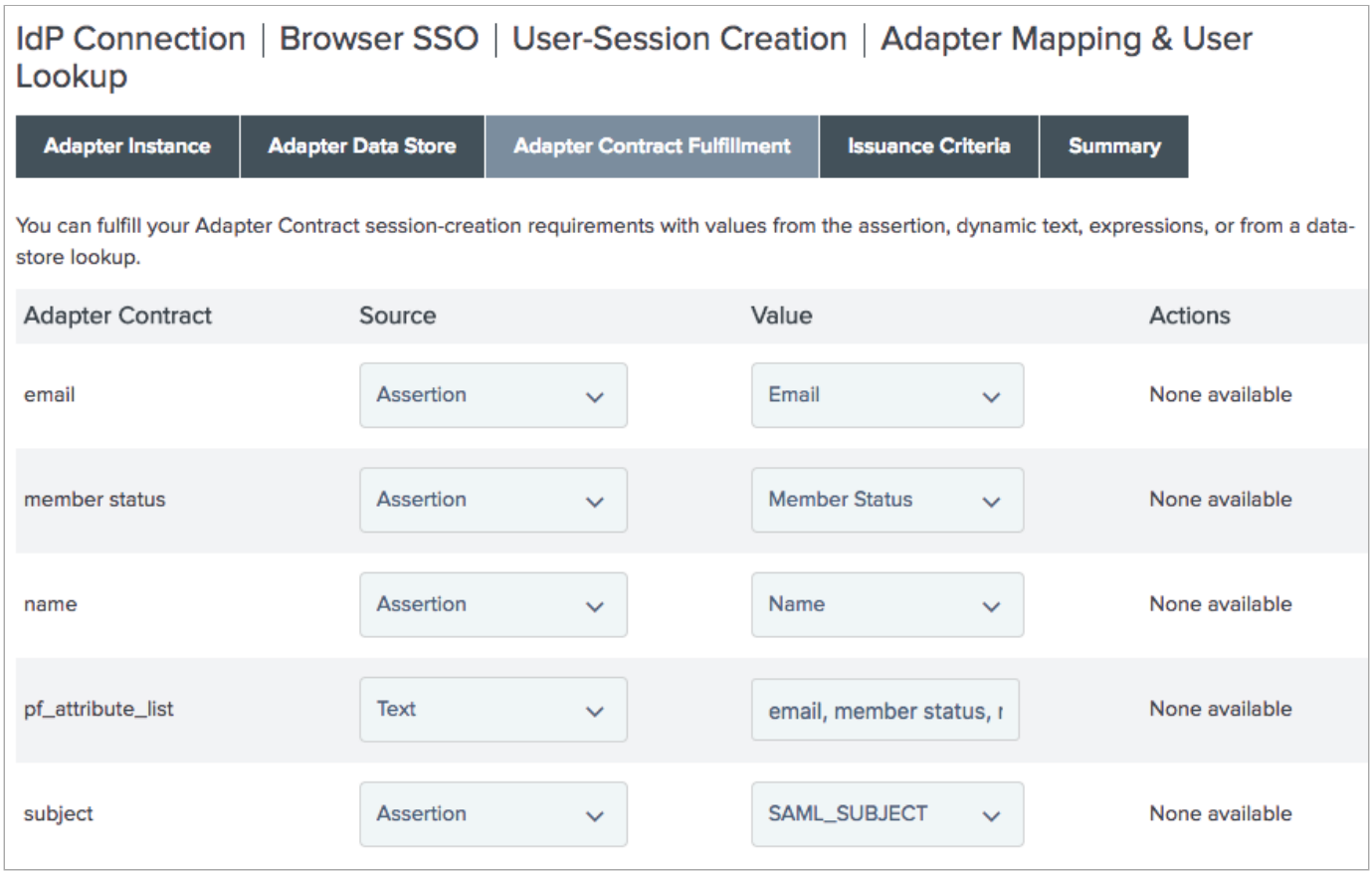
Copyright © 2025 Ping Identity Corporation

Configuring the OpenToken IIS Agent to omit the HTTP header prefix

If your IIS web application cannot accept HTTP request headers with the standard OpenToken prefix, use this alternative method.

Steps

1. In the OpenToken Adapter instance configuration, on the **Extended Contract** screen, include an attribute named `pf_attribute_list`.
2. In the identity provider (IdP) connection configuration, map the `pf_attribute_list` attribute as a text field that contains a comma-separated list of all the attributes in the adapter contract. See Configuring IdP adapter contract fulfillment in the PingFederate documentation.



Setup

Upgrading an existing deployment

If you are updating an existing deployment of the Internet Information Services (IIS) Integration Kit, back up your configuration and remove the previous agent before deploying the latest version.

Steps

1. Update the OpenToken Adapter in PingFederate as shown in Updating the OpenToken Adapter.

Copyright © 2025 Ping Identity Corporation

2. Back up your existing `C:\Program Files\Ping Identity Corporation\OpenToken IIS Agent(x-bit)\conf\pfisapi.conf` to a safe location.

   Refer to this copy when you complete the Deploying and configuring the OpenToken IIS Agent step.

3. If you are upgrading from a previous deployment, open your backup copy of `pfisapi.conf` for editing. Add the following sections:

   > **Tip**
   >
   > The `pfisapi.conf` file included in the latest Internet Information Services (IIS) Integration Kit `.zip` archive shows the placement of these parameters and includes the most up to date descriptions.

Copyright © 2025 Ping Identity Corporation

```
#--------------
# Basic Options
#--------------

# (OPTIONAL) List of protected absolute URIs. Multiple patterns can be used.
# with a separator '|'. Regular expressions are allowed.
# This overrides values entered in ProtectedResourceList
# The default value is blank.
ProtectedUriList=

# (OPTIONAL) List of excluded absolute URIs. Multiple patterns can be used.
# with a separator '|'. Regular expressions allowed.
# This will override any values entered in ExcludeList.
# The default value is blank.
ExcludeUriList=

# (OPTIONAL) Lifetime of the OpenToken cookie in seconds.
# If set to -1, the token-lifetime in the agent-config.txt file is used.
# Any other value overrides the token-lifetime in in the agent-config.txt file.
# The default value is -1.
CookieAge=-1

# (Required)
# The SameSite cookie attribute is set to this value.
# The allowed values for this setting are: Strict, Lax, None, and Nothing
# The "Strict", "Lax", and "None" value changes the SameSite cookie attribute setting.
# The "Nothing" value leaves the SameSite cookie attribute unset in the OpenToken Session Cookie.
# For the "None" value, you must use secure attributes because cross-site cookies can only be
accessed over HTTPS connections.
# If the cookie is not secure and the "None" value is selected, the SameSite cookie attribute will
not be set.
# The default value is Nothing.
SameSiteCookie=Nothing

#-----------------
# Advanced Options
#-----------------

# (OPTIONAL) Determines whether the cookie will be a session cookie or a persistent cookie.
# Setting to YES will result in SessionCookie. This overrides the value in the agent-config.txt
file.
# Possible values are 'YES' and 'NO'.
# The default value is blank.
# SessionCookie=
```

4. Set values for any new parameters. Save the file.

5. Stop IIS.

6. Remove the existing OpenToken HTTP Module from IIS.

    1. In the Internet Information Services (IIS) Manager, select the IIS server.

    2. On the **Features View** tab, double-click **Modules**.

Copyright © 2025 Ping Identity Corporation

3. On the**Modules** tab, select the OpenToken HTTP Module.
4. In the **Actions** area, click **Remove**.

7. In the Windows Control Panel, uninstall the **OpenToken IIS Agent** program.
8. Start IIS.

Updating the OpenToken Adapter

To get started with the integration, download the latest version of the OpenToken Adapter and deploy it to your PingFederate directory.

About this task

Although the OpenToken Adapter is included in PingFederate and the Internet Information Services (IIS) Integration Kit, the latest version of the OpenToken Adapter is distributed separately as part of the Java Integration Kit. Follow these steps to make sure you have the latest version available.

Steps

1. Download the Java Integration Kit `.zip` archive from the **Add-ons** tab of the PingFederate downloads page.

   Stop PingFederate.

2. If you are upgrading an existing deployment, delete the following files from `<pf_install>/pingfederate/server/default/deploy`:
   - `opentoken-adapter-<version>.jar`
   - `opentoken-java-<version>.jar`
3. From the `.zip` archive, copy `dist/opentoken-adapter-<version>.jar` to `<pf_install>/pingfederate/server/default/deploy`.

   Start PingFederate.

   If you operate PingFederate in a cluster, repeat steps 2-5 for each engine node.

Configuring an OpenToken SP Adapter instance

Configure the OpenToken Adapter to determine how PingFederate communicates with your service provider application.

Steps

1. In the PingFederate administrative console, go to **Authentication > Integration > IdP Adapters**. Click **Create new Instance**.
2. On the **Type** tab, set the basic adapter instance attributes.
   1. In the **Instance Name** field, enter a name for the adapter instance.
   2. In the **Instance ID** field, enter a unique identifier for the adapter instance.
   3. From the **Type** list, select **OpenToken Adapter**. Click **Next**.

Copyright © 2025 Ping Identity Corporation

3. On the **Instance Configuration** tab, configure the adapter instance by referring to Configuring an OpenToken SP Adapter instance in the PingFederate documentation. Click **Next**.

> **Note**
>
> If you use IIS to protect multiple sites on the same domain, in the OpenToken Adapter instance configuration, select **None** for SameSite Cookie, and select the **Secure Cookie** check box.

4. Export the configuration file:
    1. On the **Actions** tab, click **Download**, and then click **Export**.
    2. Save `agent-config.txt`. Click **Next**.
5. On the **Extended Contract** tab, add any attributes that you expect to retrieve other than the SAML subject. Click **Next**.
6. On the **Target App Info** tab, enter the basic information about your SP application. Click **Next**.
7. On the **Summary** tab, check and save your configuration. Click **Save**.
8. Create or update an identity provider (IdP) connection to use the OpenToken Adapter instance as shown in Service provider SSO configuration in the PingFederate documentation.

Deploying and configuring the OpenToken IIS Agent

Deploy the OpenToken IIS Agent on your IIS server to allow it to communicate with PingFederate.

Steps

1. If your IIS server is version 8 or later, add the IIS server role and role services.
    1. In **Server Manager**, on the **Manage** menu, click **Add Roles and Features**.
    2. In the **Add Roles and Features Wizard**, on the **Before You Begin** tab, click **Next**.
    3. On the **Installation Type** tab, select **Role-based or feature-based installation**. Click **Next**.
    4. On the **Server Selection** tab, select the IIS server. Click **Next**.
    5. On the **Server Roles** page, select **Web Server (IIS)**.
    6. On the **Features** tab, click **Next**.
    7. On the **Web Server Role (IIS) > Role Services** tab, in **Web Server > Application Development** , select **.NET 3.5 Extensibility** and **.NET 4.6 Extensibility**.

        Depending on your version of Windows, you might need to select **.NET 4.5 Extensibility** or **.NET 4.7 Extensibility**.
    8. If you receive a message that asks you to add features that are required for Web Server (IIS), click **Add Features**.
    9. Click **Next**.
    10. On the **Confirmation** tab, check that the configuration is correct. Click **Install**.

        This step can take several minutes.

Copyright © 2025 Ping Identity Corporation

11. Once the installation is complete, click **Close**.

2. Deploy the OpenToken IIS Agent.

    1. Extract the IIS Integration Kit distribution file on the IIS server, and then go to `dist/(x86)` or `dist/(x64)`.
    2. Run `setup.exe` to install the OpenToken HTTP Module into the Windows Global Assembly Cache.
    3. If your IIS server is version 7, register IIS with .NET Framework 4.0 by entering the following command at the Windows command prompt:

        ```
        <Windows_install>\Microsoft.NET\Framework\v4.0.30319\aspnet_regiis-iru
        ```

    4. Move the `agent-config.txt` that you exported in Configuring an OpenToken SP Adapter instance to `C:\Program Files\Ping Identity Corporation\OpenToken IIS Agent(n-bit)\conf` or your equivalent.

3. Configure the OpenToken IIS Agent properties file.

    1. Open `C:\Program Files\Ping Identity Corporation\OpenToken IIS Agent(n-bit)\conf\pfisapi.conf` for editing.
    2. Configure it to suit your environment based on the property descriptions in the file.
    3. If you use IIS to protect multiple sites on the same domain, set `SameSiteCookie=None` and `SecureCookie=YES`.
    4. Save the file.
    5. If you backed up a previous copy of the file in Upgrading an existing deployment, refer to that file to add new properties and restore your previous settings.
    6. Restart IIS.

    > **Note**
    >
    > You can test your configuration by deploying the sample application (`PFIsapiSample`) that is included in the installation. See `C:\Program Files\Ping Identity Corporation\OpenToken IIS Agent(n-bit)\samples` or your equivalent.

Adding the OpenToken HTTP Module in IIS

The OpenToken HTTP module allows your IIS server to communicate with PingFederate using the OpenToken format.

Steps

1. In the Internet Information Services (IIS) Manager, select the IIS server.
2. On the **Features View** tab, double-click **Modules**.
3. On the **Modules** window, in the **Actions** section, click **Add Managed Module**.
4. On the **Add Managed Module** dialog, in the **Name** field, type a name, such as OpenTokenHttpModule.
5. In the **Type** field, enter the following:

    ```
    OpenTokenModule.HttpModule, OpenTokenModule, Version=3.5.0.0, Culture=neutral,
    PublicKeyToken=f5ed9639debbca65
    ```

Copyright © 2025 Ping Identity Corporation

> **Note**
>
> The OpenToken HTTP Module doesn't automatically appear in the module selection list.

6. Click **OK**.

Release notes

Changelog

The following is the change history for the Internet Information Services (IIS) Integration Kit.

IIS Integration Kit 3.5 – March 2021

- Added support for protecting multiples sites under the same domain using one instance of the adapter.
- Fixed an issue that cause the adapter to ignore the `token-lifetime` and `secure-cookie` parameters in the `agent-config.txt` file.

IIS Integration Kit 3.4.2 – December 2020

- Fixed an issue that caused the adapter to ignore the cookie settings in the agent config file.
- Fixed an issue that caused the samesite cookie and legacy cookie to become out of sync.

IIS Integration Kit 3.4.1 – February 2020

- Added support for the SameSite cookie flag in web browsers.

IIS Integration Kit 3.4 – January 2019

- Added compatibility for IIS 8.5 and 10 environments.

IIS Integration Kit 3.3.1 – June 2018

- Addressed compatibility issue with 32-bit application pool

IIS Integration Kit 3.3 – December 2017

- Added support for using the HttpOnly flag on the session cookie (refer to the new HttpOnlyCookie attribute in `pfisapi.conf`).
- Added support for .NET Framework 4.0
- Bug fixes

IIS Integration Kit 3.2 – June 2014

- Added support for whitelisting in the properties file for IIS (`pfisapi.conf`)
- Added units to the TokenUpdateWindow field in the `pfisapi.conf` file

Copyright © 2025 Ping Identity Corporation

- Added documentation for the IIS limitation regarding HTTP POST issues with managed and native modules
- Added support for IIS 8 - Integrated Mode

IIS Integration Kit 3.1 – December 2012

- Added support for an exclusion resource list in the properties file for IIS (`pfisapi.conf`)
- Added support for the 2.5.1 version of the OpenToken Adapter and OpenToken Agent

IIS Integration Kit 3.0 – October 2011

- Upgraded to .NET Framework 4.0
- Requires IIS 7.0 or higher
- Requires integrated pipeline mode for protected applications
- Added support for both 32-and 64-bit application pool in IIS
- Added ability to control logging level
- Replaced ISAPI extension with OpenToken HTTP Module
- Removed OpenToken Exchange application
- Corrected issue with TargetResource URL being truncated in certain situations
- Corrected IIS crashing issue when attribute data is too long

IIS Integration Kit 2.2 – September 2009

- Added OpenToken Exchange service application to handle OpenToken query and POST transport methods
- For performance optimization, added a property (`TokenUpdateWindow`) in the configuration file `pfisapi.conf` to enable OpenTokens to be reused by the browser for a configurable period of time

IIS Integration Kit 2.1 – June 2009

- Added alternative method for setting session attributes as HTTP headers without a prefix
- Corrected OpenToken timestamp comparisons to prevent the occurrence of intermittent invalid OpenToken errors
- Protected-resource URL filtering is not case-sensitive

IIS Integration Kit 2.0 – December 2008

- Ported the OpenToken IIS Agent to use the OpenToken Adapter and the OpenToken .NET library
- Supports the POST Transport Method from the OpenToken Adapter
- Added URL filtering for protected resources
- Added support to allow IIS web applications to download files

Copyright © 2025 Ping Identity Corporation

IIS Integration Kit 1.1 – September 2007

- Security patch to write all HTTP standard headers and PFTOKEN custom headers only; any other HTTP custom headers will be removed
- Simplified IIS agent configuration file

IIS Integration Kit 1.0 – June 2007

- Initial Release

Known issues and limitations

The following are known issues or limtiations for the Internet Information Services (IIS) Integration Kit.

Known issues

There are no known issues.

Known limitations

- The installer does not recognize IIS 7.0 (or IIS 7.5) as uninstalled and does not stop the installation if IIS has been previously installed and then uninstalled.
- IIS Integration Kit does not support newline characters (such as '\n' and '\r') within attributes.
- When using Form POST as the transport mode in the SP OpenToken adapter, a trailing / may be needed in the URL for accessing the protected resources.
- The OpenToken name specified in the adapter setup must be unique within the given federation. This is not enforced in the user interface.
- When using the installer to upgrade by uninstalling and reinstalling the agent, it is necessary to restart IIS after uninstall and before reinstall to ensure that old DLL's are not used. You can reset IIS with the command: `resetiis /noforce`
- After uninstalling, it is necessary to remove any application mappings set up manually. The uninstall script cannot remove these mappings.
- When using cookie as the transport method for OpenToken, the domain configured in the adapter setup must match the domain configured in `pfisapi.conf`. If they do not, it is possible to end up with a persistent cookie. This is not enforced.
- When using non-session cookie as the transport method for OpenToken, there is no session cookie configuration defined in the agent config file. Set the SessionCookie type in `pfisapi.conf` by removing the # and setting it to YES or NO.
- Classic managed pipeline mode is not supported
- Attempting to access the POST data of a request in a native module in IIS 7 after the data has been accessed by a managed module (such as the OpenToken Module) prevents the native module from being able to access the data. This is a limitation of IIS 7. The issue also applies when using Query Parameter as the transport method. Using Cookie as the transport method is the only technique to resolve this issue.
- The OpenToken Module cannot be selected from the list of module types. Instead, it must be entered manually as shown in Adding the OpenToken HTTP Module in IIS.

Copyright © 2025 Ping Identity Corporation

- The OpenToken Module can only be added at the global server level. If you require implementation on a per-website basis, contact Ping Identity about PingAccess.

Download manifest

The following files are included in the Internet Information Services (IIS) Integration Kit `.zip` archive:

- `ReadMeFirst.pdf` : Contains links to this online documentation
- `Legal.pdf` : Copyright and license information.
- `dist` : Contains the agent and supporting files that the adapter needs:
  - `x64` : Contains agent installation files for 64-bit Windows architecture:
    - `setup.exe` : Installation program for the OpenToken IIS Agent.
    - `Support Files.msi` : Installation supporting files for the OpenToken IIS Agent.
    - `conf` : Contains configuration files.
    - `Module Retargetable Folder` : Contains IIS agent configuration data and a sample application.
    - `Global Assembly Cache Folder` : IIS agent DLLs.
  - `x86` : Contains agent installation files for 32-bit Windows architecture:
    - `setup.exe` : Installation program for the PingFederate OpenToken IIS Agent.
    - `Supporting Files.msi` : Installation supporting files for the OpenToken IIS Agent.
    - `conf` : Contains configuration files.
    - `Module Retargetable Folder` : Contains IIS agent configuration data and a sample application.
    - `Global Assembly Cache Folder` : Contains IIS agent DLLs.

> **Note**
>
> The integration requires the latest version of the OpenToken Adapter, which is distributed separately as part of the Java Integration Kit. Learn more in Updating the OpenToken Adapter.

Copyright © 2025 Ping Identity Corporation

Microsoft Login Integration Kit

The Microsoft Login Integration Kit allows PingFederate to use Microsoft as an identity provider (IdP). This allows users to access service provider (SP) applications by signing into Microsoft.

Features

- Supports the PingFederate PingFederate authentication API.
- Supports the JavaScript Widget for the PingFederate Authentication API.

Components

Microsoft IdP Adapter

- Allows PingFederate to communicate with the Microsoft API to process sign-on requests and get user information.

Templates

- Allow the adapter to prompt the user to sign on. The template can be presented with a browser redirect or as a pop-up window.
- Allow you to modify the appearance of the sign-on prompt.

Language packs

- Allow you to customize or localize the messages returned by the PingFederate authentication API and shown on the templates during authentication. For help, see Localizing messages for end users in the PingFederate documentation.

Intended audience

This document is intended for PingFederate administrators.

If you need help during the setup process, see the following resources:

- PingFederate documentation:
  - Identity provider SSO configuration
  - Managing IdP adapters
  - Authentication policies
  - SP connection management
- Register an application with the Microsoft identity platform in the Microsoft documentation.

System requirements

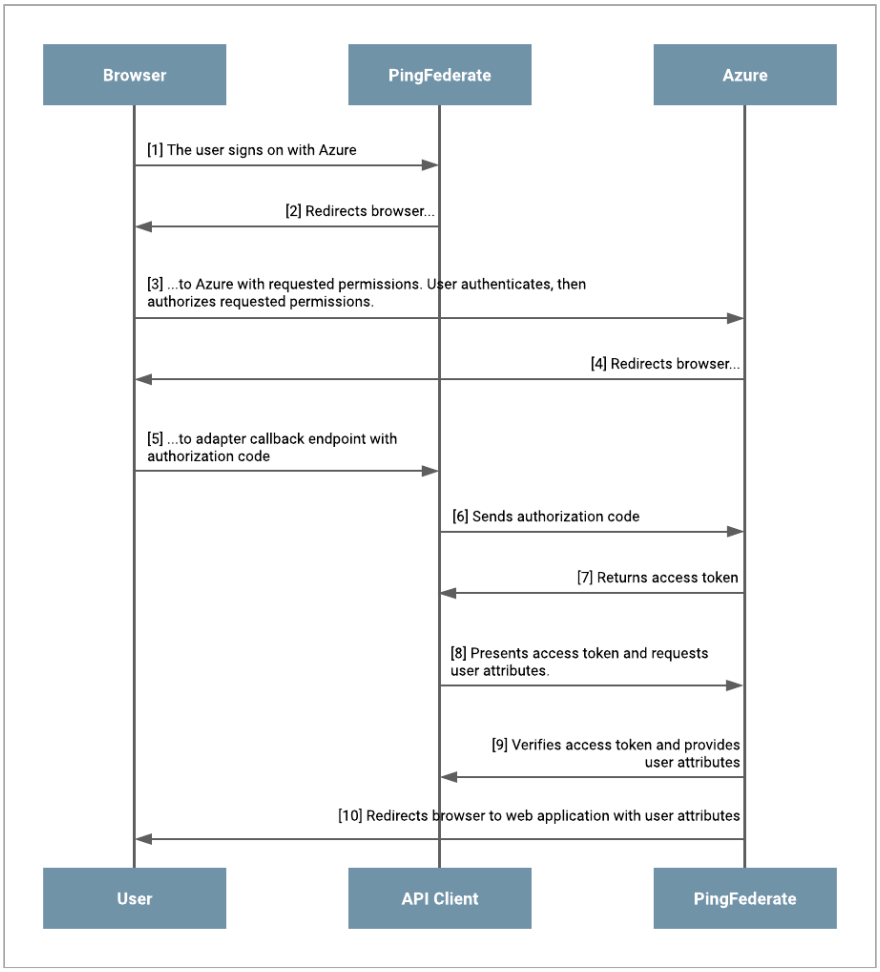
- PingFederate 11.2 or later
- A Microsoft developer account with Azure

Copyright © 2025 Ping Identity Corporation

Overview of the SSO flow

With the Microsoft Login Integration Kit, PingFederate includes the Microsoft authentication API in the sign-on flow.

The following figure illustrates a single sign-on (SSO) scenario in which PingFederate authenticates users to an SP application using the Microsoft IdP Adapter.



Description

1. The user opens a web application and chooses the Microsoft sign-on option.
2. The sign-on link points to the Microsoft IdP Adapter, which redirects the browser...
3. ...to Microsoft with a list of requested permissions. On Microsoft, the user authenticates their identity and then authorizes the requested permissions.
4. Microsoft redirects the browser...
5. ...to the Microsoft IdP Adapter authorization callback endpoint with an authorization code.

   If the user fails to authenticate or does not authorize the request, the response includes an error code instead.
6. PingFederate sends Microsoft the authorization code.
7. Microsoft returns an access token.

Copyright © 2025 Ping Identity Corporation

8. PingFederate sends Microsoft a request for user attributes, and presents the access token.
9. Microsoft verifies the access token, and provides the user information.
10. PingFederate redirects the user to the web application with the user attributes.

Authentication API Support

You can use the PingFederate authentication API to integrate the Microsoft IdP Adapter into your application.

The PingFederate Authentication API provides access to the current state of the authentication flow as a user steps through the PingFederate authentication policy. For more information, see PingFederate authentication API in the PingFederate documentation.

To integrate the Microsoft IdP Adapter into your authentication flow, configure your application based on the states, actions, and models available in the PingFederate Authentication API Explorer. For help, see Exploring the Authentication API in the PingFederate documentation.

Setup

Deploying the integration files

To get started with the integration, deploy the Microsoft Login Integration Kit files to your PingFederate directory.

About this task

> **Note**
>
> If you operate PingFederate in a cluster, the following steps refer to the console node.

Steps

1. Download the Microsoft Login Integration Kit `.zip` archive from the Ping Identity Integration Directory.
2. Stop PingFederate.
3. If you are upgrading an existing deployment, back up your customizations and delete previous versions of the integration files.
    1. Back up any Microsoft Login Integration Kit files that you customized in `<pf_install>/pingfederate/server/default/conf/`.
    2. Delete `pf-microsoft-idp-adapter-<version>.jar` from `<pf_install>/pingfederate/server/default/deploy`.
4. Extract the `.zip` archive and merge the contents of the `dist` directory with your `<pf_install>/pingfederate` directory.
5. If there is more than one version of the `pf-authn-api-sdk-<version>.jar` file in your `<pf_install>/pingfederate/server/default/lib` directory, delete all but the latest version of the file.
6. If you backed up any customized files, modify the new files with your customizations.
7. Start PingFederate.

Copyright © 2025 Ping Identity Corporation

8. If you operate PingFederate in a cluster, repeat steps 2-8 for each engine node.

Registering PingFederate as an application in Azure

To allow PingFederate to process social sign-on requests with Microsoft, add PingFederate as an OAuth application in the Azure portal.

About this task

For more information, see Register an application in the Microsoft Azure documentation.

Steps

1. Sign on to the Microsoft Azure portal and go to Azure Active Directory.
2. Go to **Manage > App registrations**. Click **New registration**.
3. On the **Register an application** page, in the **Name** field, enter a name for the application.
4. For **Supported Account Types** section, select **Personal Microsoft accounts only**.

   > **Tip**
   >
   > Advanced configurations can provide access to organizational accounts in addition to, or instead of, personal Microsoft accounts. For help, click **Help me choose** and see Register an application in the Microsoft documentation. Note that each individual organization must have **User.Read.All** set in its API permissions. External organizations might not have this set.
   > Remember your **Support Account Types** selection. You'll make the same selection in Configuring an adapter instance.

5. In the **Redirect URI** section, select **Web** and enter `https://pf_host:pf_port/ext/microsoft-authn`.

   If you set a custom value here, write it down. You'll use it again in Configuring an adapter instance.

   For more information, see Add a redirect URI in the Microsoft Azure documentation.

6. Click **Register**.
7. On the application overview page, in the **Essentials** section, note the **Application (client) ID**. If you selected the **Single tenant** option for **Supported Account Types**, also note the **Directory (tenant) ID**.

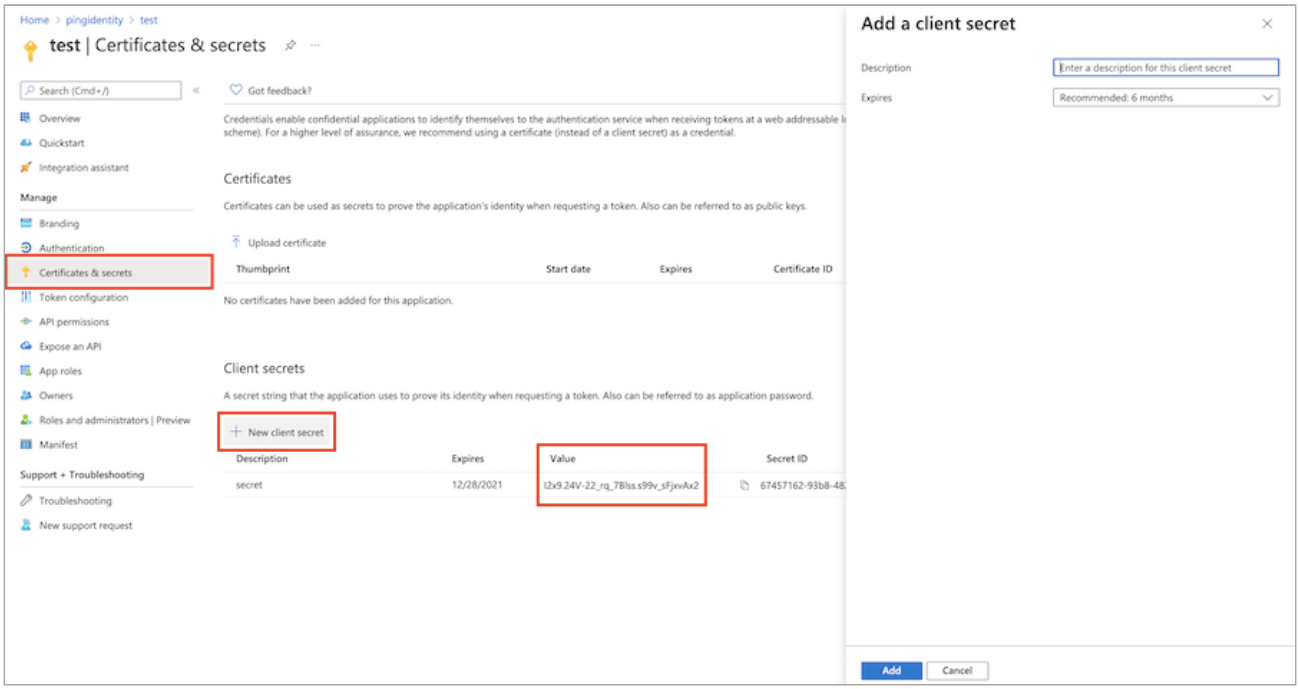
   You'll use this in Configuring an adapter instance.

8. For **Client credentials**, click **Add a certificate or secret**. Click **New client secret**.
9. Add a client secret.

   1. On the **Add a client secret** pane, enter a description and select an expiry period. Click **Add**.

      For help, see Add a client secret in the Microsoft documentation.

Copyright © 2025 Ping Identity Corporation



2. Note the client secret **Value**.

   You'll use this in Configuring an adapter instance.

Configuring an adapter instance

Configure the Microsoft IdP Adapter to determine how PingFederate communicates with Azure.

Steps

1. In the PingFederate administrative console, create a new IdP adapter instance:

   Choose from:

   - For PingFederate 10.1 or later: go to **Authentication > Integration > IdP Adapters**. Click **Create New Instance**.
   - For PingFederate 10.0 or earlier: go to **Identity Provider > Adapters**. Click **Create New Instance**.

2. On the **Type** tab, set the basic adapter instance attributes.
   1. In the **Instance Name** field, enter a name for the adapter instance.
   2. In the **Instance ID** field, enter a unique identifier for the adapter instance.
   3. From the **Type** list, select **Microsoft IdP Adapter**. Click **Next**.
3. **Optional:** On the **IdP Adapter** tab, in the **Additional Parameters** section, add any additional parameters to send in the authorization request to Microsoft, then map them to the local chained attribute:
   1. Click **Add a new row to 'Additional Parameters'**.
   2. In the **Parameter** field, enter the name of a query string parameter that you want to send.

Copyright © 2025 Ping Identity Corporation

3. In the **Local Chained Attribute** field, enter the name of the chained attribute whose value will be used as the parameter value in the authorization request to Microsoft.
4. In the **Action** column, click **Update**.
5. To add more attributes, repeat steps a-d.

4. On the **IdP Adapter** tab, finish configuring the adapter instance by referring to Microsoft IdP Adapter settings reference. Click **Next**.
5. On the **Actions** tab, test your connection to Azure. Resolve any issues that are reported, and then click **Next**.
6. On the **Extended Contract** tab, add any attributes that you want to include in the contract. Click **Next**.
7. On the **Adapter Attributes** tab, set pseudonym and masking options as shown in Set pseudonym and masking options in the PingFederate documentation. Click **Next**.
8. On the **Adapter Contract Mapping** tab, configure the contract fulfillment details for the adapter as shown in Define the IdP adapter contract in the PingFederate documentation. Click **Next**.
9. On the **Summary** tab, check and save your configuration:

   Choose from:

   - For PingFederate 10.1 or later: click **Save**.
   - For PingFederate 10.0 or earlier: click **Done**. On the **Manage IdP Adapter Instances** tab, click **Save**.

Core contract attributes

The following are the core contract attributes for the Microsoft IdP Adapter.

Attribute	Details
`businessPhones`	The user's business phone number.
`displayName`	The user's full name.
`givenName`	The user's first name.
`id`	The user's Microsoft user ID.
`jobTitle`	The user's job title.
`mail`	The user's email address.
`mobilePhone`	The user's mobile phone number.
`officeLocation`	The company office location.
`preferredLanguage`	Preferred language configured on Microsoft.

Copyright © 2025 Ping Identity Corporation

Attribute	Details
`surname`	The user's last name.
`userPrincipalName`	The user's principal name from Microsoft.
`idToken`	A JWT from Microsoft that's used in authentication.
`accessToken`	A JWT from Microsoft that's used in authorization.
`tenantId`	A header claim encoded in the `idToken` JWT that lists which tenant the user is authenticating to during sign-on.

Microsoft IdP Adapter settings reference

Field descriptions for the Microsoft IdP Adapter configuration screen.

Table 1. Standard fields

Field Name	Description
Supported Account Types	The type of accounts that users can sign in with. The single tenant and multitenant options support organizational accounts from a school or work directory. Personal accounts are not associated with an organization. Do one of the following: • For standard configurations, select **Personal accounts only**. • For advanced configurations, if you selected a different option in Registering PingFederate as an application in Azure, make the same selection here. The default selection is **Personal accounts only**.
Single Tenant ID	If you selected **Single tenant** for **Support Account Types**, enter the **Directory (tenant) ID** that you noted in Registering PingFederate as an application in Azure. Otherwise, leave this field blank. This field is blank by default.
Client ID	The **Application (client) ID** that you noted in Registering PingFederate as an application in Azure.
Client Secret	The client secret **Value** that you noted in Registering PingFederate as an application in Azure.

Copyright © 2025 Ping Identity Corporation

Field Name	Description
Authorization Callback Endpoint	The PingFederate endpoint that Microsoft uses to respond to authorization requests. If you set a custom endpoint in the **Redirect URI** field in Registering PingFederate as an application in Azure, change this field to match. The default value is: `/microsoft-authn`
Error Redirect URL	The URL that PingFederate redirects the user to when the adapter encounters an error. A blank value shows the default error page.
Unauthorized Redirect URL	The URL that PingFederate redirects to when the user does not authorize Microsoft to share their information. A blank value shows the default error page.

Table 2. Advanced fields

Field Name	Description
Include Login Hint	If selected, the incoming **User Id** value (if provided) is sent as a `login_hint` query string parameter value to Microsoft. Microsoft uses this value to pre-fill the **Username** field on the Microsoft user sign-on page.
Include Prompt	If selected, PingFederate automatically maps and sends the standard PingFederate-supported OIDC prompt parameter as the `prompt` parameter value in the authorization request to Microsoft. **Important** This applies only if PingFederate receives the prompt parameter through an authentication policy that has a value of `consent` or `login`. To send other values in the authorization request, use the **Additional Parameters** table and provide the value in an incoming chained attribute. Learn more in step 3 of Configuring an adapter instance.
Microsoft Login Base URL	The base URL used by Microsoft for any authentication calls. The default value is: `https://login.microsoftonline.com/`
Microsoft Authorization Endpoint	The endpoint used to request an authorization code from Microsoft. The default value is: `/oauth2/v2.0/authorize`
Microsoft Token Endpoint	The endpoint used by Microsoft to retrieve an access token. The default value is: `/oauth2/v2.0/token`
Microsoft Logout Endpoint	The logout endpoint used by Microsoft to end the user's session. The default value is: `/oauth2/v2.0/logout`

Copyright © 2025 Ping Identity Corporation

Field Name	Description
Microsoft User Info URL	The URL used to retrieve Microsoft user data. The default value is: `https://graph.microsoft.com/v1.0/me`
Scopes	Comma-separated scopes to request from Microsoft. The default value is: *openid*
Microsoft Sign-on Presentation	Determines how the user is directed to Microsoft for authentication. Automatic redirects are blocked by some browsers. If you select **Pop-up** and are not using PingFederate in authentication API mode, the adapter presents a template file. Options include: • Redirect • Pop-up window
Microsoft Pop-up Template	The template file that presents the Microsoft sign-on form. Applies only when **Microsoft Sign-on Presentation** is set to **Pop-up window**. The default value is: `microsoft-pop-up-template.html`
Microsoft Post Auth Template	The template file that the adapter presents after the user signs on. Applies only when **Microsoft Sign-on Presentation** is set to **Pop-up window**. The default value is: `microsoft-post-auth-template.html`
Microsoft Messages File	The language-pack file associated with Microsoft Pop-up Template. The default value is: `pingfederate-microsoft-adapter-messages`
Retry Request	Retry request if API fails with error codes configured. Options include: • selected • not selected
Maximum Retries Limit	Determines how many times PingFederate will retry a request. The default value is `5`.
Retry Error Codes	Determines which response codes are considered failures. The default value is `403`.
API Request Timeout	The amount of time in milliseconds that PingFederate allows when establishing a connection with Azure or waiting for a response to a request. A value of 0 disables the timeout. The default value is `5000`.

Copyright © 2025 Ping Identity Corporation

Field Name	Description
Proxy Settings	Defines proxy settings for outbound HTTP requests. The default value is **System Defaults**.
Custom Proxy Host	The proxy server host name to use when **Proxy Settings** is set to **Custom**. This field is blank by default.
Custom Proxy Port	The proxy server port to use when **Proxy Settings** is set to **Custom**. This field is blank by default.

Integrating social sign-on into your application

To complete your Microsoft sign-on integration, add a sign-on hyperlink to your application.

Steps

1. Create an SP connection to your application as shown in SP connection management in the PingFederate documentation. Use your IdP adapter instance and Mapping an adapter instance in the PingFederate documentation.
    - Use the Microsoft IdP Adapter instance as an authentication source. For help, see Mapping an adapter instance in the PingFederate documentation.
2. If your application is outside the PingFederate domain, configure a service provider (SP) connection.
    1. Create an SP connection that uses your IdP adapter instance as shown in SP connection management and Mapping an adapter instance in the PingFederate documentation.
    2. In your web application, create a hyperlink to allow users to sign on to the SP application. Use the following URL and replace the variables based on the descriptions in the following table.

        ```
        https://<pf_host>:<pf_port>/idp/startSSO.ping?PartnerSpId=<ConnectionId>
        ```

Variable	Description
<pf_host>	The host name or IP address of the PingFederate server.
<pf_port>	The port number for PingFederate.
<ConnectionId>	The federation identifier of the SP for the connection that uses the GitHub IdP Adapter instance.

3. If your application is inside the PingFederate domain, configure an adapter-to-adapter mapping.
    1. On the **System > Protocol Settings > Roles & Protocols** screen, select **Enable Identity Provider (IdP) role and support for the following** and **Enable Service Provider (SP) role and support for the following**.
    2. In both the **Enable Identity Provider** and **Enable Service Provider** sections, select any protocol, such as **SAML 2.0**. Click **Save**.

Copyright © 2025 Ping Identity Corporation

> **Note**
>
> PingFederate requires a protocol selection to activate the roles. The protocol that you select is not used for this integration.

3. On the **Service Provider > Adapters** screen, create or select an adapter instance that is integrated with the application as shown in SP application integration settings in the PingFederate documentation.
4. On the **Identity Provider > Adapter-to-Adapter Mappings** screen, configure the IdP-to-SP adapter mapping as shown in Adapter-to-adapter mappings in the PingFederate documentation.
5. In your web application, create a hyperlink to allow users to sign on to the SP application. Use the following URL and replace the variables based on the descriptions in the table below:

   `https://<pf_host>:<pf_port>/pf/adapter2adapter.ping?IdpAdapterId=<IdpAdapterId>&SpSessionAuthnAdapterId=<SpAdapterId>`

Variable	Description
<pf_host>	The host name or IP address of the PingFederate server.
<pf_port>	The port number for PingFederate.
<IdpAdapterId>	The instance ID of the GitHub IdP Adapter instance.
<SpAdapterId>	The instance ID of the SP adapter instance that has been integrated with the application.

Troubleshooting

Enabling debug logging

To help with troubleshooting or monitoring, you can turn on activity logging for the adapter.

About this task

These steps are optional. For general information about logging, see Enabling debug messages and console logging in the PingFederate documentation.

Steps

1. Open the `<pf_install>/pingfederate/server/default/conf/log4j2.xml` file for editing.
2. If you want to log activity for PingFederate and all adapters, do the following.
   1. Find the following section.

Copyright © 2025 Ping Identity Corporation

```
<AsyncRoot level="INFO" includeLocation="false">
        <!-- <AppenderRef ref="CONSOLE" /> -->
        <AppenderRef ref="FILE" />
</AsyncRoot>
```

2. Change `INFO` to `DEBUG`.

```
<AsyncRoot level="DEBUG" includeLocation="false">
        <!-- <AppenderRef ref="CONSOLE" /> -->
        <AppenderRef ref="FILE" />
</AsyncRoot>
```

3. If you want to see the adapter activity in the console, remove the comment tags.

```
<AsyncRoot level="INFO" includeLocation="false">
        <AppenderRef ref="CONSOLE" />
        <AppenderRef ref="FILE" />
</AsyncRoot>
```

3. If you want to log activity just for the Microsoft IdP Adapter, add the following line.

```
<Logger name="com.pingidentity.adapter.idp.microsoft" level="DEBUG"/>
```

4. Save the file.

Release notes

Changelog

The following is the change history for the Microsoft Login Integration Kit.

Microsoft Login Integration Kit 1.2 – September 2024

- Updated the Core contract attributes to include the `idToken`, `accessToken`, and `tenantId` values. The `tenantId` value is decoded from the `idToken` JSON Web Token (JWT).

Microsoft Login Integration Kit 1.1 – July 2024

- Added the ability to send a `login_hint` parameter to pre-fill the **Username** field on the Microsoft user sign-on page. Learn more in Microsoft IdP Adapter settings reference.
- Added the ability to send a prompt parameter to confirm user interaction during the authentication process. Learn more in Microsoft IdP Adapter settings reference.

Copyright © 2025 Ping Identity Corporation

- Added the ability to send other request parameters in the authorization request to Microsoft. Learn more in Configuring an adapter instance.

Microsoft Login Integration Kit 1.0.1 – August 2021

- Added the **Supported Account Types** setting to simplify matching account types with your Azure application settings. The **Tenant ID** field is now called **Single Tenant ID** for clarity. If you're upgrading, your current adapter configuration will still work.

Microsoft Login Integration Kit 1.0 – June 2021

- Initial release.
- Added a setting to use browser redirect or pop-up window for the sign-on presentation.
- Added customizable sign-on templates for the pop-up window presentation.
- Added customizable user-facing language-pack messages
- Added support for the PingFederate authentication API.
- Added support for the JavaScript Widget for the PingFederate Authentication API.

Known issues and limitations

The following are known issues or limitations with the Microsoft Login Integration Kit.

Known issues

There are no known issues.

Known limitations

There are no known limitations.

Download manifest

The following files are included in the Microsoft Login Integration Kit download.

- `Legal.pdf` – copyright and license information
- `dist/pingfederate/server/default` – contains the integration files
  - `deploy` – contains the Java libraries
    - `pf-microsoft-idp-adapter-<version>.jar` – JAR file that contains the Microsoft IdP Adapter.
  - `conf` – contains the HTML template that presents the Microsoft sign-on form.

Copyright © 2025 Ping Identity Corporation

- `language-packs` – contains files with customizable user-facing messages
  - `authn-api-messages.properties` – a version of the common PingFederate authentication API variable file that includes adapter-specific messages
    - `pingfederate-microsoft-adapter-messages.properties` – a variable file that customizes the messages that appear on the template files.
- `template` – contains user-facing HTML template files
  - `assets` – contains functional scripts and files used by the template
    - `css/microsoft.css` – a CSS file that customizes the appearance of the template files.
      - `fonts/end-user` – contains template fonts and icons
    - `icons` – contains template icons
    - `proxima-nova` – contains template fonts
      - `images` – contains template image files
    - `ping-logo.svg` – an image file with company branding
    - `microsoft-pop-up-template.html` – a template that opens a pop-up window to prompt the user to sign on.
    - `microsoft-post-auth-template.html` – a template that returns the user to the PingFederate sign-on flow after they sign on with Microsoft.

- `lib/pf-authn-api-sdk-<version>.jar` – a JAR file that contains the PingFederate Authentication API SDK

Copyright © 2025 Ping Identity Corporation

.NET Integration Kit

The .NET Integration Kit allows PingFederate to communicate user attributes with .NET-based web applications.

When PingFederate is serving an identity provider (IdP) role, it can receive user attributes from a .NET IdP application. When PingFederate is serving a service provider (SP) role, it can send user attributes to a .NET SP application.

> **Tip**
>
> For new integrations, we encourage you to consider the Agentless Integration Kit, which can integrate with a variety of platforms using a modern RESTful approach.

Components

- OpenToken Adapter
  - An adapter that allows PingFederate to send or receive user attributes in the OpenToken format. See OpenToken Adapter in the PingFederate documentation.
- OpenToken Agent
  - An agent that runs within your .NET application that allows it to send or receive user attributes in the OpenToken format.
- Sample application
  - The Integration Kit distribution also contains sample IdP and SP applications. The applications may be installed quickly for testing OpenToken processing and to provide a working demonstration of end-to-end single sign-on (SSO) and single logout (SLO). Source code and supporting files are included in the distribution and may be modified or used as a reference for application developers.

Intended audience

This document is intended for PingFederate administrators and web-application developers.

Before you start, you should be familiar with the following parts of the PingFederate documentation:

- Managing IdP adapters
- Managing SP adapters
- OpenToken Adapter

System requirements

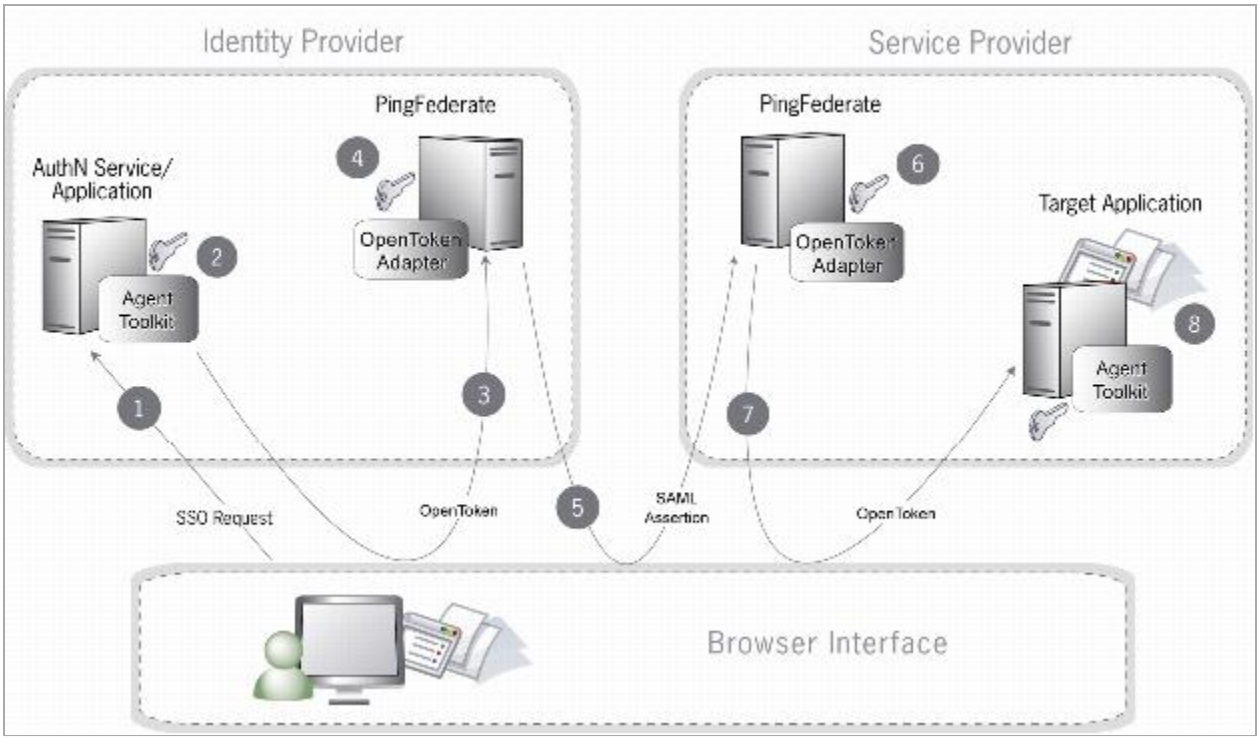
- PingFederate 9.0 or later.
- Microsoft .NET Framework 4.0 for the agent application.

Copyright © 2025 Ping Identity Corporation

Overview of the SSO flow

With the .NET Integration Kit, PingFederate exchanges user attributes with your .NET application through an OpenToken token.

The following figure shows a basic identity-provider (IdP)-intiated single sign-on (SSO) scenario in which PingFederate federation servers using the .NET Integration Kit exist on both sides of the identity federation:



Description

1. A user initiates an SSO transaction.
2. The IdP application inserts user attributes into the agent toolkit for .NET, which encrypts the data internally and generates an OpenToken token.
3. A request containing the OpenToken is redirected to the PingFederate IdP server.
4. The server invokes the OpenToken IdP Adapter, which retrieves the OpenToken, decrypts, parses, and passes the user attributes to the PingFederate IdP server. The PingFederate IdP server then generates a SAML assertion.
5. The SAML assertion is sent to the SP site.
6. The PingFederate SP server parses the SAML assertion and passes the user attributes to the OpenToken SP Adapter. The adapter encrypts the data internally and generates an OpenToken.
7. A request containing the OpenToken is redirected to the SP application.
8. The Agent Toolkit for .NET decrypts and parses the OpenToken and makes the user attributes available to the SP Application.

Copyright © 2025 Ping Identity Corporation

Setup

Updating the OpenToken Adapter

The .NET Integration Kit relies on the OpenToken Adapter that is distributed with the Java Integration Kit. Update the OpenToken Adapter to get the latest feature and security updates.

Steps

1. Download the Java Integration Kit `.zip` archive from the **Add-ons** tab of the PingFederate downloads page.
2. Stop PingFederate.
3. Delete the `opentoken-adapter-<version>.jar` file from your `<pf_install>/pingfederate/server/default/deploy` directory.
4. From the Java Integration Kit `.zip` archive, copy the contents of `dist/pingfederate` to your `<pf_install>/pingfederate` directory.

   > **Note**
   >
   > The `commons-collections`, `commons-beanutils`, and `commons-logging` libraries are provided as a convenience and should be installed only if they are not already contained in the application `CLASSPATH`.

5. Start PingFederate.
6. If you operate PingFederate in a cluster, repeat steps 2-5 for each engine node.

Deploying the OpenToken Agent

To allow your web application to send and receive user attributes in OpenToken format, deploy theplatformOpenToken agent on your application server.

Steps

1. Download the .NET Integration Kit `.zip` archive from the PingFederate server add-ons page.
2. Stop your web application.
3. If you are upgrading from a previous version of the .NET Integration Kit, delete your existing `opentoken-agent.dll` file.
4. From the integration `.zip` archive, copy `dist/opentoken-agent.dll` to a location that your .NET application can access.
5. Start your web application.
6. If you have more than one web application, repeat steps 1-5 for each application.

Copyright © 2025 Ping Identity Corporation

Custom application setup

You can configure the .NET Integration Kit to integrate PingFederate with your identity provider (IdP) or service provider (SP) application.

If you would like to see a working demonstration of the .NET Integration Kit before integrating your own applications, see Sample applications setup instead.

Configuring an OpenToken SP Adapter instance

Configure the OpenToken Adapter to determine how PingFederate communicates with your service provider application.

Steps

1. In the PingFederate administrative console, go to **Authentication > Integration > IdP Adapters**. Click **Create new Instance**.
2. On the **Type** tab, set the basic adapter instance attributes.
    1. In the **Instance Name** field, enter a name for the adapter instance.
    2. In the **Instance ID** field, enter a unique identifier for the adapter instance.
    3. From the **Type** list, select **OpenToken Adapter**. Click **Next**.
3. On the **Instance Configuration** tab, configure the adapter instance by referring to Configuring an OpenToken SP Adapter instance in the PingFederate documentation. Click **Next**.

    > **Note**
    >
    > If you use .NET to protect multiple sites on the same domain, in the OpenToken Adapter instance configuration, select **None** for SameSite Cookie, and select the **Secure Cookie** check box.

4. Export the configuration file:
    1. On the **Actions** tab, click **Download**, and then click **Export**.
    2. Save `agent-config.txt`. Click **Next**.
5. On the **Extended Contract** tab, add any attributes that you expect to retrieve other than the SAML subject. Click **Next**.
6. On the **Target App Info** tab, enter the basic information about your SP application. Click **Next**.
7. On the **Summary** tab, check and save your configuration. Click **Save**.
8. Create or update an identity provider (IdP) connection to use the OpenToken Adapter instance as shown in Service provider SSO configuration in the PingFederate documentation.

Integrating the OpenToken agent into your .NET application

To use the .NET Integration Kit, modify your application to use the OpenToken Agent.

The agent API allows your web application to directly read or write an OpenToken package.

Copyright © 2025 Ping Identity Corporation

Instantiate the OpenToken Agent by invoking a constructor and loading the `agent-config.txt` configuration file that you saved in Configuring an OpenToken SP Adapter instance. This configuration file includes the name of the cookie that the agent object will write, as well as the key to use when encrypting a new OpenToken. If the agent does not find an agent-config.txt file, it throws an exception.

Sample code

Modify your .NET application based on the following sample code.

```
using System;using System.
Collections.Generic;
using System.Text;
using opentoken;
using System.IO;
using opentoken.util;
using System.Collections;
using System.Collections.Generic;
. . . .
Agent agent = new Agent( “<PATH_TO_FILE>/agent-config.txt”);
```

IdP single sign-on integration

When PingFederate is configured as an identity provider (IdP), it needs to be able to identify a user prior to issuing a SAML assertion for that user. When using the OpenToken Adapter with PingFederate, this means that the PingFederate server attempts to read a cookie or query parameter containing an OpenToken and then use the values within to identify the user. The application that starts the SSO must include an OpenToken so that PingFederate can identify the user. Use the Agent API to write an OpenToken. The API is a .NET object that provides access to functionality for writing an OpenToken to a given HTTP response.

Writing attributes

The writeToken method takes a `System.Collections.IDictionary` collection of attributes and encodes them into an OpenToken, which is then written to the HTTP response.

The collection of attributes must contain a key named `subject`.

If any errors are encountered while creating or writing the token to the HTTP response, a `TokenException` is thrown.

The following code snippet shows the writeToken method:

Copyright © 2025 Ping Identity Corporation

```
IDictionary userInfo = new Dictionary<String, String>();
// Add userId for the logged on user as the token subject
userInfo.Add(Agent.TOKEN_SUBJECT, <userId>);
String returnUrl = "https://<{pingfed} DNS>:9031" + Request["resume"];
. . . .
try {
   UrlHelper urlHelper = new UrlHelper(returnUrl);
   agent.WriteToken(userInfo,Response,urlHelper,false);
   returnUrl = urlHelper.ToString();
}
catch(TokenException e) {
  // Handle exception
}
```

Passing multi-value attributes

The Agent Toolkit for .NET supports passing multi-value attributes to PingFederate. Each attribute appears in its own discrete `<AttributeValue>` element in the SAML 2.0 assertion or as a JSON array value in OAuth-based protocols. Multi-value attributes are passed using the `opentoken.MultiStringDictionary` collection.

The following code snippet shows how to pass multi-value attributes:

```
MultiStringDictionary userInfo = new MultiStringDictionary();
// Add userId for the logged on user as the token subject
userInfo.Add(Agent.TOKEN_SUBJECT, <userId>);

// Add an attribute GROUP with multiple values
userInfo.Add("GROUP", "Administrators");
userInfo.Add("GROUP", "Users");
String returnUrl = "https://<{pingfed} DNS>:9031" + Request["resume"];
. . . .
try {
   UrlHelper urlHelper = new UrlHelper(returnUrl);
   agent.WriteToken(userInfo,Response,urlHelper,false);
   returnUrl = urlHelper.ToString();
}
catch(TokenException e) {
   // Handle exception
}
```

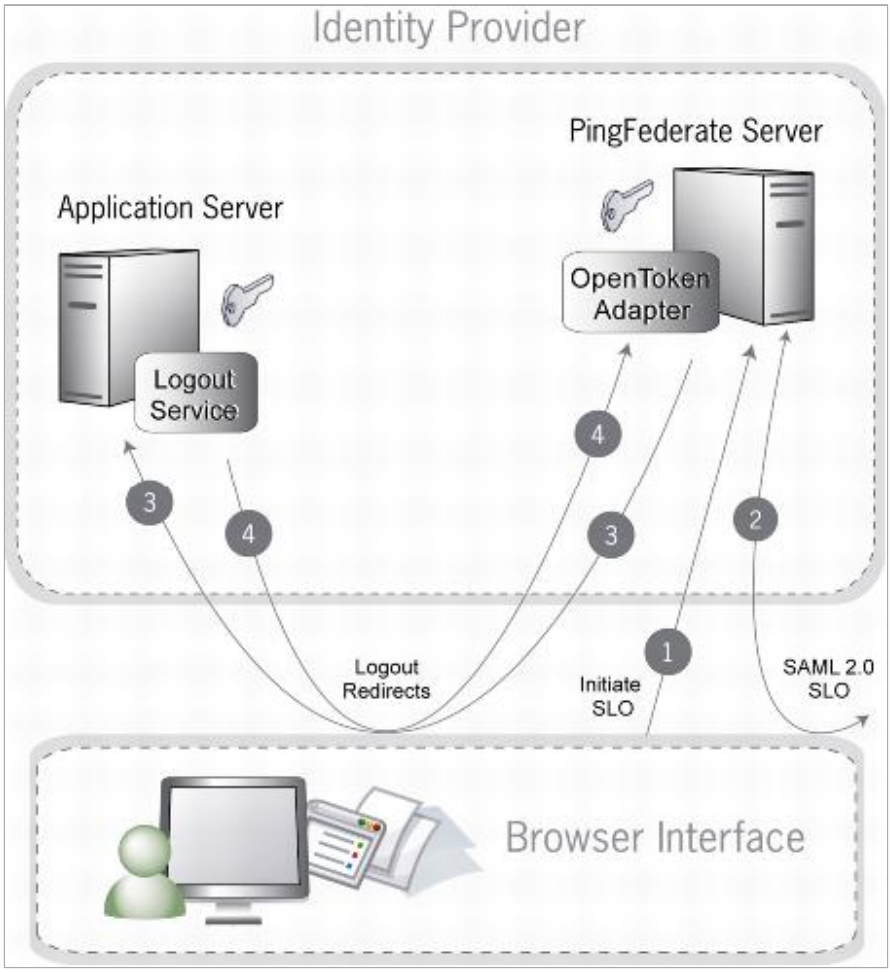
IdP single logout integration

When an IdP PingFederate server receives a request for SLO, it redirects the user's browser to the Logout Service defined in the IdP OpenToken Adapter configuration. The redirect URL includes an OpenToken containing the user attributes defined in the IdP OpenToken Adapter instance for the partner connection. The Logout Service should remove the user's session on the application server and redirect the user's browser back to the IdP PingFederate server.

Logout flow

The following diagram shows the flow of IdP-initiated SLO, but the architecture would also support SP-initiated SLO.

Copyright © 2025 Ping Identity Corporation



1. User initiates a single logout request. The request targets the PingFederate server's `/idp/startSLO.ping` endpoint.
2. PingFederate sends a logout requests and receives responses for all SPs registered for the current SSO session.
3. PingFederate redirects the request to the IdP web application's Logout Service, which identifies and removes the user's session locally.
4. The application Logout Service redirects back to PingFederate to display a logout-success page.

Processing logout requests

The following code snippet shows how to process a logout request and send it back to PingFederate through the user's browser:

```
// Remove local session
. . . .
IDictionary userInfo = new Dictionary<String, String>();
// Add userId for the logged on user as the token subject
userInfo.Add(Agent.TOKEN_SUBJECT, <userId>);
String returnUrl = "https://<{pingfed} DNS>:9031" + Request["resume"];
Response.Redirect(returnUrl);
```


Copyright © 2025 Ping Identity Corporation

SP single sign-on integration

When PingFederate is configured as an SP, it takes inbound SAML assertions and converts them to some local format (cookie or otherwise) that can be used by an application to create a user's session. For an `OpenToken`, the PingFederate adapter takes the attributes and values from the SAML assertion and stores them in an `OpenToken` cookie or query parameter in the user's browser. The user is then redirected to the target application, which can then identify the user from the `OpenToken`, using the `Agent` API.

As with the IdP, you can use the Agent API to read tokens directly. The Agent API is a .NET class that provides access to functionality for reading an `OpenToken` from a given HTTP request.

Reading attributes

The readToken method inspects the cookie (or query parameters, depending on the agent configuration), decodes the OpenToken, and returns a collection of attributes.

If there is no token, it returns a `null` result. If an errors occurs while reading the token, it returns a `null` result and a `TokenException` is thrown.

The following code snippet shows the readToken method:

```
try {
   IDictionary userInfo = agent.ReadToken(Request);
   if(userInfo != null) {
      String username = (String)userInfo[Agent.TOKEN_SUBJECT];
   }
}
catch(TokenException e) {
   // Handle exception
}
```

Receiving multi-value attributes

The Agent Toolkit for .NET supports receiving multi-value attributes from PingFederate. Multi-value attributes are passed using the `opentoken.MultiStringDictionary` collection.

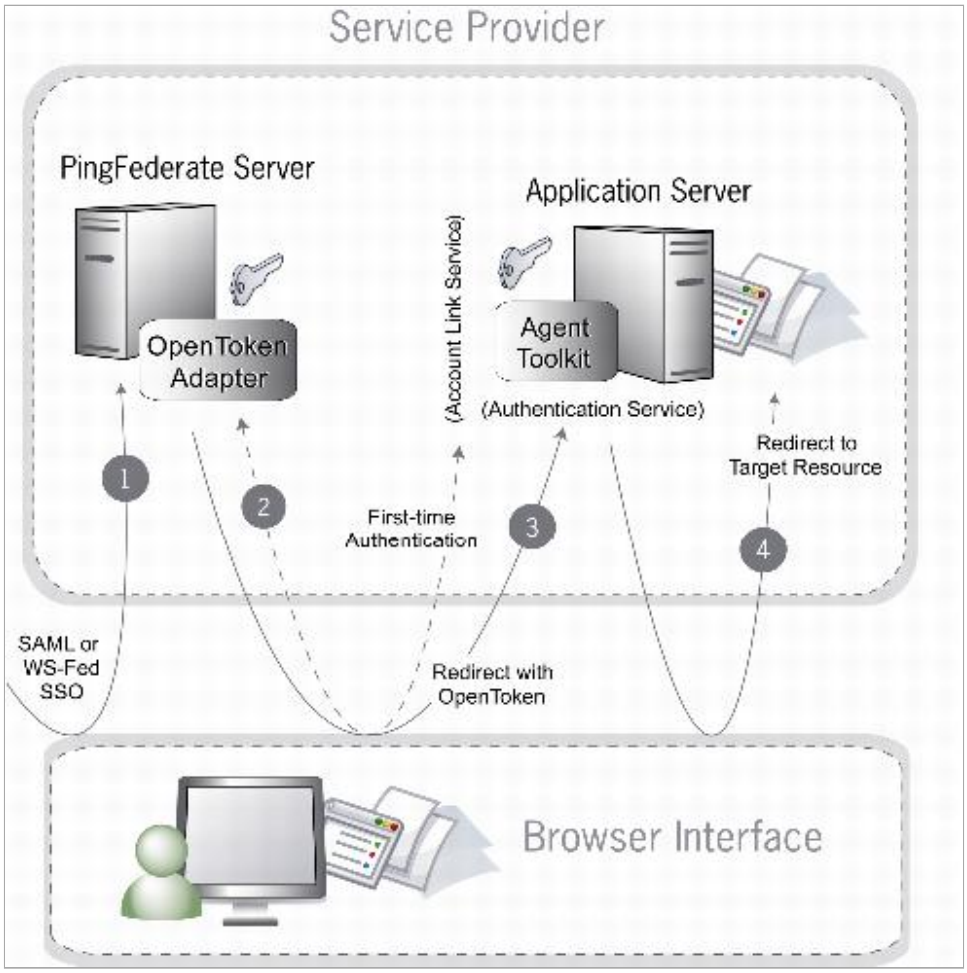
The following code snippet shows how to process multi-value attributes:

```
try {
   MultiStringDictionary userInfo =
    agent.ReadTokenMultiStringDictionary(Request);
   if(userInfo != null) {
      String username = userInfo[Agent.TOKEN_SUBJECT][0];
      List<String> groups = userInfo["GROUP"];
   }
}
catch(TokenException e) {
   // Handle exception
}
```

Copyright © 2025 Ping Identity Corporation

SP single sign-on integration using account linking

If an SP's SSO implementation employs account linking, the flow of events is somewhat different since a user must authenticate to the SP application the first time SSO is initiated (for more information, see Key concepts in the PingFederate documentation). In this case, PingFederate and the OpenToken Adapter support an integration mechanism to redirect the user to an Account Link Service to which a user can authenticate initially. Upon successful authentication, the user's browser is redirected back to PingFederate with an OpenToken, which PingFederate uses to create an account link for the user. For subsequent SSO requests, PingFederate uses the account link established in the first SSO request to identify the user. It then creates an OpenToken and sends it to the Authentication Service associated with the application.



1. PingFederate receives an assertion under either the SAML 2.0, OpenID Connect, or WS-Federation protocol.
2. If this is the first time the user has initiated SSO to this SP, PingFederate redirects the browser to the Application Server's Account Link Service, where the user must authenticate. Upon successful authentication, an OpenToken is returned to PingFederate, and an account link is established for this user within PingFederate. This account link is used on subsequent SSO transactions.
3. PingFederate retrieves the local user ID from its account link data store. Through the OpenToken Adapter, PingFederate generates an OpenToken based on the assertion and account link. PingFederate then redirects the user's browser to the web application's SSO Authentication Service, passing the OpenToken in the redirect.
4. The Authentication Service extracts the contents of the OpenToken, establishes a session for the user, and redirects the user's browser to the Target Resource (the resumePath URL sent as a query parameter).

Copyright © 2025 Ping Identity Corporation

Linking accounts

In an Account Linking event, the user's browser is redirected to the configured Account Linking service in the SP OpenToken Adapter instance. The application should capture the resumePath upon a GET request to this URL with something similar to the following:

```
IDictionary userInfo = new Dictionary<String, String>();
// Add userId for the logged on user as the token subject
userInfo.Add(Agent.TOKEN_SUBJECT, <userId>);
String returnUrl = "https://<{pingfed} DNS>:9031" + Request["resume"];
. . . .
try {
   UrlHelper urlHelper = new UrlHelper(returnUrl);
   //For sample code that instantiates and configures an Agent instance, see the
   //"Integrating the OpenToken Agent into your application" topic in the documentation
   agent.WriteToken(userInfo,Response,urlHelper,false);
   returnUrl = urlHelper.ToString();
}
catch(TokenException e) {
    // Handle exception
}
Response.Redirect(returnUrl);
```

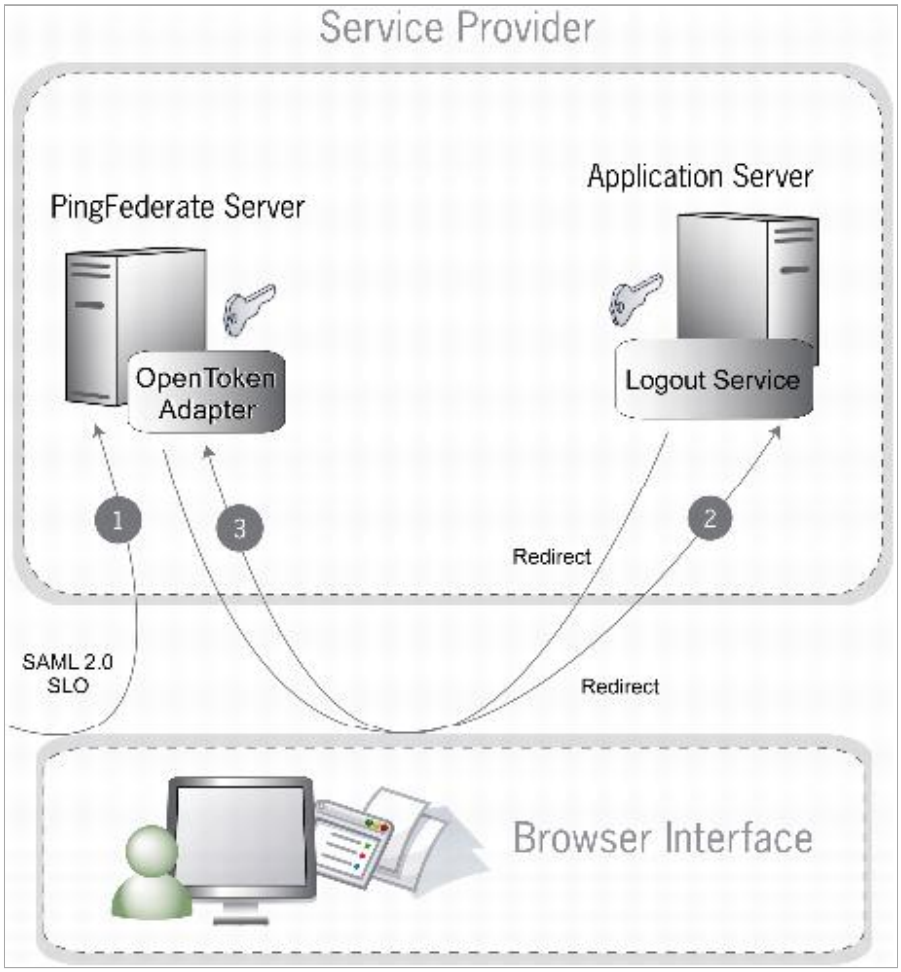
SP single logout (SLO)

When an SP PingFederate server receives a request for SLO, it redirects the user's browser to the Logout Service as configured in the SP OpenToken Adapter instance. As part of the redirect, PingFederate and the OpenToken Adapter include both an OpenToken and a resumePath query parameter.

- The OpenToken includes attributes about the user.
- The resumePath query parameter provides the target application URL.

A user can have multiple sessions. This logout sequence, as shown in the following diagram, will occur for each of the user's sessions controlled by the SP PingFederate server.

Copyright © 2025 Ping Identity Corporation



Sequence

1. PingFederate receives an SLO request under the SAML 2.0 protocol.
2. PingFederate, via the OpenToken Adapter, redirects the browser to the Application Server's Logout Service.
3. The Logout Service returns to PingFederate, indicating that the logout was successful.

The code needed to perform an SP SLO is identical to that required for an IdP SLO.

Testing

You can test the .NET Integration Kit using one of the sample applications bundled with this distribution.

See Sample applications setup.

> **Caution**
>
> Do not install the `data.zip` archive as shown in Configuring PingFederate to use the sample applications. This will overwrite the PingFederate configuration that you just completed.

Copyright © 2025 Ping Identity Corporation

Sample applications setup

You can configure the .NET Integration Kit to work with the included sample applications.

The applications provide a means of testing an end-to-end Identity Provider (IdP) and Service Provider (SP) integration with PingFederate using this integration kit.

This sample application distribution includes startup components that automatically configure PingFederate to act as both an IdP and an SP:

- The IdP server is configured to look up and send authentication information to the SP.
- The SP server is configured to forward this information to the SP sample application to create the local user session. The SP server will also be configured to send authentication requests to the IdP on behalf of local users.

Components

- Identity provider (IdP) sample application
  - Acts as an IdP application in your demonstration environment.
- Service provider (SP) sample application
  - Acts as an SP application in your demonstration environment.
- PingFederate configuration archive
  - This `data.zip` archive automatically configures PingFederate with OpenToken Adapter instances to work with the two sample applications.

System requirements

- PingFederate 9.0 or later.
- The OpenToken Adapter. See Updating the OpenToken Adapter.
- Windows Server 2008 64-bit or 32-bit.
- Microsoft Internet Information Services (IIS) with the following:
  - Running version 7 or later.
  - The correct time (recently synchronized).
  - The correct time zone.
- Microsoft .NET Framework 4.0 installed and registered with IIS.

Configuring PingFederate to use the sample applications

To see a working demonstration of the .NET Integration Kit, deploy the configuration archive and sample applications.

About this task

Copyright © 2025 Ping Identity Corporation

The sample configuration archive configures a single instance of PingFederate with an example integration that uses both the IdP and SP sample applications. It automatically creates two instances of the OpenToken Adapter.

> **Caution**
>
> Deploying the configuration archive will destroy your existing PingFederate configuration. Test it on a fresh installation of PingFederate, or back up your current configuration as shown in Exporting an archive in the PingFederate documentation.

Steps

1. Start PingFederate.
2. In the Agentless Integration Kit `.zip` archive, copy `sample/data.zip` to `<pf_install>/pingfederate/server/deploy/drop-in-deployer`.
3. If your PingFederate instance is on a different computer than the sample applications, modify your configuration.
    1. In your IdP and SP adapter instance configurations, change any URLs with `http://localhost` to point to the correct host and port for the sample applications on your Internet Information Services (IIS) server.
    2. On the **Identity Provider > Default URL** tab, change `http://localhost` to point to the correct host and port for the sample applications. Repeat for the **Service Provider > Default URL** tab.
    3. In a browser, go to the sample application URL. On the **Configuration Options** page, change the **PF Host Name** to point to the host and port of your PingFederate server.

Configuring IIS to use the sample applications

To see a working demonstration of the .NET Integration Kit, deploy the sample applications and configure your Internet Information Services (IIS) server.

Steps

1. From the .NET Integration Kit `.zip` archive, copy the `sample/IdpSample` and `sample/SpSample` directories to your IIS server.
2. Create IIS applications that point to the `IdPSample` and `SPSample` directories.
    1. On the IIS Manager navigation pane, right-click your web site, and then click **Add Application**.
    2. In the **Alias** field, enter a name, such as `PingFederate .NET IdP Sample`.
    3. From the **Application Pool** list, select an application pool that uses .NET v4.0 and runs in Integrated mode.
    4. In the **Physical path** field, enter the path to the `IdpSample` directory that you copied in step 1.
    5. Finish configuring the application. Click **OK**.
    6. Repeat steps a-e to create a similar application for the `SpSample` directory.
3. Set the default web site document to `Default.aspx`.
    1. On the IIS Manager navigation pane, select your web site.

Copyright © 2025 Ping Identity Corporation

2. In the details window, double-click **Default Document**.
3. Set `Default.aspx` as the default content page.

4. Give ASP.NET permission to write to the sample application `config` directories.
    1. In File Explorer, right-click your `IdpSample/config` directory, and then click **Properties**.
    2. On the **Config Properties** dialog, on the **Security** tab, in the **Group of user names** section, select the ASP.NET account (**IUSR** or **IIS_IUSRS**). Click **Edit**.
    3. On the **Permissions for Config** dialog, select **Write**. Click **OK**.
    4. On the **IdpSample** dialog, click **OK**.
    5. Repeat steps a-d for the `SpSample/config` directory.

Using the IdP sample application

The .NET Integration Kit identity provider (IdP) sample application demonstrates IdP-initiated single sign-on (SSO) and single logout (SLO) use cases.

About this task

The IdP sample application simulates the IdP-initiated SSO/SLO scenario in which users authenticate to an IdP locally in order to access a remote SP application. In this scenario, users may be accessing a company portal that provides links to partner applications such as local news and weather, stock market information, and HR and 401(k) benefits.

When you authenticate locally to the IdP sample application, no communication occurs between that application and PingFederate. The user authenticates using the local user store; no SAML use cases are invoked. However, when you click a link to a third-party application, such as your company's health care provider, the IdP initiates an SSO transaction.

Steps

1. Start the PingFederate and Internet Information Services (IIS) servers.
2. In a browser, open the sample application:

    `https://hostname/IdpSample`
3. On the main page, click **Login Locally**.
4. On the **Identity Provider Login** page, sign on as any of the listed users with a password of `test`.
5. Click **Login**.
6. On the **Identity Provider** page, try the following:
    1. **Optional:** To begin an IdP-iniated SSS to the SP sample application, click the **Single Sign-On**. This starts a user session on the SP and redirects you to the SP sample application. For more information, see Using the SP Sample Application.
    2. **Optional:** After signing on to the SP sample application and returning to the **Identity Provider** main page, click **Single Sign-Out** to initiate a SLO request to the SP. This ends your user session on the SP as well as your local user session.

Copyright © 2025 Ping Identity Corporation

Using the SP sample application

The .NET Integration Kit service provider (SP) sample application demonstrates SP-initiated single sign-on and single logout use cases.

Steps

1. Start the PingFederate server and the IIS server.
2. In a browser, open the sample application:

   `https://hostname/SpSample`
3. From the list, select an identity provider (IdP) connection, and then click **Single Sign-On** to begin an SP-initiated single sign-on (SSO) transaction.
4. On the **Identity Provider** login page, sign on as any of the listed users with a password of `test`.

   > **Note**
   >
   > If you have already authenticated with the identity provider, you do not need to authenticate again unless either the ForceAuthn or IsPassive option is checked in the **Advanced Options** section of the application.
5. Click **Login**.
6. To initiate a single logout (SLO) request, click **Single Sign-Out**.

Release notes

Changelog

.NET Integration Kit 2.5.4 – March 2020

- Added support for the SameSite cookie flag in web browsers.

.NET Integration Kit 2.5.3 – September 2019

- Updated the included OpenToken Adapter to version 2.5.8.
- Updated the included OpenToken Agent to version 2.5.3.
- Improved the way token timestamps are handled.

.NET Integration Kit 2.5.2 – January 2016

- Updated sample application data archive.
- Updated OpenToken Adapter to version 2.5.7.

.NET Integration Kit 2.5.1 – December 2012

- Updated to address security issue found since the previous release
- Added support for OpenToken 2.5.1 Adapter and the OpenToken 2.5.1 Agent

Copyright © 2025 Ping Identity Corporation

.NET Integration Kit 2.5 – June 2012

- Added support for the Microsoft .NET Framework 4.0
- Added support for 64-bit CPUs
- Removed support of the Microsoft .NET Framework 2.0

.NET Integration Kit 2.4 – March 2010

- Added token Replay Prevention to the OpenToken IdP Adapter Advanced Settings

.NET Integration Kit 2.3 – November 2008

- Added POST Transport Method for OpenToken when used by a Service Provider
- Added configuration to specify session cookie vs. persistent cookie
- Added option to set the "`Secure`" attribute on an OpenToken when cookie is used
- Added ability to bypass password obfuscation and strength enforcement for backward compatibility with previous .NET OpenToken agents
- Correctly handles `null` parameters for SOAP SLO
- Empty query string (`?`) is not automatically appended to the URL when redirecting to TargetResource
- TargetResource URL is URL encoded
- Corrected `not-before tolerance processing`

.NET Integration Kit 2.2 – June 2008

- Added support for SAML 2.0 `isPassive` and `ForceAuthn`
- Enforced UTF-8 encoding within OpenToken
- Symmetric key in the OpenToken agent configuration file is encrypted
- Combined the OpenToken adapter and OpenToken Java library jar files into a single adapter file for easier deployment to PingFederate

.NET Integration Kit 2.1 – April 2008

- Added `AgentConfiguration` class to simplify Agent instantiation
- Added Agent Toolkit API HTML Help file

.NET Integration Kit 2.0 – December 2007

Modified to use an open-standard, secure token called OpenToken to pass user information between an application and PingFederate. The OpenToken is passed through the user's browser as a URL query parameter or an HTTP cookie. The data within the OpenToken is a set of key/value pairs, and the data is encrypted using common encryption algorithms.

.NET Integration Kit 1.2.1 – August 2007

- Fixed buffer overflow issue in OpenToken Agent to allow dynamic buffer size for reading `PFTOKEN`
- OpenToken Agent `extractFromRequest` method corrected to only delete cookie if the `PFTOKEN` transport is cookie

Copyright © 2025 Ping Identity Corporation

- OpenToken Agent no longer writes messages to the Windows Event Log
- Modified to allow backward compatibility of the Standard Adapter 1.2.1 with PingFederate 4.0
- Bundled .NET sample application with distribution

.NET Integration Kit 1.2 – May 2007

- Added an option to encode `PFTOKEN` for handling special characters
- Added an additional constructor to allow `PFTOKEN` to use default properties for all configuration options except password, holder name, and max age
- Added `PFTOKEN` time stamp information in the log file

Known issues and limitations

Known issues

There are no known issues.

Known limitations

- The OpenToken name given to any one Adapter instance must be unique within the given federation. This however is not enforced in the user interface.
- SP and IdP Sample applications are bundled with the integration kit only from presentation point of view. There is ability to create more than one unique session under the PingFederate IdP associated with a user using the demo sample applications. The situation should not appear in production environment if you are not using the bundled demo sample application in production.
- The SP adapter has the ability to send extended attributes through cookies or query parameters along with the OpenToken. So long as these additional attributes have only one value each, this works fine, however, if there are any multi-value attribute only the first value would be sent, partly because we can only set one cookie per value name.
- After replacing the agent configuration, it is sometimes necessary to disallow and then re-allow IIS write access to the file.

Download manifest

The following files are included in the .NET Integration Kit `.zip` archive:

- `ReadMeFirst.pdf` : Contains links to this online documentation.
- `/legal` : Contains the following document:
  - `Legal.pdf` : Copyright and license information
- `/dist` : Contains the OpenToken agent for .NET:
  - `opentoken-agent.chm` : Agent Toolkit API Documentation
  - `opentoken-agent.dll` : Agent Toolkit for .NET 4.0

Copyright © 2025 Ping Identity Corporation

- `/sample`: Contains the .NET sample applications:
  - `/IdpSample`: IdP Sample Application
  - `/SpSample`: SP Sample Application
  - `data.zip`: PingFederate configuration archive for the Sample Applications

> **Note**
>
> The OpenToken Adapter is not included in this integration `.zip` archive. You can find more information in Updating the OpenToken Adapter.

Copyright © 2025 Ping Identity Corporation

MobileIron Integration Kit

The MobileIron Integration Kit includes an Identity Provider (IdP) adapter to integrate PingFederate with any instance of the MobileIron platform.

This integration enables customers to make policy decisions based on the employee's device posture, which mitigates the risk of corporate resources being accessed from employees' mobile and desktop devices.

Intended audience

This document is intended for PingFederate administrators.

If you need help during the setup process, see the following resources:

- The following sections of the PingFederate documentation:
  - Identity provider SSO configuration
  - Managing IdP adapters
  - Authentication policies
- If you want to use MobileIron to manage the PingID® app on devices, see Configuring MobileIron for PingID in the PingID documentation for more information.

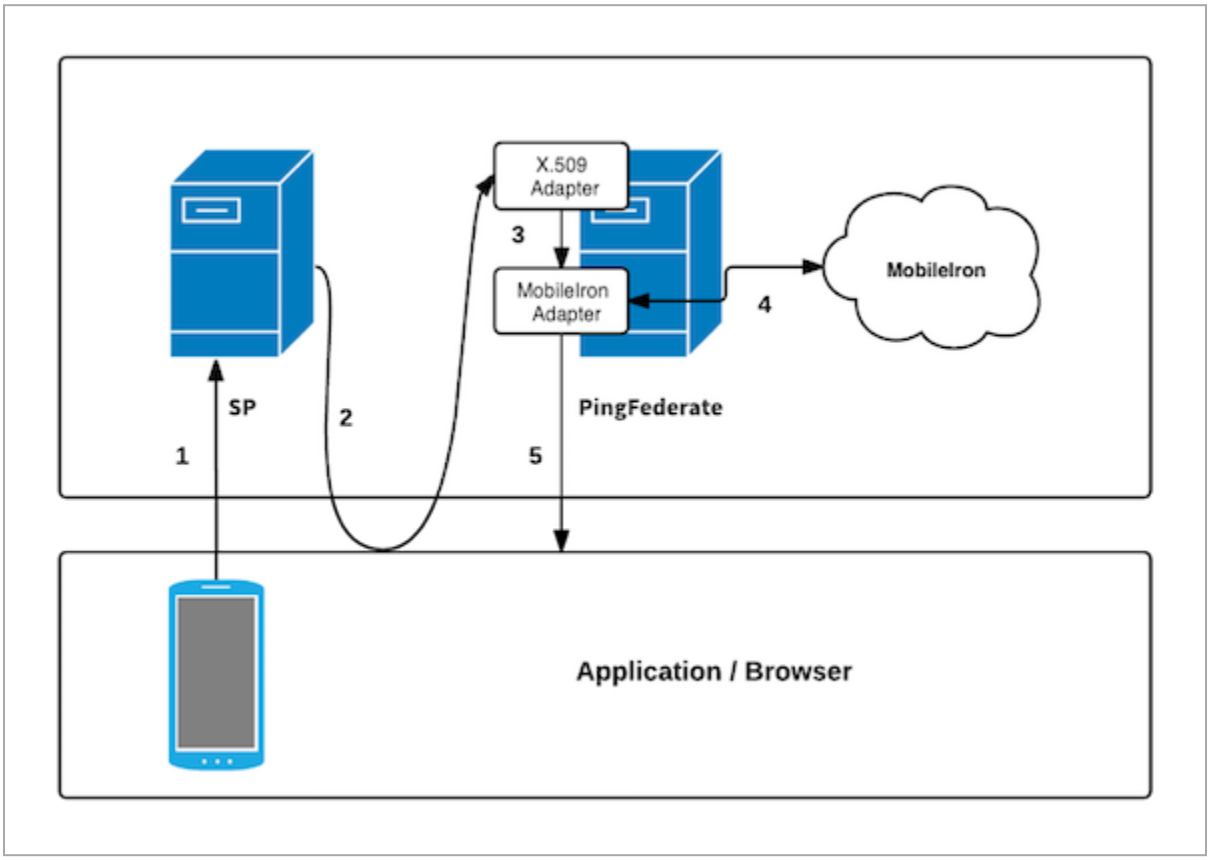
System requirements

- PingFederate 8.4.4 or later
- PingFederate X.509 Integration Kit 1.2.1
- A MobileIron Cloud, Core, or Connected Core deployment (on-premise or cloud)
- A device registered with MobileIron (for testing)

Overview of the SSO flow

The following figure shows a basic SSO scenario in which a PingFederate server authenticates users to an SP application using the MobileIron Adapter.

Copyright © 2025 Ping Identity Corporation



Processing Steps

1. A user with an MobileIron enrolled device requests access to an SP resource. The request is redirected to PingFederate to perform X.509 Authentication.
2. The browser requests the user's X.509 certificate. The PingFederate X.509 Certificate Adapter validates the certificate against a list of issuers. If no issuers are specified in the Adapter setup, it uses the server's list of trusted CAs instead.
3. The certificate is validated and the Device Identifier is parsed from the certificate. The Device Identifier is then passed to the MobileIron Adapter.
4. The Device Identifier is used to contact the MobileIron Device API to retrieve the device's posture.
5. The result of the authentication is returned, and if successful, the user is redirected to the requested resource.

Supported attributes reference

The MobileIron IdP Adapter retrieves the following attributes. Policy decisions can be applied against any of these.

Attribute	Description
`Blocked`	
`compliance`	True if the device is in compliance with its MDM security policies; false otherwise.

Copyright © 2025 Ping Identity Corporation

Attribute	Description
`compromised`	True if the device is compromised; false otherwise.
`identifier`	
`imei`	IMEI number of the device.
`iosUdid`	Unique device identifier.
`lastCheckInTime`	
`macAddress`	The Wi-Fi MAC address.
`manufacturer`	Manufacturer is automatically reported by the device during registration.
`model`	Model is automatically reported by the device during registration.
`os`	Operating system.
`osVersion`	OS version.
`ownership`	A value of `COMPANY` signifies that the device is corporate owned. Other values are: `EMPLOYEE` and `UNKNOWN`.
`phoneNumber`	
`quarantined`	True if the device is quarantined by the MDN; false otherwise.
`RFC822Name`	
`registrationTime`	
`serialNumber`	Serial number.
`status`	
`userId`	User ID.
`userUdid`	

Setup

Deploying the integration files

To get started with the integration, deploy the MobileIron IdP Adapter files to your PingFederate directory.

About this task

Copyright © 2025 Ping Identity Corporation

> **Note**
>
> If you operate PingFederate in a cluster, the following steps refer to the console node.

Steps

1. Download the MobileIron IdP Adapter `.zip` archive from the Ping Identity Integration Directory.
2. Stop PingFederate.
3. If you are upgrading an existing deployment, delete `pf-mobileiron-integration-kit-<version>.jar` from `<pf_install>/pingfederate/server/default/deploy`.
4. Extract the `.zip` archive and merge the contents of the `dist` directory with your `<pf_install>/pingfederate` directory.
5. Start PingFederate.
6. If you operate PingFederate in a cluster, repeat steps 2-5 for each engine node.

Creating a MobileIron adapter instance

About this task

This section describes how to install and configure the MobileIron Adapter.

Steps

1. Login to PingFederate as an administrator and navigate to **IdP Configuration > Adapters**.
2. Click **Create New Instance** and fill in the fields for **Name**, **ID** and select **MobileIron Adapter 1.0** for the Type of the adapter. Click **Next**.

   > **Note**
   >
   > The Name is any you choose for identifying this adapter instance. The ID is used internally and must only contain alphanumeric characters.

3. Fill in hostname for your MobileIron instance, your API user's username, and API user's password for querying the CPS API. Click **Next**.
4. The **Extended Contract** page/tab lists all the available attributes MobileIron manages for devices. These are the attributes that can be used to make authentication decisions for your users. Click **Next**.
5. Select which attribute or attributes PingFederate will use as a pseudonym. Click **Next**.
6. Configure additional attribute lookup on the **Adapter Contract Mapping** screen by clicking **Configure Adapter Contract**. Click **Next**.
7. Review the configured values on the **Summary** screen and click **Done**. Click **Save** on the following screen to save your configuration.
8. Navigate to **IdP Configuration > Adapters**.
9. Select the X509 Adapter.

Copyright © 2025 Ping Identity Corporation

10. Edit the Instance Configuration and select **Show Advanced Fields**.
11. Enable **Include Subject Alternative Names**.
12. Edit the **Extended Contract** screen.
13. Include the name of the subject alternative name key.
14. Save the configuration.

Testing the adapter

About this task

> **Note**
>
> You need a device with the MobileIron agent installed and configured with your MobileIron instance. An X.509 certificate will be provisioned to the device for authentication with the X.509 adapter. The CA root certificate must be imported into the PingFederate trusted store (see Trusted Certificate Authorities in the PingFederate *Administrator's Manual*).

Follow this procedure to test your adapter configuration:

Steps

1. Set up PingFederate to run the SP Application according to instructions in the *Quick-Start Guide*.
2. Navigate to your MobileIron instance, login and go to **Configurations > Add > Identity Certificate**.
3. Name the configuration and select Dynamically Generated from the **Certificate Distribution** dropdown. Select your Certificate Authority from the **Source** dropdown and enter the desired subject name for the **Subject** field (for example, `CN=${userCN}`).
4. Click **Add** under **Subject Alternative Name Type** and select a key and enter `${deviceMdmDeviceIdentifier}` as the value.

   > **Note**
   >
   > Version 1.3 of the X.509 adapter supports parsing URI, user principal name and RFC 822 name out-of-the-box without requiring OGNL to extract the values. Using a different key will require an OGNL script to extract the value.

5. Click **Test Configuration** and click **Continue**.
6. Select your inclusion/exclusion criteria for provisioning the certificate to the devices enrolled in MobileIron. Ping Identity recommends selecting **Custom** and selectively choosing your test device. Click **Done**.
7. On your test device, navigate to the MobileIron Go app and sync your device. It might take some time for the certificate to provision.

   > **Note**
   >
   > You may need to open the agent app and force a check-in between the device and the MobileIron instance.

Copyright © 2025 Ping Identity Corporation

8. After the profile is installed on the device, verify a certificate from your credential source is available on the device by navigating to **Devices > Select your device > Certificates**. Inspect the list of certificates installed on the device and verify a certificate from your configuration has been provisioned to the device.
9. Using the device with the installed configuration, open a browser on the device and navigate to a resource protected by the adapter.
10. When challenged for X.509 authentication, select the certificate installed by the profile. The browser will be redirected to PingFederate for X.509 validation and the device should be redirected to the protected resource.

> **Note**
>
> Depending on the device, the certificate isn't readily available in the browser for authentication. The certificate might require importing into the browser's certificate store before it can be used to authenticate a user.

Sample use cases

Here are some examples on how customers might use this kit:

- Allow access for devices running on the iOS platform.
- Deny access for devices that are not managed by MobileIron.
- Allow access for devices that are in compliance with company policy.
- Deny access for compromised or rooted devices.
- Configuration examples

Configuration examples

The following subsections are examples of using the adapter's issuance criteria to restrict authorizing users to accessing protected resources.

Restrict users based on device ownership

1. From the PingFederate home screen, navigate to your configured MobileIron adapter.
2. Select **Adapter Contract Mapping**, then **Configure Adapter Contract** to access the adapter's attribute mapping summary screen and Navigate to the **Issuance Criteria** subheader.
3. For **Source** select `adapter`
4. For **Attribute Name** select `Ownership`
5. For **Condition** select `not equal to`
6. MobileIron's device API returns one of three values for ownership:
    - `COMPANY` for Corporate owned devices
    - `EMPLOYEE` for Employee owned devices
    - `UNKNOWN`

Copyright © 2025 Ping Identity Corporation

Select which type device ownership complies with your business practices and click **Add**, **Done** twice, then **Save**.

Restrict users based on device operating system

1. From the PingFederate home screen, navigate to your configured MobileIron adapter.
2. Select **Adapter Contract Mapping** then **Configure Adapter Contract** to access the adapter's attribute mapping summary screen. Navigate to the **Issuance Criteria** subheader.
3. For **Source** select `adapter`
4. For **Attribute Name** select `os`
5. For **Condition** select `not equal to`
6. MobileIron's device API returns different values for device operating systems.
    - `IOS`
    - `ANDROID` Select which device operating systems complies with your business practices and click **Add**, **Done** twice, then **Save**.

Release notes

Changelog

MobileIron Integration Kit 1.0 - March 2018

- Initial release

Download manifest

The following files are included in the `.zip` archive:

- `Legal.pdf` – contains the licenses for the dependencies used in the project
- `dist/pingfederate/server/default` – contains the integration files
    - `pf-mobileiron-integration-kit-1.0.1.jar` – the MobileIron adapter JAR file

Copyright © 2025 Ping Identity Corporation

Overview of Office 365 integrations

PingFederate Integration with Azure Active Directory and Office 365

This guide provides detailed steps for integrating PingFederate with Microsoft Azure Active Directory and Microsoft Office 365. Learn more in Azure AD and Office 365 Integration Guide.

PingFederate Office 365 Connector

The PingFederate Office 365 Connector enables enterprises to provision users and groups to Office 365.

Office 365 Provisioner

The PingFederate Office 365 Provisioner enables enterprises to provision users and groups to Office 365.

The Office 365 Provisioner includes a quick connection template to easily set up a connection with Office 365, which can be used for single sign-on (SSO), provisioning, or both. The provisioner makes use of the Microsoft Graph API to communicate with Azure, which acts as the user and group repository for Office 365. The provisioner includes licensing support (`skuId` and `disabledPlans` attributes) and the ability for managers to be assigned to provisioned users (`manager` and `pingSourceDN` attributes).

> **Note**
>
> This provisioner is for outbound provisioning only and is not intended for inbound or hybrid environments.

Features

- Browser-based SP and IdP-initiated SSO
- Includes support for user and group life cycle management (including creates, updates, disables, and deletes).
- Includes configuration options for workflow capabilities (for example, the ability to disable updates).

Intended audience

This document is intended for PingFederate administrators.

If you need help during the setup process, see the following resources:

- The following Knowledge Base article:
  - PingFederate Office 365 FAQ
- The following sections of the PingFederate documentation:
  - Identity provider SSO configuration
  - SP connection management
  - Managing digital signing certificates and decryption keys
  - Datastores

Copyright © 2025 Ping Identity Corporation

- Configuring outbound provisioning
- Configuring outbound provisioning settings

System requirements

- PingFederate 9.0 or later.
- An existing Office 365 account.
- SSO requires the following:
  - A domain that has been created for use as a federated domain and is accessible and DNS resolvable by Microsoft.
  - Administrative access to modify DNS records for the federated domain.
  - The PingFederate server must be externally accessible.
  - A Windows platform in order to run SSO related configuration using Powershell. The Windows platform must be able to access the Azure management portal.

User and group management

The Office 365 Provisioner synchronizes users and groups from your datastore to Office 365. The following describes the behavior of each provisioning capability.

> **Tip**
>
> You can configure the following capabilities and specify which users to provision when you get to the Configure outbound provisioning part of the setup process.

Synchronizing existing users

PingFederate synchronizes users based on the `userPrincipalName` attribute in Office 365. If a user already exists in your datastore and Office 365, mapping this attribute correctly links the two records together.

For example:

- In Office 365, Janet's `userPrincipalName` is `bjensen@example.com`.
- In your datastore, Janet's `mail` is `bjensen@example.com`.
- On the **Attribute Mapping** tab of your provisioning connection configuration, map the `userPrincipalName` attribute to `mail`.
- When the provisioning connector runs, the datastore user is provisioned with a `userPrincipalName` of `bjensen@example.com`. That matches Janet's existing `userPrincipalName` in Office 365, so her information in the datastore is synchronized to her Office 365 account.

User provisioning

PingFederate provisions users when any of the following happens:

- A user is added to the datastore group or filter that is targeted by the provisioning connector.

Copyright © 2025 Ping Identity Corporation

- A user with `disabled` status is added to the datastore group or filter that is targeted by the provisioning connector, and the **Provision disabled users** provisioning option is enabled. This feature is not available in all provisioning connector versions.

You can define which users PingFederate targets for provisioning on the **Source Location** tab of your provisioning connection configuration.

User updates

PingFederate updates users when a user attribute changes in your datastore.

You can define which attributes PingFederate monitors for changes on the **Attribute Mapping** tab of your provisioning connection configuration.

User deprovisioning

PingFederate deprovisions users when any of the following happens:

- A user is deleted from the user store.
- A user is disabled in the user store.
- A user is removed from the datastore group or filter that is targeted by the provisioning connector.

The **Remove User Action** setting in the connection configuration determines whether the deprovisioning action disables or deletes the user.

Synchronizing existing groups

PingFederate synchronizes groups from the datastore to the target service based on the group name.

For example:

- In Office 365, there is a group is named `Accounting`.
- In your datastore, there is a group with a `CN` of `Accounting`.
- When the provisioning connector runs, the two groups are synchronized.

Group provisioning

PingFederate provisions groups when a group is added to the datastore filter that is targeted by the provisioning connector.

You can define which groups PingFederate targets for provisioning and monitors for changes on the **Source Location** tab in your provisioning connection configuration.

Group name updates

PingFederate renames groups when they are renamed in the datastore.

Group membership updates

PingFederate updates group memberships when memberships change in the datastore, whether the change is in the group's properties or a user's properties.

Group memberships in the datastore overwrite the group memberships in Office 365.

Copyright © 2025 Ping Identity Corporation

Group deletion

PingFederate deletes groups when any of the following happens:

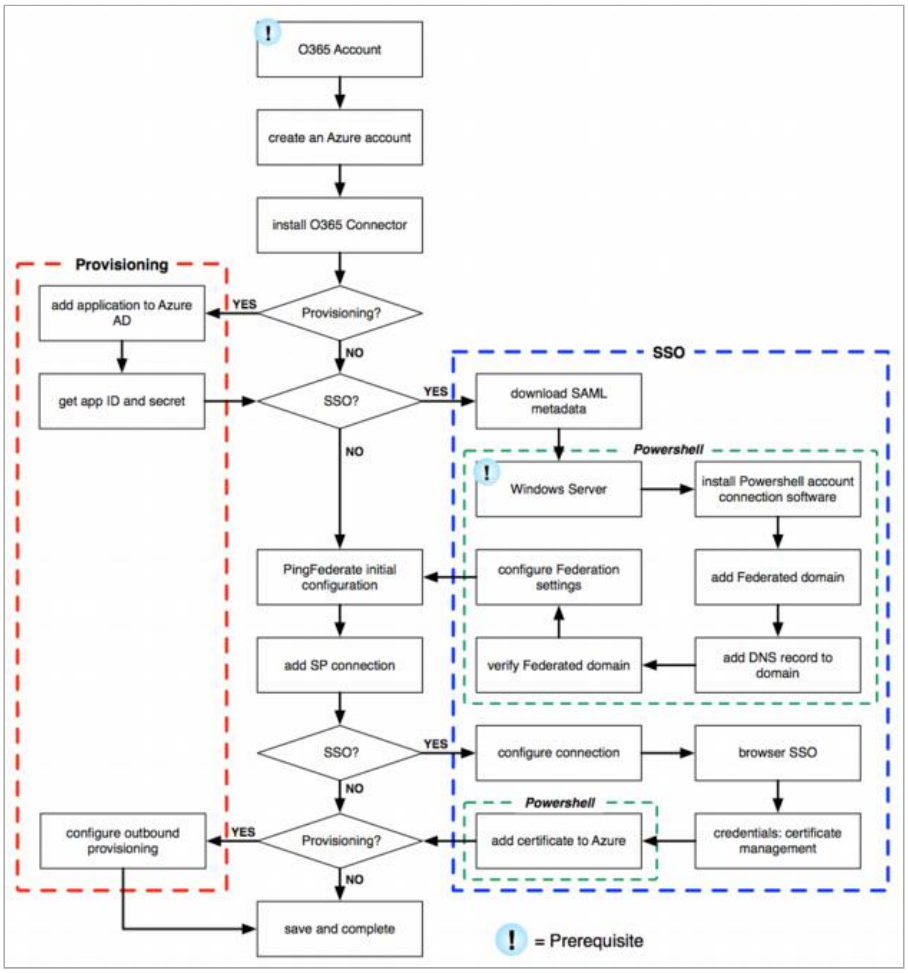
- The group is deleted in the datastore.
- The group is removed from the datastore group or filter that is targeted by the provisioning connector.

> **Important**
>
> Group deletions are permanent and cannot be undone.

Setup

Setup overview diagram

The following diagram outlines the high level steps required for installing and configuring the Office 365 Connector for SSO, provisioning, or both.



Copyright © 2025 Ping Identity Corporation

Getting started

Before you can configure this connector, you will need to complete the following steps.

- Choose an SSO configuration path
- Create an Azure account
- Download Office 365 SAML 2.0 metadata file

Choose an SSO configuration path

After installing the connector, use the table below as a reference to determine how to configure your Office 365 SSO deployment with PingFederate.

> **Tip**
> If you are upgrading from a previous version of the Office 365 Connector, see Upgrading an existing deployment.

If you want:	Then:
SSO only to Office 365	To implement this solution, follow instructions in Configure an SSO connection. Note that some instructions include WS-Federation settings to ease transition to WS-Federation at a later time if desired.
SSO and SLO to Office 365	To implement this solution, follow instructions in Preparing Active Directory for federation and Install and configure PingFederate in the Azure AD and Office 365 Integration Guide. Skip the steps marked with “SAML SSO” in this document. **Important** Although WS-Trust SLO is supported, SAML SLO is currently not supported due to a compatibility issue in the SAML implementation between PingFederate and Azure.
Active federation	For active federation, refer to the guidelines and instructions in Creating a connection to Azure Active Directory in the Azure AD and Office 365 Integration Guide.

Create an Azure account

About this task

The following steps outline how to create an Azure account for this integration. You will create the Azure account with the same credentials as those used with the Office 365 tenant to provide access to Azure Active Directory. This step is required for both provisioning and SSO.

Copyright © 2025 Ping Identity Corporation

Steps

1. Navigate to the Microsoft Azure Portal.
2. Create an account using the same credentials as the Office 365 account.

Download Office 365 SAML 2.0 metadata file

About this task

This connector's quick-connection template uses a metadata `.xml` file from Office 365 to assist in configuring many settings in the SP Connection. When asked during the SP Connection configuration steps, import the `federationmetadata.xml` that you downloaded from Office 365.

Steps

1. Download the SAML 2.0 metadata for Office 365 from Microsoft.
2. Save the `.xml` file to a desired location.

Configuring PingFederate to omit line breaks in digital signatures

Configure PingFederate to omit line breaks in digital signatures by performing the following steps.

Steps

1. Open `<pf_install>/pingfederate/bin/run.properties` in a text editor.
2. If not already present, add a new line to the file as follows: `org.apache.xml.security.ignoreLineBreaks=true`
3. Save the file.
4. Stop PingFederate and restart it to pick up the change.

Upgrading an existing deployment

If you're upgrading from a previous version of the Office 365 Provisioner, note your existing service provider (SP) connection configuration and create a new connection.

Copyright © 2025 Ping Identity Corporation

PingFederate 10.1 or later

Upgrading an existing deployment in PingFederate 10.1 or later
Steps

1. Back up your current PingFederate configuration as shown in Configuration archive in the PingFederate documentation.
2. In the PingFederate administrative console, go to **Applications > Integration > SP Connections** and select your connection.
3. Note the attribute mappings for your existing SP connection. For help, see Mapping attributes in the PingFederate documentation.
4. Delete your existing SP connection.
    1. Go to **Applications > Integration > SP Connections**.
    2. For your existing connection, click **Select action**, and then click **Delete**. Click **Confirm**.
5. Complete the steps in Deploying the integration files.
6. Complete the steps in Configure outbound provisioning.
    - From **Outbound Provisioning > Manage Channels > Channel**, on the **Attribute Mapping** tab, configure the attribute mappings based on your notes.
7. Complete the steps in Configure an SSO connection.

Copyright © 2025 Ping Identity Corporation

PingFederate 10.0 or earlier

Upgrading an existing deployment in PingFederate 10.0 or earlier
Steps

1. Back up your current PingFederate configuration as shown in Configuration archive in the PingFederate documentation.
2. In the PingFederate administrative console, go to **Identity Provider > SP Connections** and select your connection.
3. Note the attribute mappings for your existing SP connection. For help, see Mapping attributes in the PingFederate documentation.
4. Delete your existing SP connection.
    1. Go to **Identity Provider > SP Connections > Manage All**.
    2. For your existing connection, click **Select action**, and then click **Delete**. Click **Save**.
5. Complete the steps in Deploying the integration files.
6. Complete the steps in Configure outbound provisioning.
    - From **Outbound Provisioning > Manage Channels > Channel**, on the **Attribute Mapping** tab, configure the attribute mappings based on your notes.
7. Complete the steps in Configure an SSO connection.

Deploying the integration files

To get started with the integration, deploy the Office 365 Provisioner files to your PingFederate directory.

Steps

1. Download the Office 365 Provisioner `.zip` archive from the .pingidentity.com/s/marketplace-integration-home-page/[Ping Identity Integration Directory].
2. Stop PingFederate.
3. If you're upgrading an existing deployment, delete the `pf-office365-quickconnection-<version>.jar` file from your `<pf_install>/pingfederate/server/default/deploy` directory.
4. Extract the `.zip` archive and merge the contents of the `dist` directory with your `<pf_install>/pingfederate/server/default/deploy` directory.
5. Enable the PingFederate provisioning engine:
    1. Open your `<pf_install>/pingfederate/bin/run.properties` file.
    2. Change `pf.provisioner.mode` to `STANDALONE`. Save the file.

Copyright © 2025 Ping Identity Corporation

> **Note**
>
> To configure the `FAILOVER` mode instead, see Deploying provisioning failover in the PingFederate Server Clustering Guide.

6. Start PingFederate.
7. If you operate PingFederate in a cluster, repeat steps 2-4 and step 6 for each engine node.

Enabling provisioning and single sign-on in PingFederate

To use PingFederate for provisioning and single sign-on, configure an external datastore and set a SAML entity ID.

About this task

Your external datastore acts as the source of data for provisioning. PingFederate also uses an internal datastore to store the state of synchronization between the source datastore and the target datastore.

For more information, see Datastores and Configuring outbound provisioning settings in the PingFederate documentation.

PingFederate 10.1 or later

Enabling provisioning and single sign-on in PingFederate 10.1 or later
Steps

1. In the PingFederate administrative console, configure the datastore for PingFederate to use as the source of user data.

   For help, see Datastores in the PingFederate documentation.
2. Go to **System > Server > Protocol Settings > Federation Info**.
3. In the **SAML 2.0 Entity ID** field, enter a name for PingFederate to use when identifying itself to SAML partners.
4. On the **Outbound Provisioning** tab, in the **Provisioner Data Store** list, select the internal database that will store the synchronization state. Click **Save**.

   For more information, see Configuring outbound provisioning settings in the PingFederate documentation.

Copyright © 2025 Ping Identity Corporation

PingFederate 10.0 or earlier

Enabling provisioning and single sign-on in PingFederate 10.0 or earlier

Steps

1. In the PingFederate administrative console, configure the datastore for PingFederate to use as the source of user data.

   For help, see Datastores in the PingFederate documentation.

2. Go to **System > Protocol Settings > Roles & Protocols**.
3. Select the **Enable Identity Provider IdP Role and Support the Following** check box.
4. Select the **SAML 2.0** and **Outbound Provisioning** check boxes. Click **Next**.
5. Click the **Federation Info** tab.
6. In the **SAML 2.0 Entity ID** field, enter a name for PingFederate to use to identify itself to SAML partners.
7. On the **Outbound Provisioning** tab, in the **Provisioner Data Store** list, select the internal database that will store the synchronization state. Click **Save**.

   For more information, see Configuring outbound provisioning settings in the PingFederate documentation.

Configuring provisioning

To configure a connection for outbound provisioning to Office 365, complete the instructions the following sections:

1. Add application to Azure AD
2. Configure outbound provisioning
3. Configure license management
4. Configure manager assignment

Add application to Azure AD

Add an application to Azure Active Directory to create and expose Microsoft Graph API endpoints for provisioning.

Steps

1. Complete the steps in Register an application with the Microsoft identity platform in the Microsoft identity platform documentation.
2. Note your Azure application ID and secret.

Copyright © 2025 Ping Identity Corporation

3. To allow the provisioner to manage all users, including deleting users or modifying administrators, assign the "User administrator" role to your Azure AD application.
    1. Complete the steps in "Authorization_RequestDenied" error message when you try to change a password if you use Graph API in the Microsoft documentation.
4. Add the following application permissions to your application by completing the steps in .microsoft.com/en-us/azure/active-directory/develop/quickstart-configure-app-access-web-apis//[Add permissions to access web APIs]:
    - `Application.ReadWrite.All`
    - `Group.ReadWrite.All`
    - `Organization.Read.All`
    - `User.ReadWrite.All`

Configure outbound provisioning

About this task

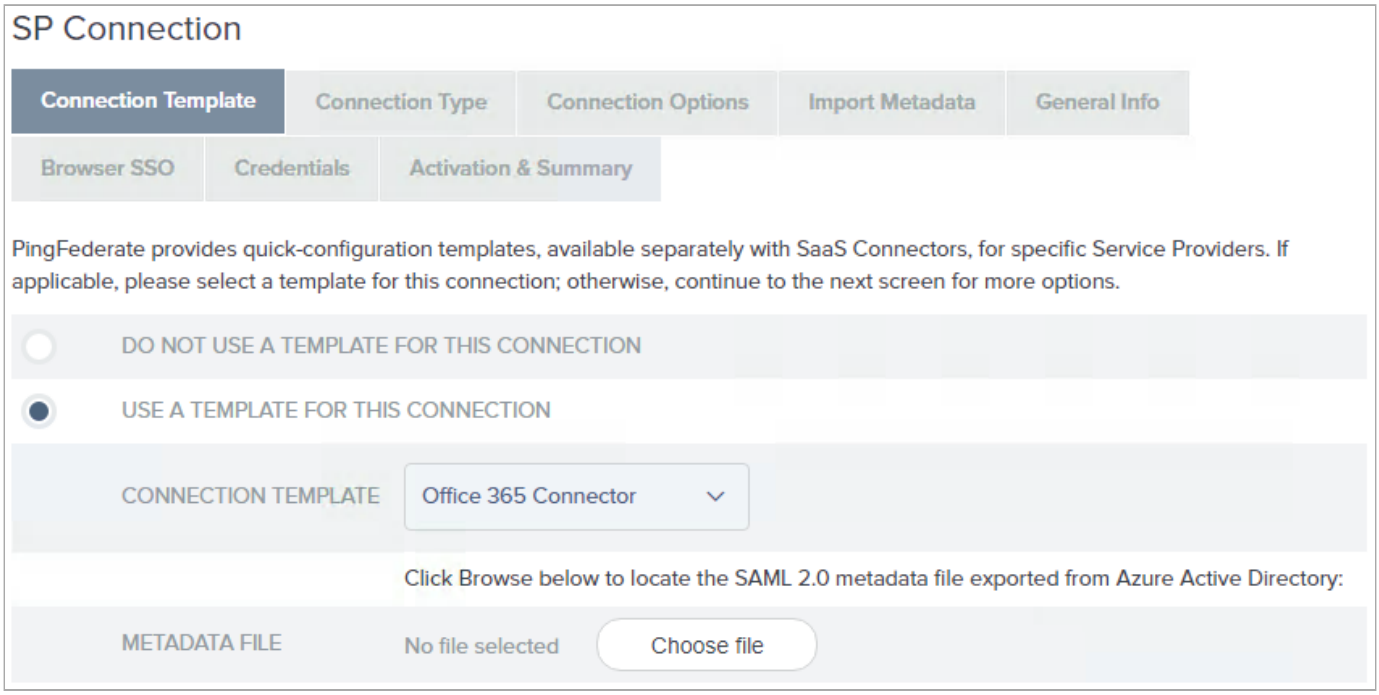
Outbound provisioning details are managed within an SP connection. You can configure outbound provisioning with or without Browser SSO, WS-Trust STS, or both when you create a new SP connection. You also have the option to add outbound provisioning to an existing SP connection.

For SSO instructions, see Configure SSO.

Steps

1. Create a new SP connection or select an existing SP connection from the **SP Configuration** menu.
2. On the **Connection Template** screen, select the **Use a template for this connection** option and choose **Office 365 Connector** from the **Connection Template** drop-down list. You will be asked to provide the `federationmetadata.xml` file you obtained earlier in Download Office 365 SAML 2.0 metadata file.

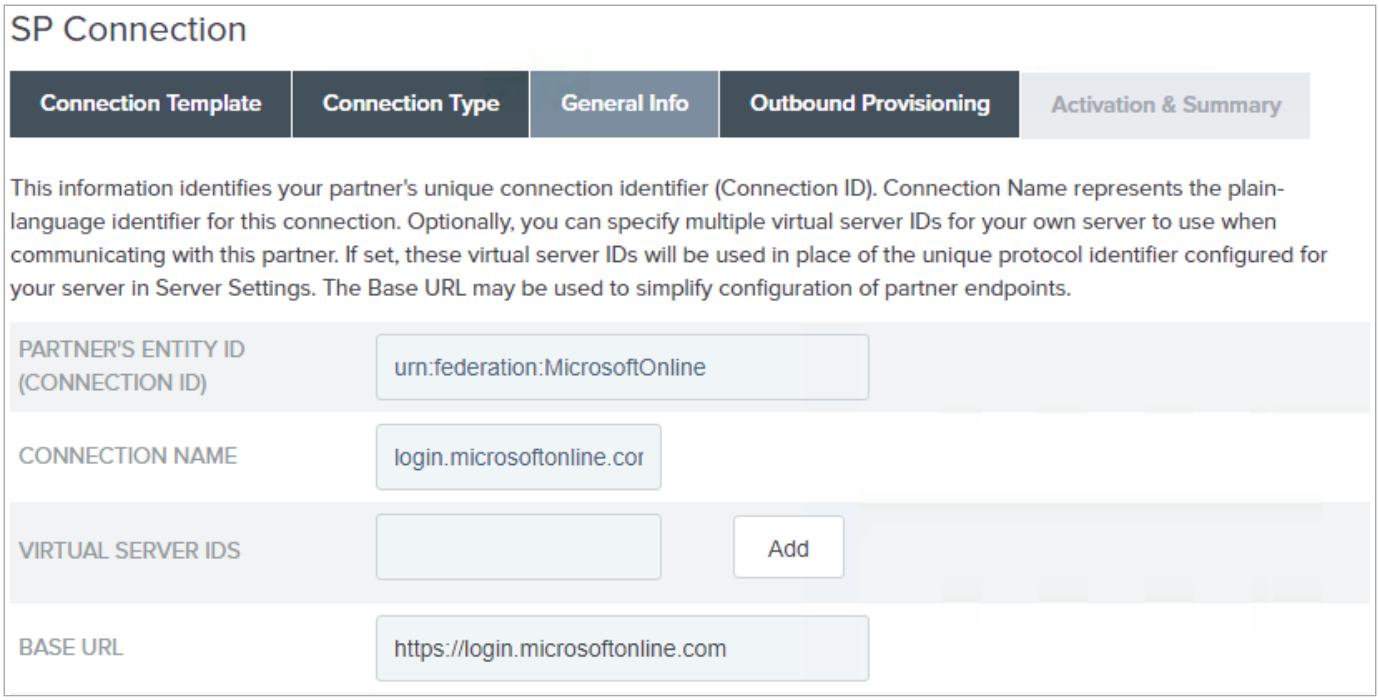
Copyright © 2025 Ping Identity Corporation



> **Tip**
>
> If this selection is not available, verify the connector installation and restart PingFederate.

3. On the **Connection Type** screen, ensure the **Outbound Provisioning** checkbox is selected, and the **Browser SSO Profiles** checkbox is cleared (if appropriate).

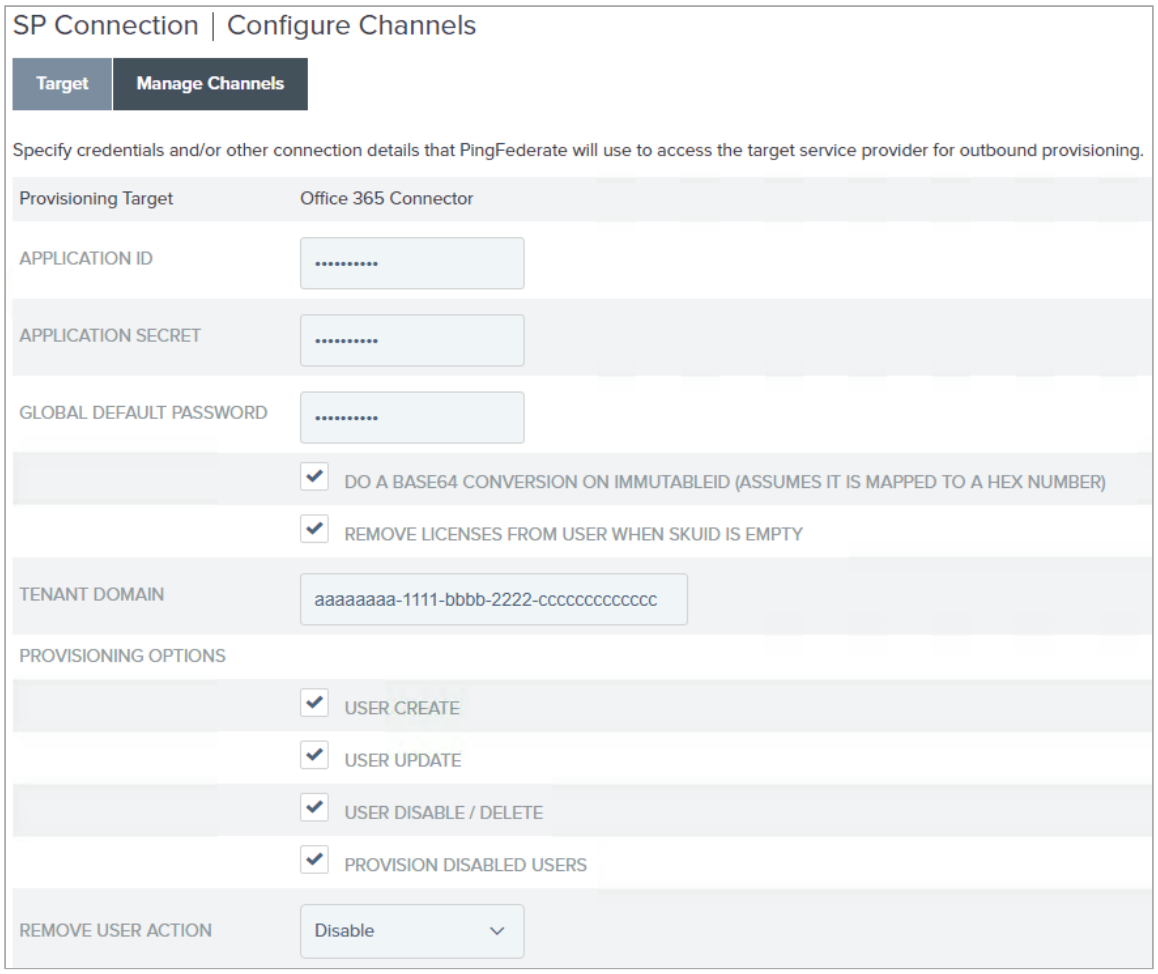
4. On the **General Info** screen, the default values are taken from the metadata file you selected in an earlier step. We recommend using the metadata default values.



5. Follow the connection wizard to configure the connection.

Copyright © 2025 Ping Identity Corporation

6. On the **Outbound Provisioning** screen, click **Configure Provisioning**.
7. On the **Target** screen, enter the values for each field as required by the Office 365 Connector.



Field Name	Value
Application ID	The application ID for the application created in Azure. For more information, see Add application to Azure AD.
Application Secret	The secret generated during application creation in Azure. For more information, see Add application to Azure AD.
Global Default Password	The default password. Only used if the password attribute is not mapped, or value of the mapped field is empty.
Do a Base64 Conversion on ImmutableID	**True** (default) is recommended. Set to false if the ImmutableID is not base64. The conversion assumes it is mapped to a hex number.

Copyright © 2025 Ping Identity Corporation

Field Name	Value
Remove Licenses from User when SkuId is Empty	**False** (default) – When disabled, if you choose to not configure the **skuId** field in your configuration's **Attribute Mapping** screen, or if the user's **skuId** field is cleared in the datastore, the user's licenses will not be removed from their account. **True** – When enabled, if you choose to not configure the **skuId** field in your configuration's **Attribute Mapping** screen, or if the user's **skuId** field is cleared in the datastore, the user's licenses will be removed from their account.
Tenant Domain	The tenant domain configured in Azure, which is retrieved by going to the application properties and selecting view endpoints, and copying the ID from the URL under **Windows Azure AD Graph API Endpoint**.
Provisioning Options	
User Create	**True**(default) – Users will be created in Office 365. **False** – Users will not be created in Office 365. **Note** The `provisioner.log` will display a warning within the create user workflow that the user was not created in Office 365.
User Update	**True** (default) – Users will be updated in Office 365. **False** – Users will not be updated in Office 365. **Note** The `provisioner.log` will display a warning within the update user workflow that the user was not updated in Office 365.
User Disable / Delete	[**True** (default) – Users will be disabled or deleted in Office 365. **False** – Users will not be disabled or deleted in Office 365. **Note** The `provisioner.log` will display a warning indicating that the user was not disabled or deleted in Office 365.
Provision Disabled Users	This option is only relevant if **User Create** is True. **True** (default) – Office 365 users will be created in a disabled state. **False** – Office 365 users will not be created in a disabled state. This is desirable for scenarios where there are disabled users in the data store, not intended for creation in Office 365 during initial synchronization. **Note** The `provisioner.log` will display a warning within the create user workflow indicating that the user was not created in Office 365.

Copyright © 2025 Ping Identity Corporation

Field Name	Value
Remove User Action	Select a deprovision method (**Disable** or **Delete**). Deprovisioning is triggered when previously provisioned users no longer meet the condition set in the **Source Location** screen, or when a user has been suspended or deleted from the data store. This option is only applicable if **User Disable / Delete** is set to True. **Disable** (default) – when selected, if you delete a user from Active Directory, the user will be disabled in Office 365 (also known as a soft delete). **Delete** – when selected, if you delete a user from Active Directory, the user will be deleted in Office 365 (also known as a hard delete). **Note** When a user is deleted in Azure Active Directory, the deleted user is retained for 30 days from the deletion date. During that time, the user and its properties can be restored under **Users > Deleted users**.

Important

For user provisioning to succeed, the users' userPrincipalName domain must match a verified domain in Azure.

8. Click **Next** to continue the provisioning configuration.

   For more information, see the following sections under Configuring outbound provisioning:

   - Manage channels
   - Specifying channel information
   - Identifying the source datastore
   - Modify source settings
   - Specify a source location
   - Map attributes
   - Review channel settings

   Note

   Credentials will be verified when the channel and SP connection is set to Active and provisioning is initiated.

   Tip

   If you are not ready to complete the provisioning configuration, you can click **Save** and return to the configuration page later. To return to the configuration page, select the connection from **Identity Provider > SP Connections > Manage All**.

Copyright © 2025 Ping Identity Corporation

Supported attributes reference

The following table consists of the attributes that can be mapped on a user during provisioning.

`userPrincipalName`	The user principal name (UPN) of the user. The UPN is an Internet-style login name for the user based on the Internet standard `RFC822`. **Note** This must match the domain configured in the Azure environment.
`displayName`	The name displayed in the address book for the user. This property is required when a user is created and it cannot be cleared during updates.
`mailNickname`	The mail alias for the user. This property must be specified when a user is created.
`city`	The city in which the user is located.
`country`	The country/region in which the user is located; for example, "US" or "UK".
`department`	The name for the department in which the user works.
`facsimileTelephoneNumber`	The telephone number of the user's business fax machine.
`givenname`	The given name (first name) of the user.
`jobTitle`	The user's job title.
`mobile`	The primary cellular telephone number for the user. **Note** To update a user's mobile number, the Office 365 Connector requires elevated permissions. To elevate these permissions please see O365 Connector: Mobile attribute updates. If you do not wish to elevate these permissions, please make the mobile attribute Create-Only when configuring the connection's attribute mappings.
`physicalDeliveryOfficeName`	The office location in the user's place of business.
`postalCode`	The postal code for the user's postal address. The postal code is specific to the user's country/region. In the United States of America, this attribute contains the ZIP code.
`perferredLanguage`	The preferred language for the user. Should follow `ISO 639-1` Code; for example "en-US".
`state`	The state or province in the user's address.

Copyright © 2025 Ping Identity Corporation

`streetAddress`	The street address of the user's place of business.
`surname`	The user's surname (family name or last name).
`telephoneNumber`	The primary telephone number of the user's place of business.
`usageLocation`	Required for the licensing feature. Needs to be mapped to an attribute that contains the `ISO-3166` formatted country (a two letter country code) of license usage. Required for users that will be assigned licenses due to legal requirement to check for availability of services in countries. Examples include: "US", "JP", and "GB".
`userType`	A string value that can be used to classify user types in your directory, such as "Member" and "Guest".
`password`	Map password to a field so the content will become the user's initial password instead of the less secure default. The field can also be set to a static default value. This field is required when a user is created. It can not be updated, but the user can be forced to update their password on their next login by setting their **resetPassword** field to true. The password must satisfy minimum requirements as specified by the user's `passwordPolicies` property. By default, a strong password is required.
`resetPassword`	Determines if a user needs to do a password reset the next time they login. Default value is true, but can be mapped to an attribute.
`manager`	Required for the manager feature. Sets the user DN of the associated manager.
`pingSourceDn`	Required for the manager feature. A custom field that we set on a user in Azure, which holds the user's DN from AD and is used to lookup users in Azure in order to set the manager field on a user in Azure. Sets the user DN. Users and managers must be created or updated with the `pingSourceDN` information for the manager association to succeed.
`skuId`	Required for the licensing feature. Can be mapped in PingFederate to a single or multi-valued attribute in LDAP. Used for the IDs or names of the license(s) assigned to users. The **usageLocation** field must also be set for a license to be successfully assigned.
`disabledPlans`	Part of the licensing feature. Can be mapped in PingFederate to a single or multi-valued attribute in LDAP. Used for the IDs or names of disabled plans for individual users' licenses.
`immutableId`	This property is used to associate an on-premises Active Directory user account to their Azure AD user object. This property must be specified when creating a new user account in the Graph if you are using a federated domain for the user's `userPrincipalName` (UPN) property. This field can not be updated by the Office 365 Connector.

Copyright © 2025 Ping Identity Corporation

`otherMails`	A list of additional email addresses for the user.

Configure license management

About this task

The Office 365 Connector supports the ability to manage the Office 365 licenses assigned to a user.

Steps

1. The **usageLocation** field must be set to a static value, or mapped to an attribute containing the `ISO-3166` two-character country code for the location of the user.
2. The **skuId** field must either be set to a single static value or an attribute containing one or more license keys to be assigned to the user.

   > **Note**
   >
   > Each license specified in the **skuId** field can be either the actual ID of that license or the specified name of the license.

3. The **disabledPlans** field may either be set to a single static value or an attribute containing product keys to be disabled for the user. The **disabledPlans** field cannot contain a list of product keys. To provide a list of product keys, set the **disabledPlans** field to a multi-valued LDAP attribute.

   > **Note**
   >
   > If no **disabledPlans** are specified, the user will have access to all products available through their assigned licenses specified in their **skuId** field.

   > **Note**
   >
   > Each product specified in the **disabledPlans** field can be either the actual ID of that product or the name of the product.

4. Ensure the appropriate option for the **Remove Licenses from User when skuID is Empty** connection field is configured on the SP Connection configured for the Office 365 Connector:
   - **False** (default) - When disabled, if you choose to not configure the **skuId** field in your configuration's **Attribute Mapping** screen, or if the user's **skuId** field is cleared in the datastore, the user's licenses will not be removed from their account.
   - **True** - When enabled, if you choose to not configure the **skuId** field in your configuration's **Attribute Mapping** screen, or if the user's **skuId** field is cleared in the datastore, the user's licenses will be removed from their account.

Configure manager assignment

About this task

The Office 365 Connector supports the ability to assign a manager to a user.

Copyright © 2025 Ping Identity Corporation

Steps

1. The **pingSourceDn** field must be mapped to an attribute containing a unique identifier for the user.

   > **Note**
   >
   > By default this is mapped to the **distinguishedName** field. We recommend leaving this field mapped to the default mapping.

2. The **manager** field must either be set to a static value, or mapped to an attribute containing the value of the assigned manager's **pingSourceDn** field.

   > **Note**
   >
   > By default this is mapped to the **manager** field. We recommend leaving this field mapped to the default mapping.

ExampleAn example of how assigning a manager to a user works

> **Note**
>
> The following assumes the default mappings for the **pingSourceDn** and **manager** fields.

1. The manager is provisioned to Azure.
2. The employee is provisioned to Azure.
3. The employee's manager, under their **Organization** tab in Active Directory (AD), is set to the manager in AD.
4. The Office 365 Connector will assign the manager as the employee's manager in Azure.

   > **Tip**
   >
   > To update or clear the employee's manager in Azure, change or clear the employee's manager under their **Organization** tab in AD.

Configuring single sign-on

The following section describes the steps for configuring single sign-on (SSO) to Office 365.

> **Note**
>
> Configuring SSO is optional for outbound provisioning.

> **Important**
>
> A SAML connection is limited in functionality for this integration. For more information, see Choose an SSO configuration path. If SSO, SLO, and/or active federation are required, see Azure AD and Office 365 Integration Guide for configuration steps.

Copyright © 2025 Ping Identity Corporation

Install Powershell account connection software

About this task

Some SSO configuration settings can only be updated through the execution of Powershell commands. Connection to Azure from Windows is done through installation of Microsoft software, allowing the Powershell commands executed on the server to modify the SSO configuration in Azure. Follow the links below for instructions on how to install the required software.

Steps

1. Install **Microsoft Online Services Sign-In Assistant** using instructions from Microsoft.
1. Install the Azure AD Module.

Add federated domain

Steps

1. On the Windows machine with the account connection software, run the **Windows Azure Active Directory Module for Windows Powershell** application.

   > **Tip**
   >
   > Any instructions beginning with “PS>” indicate that the command to the right of the mentioned text is to be executed at the Powershell command line in the **Windows Azure Active Directory Module for Windows Powershell** command prompt.

2. Connect to Azure Active Directory:

   ```
   PS> Connect-MsolService
   ```

3. Enter the username and password for the Azure account with administrative privileges.

   > **Tip**
   >
   > Text surrounded by “<” and “>” is intended to be replaced with a substitution indicated by the text between the symbols. For example, if the name of your federated domain is `myfederateddomain.com`, then `<federated_domain_name>` should be replaced by `myfederateddomain.com` before execution of the command.

4. Add the federated domain:

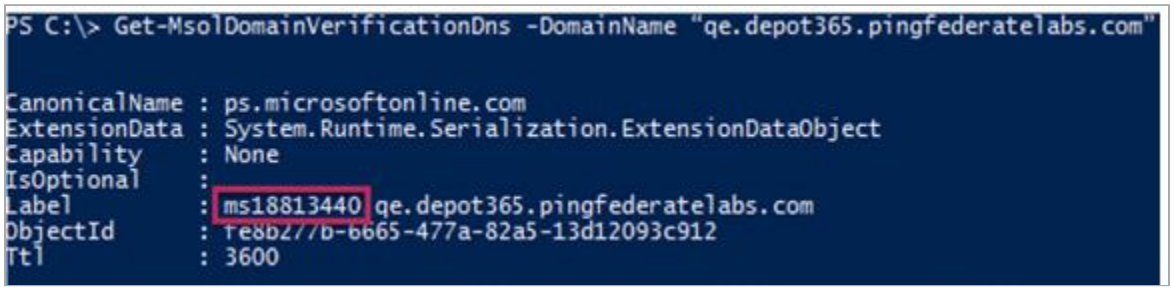
   ```
   PS> New-MsolDomain -name “<federated_domain_name>” -Authentication Federated
   ```

5. Get the domain label prefix value to aid in domain verification:

   ```
   PS> Get-MsolDomainVerificationDns -DomainName “<federated_domain_name>”
   ```

   Note the prefix of the value in the Label field and save for later use. (for example, the prefix will be in the format `ms##`)

Copyright © 2025 Ping Identity Corporation



> **Tip**
>
> To preserve some of the setup completed in the previous steps, leave the **Windows Azure Active Directory Module for Windows Powershell**command prompt window open until configuration of the Office 365 Connector has been completed.

DNS updates

Steps

1. Add a DNS redirect from a sub-domain of the federated domain to point to the PingFederate server so that `<sub-domain>.<federated_domain_name>` points to `<PingFederate_domain_name_or_IP>` where `<sub-domain>` is a unique identifier for the PingFederate server.

   For example, redirect `pf.myfederateddomain.com` to `pfnode.mycompany.com` where `pfnode.mycompany.com` resolves to the PingFederate server.

2. To assist in verification of the domain, add a TXT record to the DNS settings of the federated domain. Insert the domain label prefix recorded recorded in the Add federated domain step where indicated below.

   ```
   type: TXT, alias/host name: @, destination/points to address: MS=<domain_label_prefix>, ttl: 1hour
   ```

Verify federated domain

Confirm ownership of the federated domain.

For more information on the verification command, see the online documentation Confirm-MsolDomain.

Execute the Powershell commands below in the command prompt window used in Add federated domain.

Copyright © 2025 Ping Identity Corporation

```
PS> $domainName = “<federated_domain_name>”
  PS> $hostName = “<sub-domain>.$domainName”
  PS> $port = 9031
  PS> ${pingfed} = “https://$\{hostName}:$port
  PS> $brandName = “<federated_domain_alias>”
  PS> $issuer = “<PingFederate_SAML_2_entity_id>”
  PS> $spId = “urn:federation:MicrosoftOnline”
  PS> $activeLogOn = “$pingfederate/idp/sts.wst”
  PS> $logOff = “$pingfederate/idp/startSLO.ping”
  PS> $metaData = “$pingfederate/pf/sts_mex.ping?PartnerSpId=$spId”
  PS> $passiveLogOnPF=”$pingfederate/idp/SSO.saml2”
  PS> Confirm-MsolDomain -DomainName “$domainName”
     -FederationBrandName  “$brandName” -IssuerUri “$issuer”
     -LogOffUri $logOff  -PassiveLogOnUri  “$passiveLogOnPF”
```

Configure federation settings

Set the federation type to SAML.

```
PS> Set-MsolDomainAuthentication -DomainName “$domainName”
            -Authentication Federated -PreferredAuthenticationProtocol Samlp
```

Obtain PingFederate signing certificate

About this task

Below are steps to obtain the signing certificate that is used for the digital signature verification of the PingFederate IdP. When asked during Add the signing certificate to Azure, import the certificate that you have downloaded from PingFederate. For more information on managing digital signing certificates and decryption keys, see the PingFederate documentation.

To export an existing certificate:

Steps

1. Log into the **PingFederate Administrative Console** as an administrative user for your organization.
2. Go to **Server Configuration > Certificate Management > Signing & Decryption Keys & Certificates**.
3. Click **Export** under **Action** for the certificate you want to export.
4. On the **Export Certificate** screen, select **Certificate Only** export and then click **Next**.
5. On the **Export & Summary** screen, click **Export** to save the certificate file on your system.
6. When finished, click **Done**.
7. Click **Cancel** to go back to the **Server Configuration** screen.

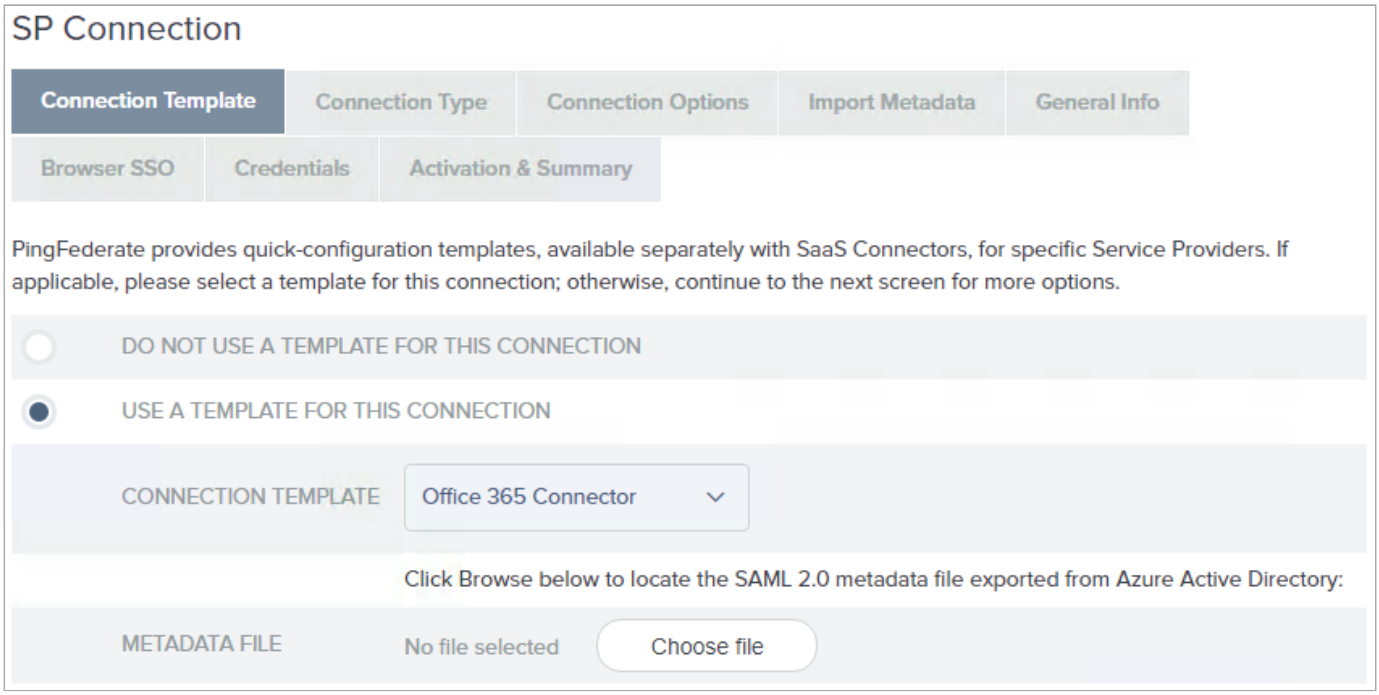
Copyright © 2025 Ping Identity Corporation

Configure an SSO connection

Steps

1. Create a new SP connection or select an existing SP connection from the **SP Configuration** menu.
2. On the **Connection Template** screen, select **Use a template for this connection** and choose **Office 365** from the **Connection Template** drop-down list.

   You will be asked to provide the `federationmetadata.xml` file you obtained earlier in Download Office 365 SAML 2.0 metadata file.

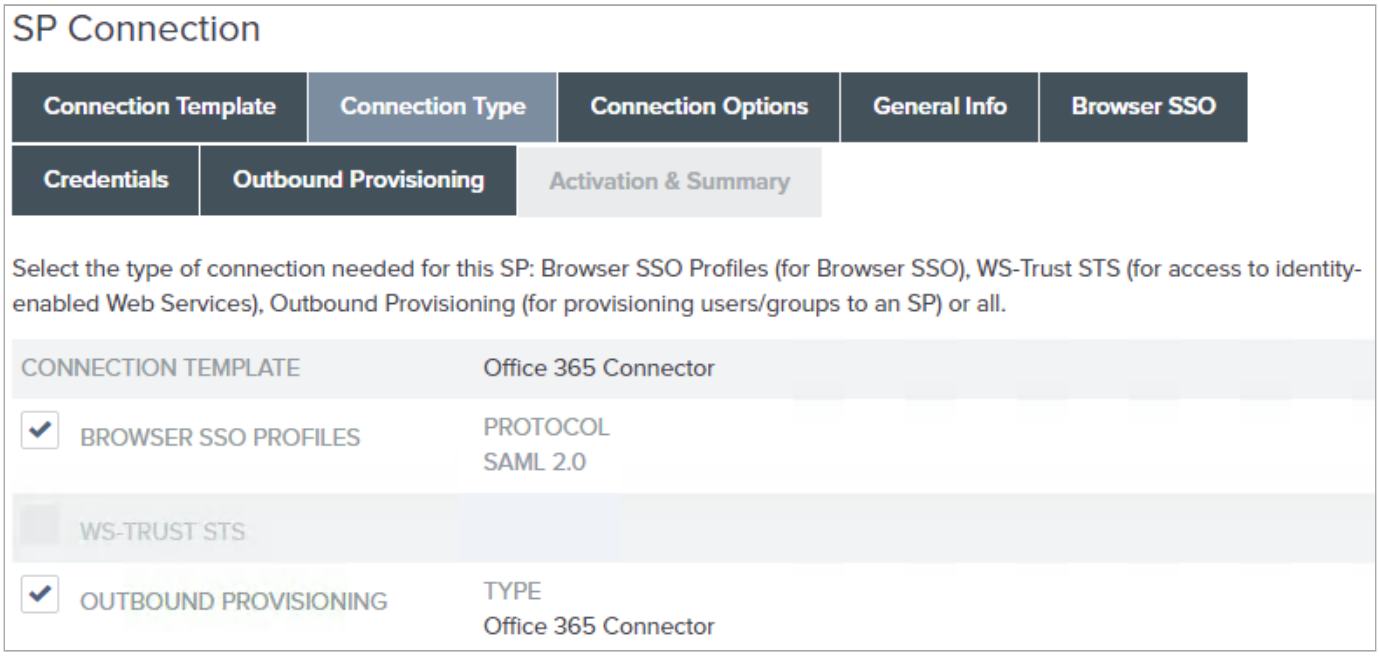


> **Tip**
>
> If this selection is not available, verify the connector installation and restart PingFederate.

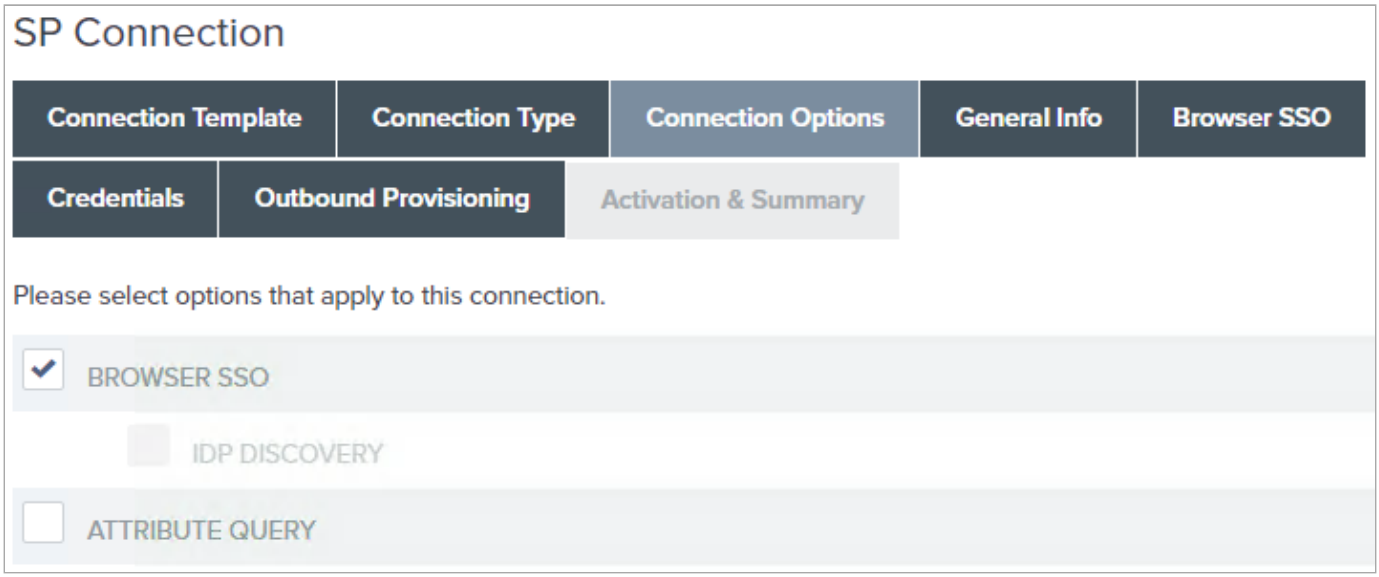
3. On the **Connection Type** screen, ensure that the **Browser SSO Profiles** checkbox is selected and click **Next**.

> **Note**
>
> If outbound provisioning will also be used, select **Outbound Provisioning** profile as well. The screenshot below shows an example where both are selected.

Copyright © 2025 Ping Identity Corporation



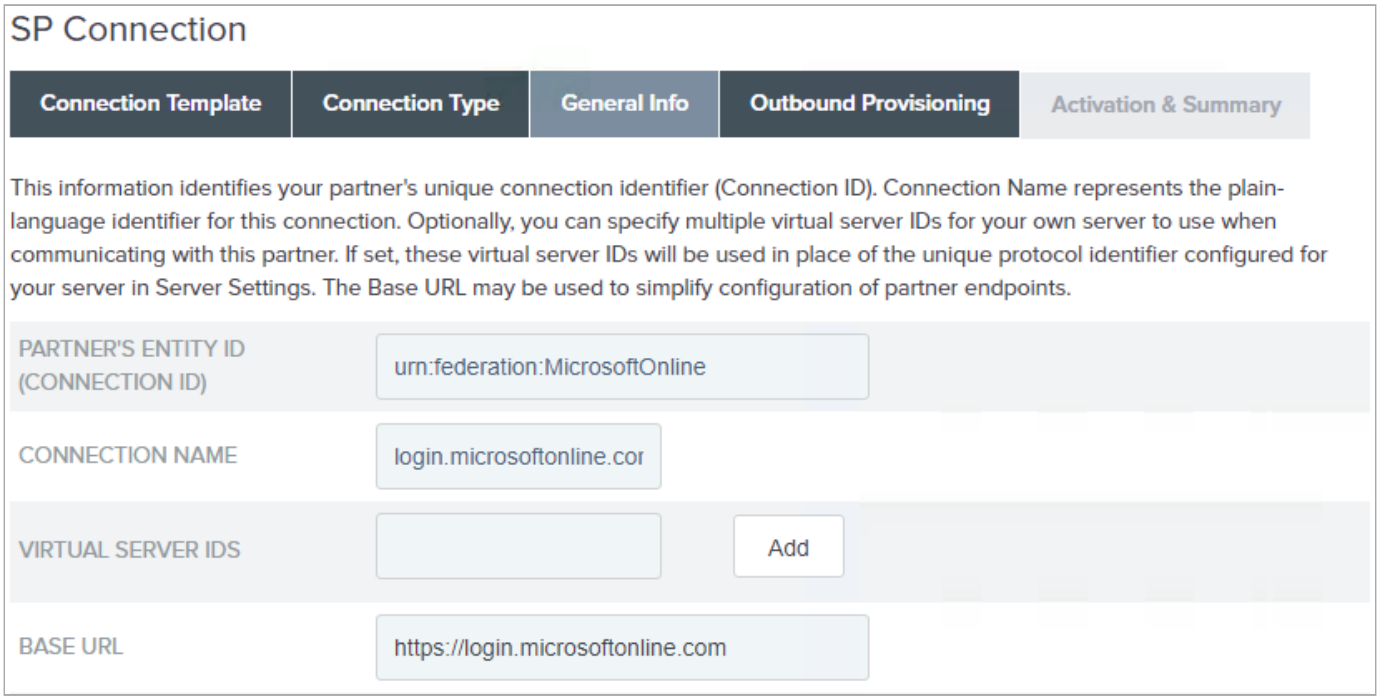
4. On the **Connection Options** screen, ensure **Browser SSO** is selected and click **Next**.



5. On the **General Info** screen, ensure that the **Partner's Entity ID (Connection ID)** and the **Connection Name** are accurate. Change details if required and click **Next**.

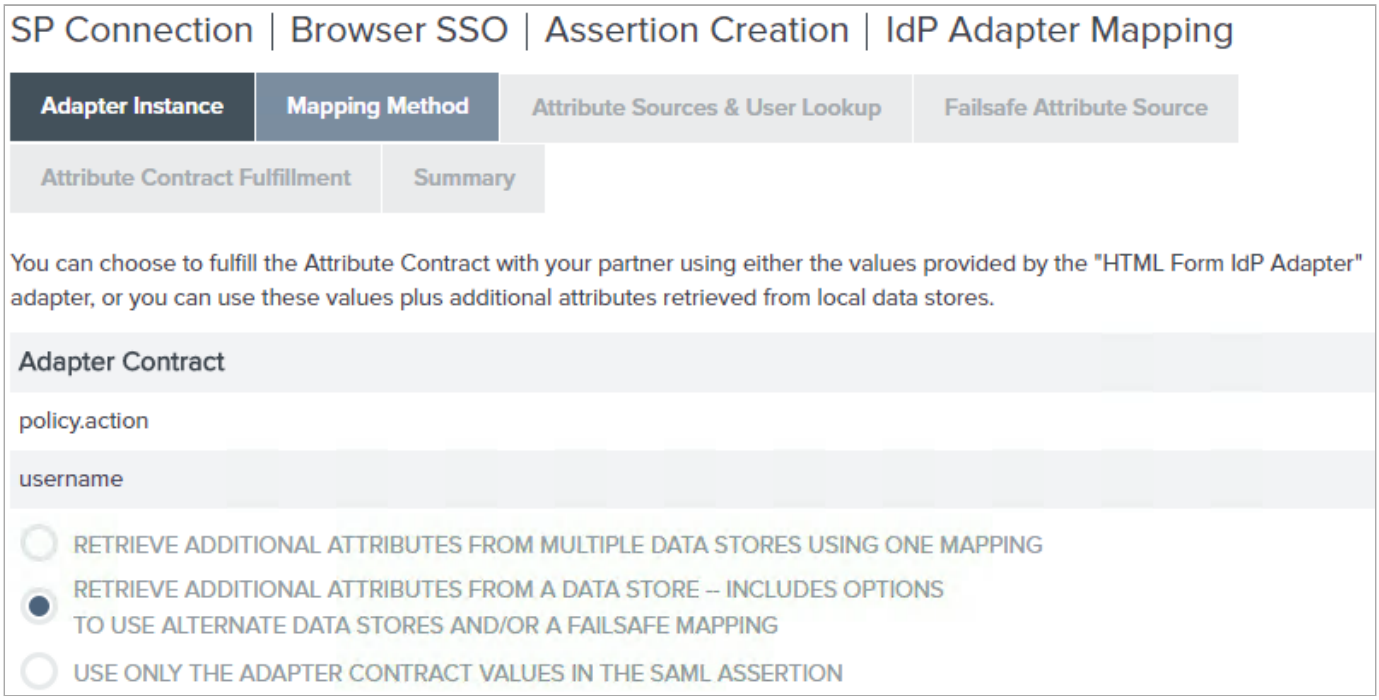
> **Note**
>
> By default, some fields are pre-populated as a result of using the Office 365 Connector template.

Copyright © 2025 Ping Identity Corporation



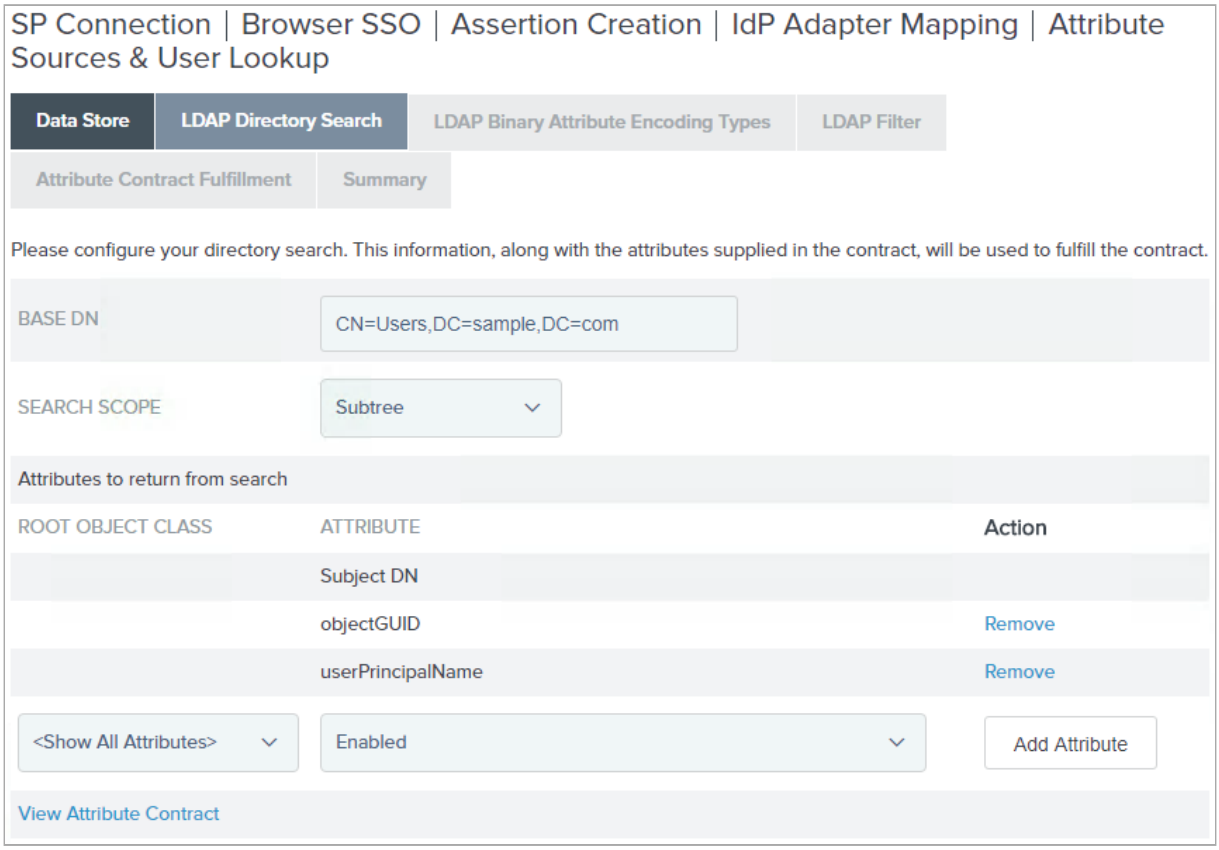
6. On the **Browser SSO** screen, click **Configure Browser SSO**.
7. On the **Assertion Creation** screen, click **Configure Assertion Creation**.
8. On the **IdP Adapter Mapping** screen, click **Map New Adapter Instance**. If an HTML form adapter form already exists, select it from the drop down list and click **Next**. Otherwise, perform the following steps to create a new HTML form adapter:
    1. If an LDAP instance has not been configured in PingFederate, follow the instructions in Configuring an LDAP connection.
    2. If a credential validator has not already been created, follow the instructions in Configure the LDAP Username Password Credential Validator.
    3. Complete the creation of the HTML form adapter using the instructions in Configuring an HTML Form Adapter instance.
    4. Once the above are completed, return to the **IdP Adapter Mapping** screen and click **Next**.
9. On the **Mapping Method** screen, select **Retrieve additional attributes from a data store—includes options to use alternate data stores and/or a failsafe mapping.** Click **Next**.

Copyright © 2025 Ping Identity Corporation



10. Click **Add Attribute Source**.
11. Fill in the **Attribute Source Description** field with an identifier of your choosing. Select the desired source datastore in the **Active Data Store** drop down list, then click **Next**.
12. On the **LDAP Directory Search** screen, enter the following values:
    - Base DN: where the users are found in the source datastore
    - Search Scope: select the appropriate value
    - Attributes to return from search:
        - **objectGUID**
        - **userPrincipalName**

Copyright © 2025 Ping Identity Corporation

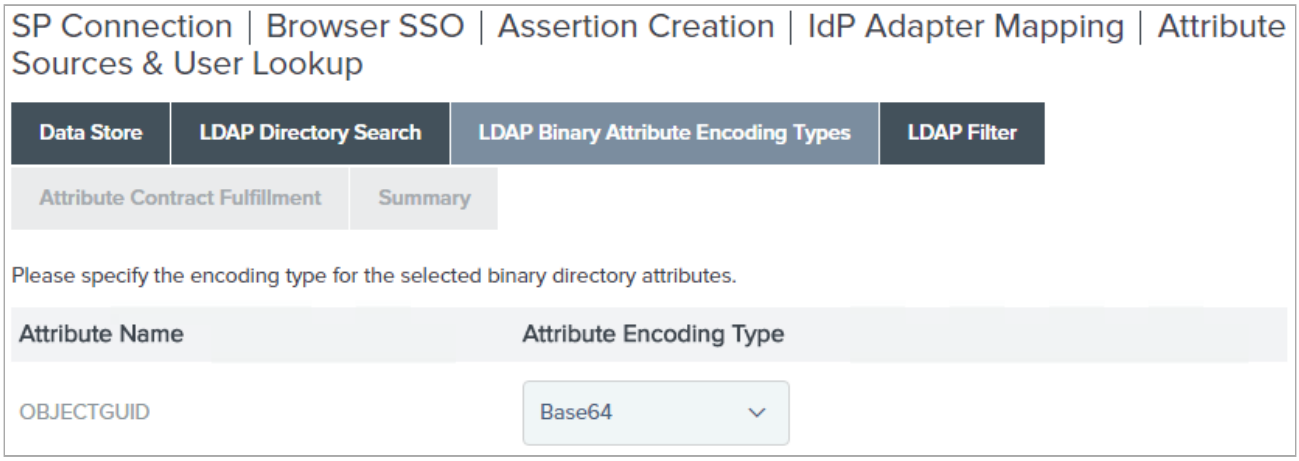


13. Click **Next**.
14. If you are in the **LDAP Binary Attribute Encoding Types** screen, confirm `objectGUID` is set to **Base64**, click **Next**.

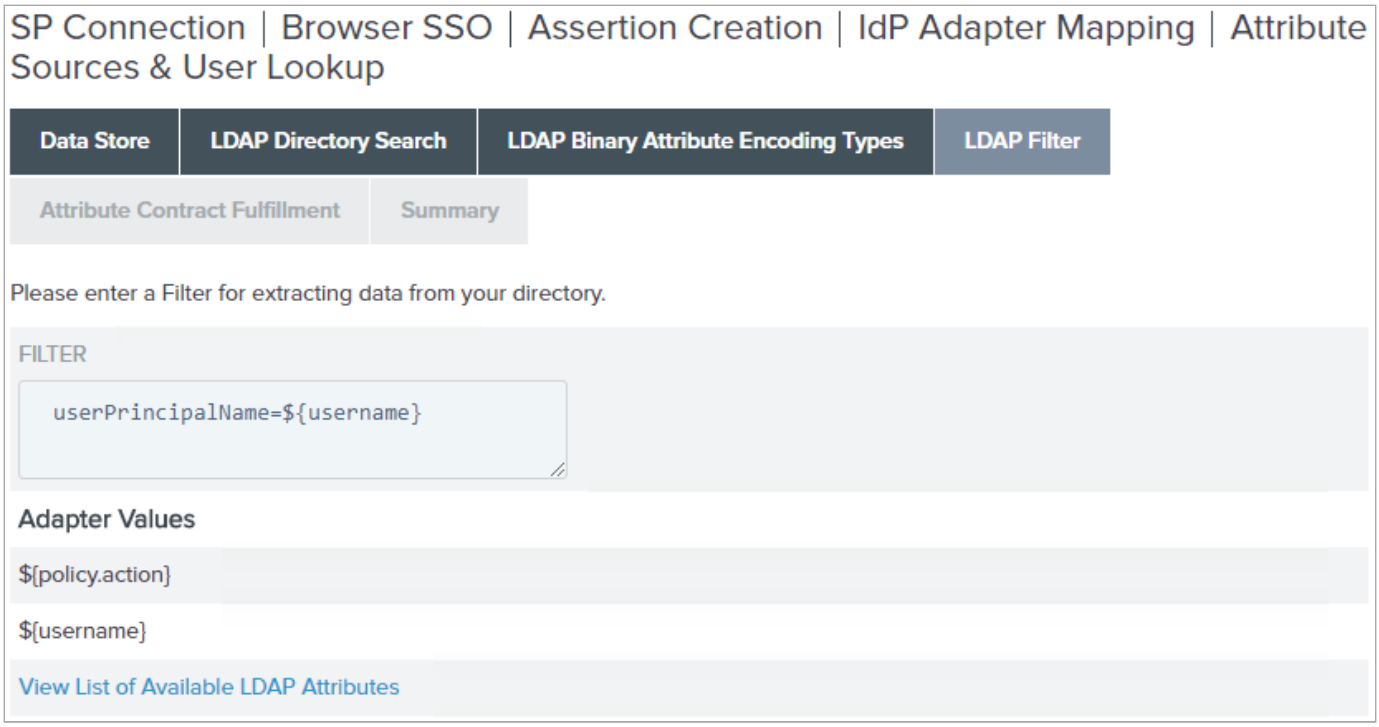
    If you are NOT in the **LDAP Binary Attribute Encoding Types** screen, then `objectGUID` is not currently retrieved in binary format and the datastore settings must be updated. To update `objectGUID` in LDAP perform the following steps:

    1. Open a new private browser session and log in to the PingFederate Admin Console
    2. Click **Data Stores**, then **Manage Datastores**
    3. Select your source datastore
    4. Click **LDAP Configuration**
    5. Click **Advanced**
    6. Select the LDAP Binary Attributes tab
    7. Enter `objectGUID` in the BINARY ATTRIBUTE NAME field and click **Add**
    8. Click **Done**, **Done**, and **Save**
    9. Return to the **LDAP Binary Attribute Encoding Types** screen
    10. Confirm `objectGUID` is set to **Base64** and click **Next**

Copyright © 2025 Ping Identity Corporation



15. On the **LDAP Filter** screen, enter `userPrincipalName=${username}` in the **Filter** field.



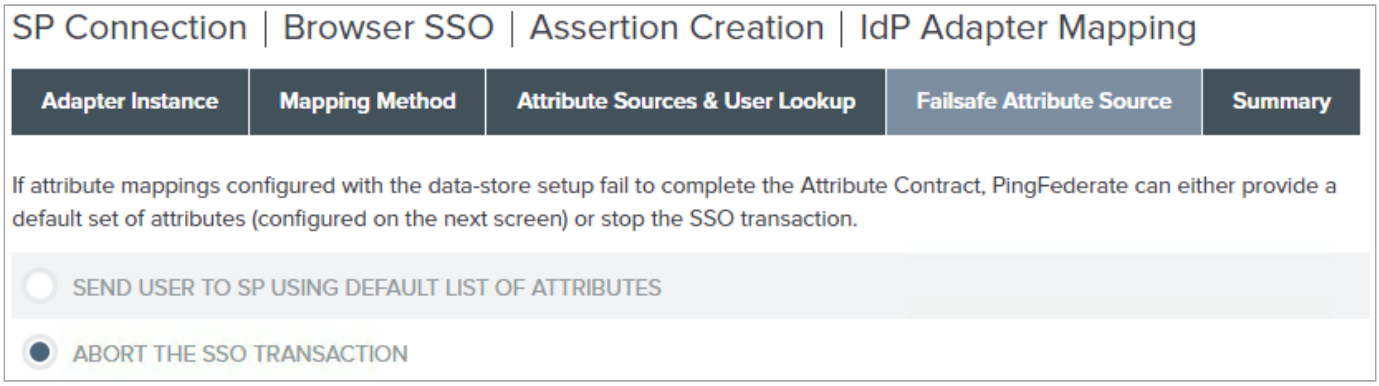
16. Click **Next**.
17. On the **Attribute Contract Fulfillment** screen, set the following values:

Attribute Contract	Source	Value
IDPEmail	LDAP	userPrincipalName
SAML_NAME_FORMAT	Text	urn:oasis:names:tc:SAML:2.0:nameid-format:persistent
SAML_SUBJECT	LDAP	objectGUID

18. Click **Next**.

Copyright © 2025 Ping Identity Corporation

19. On the **Attribute Source Summary** screen, click **Done**.
20. On the **Attribute Sources & User Lookup** screen, click **Next**.
21. On the **Failsafe Attribute Source** screen, select **Abort the SSO transaction** and click **Next**.



22. On the **IdP Adapter Mapping** summary screen, click **Done**.
23. On the **Authentication Source Mapping** screen, click **Done**.
24. On the **Assertion Creation** screen, click **Done**.
25. On the **Protocol Settings** screen, click **Configure Protocol Settings**.
26. On the **Assertion Consumer Service URL** screen, delete the binding `urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST-SimpleSign` and click **Done**. PingFederate does not support this binding and as such, it may result in validation errors.
27. On the **Protocol Settings** screen, click **Done**.
28. On the **Browser SSO** screen, click **Next**.
29. On the **Credentials** screen, click **Configure Credentials**.
30. On the **Credentials > Digital Signature Settings** screen, select the signing certificate and click **Next**.
31. On the **Signature Verification Settings** screen, click **Manage Signature Verification Settings**.
32. On the **Trust Model** screen, select the appropriate value and complete the steps for configuring the trust model and signature verification according to instructions in Managing signature verification settings.
33. On the **Signature Verification Summary** screen, click **Done**.
34. On the **Credentials** screen, click **Next**.
35. On the **Activation & Summary** screen, set **Connection Status** to **Active**, then click **Save**.

Add the signing certificate to Azure

About this task

Copyright © 2025 Ping Identity Corporation

The active signing certificate in PingFederate must be saved in Azure to secure the SSO communications between PingFederate and Office 365. SSO transactions cannot take place without the correct certificate added to Azure. Use the following procedure to add the signing certificate previously exported in Obtain PingFederate signing certificate to Azure.

Steps

1. Open the exported certificate using a text editor.
2. Copy the certificate text to the clipboard without header, footer, whitespace or carriage returns.
3. Execute the Powershell commands below in the command prompt window used in Configure federation settings.

```
PS> $cert = “<SAVED_CERTIFICATE_TEXT>”
PS> Set-MsolDomainFederationSettings -DomainName “$domainName” -SigningCertificate “$cert”
```

Troubleshooting

The following table lists potential problems administrators might encounter during the setup or deployment of the Office 365 Connector, along with possible solutions.

For more information and solutions, see PingFederate Office 365 FAQ in the Knowledge Base.

Problem	Possible Solution
The Exception appears in `server.log`: `{"code":"Authorization_RequestDenied", "message": {"lang":"en","value":"Insufficient privileges to complete the operation."}"`	In some cases the app created to provision using the Microsoft Graph API may not have the necessary permissions to access and modify all attributes. If you are seeing some provisioning events failing with an `Insufficient privileges to complete the operation.` exception elevating the permission level of the app may fix this. More information on this issue, as well as the script needed to elevate the permissions using Azure AD Module for Windows PowerShell can be found at Microsoft Support.

Release notes

Changelog

The following is the change history for the Office 365 Provisioner.

Office 365 Provisioner 2.3 – December 2021

- Migrated from the deprecated Azure Graph API to the Microsoft Graph API

Copyright © 2025 Ping Identity Corporation

Important

Update your Azure AD application permissions as shown in Add application to Azure AD.

Office 365 Provisioner 2.2 – June 2018

- Added support for hard deleting users
- Added deprovisioning configuration options for CRUD capabilities

Office 365 Provisioner 2.1 – March 2017

- Added configuration options for CRUD capabilities
- Added support for proxy connections

Office 365 Provisioner 2.0.2 – January 2017

- Fixed deserialization issue due to a SaaS API change

Office 365 Provisioner 2.0.1 – July 2016

- Fixed Group membership issue

Office 365 Provisioner 2.0 – January 2016

- Added support for provisioning additional user attributes
- Added support for deleting groups
- Added additional license configuration support
- Azure Active Directory Graph API updated from version v1.5 to v1.6
- Improved exception handling and reporting
- Minor bug fixes
- Updates to user group mappings resulting in the removal of the memberOf attribute

Office 365 Provisioner 1.1.2 – December 2015

- Fixed exception handling issue

Office 365 Provisioner 1.1.1 – June 2015

- Fixed compatibility issues

Office 365 Provisioner 1.1 – February 2015

- Support added for provisioning users with licenses
- Capability to assign managers to provisioned users

Copyright © 2025 Ping Identity Corporation

- Capability to update userPrincipalName
- Support for rename group
- Attribute changes require administrators to refresh the target and attribute mapping screens
- Support for non-base64 immutableId
- Resource and memberOf attributes removed

Office 365 Provisioner 1.0 – May 2014

- Initial Release
- Support for SSO and SLO
- Support for User Provisioning
- Support for Group Provisioning

Known issues and limitations

The following are known issues or limitations for the Office 365 Provisioner.

- User attributes cannot be cleared once set, they can only be updated.
- Due to a limitation with PingFederate 8.1 and earlier versions, when configuring two SP connections with the same provisioner, the second connection built may be pre-populated with the channel from the first connection. To avoid conflicts, delete this pre-populated channel and create a unique channel for each connection.
- Cookies must be enabled in the selected browser for SLO to work.
- Updating the mobile attribute requires that the service principal representing the provisioner (the place the user gets the client key and secret) be assigned a role with Company Administrator privileges (using Powershell). See O365 Connector: Mobile attribute updates for more information.
- Updating the Password attribute is not supported.
- User updates containing a manager that has not yet been provisioned / updated by the new version will fail, as the manager will not have the new extended attribute holding their distinguished name from AD
- If the DoBase64Conversion field is switched to “false”, expect conflicts / failures on federated domains containing pre-existing users provisioned by dirsync / V1.0
- Only outbound provisioning is supported.
- Syncing with existing groups is not supported.
- SAML SLO is not supported. (WS-Fed SLO is supported and set as default).
- After deleting a user, Azure AD prevents the same user from being created again due to a conflicting `immutableId` value. This issue only occurs when **Remove Action** is set to **Delete**.

Copyright © 2025 Ping Identity Corporation

Download manifest

The following files are included in the Office 365 Provisioner `.zip` archive.

The distribution `.zip` archive for the connector contains the following:

- `/legal`:
  - `Legal.pdf` – copyright and license information.
- `/dist` – contains libraries needed for the connector:
  - `pf-office365-quickconnection-<version>.jar` – PingFederate Office 365 Connector

Copyright © 2025 Ping Identity Corporation

OpenToken Token Translator

The PingFederate OpenToken Token Translator provides a Token Processor and a Token Generator for use with the PingFederate WS-Trust Security Token Service (STS).

The Token Processor allows an Identity Provider (IdP) STS to accept and validate an OpenToken from a Web Service Client (WSC) and then map user attributes into a SAML token for the WSC to send to a Web Service Provider (WSP). The Token Generator allows a Service Provider (SP) STS to issue an OpenToken for a WSP, including mapped attributes from an incoming SAML token.

> **Note**
>
> Ping Identity provides a Java STS-Client Software Development Kit (SDK) for enabling web service applications (client or provider) to interact with the PingFederate STS. You can download the Java Client SDK from the PingFederate server add-ons page.

OpenToken is an open-standard, secure session cookie used to pass user information between an application and PingFederate. For STS purposes, the OpenToken is passed as a Web Services Security (WSS) binary security token in WS-Trust messages. The data within the OpenToken is a set of key/value pairs, encrypted using common encryption algorithms, as illustrated below:



This translator package includes a Java Application Programmer Interface (API) for WSC and WSP developers to use for writing or reading an OpenToken, respectively.

Intended audience

This document is intended for PingFederate administrators.

If you need help during the setup process, see the following resources:

- The following sections of the PingFederate documentation:
  - WS-Trust STS configuration

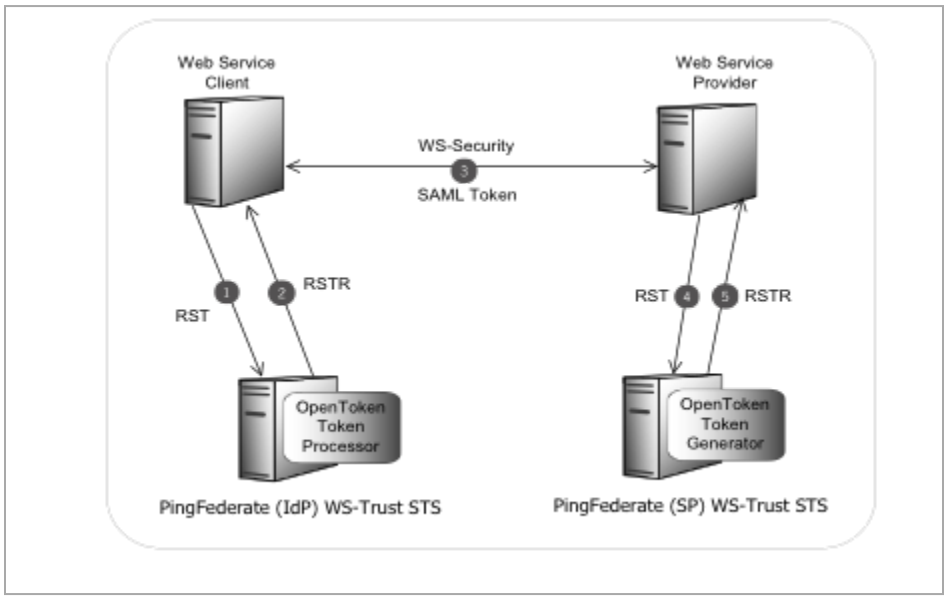
System requirements

- PingFederate 9.3 or later
- To use the Java API, J2SE Java Runtime Environment 1.5 or later is required on the WSC and WSP
- To use strong Advanced Encryption Standard (AES) encryption with a key size of more than 128 bits, the Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files must be installed in your JDK on PingFederate, as well as the WSC and WSP.

Copyright © 2025 Ping Identity Corporation

WS-Trust STS processing

The following figure shows a basic Web Services scenario using the PingFederate WS-Trust STS in the role of both IdP and SP:



Processing Steps

1. A WSC sends a Request Security Token (RST) message containing an OpenToken as a SOAP request to the PingFederate STS IdP endpoint.
2. The OpenToken Token Processor validates the OpenToken and, if valid, maps attributes from the OpenToken into a SAML token. PingFederate issues the SAML token based upon the SP connection configuration and embeds the token in a Request Security Token Response (RSTR) which is returned to the WSC.
3. The WSC binds the issued SAML token into a WSS header and sends it via a SOAP request to the WSP.
4. The WSP sends an RST Issue request containing the SAML token to the PingFederate STS SP endpoint. PingFederate validates the SAML token and, if valid, maps attributes from the SAML token in to an OpenToken. Ping Federate issues the Open Token based upon the OpenToken Token Generator configuration and embeds the token in an RSTR which is returned to the WSP.
5. The WSP receives the OpenToken in the RSTR for local domain processing.

Setup

IdP installation and setup

About this task

This section describes how to install and configure the OpenToken Token Processor for PingFederate acting as an IdP.

Copyright © 2025 Ping Identity Corporation

Steps

1. Copy the `pf-opentoken-token-translator-<version>.jar` file from the deploy directory of this distribution to the `<pf-install>/pingfederate/server/default/deploy` directory of your PingFederate server installation.

   > **Note**
   >
   > If you have a previous version of the OpenToken Token Translator file installed, please delete it from the above location and replace it with the version referenced.

2. Log on to the PingFederate administrative console and click **Token Processors** under **IdP Configuration** on the main menu.

   If you do not see **Token Processors** on the Main Menu, enable WS-Trust by going to the **Server Settings Roles & Protocols** screen and selecting **WS-Trust** for the IdP Role.

   > **Note**
   >
   > To enable token exchange, you may be prompted to provide SAML 1.x and SAML 2.0 federation identifiers for the STS on the Federation Info screen. Refer to the Federation Info screen's Help page for more information.

3. On the **Manage Token Processor Instances** tab, click **Create New Instance**.
4. On the **Type** tab, enter an **Instance Name** and **Instance ID**, and select **OpenToken Token Processor** as the **Type**.
5. Click **Next**.
6. On the **Instance Configuration** screen, enter a strong password for generating the encryption key.
7. Click **Show Advanced Fields** to set other encryption and validation options.

   For more information, see the screen description column.

8. Click **Next**.
9. On the **Actions** screen, click **Download** and then **Export** to save the `agent-config.txt` file.

   The WSC application that generates the OpenToken will need this information.

10. Click **Next**.
11. On the **Extended Contract** tab, add any attributes that you want to map into the SAML assertion, in addition to the subject.
12. Click **Next**.
13. (Optional) On the **Token Attributes** screen, select any or all attributes whose values should be masked in PingFederate log files.

    Additionally, you can select **Mask all OGNL-expression generated log values**.

14. Click **Next**.
15. On the **Summary** tab, verify that the information is correct and click **Done**.
16. On the **Manage Token Processor Instances** screen, click **Save**.

Copyright © 2025 Ping Identity Corporation

OpenToken Token Processor Settings Reference

Field descriptions for the OpenToken Token Processor configuration screen.

Standard fields

Field	Description
Password	The password to use for generating the encryption key. Also known as the shared secret.
Confirm Password	Must match the value entered in the **Password** field.

Advanced fields

Field	Description
Cipher Suite	The algorithm, cipher mode, and key size that should be used for encrypting the token. The default selected value is **AES-128/CBC**.
Token Lifetime	The duration in seconds for which the token is valid. Valid range is 1 to 28800. The default value is `300` (5 minutes).
Session Lifetime	The duration in seconds for which the token may be re-issued without authentication. Valid range is 1 to 259200. The default value is `43200` (12 hours).
Not Before Tolerance	The amount of time in seconds to allow for clock skew between servers. Valid range is 0 to 3600. The default value is `0`.
Force SunJCE Provider	If selected, the SunJCE provider is forced for encryption and decryption.
Use Verbose Error Messages	If selected, use verbose TokenException messages.
Obfuscate Password	If selected, the password will be obfuscated and password-strength validation will be applied. Clearing the checkbox allows backward compatibility with previous OpenToken agents.
Skip Malformed Attribute Detection	If not selected, it prevents insecure content from affecting the security of your application/agent. It is recommended to update your applications with the latest version of the agent. It is recommended to not change the value of this flag.

Copyright © 2025 Ping Identity Corporation

SP installation and setup

Steps

1. Copy the `pf-opentoken-token-translator-<version>.jar` file from the dist directory of this distribution to the `<pf-install>/pingfederate/server/default/deploy` directory of your PingFederate server installation.

   > **Note**
   >
   > If you have a previous version of the OpenToken Token Translator file installed, please delete it from the above location and replace it with the version referenced.

2. Log on to the PingFederate administrative console and click **Token Generators** under **SP Configuration** on the main menu.

   If you don't see **Token Generators** on the main menu, enable WS-Trust by going to the **Server Settings Roles & Protocols** screen and selecting **WS-Trust** for the **SP Role**.

   > **Note**
   >
   > To enable token validation, you may be prompted to provide SAML 1.x and SAML 2.0 federation identifiers for the STS on the Federation Info screen. Refer to the Federation Info screen's Help page for more information.

3. On the **Manage Token Generator Instances** tab, click **Create New Instance**.
4. On the **Type** tab, enter an **Instance Name** and **Instance ID**, and select **OpenToken Token Generator** as the **Type**.
5. Click **Next**.
6. On the **Instance Configuration** page, enter in a strong password for generating the encryption key. Optionally, you can click **Show Advanced Fields** to set other encryption and validation options. Refer to the screen Description column for more information.
7. Click **Next**.
8. On the **Actions** page, click **Download** and then **Export** to save the `agent-config.txt` file.

   The WSP application that receives the OpenToken will need this information later.

9. Click **Next**.
10. On the **Extended Contract** tab, add any attributes that you want to map from the SAML assertion, in addition to the subject.
11. Click **Next**.
12. On the **Summary** tab, verify that the information is correct and click **Done**.
13. On the **Manage Token Generator Instances** screen, click **Save**.

OpenToken Token Generator Settings Reference

Field descriptions for the OpenToken Token Generator configuration screen.

Copyright © 2025 Ping Identity Corporation

Table 1. Standard fields

Field	Description
Password	The password to use for generating the encryption key. Also known as the shared secret.
Confirm Password	Must match the value entered in the **Password** field.

Table 2. Advanced fields

Field	Description
Cipher Suite	The algorithm, cipher mode, and key size that should be used for encrypting the token. The default selected value is **AES-128/CBC**.
Token Lifetime	The duration in seconds for which the token is valid. Valid range is 1 to 28800. The default value is `300` (5 minutes).
Session Lifetime	The duration in seconds for which the token may be re-issued without authentication. Valid range is 1 to 259200. The default value is `43200` (12 hours).
Not Before Tolerance	The amount of time in seconds to allow for clock skew between servers. Valid range is 0 to 3600. The default value is `0`.
Force SunJCE Provider	If selected, the SunJCE provider is forced for encryption and decryption.
Use Verbose Error Messages	If selected, use verbose TokenException messages.
Obfuscate Password	If selected, the password will be obfuscated and password-strength validation will be applied. Clearing the checkbox allows backward compatibility with previous OpenToken agents.
Skip Trimming of Backslashes	If not selected, it prevents insecure content from affecting the security of your application and the agent. Update your applications with the latest version of the agent. It is recommended to not change the value of this flag to avoid a negative security impact, such as someone maliciously adding slashes to exploit the system

The OpenToken agent API

WSC and WSP application developers can use the Java OpenToken Agent API (included in this distribution) to write and read an OpenToken.

Copyright © 2025 Ping Identity Corporation

The API provide access to functionality for writing an OpenToken at the WSC to be exchanged for a SAML token, and for reading an OpenToken at the WSP that was issued from a SAML token.

> **Note**
>
> The Agent API may be used in conjunction with the Java STS Client SDK for interacting with the PingFederate WS-Trust endpoints. You can download the Java Client SDK from the PingFederate server add-ons page. Examples in this section implement the SDK.

> **Important**
>
> If you have already installed a previous version of the OpenToken Agent, we recommend removing the existing Agent JAR file and installing the current version to address potential security issues.

Install the OpenToken agent API

Complete the following to install the Java OpenToken Agent API.

Steps

1. Copy the following JAR files to a location in the CLASSPATH of the Java application:
   - `opentoken-agent-2.7.2.jar`
   - `commons-collections-3.2.2.jar`

     > **Note**
     >
     > Version 3.2.2 or later of the Apache Commons Collection is required.
2. Ensure the following Apache Commons libraries are also available in the application CLASSPATH:
   - `commons-beanutils-1.9.4.jar`
   - `commons-logging-1.1.1.jar`

Read an OpenToken as a WSP

The Agent API provides access to functionality for reading an OpenToken received in an Issue request from the PingFederate STS.

Java Sample Code

The code snippet below demonstrates using the PingFederate STS Java Client SDK to retrieve the OpenToken issued from the PingFederate STS and using the OpenToken Agent API to read the OpenToken. If any errors are encountered while creating the token, a TokenException is thrown:

Copyright © 2025 Ping Identity Corporation

```
// Configure STS Client
              (SP-side, IdP connection) AgentConfiguration spAgentConfiguration = new
              AgentConfiguration(); spAgentConfiguration.setPassword("Password1");
              spAgentConfiguration.setCipherSuite(Token.CIPHER_SUITE_AES128CBC);    // Instantiate the
OpenToken agent
   Agent spAgent = new Agent(spAgentConfiguration);
// Configure STS
              Client STSClientConfiguration spStsConfig = new STSClientConfiguration();
              spStsConfig.setStsEndpoint("https://sp.domain.com:9031/sp/sts.wst");
              spStsConfig.setOutTokenType(TokenType.BINARY);
              spStsConfig.setOutTokenValueType(TokenType.OPENTOKEN);    // Instantiate STS Client
   STSClient spStsClient = new STSClient(spStsConfig);
   // Send RST Issue request to STS
   Element opentoken = spStsClient.issueToken(new Saml20Token(samlToken));
   // Read OpenToken
String otk = textContent(opentoken);
MultiMap otkValues = new MultiValueMap();
otkValues = spAgent.readTokenToMultiMap(otk);
```

Write an OpenToken as a WSC

The OpenToken Agent API provides access to functionality for writing an OpenToken as a WSC to include in an Issue request to the PingFederate STS.

Java Sample Code

The writeToken method of the Agent class takes an org.apache.commons.collections.MultiMap collection of attributes and encodes them into an OpenToken.

> **Note**
>
> The collection of attributes must contain a key named “subject” for a valid token to be generated.

If any errors are encountered while creating the token, a TokenException is thrown.

The code snippet below demonstrates generating an OpenToken and using the PingFederate STS Java Client SDK to send the OpenToken to the PingFederate STS:

Copyright © 2025 Ping Identity Corporation

```
// Configure the Opentoken
              agent AgentConfiguration agentConfiguration = new AgentConfiguration();
              agentConfiguration.setPassword("2Federate");
              agentConfiguration.setCipherSuite(Token.CIPHER_SUITE_AES128CBC);    // Instantiate the
OpenToken agent
   Agent agent = new Agent(agentConfiguration);
   // Set OpenToken attributes
   MultiMap values = new MultiValueMap();
   values.put(Agent.TOKEN_SUBJECT, "joe");
   values.put("foo", "bar");
   String tokenData = agent.writeToken(values);
// Configure STS
              Client STSClientConfiguration idpStsConfig = new STSClientConfiguration();
              idpStsConfig.setAppliesTo("http://sp.domain.com");
              idpStsConfig.setStsEndpoint("https://idp.domain.com:9031/idp/sts.wst");
              idpStsConfig.setInTokenType(TokenType.BINARY);
              idpStsConfig.setInTokenValueType(TokenType.OPENTOKEN);    // Instantiate STS Client
   STSClient idpStsClient = new STSClient(idpStsConfig);
   // Send RST Issue request to STS
   Element samlToken = idpStsClient.issueToken(tokenData);
```

Release notes

Changelog

The following is the change history for the OpenToken Token Translator.

OpenToken Token Translator 1.0.2 - August 2022

- Updated dependencies to address security issues.

OpenToken Token Translator 1.0.1 - December 2012

- Updated to address security issue found since the previous release
- Added support for OpenToken 2.5.1 Adapter and the OpenToken 2.5.1 Agent

OpenToken Token Translator 1.0 - April 2009

- Initial Release

Download manifest

The following files are included in the OpenToken Token Translator `.zip` archive:

- `Legal.pdf` - copyright and license information

Copyright © 2025 Ping Identity Corporation

- `agent` - contains the OpenToken Agent and supporting libraries
  - `opentoken-agent-<version>.jar` - a JAR file that contains the OpenToken Agent
  - `lib` - contains the libraries needed by the OpenToken Agent
    - `commons-beanutils-1.9.4.jar` - the Apache Commons Bean Utility library
    - `commons-collections-3.2.2.jar` - the Apache Commons Collections library
    - `commons-logging-1.1.1.jar` - the Apache Commons Logging library
- `dist/pingfederate/server.default` - contains the integration files
  - `deploy` - contains the Java libraries
    - `pf-opentoken-token-translator-<version>.jar` - a JAR file that contains the OpenToken Token Translator JAR.

Copyright © 2025 Ping Identity Corporation

Oracle Access Manager (OAM) Integration Kit

The PingFederate Oracle Access Manager (OAM) Integration Kit adds Identity Provider (IdP) and Service Provider (SP) Adapters to PingFederate.

The OAM IdP Adapter allows an IdP enterprise to extend an existing OAM investment by using the SAML or WS-Federation protocols to expand the reach of the OAM domain to partner applications. The OAM SP Adapter allows an SP enterprise to accept SAML or WS-Federation assertions and provide SSO to OAM-protected applications

Intended audience

This document is intended for PingFederate administrators and web application developers.

If you need help during the setup process, see the following sections of the PingFederate documentation:

- Managing SP adapters
- Configuring an OpenToken SP Adapter instance

System requirements

- PingFederate 8.x or later
- OAM Server 11g R2
- OAM Access SDK11.1.2.3.0 (installed on the same machine running the PingFederate server)
- OAM 11g Webgate running on Apache 2.4
- Redhat 6.7 (if using the precompiled module included in this distribution

Setup

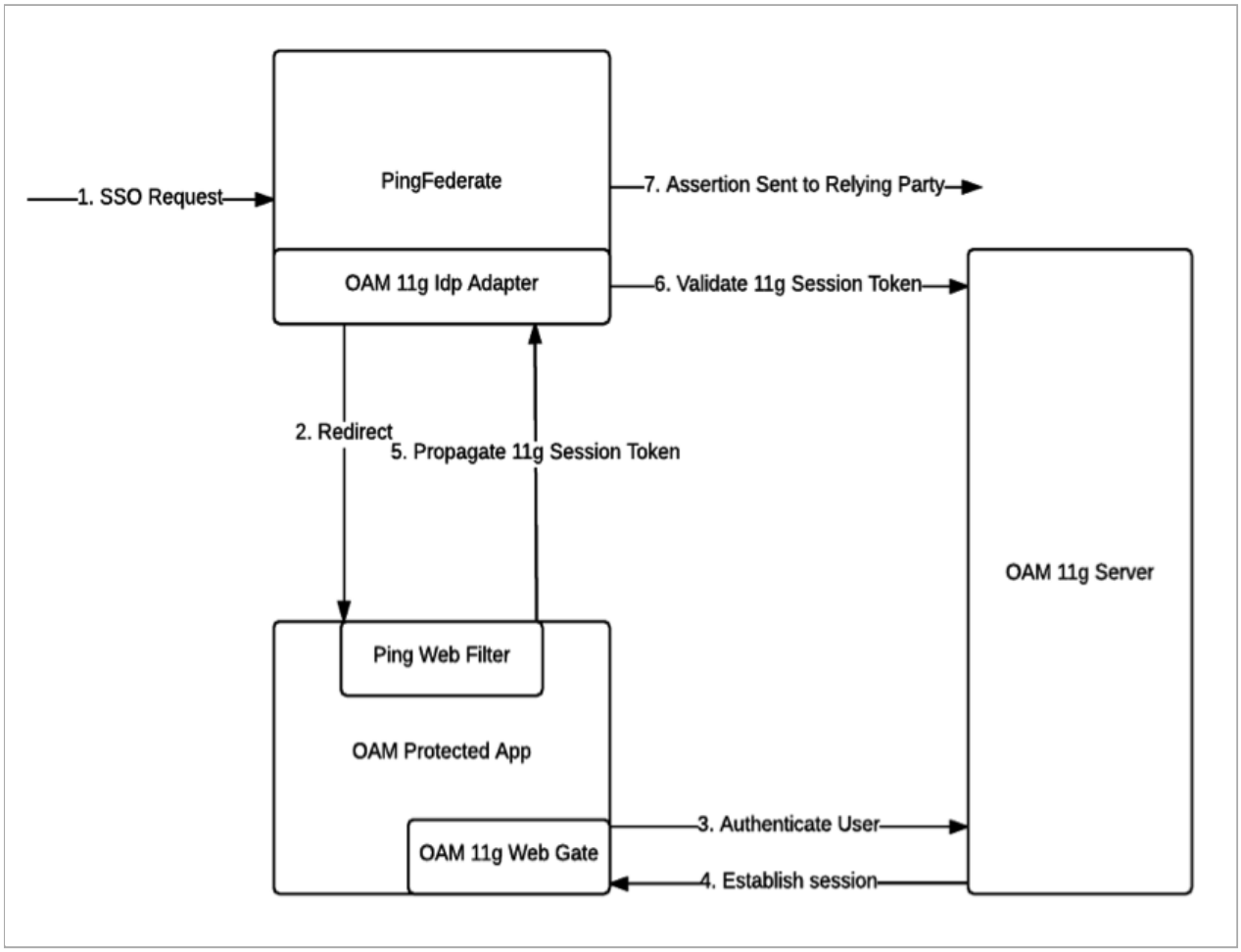
IdP implementation

This section describes using the OAM Integration Kit as an identity provider (IdP).

The OAM IdP Adapter uses the Access Server SDK to decrypt the OAM session cookie and pass attributes to the PingFederate server. You can then add attribute values to the Attribute Contract in the PingFederate administrative console and transfer them to a partner application in a SAML assertion. (For more information, see: Creating an Attribute Contract in the PingFederate Administrator's Manual.)

The following figure illustrates the request flow and how the OAM IdP Adapter is used to facilitate generating a SAML WS-Federation assertion from the ObSSOCookie:

Copyright © 2025 Ping Identity Corporation



OAM IdP Overview

Processing Steps

1. User initiates single sign on through PingFederate.
2. The OAM IdP Adapter redirects the user to an OAM Protected Resource.
3. OAM Webgate authenticates the user.
4. After successful authentication an OAM 11g session is established and a host level cookie is created for the Webgate.
5. User is allowed access to the OAM protected resource at which point the Ping Web Filter intercepts this request and sends the host level OAM Session token to PingFederate.
6. OAM IdP Adapter validates the session token using Access Server APIs.
7. The user information is passed to PingFederate, which can create an assertion and send it to the required relying party (aka service provider).

    - OAM IdP configuration
    - Apache module installation
    - Apache module configuration
    - PingFederate IdP configuration

Copyright © 2025 Ping Identity Corporation

- IdP adapter instance configuration
- IdP adapter testing

Upgrade instructions

About this task

Use the following instructions to upgrade an existing OAM 3.0 installation to the current version.

Steps

1. Stop PingFederate.
2. Delete the previous adapter: `<PF_INSTALL>/server/default/deploy/pf-oam-adapter-*.jar`
3. From the extracted distribution `.zip` archive:
   - Copy `dist/mod_pfoam.so` to `<apache installation>/modules`
   - Copy `dist/pf-oam-adapter-3.1.0.jar` to `<PF_INSTALL>/server/default/deploy`
4. Copy the following files from the Access Server SDK to `<PF_INSTALL>/server/default/deploy`:
   - oamasdk-api.jar
   - opss_standalone/modules/
     - oracle.idm_11.1.1/identitystore.jar
     - oracle.pki_11.1.1/oraclepki.jar
     - oracle.jps_11.1.1/* (all files)
     - oracle.osdt_11.1.1/* (all files)

     > **Note**
     >
     > The files listed above pertain to the specified version of the OAM SDK in the System Requirements. Other versions may require different files.
5. Start PingFederate.
6. Verify configuration within PingFederate for all OAM IdP Adapters.

OAM IdP configuration

About this task

These steps are necessary to configure OAM for operation as an identity provider (IdP) with PingFederate.

Copyright © 2025 Ping Identity Corporation

> **Note**
>
> You can find more information on the Webgate configuration files in OAM documentation for configuring Webgates.

Steps

1. Create an OAM Apache 11g Webgate (or use an existing one).
2. Create a new folder in the PingFederate Server, to store the Webgate configuration files. This folder will henceforth be referred to as the Agent Config Location. This path must be specified during the PingFederate Configuration.
3. Copy the Webgate configuration files to the Agent Config Location folder.

Apache module installation

About this task

This procedure shows you how to add the `mod_pfoam.so` module to your Apache installation.

Steps

1. From the extracted distribution `.zip` archive:
    1. Copy `dist/mod_pfoam.so` to `<apache installation>/modules`.
    2. Copy `conf/httpd-pfoam.conf` to `<apache installation>/conf/extra`.
2. Open your Apache server configuration file (`httpd.conf`).
3. Look for the first `LoadModule` directive in the `httpd.conf` file.
4. Immediately before that directive insert this line:

    ```
    LoadModule pfoam module
    ```

5. At the end of the `httpd.conf` file insert this line:

    ```
    Include conf/extra/httpd-pfoam.conf
    ```

Apache module configuration

The configuration options for the Apache module are listed in the table below. Update the directives as needed in the module's configuration file: `<apache installation>/conf/extra/httpd-pfoam.conf`

Table 1. Configuration directives

Field	Description	Default Value
OAMCookieName	Cookie name containing the OAM 11g Session Token. Example: `OAMAuthnCookie_webgate.mydomain.com:80`	n/a

Copyright © 2025 Ping Identity Corporation

Field	Description	Default Value
PFResumePath	Parameter containing the relative sso url passed from PingFederate	resumePath
SessionTokenParameterName	Parameter Name used to pass OAM session token to PingFederate	OAMAuthnCookie
PFBaseUrl	Base URL for PingFederate used in conjunction with resumePath. Example: `https://mydomain.com:9031`	n/a

> **Important**
>
> Restart the Apache server after making configuration changes.

PingFederate IdP configuration

Steps

1. Unzip the distribution `.zip` archive and copy this file to the server/default/deploy folder in your PingFederate server installation:

   `dist/pf-oam-adapter-3.1.0.jar`

2. Copy the following file to the Agent Config Location folder, which was created in Step 2 of OAM IdP configuration:

   `conf/jps-config.xml`

3. Add the following to `run.properties` within `<PF_HOME>/bin` folder:

   `oracle.security.jps.config=<AGENT_CONFIG_LOCATION>/jps-config.xml`

   > **Important**
   >
   > Ensure that the Agent Config Location path uses forward slashes (/), as shown above.

4. Install and configure the OAM Access Server SDK. For information on the Access Server SDK, refer to your OAM documentation.

   > **Note**
   >
   > The Access Server SDK functions as a gate to the OAM Access Server and some files will need to be copied to the server where PingFederate is running.

5. Copy the following files from the Access Server SDK to `<PF_INSTALL>/server/default/deploy`:
   - `oamasdk-api.jar`
   - `opss_standalone/modules/`
     - `oracle.idm_11.1.1/identitystore.jar`

Copyright © 2025 Ping Identity Corporation

- `oracle.pki_11.1.1/oraclepki.jar`
- `oracle.jps_11.1.1/* (all files)`
- `oracle.osdt_11.1.1/* (all files)`

> **Note**
>
> The files listed above pertain to the specified version of the OAM SDK in the System Requirements. Other versions may require different files

6. Start or restart the PingFederate server.

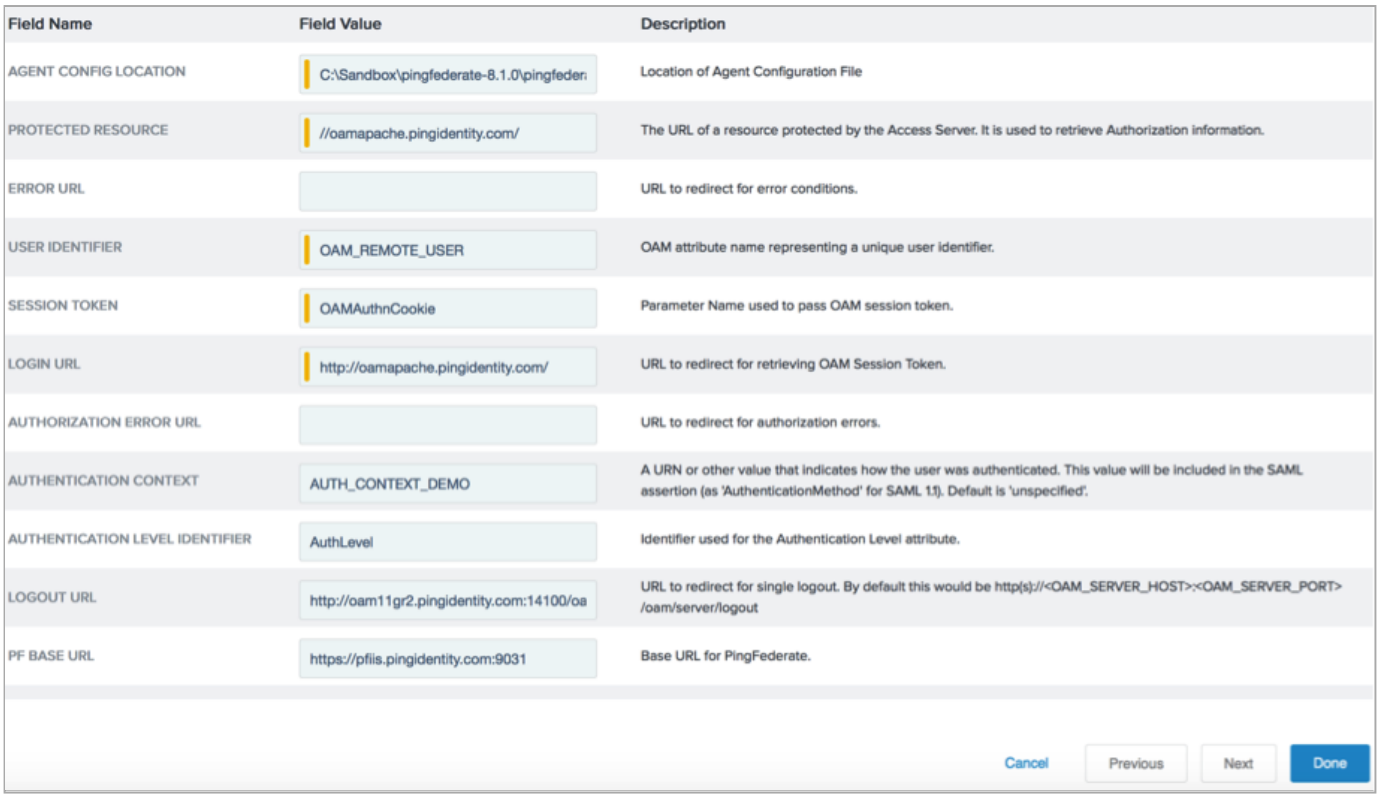
IdP adapter instance configuration

About this task

After installing the OAM Integration Kit and the Access Server SDK library, you can configure your SP connection to use an instance of the OAM Adapter. The first part of this process is configuring the adapter instance.

Steps

1. Log on to the PingFederate administrative console and click **Adapters** under **IdP Configuration** on the Main Menu screen.
2. On the Manage IdP Adapter Instances screen, click **Create New Instance**.
3. Enter the Adapter Name and Adapter ID. Select OAM 11g IdP Adapter 3.1.0 as the Adapter Type and click **Next**.
4. On the IdP Adapter screen, enter the values for adapter configuration as described on the screen and click **Next**.



Copyright © 2025 Ping Identity Corporation

> **Note**
>
> The Authentication Level Identifier is taken from the user's session token. The default/recommended value is authLevel. For the user's Authentication Level to be sent in the assertion, you must add the Authentication Level Identifier to the Adapter Contract (see step 5 below).

5. Optionally, on the **Extended Adapter Contract** screen, you can configure additional attributes for the adapter. For instance, you can use the extended adapter contract for Policy Server response-object attributes. For more information, see Extending an IdP adapter contract in the PingFederate documentation.
6. Click **Next**.
7. Select **userId** as the unique id. You may also select any extended attributes specified in the previous screen.
8. On the **Summary** screen, verify that the information is correct and click **Done**.
9. On the **Manage Adapter Instances** screen, click **Save** to complete the adapter configuration.

Result

You can now use this adapter instance for an SP connection. For information on setting up or modifying a connection, see SP connection management in the PingFederate documentation.

IdP adapter testing

About this task

You can test this adapter using the SP sample application that ships with PingFederate. Follow this procedure to verify adapter functions:

Steps

1. Set up PingFederate to run the SP sample application according to instructions in the Sample Application Quick Start Guide.
2. Configure an instance of the OAM Adapter (see OAM IdP configuration).
3. Reconfigure the SP connection to the sample application to use the OAM Adapter Instance by deleting the existing adapter instance and map the OAM Adapter instance in its place. See Managing mappings in the PingFederate documentation.

> **Note**
>
> Use the default setting on the Assertion Mapping screen. On the Attribute Contract Fulfillment screen, map SAML_SUBJECT to the Adapter value userId. If you have extended the Adapter Contract and wish to send the extended-attribute value to the SP during SSO, you will need to add a corresponding attribute to the Attribute Contract for the SP connection. Then map this attribute to the additional adapter attribute value (for example, authLevel).
> For any attributes in the Attribute Contract for which there are no related Adapter attributes, select Text in the Source drop-down list for each attribute and enter "test" (or any other text) in the associated text boxes.

4. On a web page protected by the OAM Access Gate, create an "SSO" link to the PingFederate startSSO endpoint, including the sample SP's connection ID, in the following format:

Copyright © 2025 Ping Identity Corporation

`http[s]://<PF_host>:<port>/IdP/startSSO.ping?PartnerIdPId=<connection_id>`

where:

- <PF_host> is the machine running the PingFederate server
- <port> is the PingFederate port and
- <connection_id> is the Connection ID of the SP connection to the sample application.

5. Access the protected web page by authenticating through OAM Webgate, and click the SSO link.

Result

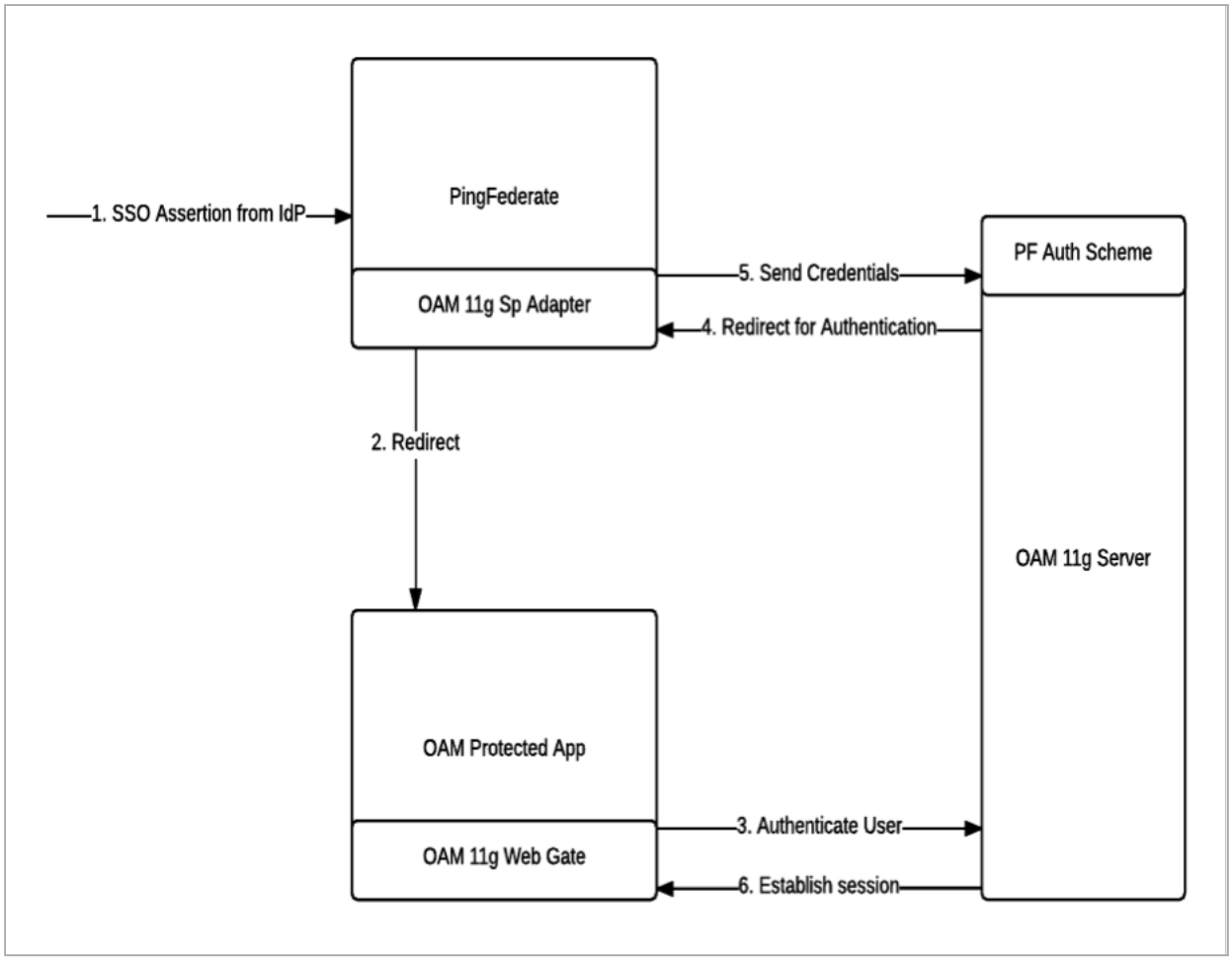
You will be logged on to the sample SP application. If you have modified the connection Attribute Contract to include Authentication Level and extended the Adapter Contract, you should see the authLevel displayed in the “User Attributes” table.

SP implementation

The OAM SP Adapter uses an authentication scheme deployed within Oracle Access Manager to create a session for the user.

The following figure illustrates the request flow and how the OAM SP Adapter is used to facilitate using a SAML WS-Federation assertion to create an OAM session:

Copyright © 2025 Ping Identity Corporation



OAM SP Implementation

Processing Steps

1. An SSO assertion is sent to PingFederate acting as an SP.
2. The OAM Sp Adapter redirects the user to an OAM Protected Resource secured with a PingFederate custom authentication scheme.
3. OAM Webgate sends a request to authenticate the user.
4. OAM Server redirects the authentication request to PingFederate.
5. OAM SP Adapter sends the required credentials back to the OAM Server.
6. The OAM Server validates the credentials and an 11g session is established.
    - OAM SP configuration
    - PingFederate SP configuration
    - SP adapter instance configuration
    - SP adapter testing

Copyright © 2025 Ping Identity Corporation

Upgrade instructions

About this task

Use the following instructions to upgrade an existing OAM 3.0 installation to the current version.

Steps

1. Stop PingFederate.
2. Delete the previous adapter: `<PF_INSTALL>/server/default/deploy/pf-oam-adapter-*.jar`
3. From the extracted distribution `.zip` archive, copy `dist/pf-oam-adapter-3.1.0.jar` to `<PF_INSTALL>/server/default/deploy`
4. Edit `<PF_INSTALL>/bin/run.properties` to remove the `TargetResource` query parameter from the property `pf.oam.ik.ssoUrl`. For example, the URL should look similar to: `https://<PF_HOST>:<PF_PORT>/sp/startSSO.ping?PartnerIdpId=<PARTNER_ID>`
5. Start PingFederate.
6. Verify configuration within PingFederate for all OAM SP Adapters.

OAM SP configuration

Steps

1. From the extracted distribution `.zip` archive, deploy the authentication plug-in jar, `dist/PingOpenTokenAuthPlugin.jar`, within OAM 11g and create an Authentication Module. For information on authentication plugins see About the Custom Plug-in Life Cycle in the OAM documentation.
2. The authentication plugin requires the opentoken configuration file (agent-config.txt) which can be obtained through the SP adapter configuration as described in the section below. Specify the location of this file for the authentication plugin property *opentokenConfigFile*.
3. Create or update an authentication scheme to use the plug-in deployed in Step 1. Use the following values for the authentication scheme parameters.

Table 1. Configuration directives

Parameter	Value
Challenge Method	Form
Challenge Redirect URL	`/oam/server/`
Authentication Module	Select the authentication module from step 1.
Challenge URL	`\http(s)://<PF_HOST:PF_PORT>/ext/pf-oam-authn/sso.ping`
Context Type	external

Copyright © 2025 Ping Identity Corporation

4. Configure an OAM Webgate to use the updated authentication scheme.

PingFederate SP configuration

Steps

1. Unzip the distribution `.zip` archive and copy the following file to the `server/default/deploy` folder in your PingFederate server installation:

   `dist/pf-oam-adapter-3.1.0.jar`

2. Add the following to `<PF_INSTALL>/bin/run.properties`:

   `pf.oam.ik.ssoUrl=<PF_SSO_URL>`

   where PF_SSO_URL is the Sp-initiated Single sign on URL. For example:

   ```
   https://<PF_HOST>:<PF_PORT>/sp/startSSO.ping?PartnerIdpId=<PARTNER_ID>
   ```

   > **Note**
   >
   > The Target Resource URL will be determined by the adapter based on the request context.

3. Start or restart the PingFederate Server.

SP adapter instance configuration

About this task

After installing the OAM Integration Kit, you can configure your SP connection to use an instance of the OAM SP Adapter. The first part of this process is configuring the adapter instance.

Steps

1. Log on to the PingFederate administrative console and click **Adapters** under **SP Configuration** on the Main Menu screen.
2. On the **Manage SP Adapter Instances** screen, click **Create New Instance**.
3. Enter the **Adapter Name** and **Adapter ID**. Select **OAM 11g SP Adapter 3.1.0** as the Adapter Type and click **Next**.
4. On the **SP Adapter** screen, enter the values for adapter configuration as described on the screen and click **Next**.

Copyright © 2025 Ping Identity Corporation

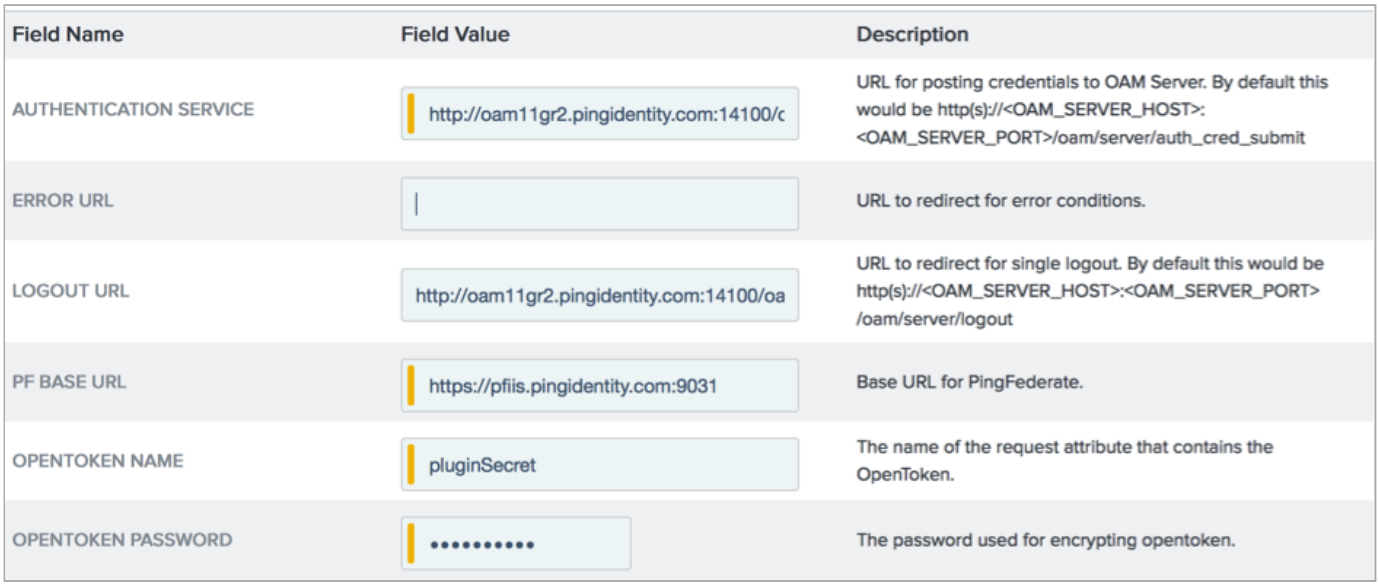


Figure 1. SP Instance Configuration

5. Download the opentoken configuration file (agent-config.txt). This will be used during authentication plugin configuration for the OAM server. Click **Next**.
6. Optionally, on the Extended Adapter Contract screen, you can configure additional attributes for the adapter. (See the Extending an Adapter Contract in the PingFederate Administrator's Manual.)

> **Note**
>
> Extended attributes are not supported in this version of OAM Integration Kit.

7. Click **Next**.
8. On the **Summary** screen, verify that the information is correct and click **Done**.
9. On the **Manage Adapter Instances** screen, click **Save** to complete the adapter configuration.

Result

You can now use this adapter instance for an IdP connection. For information on setting up or modifying a connection, see Managing IdP adapters in the PingFederate documentation.

SP adapter testing

About this task

You can test this adapter using the IdP sample application that ships with PingFederate. Follow this procedure to verify adapter functions:

Steps

1. Set up PingFederate to run the IdP sample application according to instructions in the Sample ApplicationQuick Start Guide.
2. Configure an instance of the OAM SP Adapter (see OAM SP configuration).

Copyright © 2025 Ping Identity Corporation

3. Reconfigure the IdP connection to the sample application to use the OAM Adapter instance by deleting the existing adapter instance for the connection and mapping the OAM Adapter instance in its place (see Configuring Adapter Mapping and User Lookup in the PingFederate Administrator's Manual).
4. From the Main Menu, click **Adapters** under **My SP Configuration** on the Main Menu screen.
5. Delete the Adapter Instance that was previously used by the sample-application connection.
6. Configure an OAM 11g Webgate to use the custom authentication plug-in.
7. Access an OAM protected resource within the OAM 11g Webgate from the previous step. You should arrive at the IdP sample application's login page.
8. Add at least one of the users in the username drop-down list to the OAM Identity Manager. Refer to your OAM documentation for more information.

   Alternatively, you can add users already in OAM Identity Manager to the sample application's user-properties file (see the *Quick Start Guide* for the location of this file).
9. Add the same user(s) to the Authorization Rule in the Policy Domain governing the protected Web page.
10. On the IdP sample application's login page, log in with a username managed by OAM.

    Result

    You should be allowed access to OAM-protected Web page.

Release notes

Changelog

OAM Integration Kit 3.1 – April 2018 (Current Release)

- Added support for dynamic target resources: The adapter now uses the target resource contained within the request to determine where the user should be redirected to after successful SSO.
- Bug fix: The adapter will include the value of the Access Manager parameter OAM_REQ in the response if it was present in the initial request (as required by the OAM Server)
- Security update

OAM Integration Kit 3.0 – May 2016

- Initial Release

Known issues and limitations

Known Issues

- If the following error is raised in the server.log, perform the steps listed to resolve the issue:

Copyright © 2025 Ping Identity Corporation

```
ERROR [SystemErr] oracle.security.am.common.nap.ObMessageChannelImpldoSSLHandShake
SEVERE: SSL handshake error sun.security.validator.ValidatorException:
PKIX path validation failed: java.security.cert.CertPathValidatorException:
Algorithm constraints check failed on disabled signature algorithm: MD5withRSA
```

Steps to resolve:

1. Edit `<JAVA_HOME>/jre/lib/security/java.security`.
2. Remove the value `MD5` from the property: `jdk.certpath.disabledAlgorithms`.
3. Remove the value `MD5withRSA` from the property: `jdk.tls.disabledAlgorithms`.
4. Save the file and restart PingFederate.

Known Limitations

- The PingFederate instance, acting as the Service Provider Server and implementing the OAM Integration Kit, must be in the same domain as the OAM Access Server.
- The OAM Access Server may return additional values not specified in the IdP adapter's extended contract. For instance, for an LDAP lookup, the OAM Access Server may send a userDN. In order to perform attribute masking on these additional values, you must add the fields to the extended adapter contract, then and mask the attribute there.
- The OAM adapters are designed for Single Sign-On only. Single Log-Out is not supported in this release.
- Any attribute being mapped to SAML_SUBJECT, such as userId, will not be masked in the server.log file even if it is marked as such. This is because SAML standard already provides facilities for ensuring privacy and confidentiality of the SAML_SUBJECT, and can never be masked by PingFederate.
- Failing to run the configureAccessGate or similar utility in the OAM SDK will result in an error saying that OBACCESS_INSTALL_DIR has not been set. Refer to your OAM SDK documentation for more information.
- Placing the authn_pingfed.dll in the wrong directory will result in the same OBACCESS_INSTALL_DIR error. Be sure to copy authn_pingfed.dll plug-in to the Access Server lib path, not the Access Server SDK lib path.
- The authn_pingfed.dll packaged with the kit has been tested against OAM 6.1 (NetPoint COREid 6.1).
- Any attributes returned by an OAM Action will always be masked in the server.log file as they are being gathered by the adapter.
- Due to the difficulty and complexity of setting up a OAM environment, limited configuration elements were tested with regard to this installation. Please consult your OAM Access Server User Manuals for details on configuring these products.

Download manifest

The distribution `.zip` archive for the OAM Integration Kit contains the following:

- ReadMeFirst.pdf
- /dist - contains libraries needed to run the adapter
  - pf-oam-adapter-3.1.0.jar - OAM Adapter JAR file

Copyright © 2025 Ping Identity Corporation

- mod_pfoam.so - Apache 2.4 Module, compiled on Redhat 6.7
- PingOpenTokenAuthPlugin.jar - OAM Authentication plugin used for SP use case

• /conf - contains libraries needed to run the adapter:

- httpd-pfoam.conf - Sample Apache configuration file for mod_pfoam.so
- jps-config.xml - OAM configuration file

Copyright © 2025 Ping Identity Corporation

One-Time Passcode Integration Kit

The One-Time Passcode Integration Kit allows PingFederate to send one-time passcodes (OTPs) through a variety of notification publishers. The OTPs can be used as a second authentication factor for single sign-on.

Features

- Includes a customizable user interface that allows users to select an OTP delivery method and enter a passcode.
- Supports the PingFederate Authentication API, allowing you to integrate it into your existing web-based sign-on application.
- Supports existing PingFederate notification publishers.

Components

- One-Time Passcode IdP Adapter
  - Allows PingFederate to trigger a notification publisher to send an OTP to a user during sign-on.
  - Accesses user contact information from data store attributes or from earlier in the authentication flow.
  - Validates passcodes entered by users.
- Language packs
  - Allow you to customize the messages shown to the user during authentication.
- Templates
  - Allow you to modify the appearance of pages shown to the user during authentication.
- Twilio Programmable API Notification Publisher
  - Allows you to send OTP messages through SMS and voice with a customizable message body.
- Twilio Verify API Notification Publisher
  - Allows you to send OTP messages through SMS, voice, and email (with the SendGrid addon). This option supports localization but not customizable messages.

Intended audience

This document is intended for PingFederate administrators.

Before starting, you should familiarize yourself with the following sections of the PingFederate documentation:

- Authentication applications and the authentication API
- Managing notification publisher instances
- Managing IdP adapters
- Authentication policies

Copyright © 2025 Ping Identity Corporation

System requirements

- PingFederate 11.3 or later

Overview of one-time passcodes

The One-Time Passcode Integration Kit generates passcodes that the user must enter to complete the authentication flow.

Passcode storage

PingFederate does not store one-time passcodes (OTPs) at any time. Instead, passcodes are generated based on several components that are stored in memory, including a nonce value, a transaction ID, and a secret value associated with the adapter instance.

Passcode duration

Passcodes are associated with a particular PingFederate session. Although the passcodes themselves do not expire, they can only be used once and only for the associated session and authentication flow. When the authentication flow ends, the passcode is no longer valid.

Overview of the Twilio notification publishers

The One-Time Passcode Integration Kit includes two Twilio-based notification publishers. They use two Twilio APIs that offer different features and behavior.

Twilio Programmable API Notification Publisher

- Uses the Twilio Programmable API, a generic messaging API.
- Supports SMS messages and voice calls. Learn more in Programmable SMS and Programmable Voice in the Twilio documentation.
- Allows you to manually customize or localize messages using the templates and message files included in this integration.
- Allows you to redact the last four digits of phone numbers that appear in the Twilio administrator console. Learn more in Message Redaction Supports Phone Number and Content Privacy for Sensitive Applications in the Twilio blog and the Twilio Message Redaction request form.

Twilio Verify API Notification Publisher

- Uses the newer Twilio Verify API, which is designed to send one-time passcodes (OTPs). Learn more in Verify in the Twilio documentation.
- Supports SMS messages, voice calls, and email.
- SMS and voice calls do not support message customization, but they do support automatic localization of the default messages.

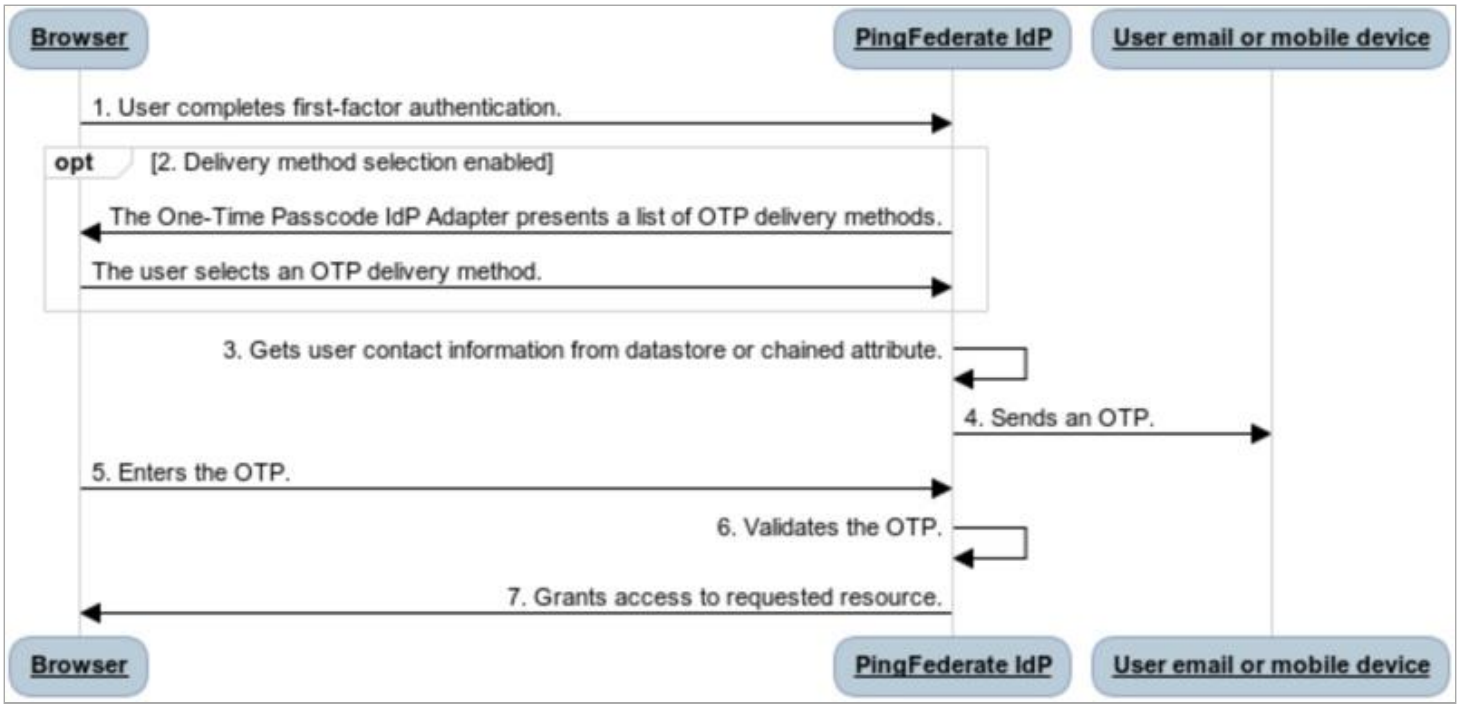
Copyright © 2025 Ping Identity Corporation

- Email support is provided through an integration between Twilio and SendGrid. It allows some customization. Learn more in Send Serverless Emails using SendGrid and Twilio Functions in the Twilio documentation.
- Shows phone numbers in clear text in the Twilio administrator console for 30 days after each transaction. Phone numbers are encrypted in the backend server at all times.

Overview of the SSO flow

With the One-Time Passcode Integration Kit, PingFederate sends a one-time passcode (OTP) to the user as part of the sign-on flow.

The following figure shows how the One-Time Passcode IdP Adapter integrates with the sign-on process:



Description

1. The user initiates single sign-on (SSO) with PingFederate and completes the first-factor authentication step, such as an HTML Form Adapter instance.
2. If the adapter is configured to allow user selection:
    1. The adapter presents a list of OTP delivery methods to the user.
    2. The user selects an OTP delivery method.
3. The adapter gets the user's contact information from a datastore or an attribute passed from earlier in the authentication flow.
4. The adapter sends the user an OTP in an email, SMS message, or automated voice call.
5. The user enters the OTP in the browser.

Copyright © 2025 Ping Identity Corporation

6. The adapter validates the OTP.
7. PingFederate grants access to the requested resource.

PingFederate Authentication API support

You can use the PingFederate Authentication API to integrate the One-Time Passcode IdP Adapter into your application.

The PingFederate Authentication API provides access to the current state of the authentication flow as a user steps through the PingFederate authentication policy. Learn more in PingFederate Authentication API in the PingFederate documentation.

To integrate the One-Time Passcode IdP Adapter into your authentication flow, configure your application based on the information in this section.

> **Tip**
>
> You can also explore the process using the PingFederate Authentication API Explorer. Learn more in Exploring the Authentication API in the PingFederate documentation.

Models, objects, and error codes

When using the One-Time Passcode Integration Kit through the PingFederate Authentication API, the adapter uses the following state models, action models, objects, and error codes.

Status	Request model	Action	Description
`DEVICE_SELECTION_REQUIRED`	**`devices`** The list of devices associated with the user. **`user`** The authenticating user's username. **`userData`** The user info object.	• `selectDevice` • `cancelAuthentication`	Indicates that device selection is required because the user might have more than one device. To continue, the user must select a device for multi-factor authentication (MFA).

Copyright © 2025 Ping Identity Corporation

Status	Request model	Action	Description
`OTP_REQUIRED`	**`devices`** The list of devices associated with the user. **`user`** The authenticating user's username. **`selectedDeviceRef`** The device identifier. **`userData`** The user info object.	• `checkOtp` • `cancelAuthentication` • `selectDevice` • `resendOtp`	Indicates that a one-time passcode (OTP) is required. To continue, the user must enter the OTP sent to them through either SMS, voice call, or email.
`OTP_VERIFIED`	This state has no model.	• `continueAuthentication`	Indicates that the user has completed MFA using an OTP.
`MFA_FAILED`	**`code`** The error code. **`message`** The developer-facing error message. **`userMessage`** The user-facing error message.	• `cancelAuthentication`	Indicates a dead end in the authentication flow. The API client can proceed in the flow by calling `cancelAuthentication`. The adapter returns a `FAILURE` status.

Status	Request model	Action	Description
`selectDevice`	**`deviceRef`** ***(required)*** The device identifier.	• Error: `VALIDATION_ERROR` ErrorDetail: `INVALID_DEVICE`	Starts an authentication flow with the specified `deviceId`. For example: `{ "deviceRef": { "id": "<device ID>" } }`

Copyright © 2025 Ping Identity Corporation

Status	Request model	Action	Description
`checkOtp`	**`otp`** The OTP submitted by the user.	• Error: `VALIDATION_ERROR` ErrorDetail: `INVALID_OTP`	Validates the submitted OTP.
`resendOtp`	This action has no model.	• Error: `REQUEST_FAILED` ErrorDetail: `OTP_RESEND_LIMIT`	Re-sends an OTP to the previously selected device.
`continueAuthentication`	This action has no model.	This action has no errors.	This action continues the current authentication flow.
`cancelAuthentication`	This action has no model.	This action has no errors.	This action cancels the current authentication step.

Objects

Parameter Name	Type	Description
id	String	The unique identifier for this object.
type	String	The device delivery method type. The available options are `SMS`, `VOICE`, and `EMAIL`.
target	String	The device's masked email address or phone number.

Parameter Name	Type	Description
username	String	The user's username that was mapped into the adapter.

userData object

Object with dynamic data populated based on adapter configuration.

Parameter Name	Type	Description
id	String	The resource's identifier.

Copyright © 2025 Ping Identity Corporation

Error codes

An error code is returned if the call flow state hasn't reached a dead end and the user can still authenticate with a device. In cases where a flow reaches a dead end, the `MFA_FAILED` state is returned with a corresponding code.

Error code	Message	HTTP status
`VALIDATION_ERROR`	One or more validation errors occured.	`400`
`REQUEST_FAILED`	The request couldn't be completed. There was an issue processing the request.	`400`

Error code	Message	userMessageKey	Parent code
`INVALID_OTP`	An invalid or expired OTP was provided.	`authn.api.invalid.otp`	`VALIDATION_ERROR`
`OTP_RESEND_LIMIT` **Note** This error code can also be returned by the `MFA_FAILED` state.	The OTP has been re-sent the maximum number of times.	`authn.api.otp.resend.limit`	`REQUEST_FAILED`
`INVALID_DEVICE`	An invalid device was provided.		`VALIDATION_ERROR`

Error code	Message	userMessageKey
`OTP_RESEND_LIMIT` **Note** This error code can also be returned if the call flow state hasn't reached a dead end. To learn more, refer to the previous table, **Detail level error codes**.	The OTP has been re-sent the maximum number of times.	`authn.api.otp.resend.limit`
`INVALID_DEVICE`	An invalid device was provided.	

Setup

Copyright © 2025 Ping Identity Corporation

Deploying the integration files

To get started with the integration, deploy the One-Time Passcode Integration Kit files to your PingFederate directory.

Steps

1. Stop PingFederate.
2. If you are upgrading from a previous deployment, do the following:
    1. Back up any One-Time Passcode Integration Kit files that you customized in `<pf_install>/pingfederate/server/default/conf/`.
    2. Delete `pf-one-time-passcode-adapter-<version>.jar` from `<pf_install>/pingfederate/server/default/deploy`.
    3. Delete `pf-authn-api-sdk-<version>.jar` from `<pf_install>/pingfederate/server/default/lib` if it is older than the version in the integration files.
3. In the integration `.zip` archive, copy the contents of `dist` to `<pf_install>/pingfederate/server/default`.
4. If you want to use the SMTP Notification Publisher, merge the contents of the `pingfederate-smtp-otp-messages.properties` file into the `<pf_install>/pingfederate/server/default/conf/language-packs/pingfederate-email-messages.properties` file.
5. If you backed up templates or messages from a previous deployment, modify the new files with your customizations.
6. Start PingFederate.
7. If you operate PingFederate in a cluster, repeat steps 1 - 6 for each instance of PingFederate.

Configuring a Twilio notification publisher instance

You can configure one of the Twilio notification publishers to allow PingFederate to send one-time passcodes (OTPs) using email, SMS message, or voice call.

About this task

> **Note**
>
> Instead of using a Twilio notification publisher, you can send email notifications directly from PingFederate by completing the steps in Configuring an SMTP Notification Publisher instance in the PingFederate documentation.

Steps

1. Complete the steps in Getting your Twilio API credentials.
2. Sign on to the PingFederate administrative console.
3. On the **System > Notification Publishers** tab, click **Create New Instance**.
4. On the **Type** tab, set the basic notification publisher instance attributes.
    1. In the **Instance Name** field, enter a name for the notification publisher instance.

Copyright © 2025 Ping Identity Corporation

2. In the **Instance ID** field, enter a unique identifier for the notification publisher instance.
3. In the **Type** list, select **Twilio Verify API Notification Publisher** or **Twilio Programmable API Notification Publisher**. Click **Next**.

5. Configure the notification publisher by referring to Twilio Programmable API Notification Publisher settings reference or Twilio Verify API Notification Publisher settings reference. Click **Next**.
6. On the **Actions** tab, click **Test**. Resolve any issues that are reported, and then click **Next**.
7. On the **Summary** tab, check that the configuration is correct. Click **Done**.
8. On the **Notification Publishers** tab, click **Save**.

Getting your Twilio API credentials

To use the Twilio notification publishers, configure your Twilio account and note your credentials.

Steps

1. Sign on to the Twilio Console.
2. Create an account as described in View and Create New Accounts in Twilio Console in the Twilio documentation.
3. Note the **Account SID** and **Auth Token**.

   Learn more in Auth Tokens and How to Change Them in the Twilio documentation.
4. If you want to use the Twilio Verify API Notification Publisher, do the following:
   1. On the Twilio Verify Services page in the Twilio Console, create a service.
   2. Note the **Service ID**.
   3. In the **Code Customization** section, enable **Custom Code**.

Twilio Programmable API Notification Publisher settings reference

Field descriptions for the **Twilio Programmable API Notification Publisher** configuration page.

Field	Description
Account SID	The **Account SID** that you noted in Getting your Twilio API credentials.
Auth Token	The **Auth Token** that you noted in Getting your Twilio API credentials.

Copyright © 2025 Ping Identity Corporation

Field	Description
From Number **Note** This field is required if you select **Voice** as the **Delivery Method**.	The Twilio phone number, sender ID, or Wireless SIM that sends the message to the user. This value appears as the sender. Phone numbers must be in E.164 format. Learn more in E.164 in the Twilio documentation. **Note** Alphanumeric sender IDs are not available in all countries. You can find a list of supported countries in International support for Alphanumeric Sender ID in the Twilio Help Center.
Messaging Service ID	The Twilio Messaging Service SID.
Delivery Method	The method used to send the one-time passcode (OTP). The default value is **SMS**. **Note** If the **Delivery Method** is **SMS**, you must configure a value for at least one of the following fields: • **From Number** • **Messaging Service ID**
Block VoIP Numbers	When selected, the Twilio Lookup API is used to ensure SMS messages are not sent to VoIP phone numbers. This is a paid service. Learn more in Lookup in the Twilio documentation. This checkbox is cleared by default.
Blocked Numbers	OTPs are not sent to phone numbers that match these values. Separate multiple values with a comma. The default values are the following toll-free numbers: `+1800,+1888,+1877,+1866,+1855,+1844,+1833`.
Test Phone	The phone number used to test the configuration on the **Actions** tab.

Field	Description
Messages File	The name of the messages file to use for notifications. The adapter uses the corresponding `.properties` file in `<pf_install>/pingfederate/server/default/conf/language-packs`. The default value is `twilio-programmable-api-publisher-messages`.

Copyright © 2025 Ping Identity Corporation

Field	Description
Voice Option	The Twilio voice to use for voice calls. Select Twilio's basic **Man** or **Woman** voice, or select **Other** to specify one of the more advanced voices. You can find a list of languages and locales supported by each voice in What Languages can the <Say> TwiML Verb Speak? in the Twilio Help Center. The default setting is **Woman**.
Other Voice Option	If you selected a **Voice Option** of **Other**, specify the specific voice to use. For example, `alice` or `Polly.Joanna`. You can find a list of voice names in What Languages can the <Say> TwiML Verb Speak? in the Twilio Help Center. This field is blank by default.
API Request Timeout	The amount of time in milliseconds that PingFederate allows when establishing a connection with the Twilio API or waiting for a response to a request. A value of `0` disables the timeout. The default value is `5000`.
Connection Timeout	The amount of time in milliseconds that PingFederate allows to establish a connection with the Twilio API. A value of `0` disables the timeout. The default value is `5000`.
Proxy Settings	Defines proxy settings for outbound HTTP requests. The default value is **System Defaults**.
Custom Proxy Host	The proxy server host name to use when **Proxy Settings** is set to **Custom**. This field is blank by default.
Custom Proxy Port	The proxy server port to use when **Proxy Settings** is set to **Custom**. This field is blank by default.

Twilio Verify API Notification Publisher settings reference

Field descriptions for the **Twilio Verify API Notification Publisher** configuration page.

Field	Description
Account SID	The **Account SID** that you noted in Getting your Twilio API credentials.
Auth Token	The **Auth Token** that you noted in Getting your Twilio API credentials.
Service ID	The **Service ID** that you noted in Getting your Twilio API credentials.
Delivery Method	The method used to send the one-time passcode (OTP).

Copyright © 2025 Ping Identity Corporation

Field	Description
Block VoIP Numbers	When selected, the Twilio Lookup API is used to ensure SMS messages are not sent to VoIP phone numbers. This is a paid service. Learn more in Lookup in the Twilio documentation. This checkbox is cleared by default.
Blocked Numbers	OTPs are not sent to phone numbers that match these values. Separate multiple values with a comma. The default values are the following toll-free numbers: `+1800,+1888,+1877,+1866,+1855,+1844,+1833`.
Test Mail	The email address used to test the configuration on the **Actions** tab.
Test Phone	The phone number used to test the configuration on the **Actions** tab.

Field	Description
API Request Timeout	The amount of time in milliseconds that PingFederate waits for the Twilio API to respond to requests. A value of `0` disables the timeout. A negative value uses the PingFederate system defaults. The default value is `5000`.
Connection Timeout	The amount of time in milliseconds that PingFederate allows to establish a connection with the Twilio API. A value of `0` disables the timeout. A negative value uses the PingFederate system defaults. The default value is `5000`.
Proxy Settings	The proxy settings to use for outbound HTTP requests originating from the notification publisher. This setting allows you to override the PingFederate system-default proxy settings. The default selection is **System Defaults**.
Customer Proxy Host	The proxy server hostname to use when **Proxy Settings** is set to **Custom**. This field is blank by default.
Custom Proxy Port	The proxy server port to use when **Proxy Settings** is set to **Custom**. This field is blank by default.

Configuring an adapter instance

Configure the One-Time Passcode IdP Adapter to allow PingFederate to use notification publishers to send one-time passcodes (OTPs) to users during sign-on.

Steps

1. Sign on to the PingFederate administrative console.

Copyright © 2025 Ping Identity Corporation

2. On the **Identity Provider > Adapters** tab, click **Create New Instance**.
3. On the **Type** tab, set the basic adapter instance attributes.
    1. In the **Instance Name** field, enter a name for the adapter instance.
    2. In the **Instance ID** field, enter a unique identifier for the adapter instance.
    3. In the **Type** list, select **One-Time Passcode IdP Adapter**. Click **Next**.
4. On the **IdP Adapter** tab, in the **Notification Delivery Methods** section, configure a list of delivery methods by assigning contact information and message templates to your notification publishers.
    1. Click **Add a new row to 'Notification Delivery Methods'**.
    2. In the **Notification Publisher** list, select the notification publisher that you want to use.
    3. In the **Contact Attribute** list, enter the attribute that contains the user's contact information for this delivery method. For example, `email`, or `phone`.
    4. In the **Language Properties and Templates Key** field, enter a key. You can find an explanation of keys in Language property and template keys.
    5. In the **Action** column, click **Update**.
    6. To add more delivery methods, repeat steps a - e.
    7. Change the order of the delivery methods to define the sequence that the adapter attempts the deliveries when **Device Selection** is set to **Automatic**.
5. On the **IdP Adapter** tab, in the **User Attribute Mapping** section, configure user data mapping from the chained attribute to the target attribute.
    1. Click **Add a new row to 'User Attribute Mapping'**.
    2. In the **Source User Attribute** field, enter the name of the source attribute that contains the user information.
    3. In the **Target User Attribute** field, enter the name of the target attribute to use in templates and authentication API responses.
    4. In the **Action** column, click **Update**.
    5. To add more user information attributes, repeat steps a - d.

       Example:

Chained Attribute	Target Attribute
firstName	first_name
lastName	last_name
title	title

Copyright © 2025 Ping Identity Corporation

If the chained attributes contain acceptable values for the `firstName`, `lastName`, and `title` attributes, the One-Time Passcode IdP Adapter assigns the values to `first_name`, `last_name`, and `title`.

In the templates, these attributes are accessible using the following velocity template variables:

- $userData.first_name
- $userData.last_name
- $userData.title

In the Authentication API `DEVICE_SELECTION_REQUIRED` and `OTP_REQUIRED` response model states, the `userData` object contains the mapped attributes.

6. Configure the adapter instance by referring to One-Time Passcode IdP Adapter settings reference. Click **Next**.
7. On the **Actions** tab, test your connection to the database. Resolve any issues that are reported, and then click **Next**.
8. On the **Extended Contract** tab, add any attributes that you expect to retrieve in addition to the core contract attributes. Click **Next**.
9. Complete the adapter configuration.
10. On the **Summary** tab, check that the configuration is correct. Click **Done**.
11. On the **Manage IdP Adapter Instances** tab, click **Save**.
12. Modify your PingFederate authentication policy to include your One-Time Passcode IdP Adapter instance. Position it after a first-factor authentication step, such as the HTML Form IdP Adapter.

    You can find general information about configuring authentication policies in Policies in the PingFederate documentation.

Language property and template keys

The **Language Property and Template Keys** that you define in the One-Time Passcode IdP Adapter instance configuration affect the messages and prompts shown to end users.

The adapter uses these keys in different ways depending on how the adapter is accessed and which notification publisher it interacts with.

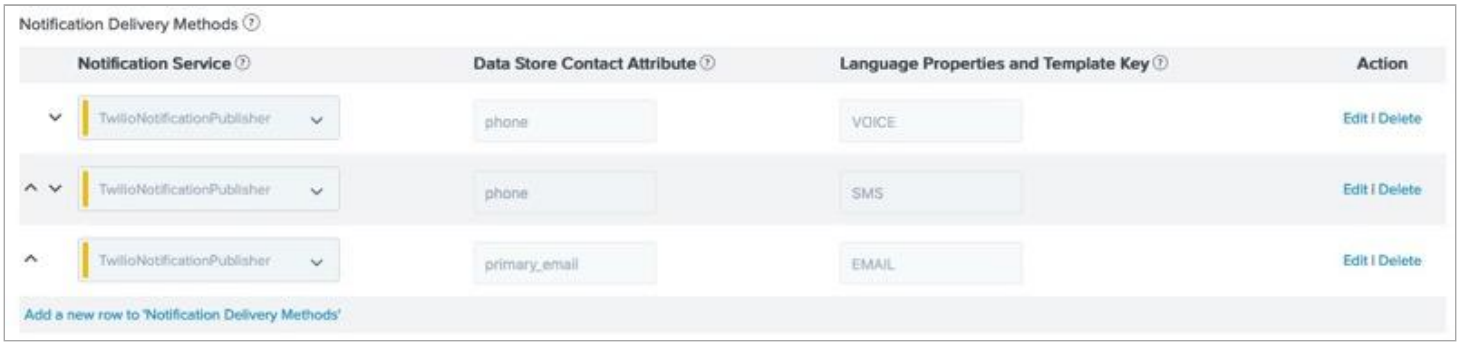
Using the PingFederate authentication API

When the PingFederate authentication API provides your web application with a list of notification delivery methods, it includes the **Language Property and Template Keys** value as the device type.

By having your web application read the device type, you can set different icons or formatting for each delivery method.

In the example below, the authentication API is prompting the web application to select a delivery method. The methods are defined with these three keys: `VOICE`, `SMS`, and `EMAIL`.

Copyright © 2025 Ping Identity Corporation




```
{
  "id": "G1j8b",
  "pluginTypeId": "7r5wldvoQS8iEJEpdMYqmA",
  "status": "DEVICE_SELECTION_REQUIRED",
  "devices": [
    {
      "id": "7337a6ac-98b2-4c02-9afd-82aa9f6b55c7",
      "type": "VOICE",
      "target": "+155*\**55"
    },
    {
      "id": "2ac3e6a8-5b3e-4247-b7d8-4d12cec8b9da",
      "type": "SMS",
      "target": "+155**\*55"
    },
    {
      "id": "b6cbaeff-bea2-4cfd-a0fd-3ffe6907df39",
      "type": "EMAIL",
      "target": "an****@example.com"
    }
  ],
  "user": {
    "username": "anonymous"
  },
  "_links": {
    "cancelAuthentication": {
      "href": "https://pf-host:9031/pf-ws/authn/flows/G1j8b"
    },
    "selectDevice": {
      "href": "https://pf-host:9031/pf-ws/authn/flows/G1j8b"
    },
    "self": {
      "href": "https://pf-host:9031/pf-ws/authn/flows/G1j8b"
    }
  }
}
```


User interface

The template files included in the integration kit show information about your OTP delivery methods. To do this, they use lines from the `pingfederate-otp-adapter-messages.properties` file that begin with the **Language Properties and Templates Key** that you define for each delivery method.

For example, you configure a delivery method with a **Language Properties and Templates Key** value of `sms` . When asking the user to enter a passcode for that delivery method, the adapter uses the following lines to generate the page:

Copyright © 2025 Ping Identity Corporation

```
sms.otp.adapter.otp.required.template.header=SMS Authentication
sms.otp.adapter.otp.required.template.message=Enter the passcode you received to continue.
sms.otp.adapter.otp.required.template.otp.sent.text=SMS sent to:
sms.otp.adapter.otp.required.template.resend.button.title=Resend
```

By default, the `pingfederate-otp-adapter-messages.properties` file has messages defined for the following keys:

- `email`
- `sms`
- `voice`

You can modify the existing messages, or add your own using a new prefix.

> **Note**
>
> When using the SMTP Notification Publisher or the Twilio Programmable API Notification Publisher, the key determines the template file that forms the message body.
> `.otp.adapter.template.html` is a file name suffix appended to the prefix key you configured in the **Language Properties and Templates Key** field.

SMTP Notification Publisher

When using the SMTP Notification Publisher, the key determines the template file that forms the body of the email message.

For example, if you enter `pingfederate.smtp` in the **Language Properties and Templates Key** field, the adapter will use the `<pf_install>/pingfederate/server/default/conf/template/mail-notifications/pingfederate.smtp.otp.adapter.template.html` file as the body of the email message.

If you are also using the user interface, the adapter will check the `pingfederate-otp-adapter-messages.properties` file for message lines that begin with `pingfederate.smtp`. Modify the file accordingly.

Twilio Programmable API Notification Publisher

When using the Twilio Programmable API Notification Publisher, the key determines the template file that forms the body of the notification SMS or voice call.

For example, if you enter `twilio.sms` in the **Language Properties and Templates Key** field, the adapter will use the `<pf_install>/pingfederate/server/default/conf/template/twilio.sms.otp.adapter.template.html` file as the body of the SMS message.

If you are also using the user interface, the adapter will check the `pingfederate-otp-adapter-messages.properties` file for message lines that begin with `twilio.sms`. Modify the file accordingly.

One-Time Passcode IdP Adapter settings reference

Field descriptions for the **One-Time Passcode IdP Adapter** configuration page.

Copyright © 2025 Ping Identity Corporation

Field	Description
Device Selection	***Automatic*** The adapter uses the method from the **Preferred Delivery Method Attribute** or uses the first delivery method that it matches in the **Notification Delivery Methods** list. ***User Choice*** The adapter prompts the user to choose the delivery method.
Preferred Delivery Method Attribute	The source attribute that contains the user's preferred one-time passcode (OTP) delivery method. If the user has a valid preferred delivery method attribute, it overrides the **Automatic** and **User choice** options above. For example, you enter `OTPPreference` in this field. You also create a user attribute called `OTPPreference` in your datastore or pass it to this adapter as a chained attribute. When Alice signs on, the adapter checks her `OTPPreference` attribute. The value is `sms`, which matches one of the **Language Properties and Template Key** entries in the **Notification Delivery Methods** table. The adapter automatically sends the OTP to Alice by SMS message.
Attribute Source	The source of the attribute in the **Preferred Delivery Method Attribute** field and the attributes listed in the **Contact Attribute** column of the **Notification Delivery Methods** table. Select a datastore, or select **Chained Attributes** if the adapter receives the attributes from earlier in the authentication flow.
Search String	The string that the adapter uses to search the datastore to find the user. • For JDBC, enter a "select" statement. For example, `select email, phone from <db.table> where username=${userid}`. • For LDAP, enter an LDAP filter. For example, `sAMAccountName=${userid}`. • For a PingOne datastore, enter the attribute. For example, `username=${userid}` or `id=${userid}`s. • For REST API datastores, enter the resource path appended to the base URL of the REST API datastore. For example, `/users?uid=${userid}`. The `${userid}` variable contains the user ID. Your adapter instance receives this from earlier in your PingFederate authentication flow.
Base DN	The base DN that the adapter uses when connecting to an LDAP datastore.
Test User ID	The user ID used to test the configuration on the **Actions** tab.
Failure Mode	This setting determines whether the adapter should block the user's sign-on attempt or bypass the OTP requirement when the adapter can't find the user or contact information in the datastore or chained attributes.

Copyright © 2025 Ping Identity Corporation

Field	Description
HTML Template Prefix	A file prefix identifying the customizable HTML templates that the adapter instance uses. The default value is: `otp.adapter.` **Important** You must keep the following files in the `<pf_home>/server/default/conf/template` folder: • `<HTML Template Prefix>device.selection.template.html` • `<HTML Template Prefix>otp.verified.template.html` • `<HTML Template Prefix>otp.required.template.html` • `<HTML Template Prefix>mfa.failed.template.html`
Visible Leading Characters Length	The length of the leading characters to display for email and phone number formats that don't pass the built-in validation check.
Visible Trailing Characters Length	The length of the trailing characters to display for email and phone number formats that don't pass the built-in validation check.
OTP Length	Length of the OTP generated by the adapter. The default value is `6`.
Max OTP Attempts	The maximum number of times the user is allowed to try entering the OTP before authentication fails. The default value is `3`.
Max OTP Resends	The maximum number of times the user is allowed to request a specific OTP to be sent. After reaching this limit, the `Resend` button on the passcode entry prompt no longer resends the passcode. The default value is `15`.
Show Success Screens	Determines whether the adapter shows an authentication success screen to the user. This checkbox is selected by default.
Show Error Screens	Determines whether the adapter shows an authentication error screen to the user. This checkbox is selected by default.
OTP Generator Field	A read-only value used by the adapter. **Caution** Do not edit this field. This field is hidden in PingFederate 10.0 and later.

Copyright © 2025 Ping Identity Corporation

Field	Description
LDAP Search Scope	When the attribute source is an LDAP datastore, this setting determines the scope of the user search. ***Single Level*** Searches the immediate children of the base object, but excludes the base object itself. ***Include Subtree (default)*** Searches all child objects as well as the base object.

Customizing templates

The One-Time Passcode Integration Kit includes templates that can present one-time passcode (OTP) authentication challenges and other messages.

- You can customize the HTML portions of these templates to suit your organization's branding and presentation needs.
- You can use the **HTML Prefix Field** to configure adapter instance-specific templates. Learn more in the One-Time Passcode IdP Adapter settings reference.
- You can access the user information attributes configured in the **User Attribute Mapping** table during the Configuring an adapter instance procedure in the templates using the `$userData.<target_attribute_name>` velocity variable.

The following table lists the available templates and their uses. You can find more information about templates in Customizable user-facing pages in the PingFederate documentation.

Template files included in the One-Time Passcode Integration Kit

Template file	Description
`otp.adapter.device.selection.template.html`	Presents the user with a form to select a device on which to receive the OTP.
`otp.adapter.mfa.failed.template.html`	Presents the user with a form to indicate failed verification of the OTP they entered previously.
`otp.adapter.otp.required.template.html`	Presents the user with a form to input the OTP.
`otp.adapter.otp.verified.template.html`	Presents the user with a form to indicate successful verification of the OTP they entered previously.

Release notes

Copyright © 2025 Ping Identity Corporation

Changelog

The following is the change history for the One-Time Passcode Integration Kit.

One-Time Passcode Integration Kit 1.1 - April 2025

- Added a default masking mechanism for arbitrary emails and phone number formats that fail the built-in validation.

  Learn more in the **Visible Leading Characters Length** and **Visible Trailing Characters Length** table entries in the One-Time Passcode IdP Adapter settings reference.

- Added the ability to customize velocity templates for specific adapter instances.

  Learn more in the **HTML Template Prefix** table entry in the One-Time Passcode IdP Adapter settings reference.

- Added the ability to pass dynamic user information to velocity templates.

  Learn more in step 5 of the Configuring an adapter instance procedure.

- Added the ability to use a **Messaging Service SID** with the Twilio Programmable API Notification Publisher.

  Learn more in the **Messaging Service SID** table entry in the Twilio Programmable API Notification Publisher settings reference.

- Upgraded the Twilio SDK jar used in the Twilio Programmable API Notification Publisher and the Twilio Verify API Notification Publisher.

One-Time Passcode Integration Kit 1.0.2 — August 2023

- Fixed an issue that prevented the One-Time Passcode IdP Adapter from creating LDAPS datastore connections.
- Fixed an issue with an invalid URL in the adapter CSS.
- Upgraded the Apache Commons Text library to version 1.10.0 to follow best practices for CVE-2022-42889.

> **Note**
>
> The One-Time Passcode IdP Adapter was not vulnerable to this issue.

One-Time Passcode Integration Kit 1.0.1 — June 2021

- Fixed an issue that caused an error when a PingFederate node loaded the adapter while a user was authenticating.

One-Time Passcode Integration Kit 1.0 — June 2020

- Initial release.
- Added the ability to use the adapter with a UI or through the PingFederate authentication API.
- Added the ability to look up user device details in a data store attached to PingFederate.
- Added the ability to use chained attributes from earlier in the authentication flow.
- Added the ability to allow users to choose their delivery method.
- Added the ability to stop or continue authentication when the data store lookup fails.

Copyright © 2025 Ping Identity Corporation

- Added the ability to send one-time passcodes through PingFederate notification publishers.
- Added the Twilio Verify API Notification Publisher.
- Added the Twilio Programmable API Notification Publisher.

Known issues and limitations

The following are known issues or limitations for the One-Time Passcode Integration Kit.

Known issues

There are no known issues.

Known limitations

- When PingFederate is operating in standalone mode, changes you make to an existing notification publisher configuration don't take effect until you save the corresponding adapter instance again.
- Passcodes are associated with a particular PingFederate session, which causes the following known limitations:
    - The one-time passcode (OTP) timeout value cannot be controlled. The OTP won't expire as long as it isn't used or the PingFederate session isn't expired.
    - You can't refresh the OTP value using resend request.
- This integration kit is intentionally designed as a simpler version of multi-factor authentication (MFA) without full features and will not add any new feature in the future.
    - For full-featured MFA operation, PingOne MFA or PingID is recommended.

Download manifest

The following files are included in the One-Time Passcode Integration Kit:

- `ReadMeFirst.pdf` : Contains links to this online documentation.
- `legal` : A directory that contains this document:
    - `Legal.pdf` : Copyright and license information.
- `dist` : A directory that contains integration files.
    - `conf` : A directory that contains configuration files.
        - `language-packs` : A directory that contains language template files.
            - `pingfederate-otp-adapter-messages.properties` : A variable file that customizes the messages that appear in template files.
            - `pingfederate-smtp-otp-messages.properties` : A variable file that customizes the messages that appear on the SMTP template file.

Copyright © 2025 Ping Identity Corporation

- `twilio-programmable-api-publisher-messages.properties` : A variable file that customizes the messages that appear on the Twilio SMS template file.

- `template` : A directory that contains language template files.
  - `assets` : A directory that contains style, font, and image files for the template files.
    - `css/otp-adapter.css` : A CSS file that customizes the appearance of the template file.
    - `fonts/end-user` : Contains fonts used in the template file.
      - `icons` : Contains an icon font used in the template file.
      - `proxima-nova` : Contains a text font used in the template file.
    - `images` : Contains sample logos and delivery method icons used in the template file.
  - `mail-notifications` : A directory that contains email template files.
    - `pingfederate.smtp.otp.adapter.template.html` : A sample template file that the SMTP Notification Publisher can use to send a one-time passcode in an email.
  - `otp.adapter.device.selection.template.html` : A sample template file that allows the user to select a delivery method for the passcode.
  - `otp.adapter.mfa.failed.template.html` : A sample template file that shows a message when the passcode fails.
  - `otp.adapter.otp.required.template.html` : A sample template file that allows a user to enter a passcode.
  - `otp.adapter.otp.verified.template.html` : A sample template file that shows a message when the passcode succeeds.
  - `twilio.sms.otp.adapter.template.html` : A sample template file that the Twilio Programmable API Notification Publisher can use to send a one-time passcode in an SMS message.
  - `twilio.voice.otp.adapter.template.html` : A sample template file that the Twilio Programmable API Notification Publisher can use to send a one-time passcode in a voice message.

- `deploy` : A directory that contains the Java libraries needed to run the adapter and Twilio notification publishers.
  - `pf-one-time-passcode-adapter-<version>.jar` : A JAR file that contains the One-Time Passcode IdP Adapter.
  - `twilio-programmable-api-notification-plugin-<version>.jar` : A JAR file that contains the Twilio Programmable API Notification Publisher.
  - `twilio-verify-api-notification-plugin-<version>.jar` : A JAR file that contains the Twilio Verify API Notification Publisher.
- `lib` : A directory that contains PingFederate core library files.
  - `pf-authn-api-sdk-<version>.jar` : A JAR file that contains the PingFederate Authentication API SDK.

Copyright © 2025 Ping Identity Corporation

PHP Integration Kit

The PHP Integration Kit allows PHP applications to integrate with a PingFederate server acting as either an identity provider (IdP) or service provider (SP).

> **Note**
>
> Updates and support have ended for this integration, because this integration relies on the "mcrypt" extension, which was deprecated after PHP 7.1. The availability of this user guide and the related product download is intended only for those who have existing solutions using this integration.

> **Tip**
>
> For new integrations, we encourage you to use the Agentless Integration Kit, which offers samesite cookie support and can integrate with a variety of platforms using a modern RESTful approach.

Components

- OpenToken adapter and agent
  - Allows developers to integrate their PHP applications with a PingFederate server acting as either an identity provider (IdP) or a service provider (SP). The integration kit allows an IdP server to receive user attributes from a PHP IdP application. On the SP side, the integration kit allows a PHP SP application to receive user attributes from the SP server.
- Sample applications
  - The Integration Kit distribution also contains sample IdP and SP applications. The applications may be installed quickly for testing OpenToken processing and to provide a working demonstration of end-to-end single sign-on (SSO) and single logout (SLO). Source code and supporting files are included in the distribution and may be modified or used as a reference for application developers.

Intended audience

This document is intended for PingFederate administrators and web application developers.

Before starting, we recommend that you familiarize yourself with the following sections of the PingFederate documentation:

- Identity provider SSO configuration
- Service provider SSO configuration
- OpenToken Adapter

System requirements

- PingFederate 8.0 or later.
- A web server running PHP 5.4.8 to 7.1.

Copyright © 2025 Ping Identity Corporation

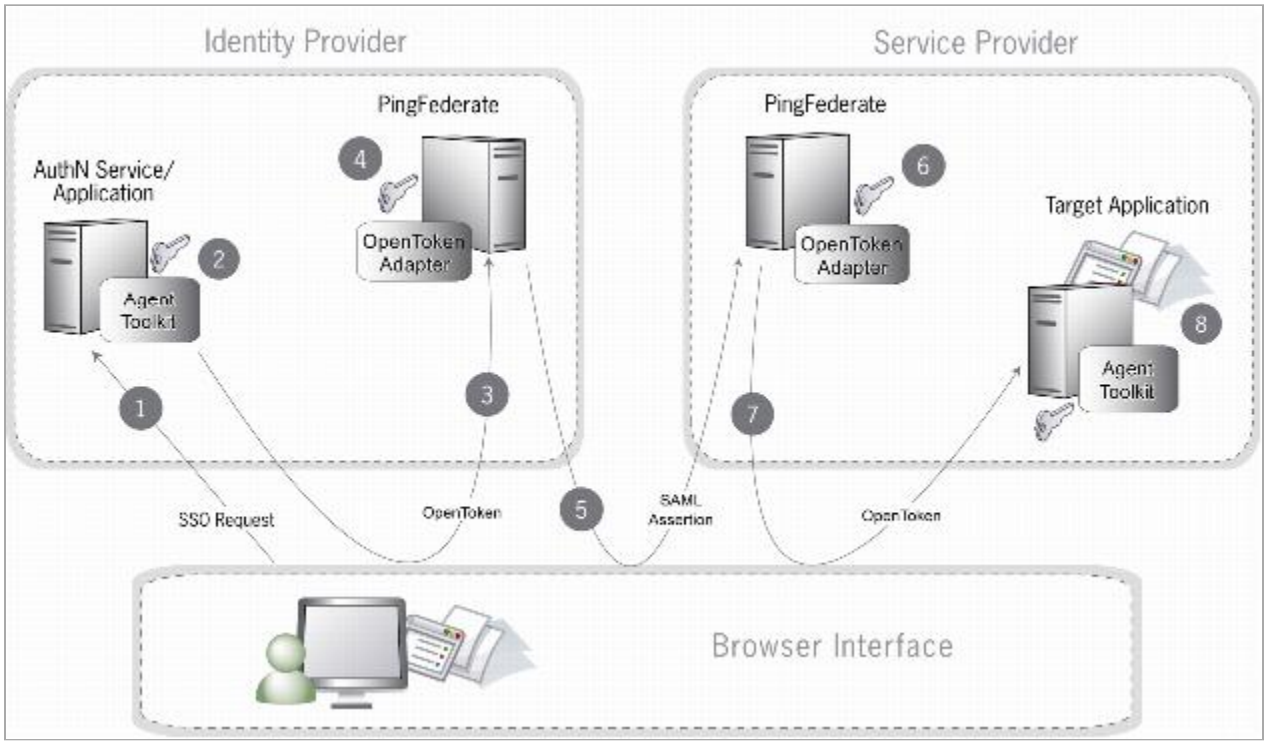
- The following extensions must be available on the PHP server:
  - `cURL` – enables IdP and SP applications to communicate with PingFederate. This is required for both the Windows and Linux installations of PHP 5.4.8.
  - `mcrypt` – used to encrypt and decrypt the OpenToken. Information and download: https://php.net/mcrypt and http://mcrypt.sourceforge.net.
  - `mhash` – used to generate `hmacs` and secure keys from passwords. Information and download: https://php.net/mhash and http://mhash.sourceforge.net.
  - `zlib` – used for data compression. The PHP kit uses the `gzuncompress` method to uncompress data. Information and download: https://php.net/manual/en/ref.zlib.php.

Overview of the SSO flow

The PHP Integration Kit consists of two parts:

- The OpenToken Adapter, which runs within the PingFederate server
- The Agent Toolkit for PHP, which resides within the PHP user application

The following figure shows a basic IdP-initiated single sign-on (SSO) scenario in which PingFederate federation servers using the PHP Integration Kit exist on both sides of the identity federation:



Processing Steps

1. A user initiates an SSO transaction.

Copyright © 2025 Ping Identity Corporation

2. The IdP application inserts user attributes into the Agent Toolkit for PHP, which encrypts the data internally and generates an `OpenToken`.
3. A request containing the `OpenToken` is redirected to the PingFederate IdP server.
4. The server invokes the OpenToken IdP Adapter, which retrieves the `OpenToken`, decrypts, parses, and passes the user attributes to the PingFederate IdP server. The PingFederate IdP server then generates a Security Assertion Markup Language (SAML) assertion.
5. The SAML assertion is sent to the SP site.
6. The PingFederate SP server parses the SAML assertion and passes the user attributes to the OpenToken SP Adapter. The Adapter encrypts the data internally and generates an `OpenToken`.
7. A request containing the OpenToken is redirected to the SP application.
8. The Agent Toolkit for PHP decrypts and parses the OpenToken and makes the user attributes available to the SP Application.

> **Note**
>
> PingFederate can be configured to look up additional attributes from either an IdP or SP data store. For more information, see Datastores in the PingFederate documentation.

Setup

Updating the OpenToken Adapter

The PHP Integration Kit relies on the OpenToken Adapter that is distributed with the Java Integration Kit. Update the OpenToken Adapter to get the latest feature and security updates.

Steps

1. Download the Java Integration Kit `.zip` archive from the **Add-ons** tab of the PingFederate downloads page.
2. Stop PingFederate.
3. Delete the `opentoken-adapter-<version>.jar` file from your `<pf_install>/pingfederate/server/default/deploy` directory.
4. From the Java Integration Kit `.zip` archive, copy the contents of `dist/pingfederate` to your `<pf_install>/pingfederate` directory.

   > **Note**
   >
   > The `commons-collections`, `commons-beanutils`, and `commons-logging` libraries are provided as a convenience and should be installed only if they are not already contained in the application `CLASSPATH`.

5. Start PingFederate.
6. If you operate PingFederate in a cluster, repeat steps 2-5 for each engine node.

Copyright © 2025 Ping Identity Corporation

Custom application setup

You can configure theproductto integrate PingFederate with your identity provider (IdP) or service provider (SP) application.

If you would like to see a working demonstration of theproductbefore integrating your own applications, see Sample application setup instead.

Configuring an OpenToken Adapter instance

Configure the OpenToken Adapter to determine how PingFederate communicates with your application.

About this task

These steps are for creating a service provider (SP) adapter instance. You can complete the equivalent steps to create an identity provider (IdP) adapter instance.

Steps

1. Sign on to the PingFederate administrative console.
2. On the **Service Provider > Adapters** screen, click **Create New Instance**.
3. On the **Type** screen, set the basic adapter instance attributes.
    1. In the **Instance Name** field, type a name for the adapter instance.
    2. In the **Instance ID** field, type a unique identifier for the adapter instance.
    3. On the **Type** list, select **OpenToken Adapter**, and then click **Next**.
4. On the **Instance Configuration** screen, configure the adapter instance by referring to Configuring an OpenToken SP Adapter instance in the PingFederate documentation. Click **Next**.
    - In your adapter instance configuration, in the **Advanced Fields** section, clear the **Obfuscate Password** check box. The PHP agent does not support encrypted passwords. The password is Base64 encoded.
5. Export the configuration file.
    1. On the **Actions** screen, click **Download**, and then click **Export**.
    2. Save `agent-config.txt`. Click **Next**.
6. On the **Extended Contract** screen, add any attributes that you expect to retrieve in addition to core contract attributes. Click **Next**.
7. On the **Summary** screen, check that the configuration is correct, and then click **Done**.
8. On the **Manage SP Adapter Instances** screen, click **Save**.
9. Create or update an identity provider (IdP) connection to use the OpenToken Adapter instance. See Service provider SSO configuration in the PingFederate documentation.

Copyright © 2025 Ping Identity Corporation

Configuring your PHP server

To allow your PHP server to work with the PHP Integration Kit, check that the correct extensions are installed and configure your agent config file.

Steps

1. On your PHP server, use the phpinfo() function to check that the cURL, mcrypt, zlib, and mhash extensions are installed.
2. Check that your PHP server is configured to process files with a .php extension.

   On the Apache HTTPD server, you can use the AddType command to add the extension:

   ```
   AddType application/x-httpd-php .php
   ```

   This step is not needed on the built-in PHP web server.
3. From the integration `.zip` archive, copy the `dist/pingidentity/opentoken` directory to your PHP `include` path as `/opentoken`.
4. Change `AGENT_CONFIG_FILE` in the `pingidentity/opentoken/helpers/config.php` file to point to the location of the `agent-config.txt` file that you saved in Configuring an OpenToken Adapter instance.

   On Windows, use the network-path syntax:

   ```
   \\<host>\<path>
   ```
5. To avoid internationalization compatibility issues, enforce UTF-8 encoding by explicitly specify `default_charset=utf8` in your PHP server's `php.ini` file.

IdP single sign-on integration

When PingFederate is configured as an IdP, it needs to be able to identify a user prior to issuing a SAML assertion for that user. When using the OpenToken Adapter with PingFederate, this means that the PingFederate server attempts to read a cookie or query parameter containing an OpenToken and then use the values within to identify the user. The application that starts the SSO must include an OpenToken so that PingFederate can identify the user. Use the Agent API to write an OpenToken. The Agent API is a PHP object that provides access to functionality for writing an OpenToken to a given HTTP response.

The Agent Toolkit for PHP makes use of Namespaces, which enable the Agent API to be auto-loaded with other applications. For more information on auto-loading, see Autoloading Classes in the PHP documentation.

> **Note**
>
> The sample code in this guide assumes that the Agent APIs are already loaded.

The following is sample code for auto-loading. Other auto-loading frameworks such as Zend can be used instead.

```
spl_autoload_extensions(".php");
spl_autoload_register();
```

Instantiating the agent object is done simply by invoking a constructor, as in the example below:

Copyright © 2025 Ping Identity Corporation

```
<?php
   use pingidentity\opentoken\agent;
   $myagent = new Agent();
?>
```

When the agent object is instantiated, it uses the `config.php` file to find the configuration data generated when the OpenToken Adapter was configured. This configuration data includes the name of the cookie that the agent object will write, as well as the key to use when encrypting a new OpenToken. If the file specified in `config.php` is not found, the agent constructor will throw an exception.

The writeTokenToHTTPResponse method takes an array of attributes and encodes them into an OpenToken, which is then written to the HTTP response.

> **Note**
>
> The array of attributes parameter must contain a key named “subject” in order for a valid token to be generated.

If any errors are encountered while creating the token or writing it out to the response, the `lastError` attribute of the agent instance will contain a message with a description of the error.

Sample code

The code snippet below demonstrates the use of the `writeTokenToHTTPResponse` method:

```
<?php
   # Use and instantiate an agent
   use pingidentity\opentoken\agent;
   $myagent = new Agent();

   # Setup an array of values
   $myvalues = array(TOKEN_SUBJECT => “$userId”);

   # Generate and write the token to a cookie (assuming that's
   # our configuration)
   $myagent->writeTokenToHTTPResponse($myvalues);
?>
```

Passing multi-value attributes

The Agent Toolkit for PHP supports passing multi-value attributes to PingFederate that will each appear in its own discrete `<AttributeValue>` element in the SAML 2.0 assertion or as a JSON array value in OAuth-based protocols. Multi-value attributes are passed using the `MultiStringArray` PHP class distributed with the Agent Toolkit for PHP.

The following code snippet demonstrates how to pass multi-valued attributes using the Agent Toolkit for PHP:

Copyright © 2025 Ping Identity Corporation

```
<?php
   # Use and instantiate an agent
   use pingidentity\opentoken\agent;

use pingidentity\opentoken\helpers\multistringarray;
   $myagent = new Agent();

   # Setup an array of values
   $myvalues = MultiStringArray()

   # Single Value Attribute
   $myvalues->add(TOKEN_SUBJECT, <$userId>);

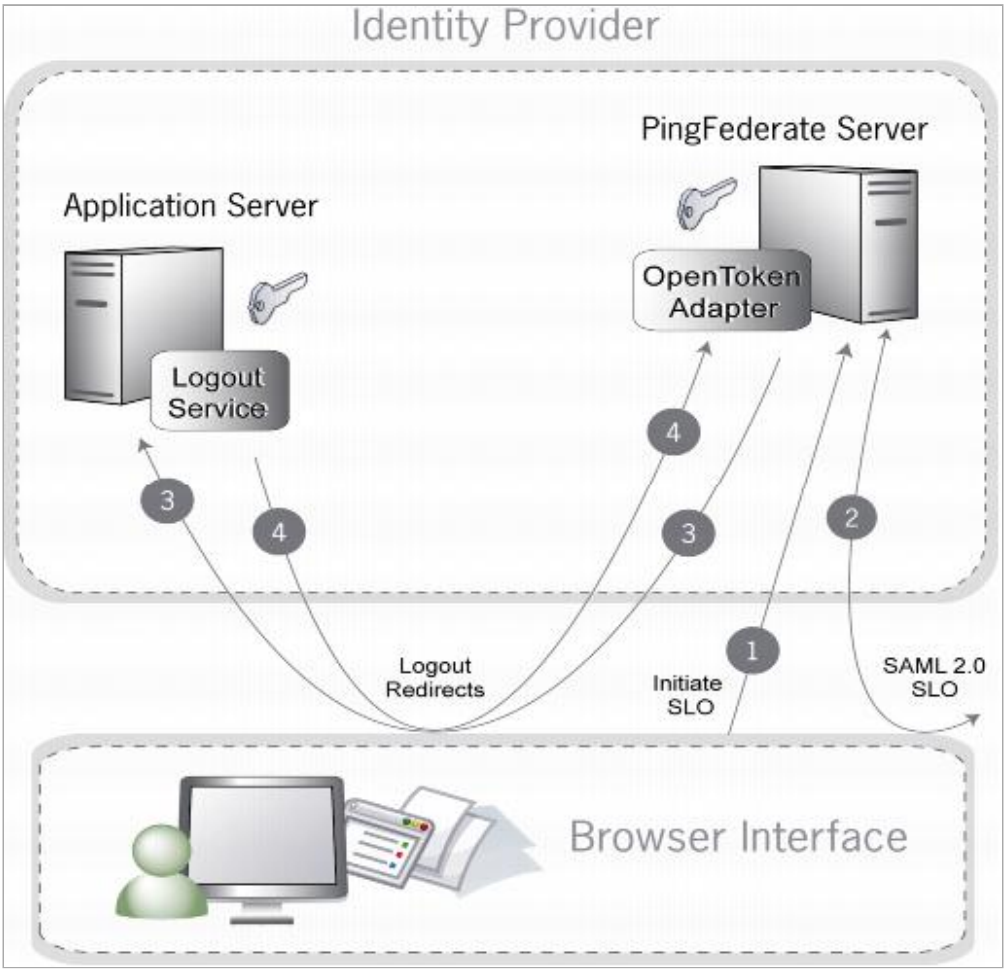
   #Multiple Value Attribute
   $myvalues->add(“MultiValueAttr”, “value1”);
   $myvalues->add(”MultiValueAttr”, ”value2”);
   $myvalues->add(“MultiValueAttr”, “value3”);

   # Generate and write the token to a cookie (assuming that's
   # our configuration)
   $myagent->writeTokenToHTTPResponse($myvalues);
?>
```

IdP single logout integration

When an IdP PingFederate server receives a request for SLO, it redirects the user’s browser to the Logout Service defined in the IdP OpenToken Adapter configuration. The redirect URL includes an OpenToken containing the user attributes defined in the IdP OpenToken Adapter instance for the partner connection. The Logout Service should remove the user’s session on the application server and redirect the user’s browser back to the IdP PingFederate server. The diagram below shows the flow of IdP-initiated SLO, but the architecture would also support SP-initiated SLO.

Copyright © 2025 Ping Identity Corporation



Processing Steps

1. User initiates a single logout request. The request targets the PingFederate server's `/idp/startSLO.ping` endpoint.
2. PingFederate sends a logout request and receives responses from all SPs registered for the current SSO session.
3. PingFederate redirects the request to the IdP Web application's Logout Service, which identifies and removes the user's session locally.
4. The application Logout Service redirects back to PingFederate to display a logout-success page.
    - Sample code

Sample code

Below is an example code snippet for processing a logout request and redirecting the user's browser back to PingFederate:

Copyright © 2025 Ping Identity Corporation

```
<?php
    # Use and instantiate an agent
    use pingidentity\opentoken\agent;
    $myagent = new Agent();

    # Setup an array of values
    $myvalues = array(TOKEN_SUBJECT => “user1”);

    # Generate and write the token to a string (assuming that's
    # our configuration)
    $queryParam = $myagent->writeTokenToHTTPResponse($myvalues);

    $resumePath = $_GET['resume'];
    $returnUrl = "https://<SERVER_NAME>:9031/" . $resumePath;

    session_destroy();
    ob_end_clean();

    if ($queryParam)
    {
      if (strpos($returnUrl, "?") == FALSE)
      {
        $returnUrl = $returnUrl . "?";
      }
      $returnUrl = $returnUrl . $queryParam;
    }

    header("Location: " . $returnUrl);
?>
```

SP single sign-on integration

When PingFederate is configured as an SP, it takes inbound SAML assertions and converts them to some local format (cookie or otherwise) that can be used by an application to create a user’s session. For an `OpenToken`, the PingFederate adapter takes the attributes and values from the SAML assertion and stores them in an `OpenToken` cookie or query parameter in the user’s browser. The user is then redirected to the target application, which can then identify the user from the included `OpenToken`. The application can use either the Agent object to do the decoding explicitly, or include the opentoken.php file to have that processing done automatically.

To do the processing automatically, include the `pingidentity/opentoken/helpers/opentoken.php` file inside the page responsible for logging the user in. The opentoken.php file will automatically parse inbound tokens using the agent configuration generated by PingFederate during the PHP Integration Kit deployment process, and provide them as global variables.

These global variables are listed below:

Variable Name	Description/Contents
$opentoken_subject	String representing the authenticated subject in the OpenToken
$opentoken_haveValidToken	`Boolean` indicating whether the received OpenToken was valid

Copyright © 2025 Ping Identity Corporation

Variable Name	Description/Contents
$opentoken_lastError	String containing the last error message from decoding the `OpenToken`
$opentoken_values	Keyed-array of values contained in the `OpenToken`. `Null` if no valid token was received

Sample code

A simple example of this mode of operation is demonstrated in the PHP code below:

```
<?php
   include_once “pingidentity/opentoken/helpers/opentoken.php”
   if ($opentoken_haveValidToken == true)
   {
      echo “Welcome “, $opentoken_subject, “!<p>”;
      print_r($opentoken_values);
   }
   else
   {
      echo “You are not logged in: “, $opentoken_lastError, “<p>”;
   }
?>
```

As with the IdP, you can use the pingidentity/opentoken/agent.php API to read tokens directly. The agent.php API is a PHP object that provides access to functionality for reading an `OpenToken` from a given HTTP request.

Instantiating the agent object is done simply by using the PHP Agent Namespace and invoking the constructor, as in the example below:

```
<?php
   use pingidentity\opentoken\agent;
   $myagent = new Agent();
?>
```

When the Agent object is instantiated, it uses the config.php file to find the configuration data generated when the OpenToken Adapter was configured. This configuration data includes the name of the cookie that the agent object will read. If the file specified in config.php is not found, the Agent constructor will throw an exception.

The readTokenFromHTTPRequest method inspects the cookie (or query parameters, depending on the configuration of the agent instance) and decodes the `OpenToken`, returning an array of attributes or null if no token is found or an error is encountered. In the case of an error, the `lastError` attribute will contain the error message.

The following code demonstrates the use of this method:

Copyright © 2025 Ping Identity Corporation

```
<?php
   # Use and instantiate an agent
   use pingidentity\opentoken\agent;
   $myagent = new Agent();

   # Use the read method to extract the token values
   $myvalues = $myagent->readTokenFromHTTPRequest();
   if ($myvalues)
   {
      # Found some values – print each of them
      foreach ($myvalues as $key => $value)
      {        echo $key . " = " . $value . "<br>";
      }
   }
   else
   {
      # No values – print the error message
      echo "Error reading token: " . $myagent->lastError;
   }
?>
```

Receiving multi-value attributes

The Agent Toolkit for PHP receives multi-value attributes passed in the SAML assertion from PingFederate using the MultiStringArray PHP class distributed with the Agent Toolkit for PHP.

The following code snippet demonstrates how to get the multi-value attributes using the Agent Toolkit for PHP:

```
<?php
   # Use and instantiate an agent
   use pingidentity\opentoken\agent;
 use pingidentity\opentoken\helpers\multistringarray;

     $myagent = new Agent();

     # read OpenToken into a MultiStringArray ($userInfo)
     $userInfo = $myagent->readTokenFromHttpRequestToMultiStringArray();

     # retrieve single value attribute
     $username = $userinfo->get(TOKEN_SUBJECT, 0);

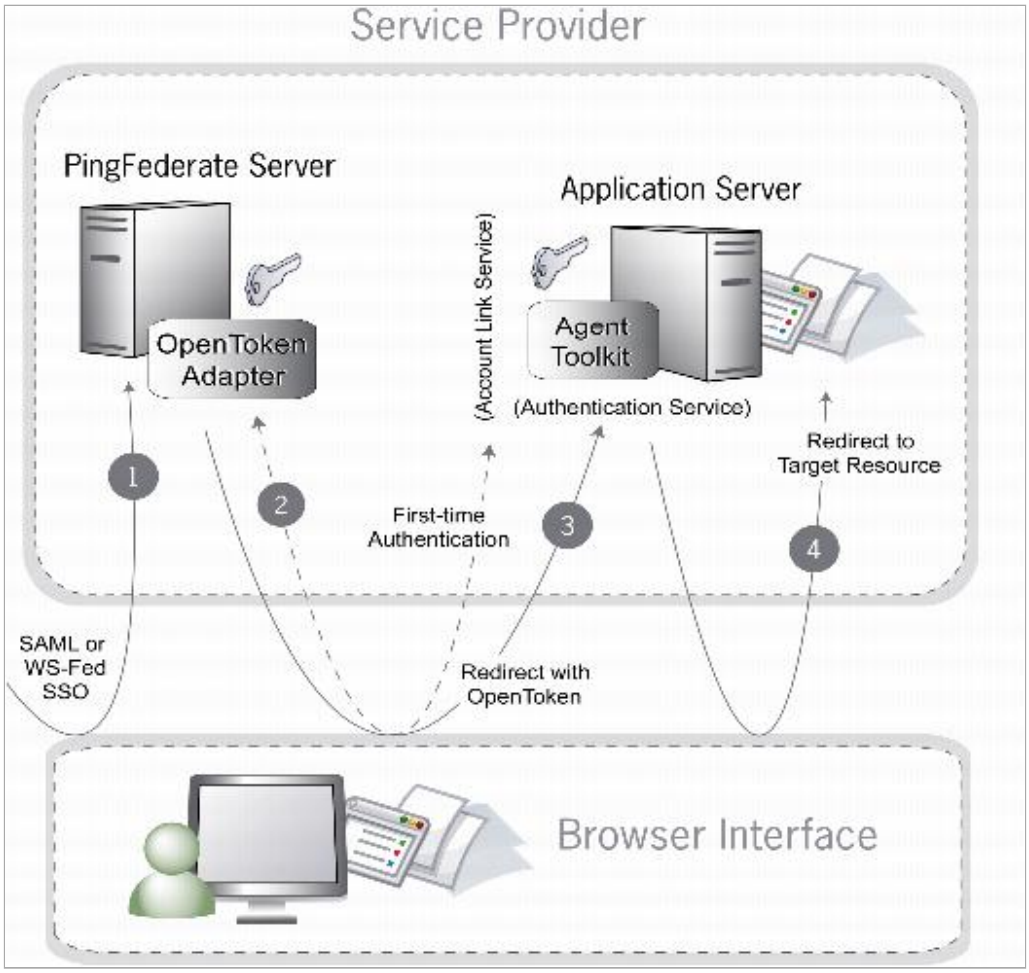
     # retrieve multiple value attribute. Returns an array of values.
     $multiValues = $userinfo->get("MultiValueAttr");

?>
```

Copyright © 2025 Ping Identity Corporation

SP single sign-on integration using account linking

If an SP's SSO implementation employs account linking, the flow of events is somewhat different since a user must authenticate to the SP application the first time SSO is initiated (for more information, see Key concepts in the PingFederate documentation). In this case, PingFederate and the OpenToken Adapter support an integration mechanism to redirect the user to an Account Link Service to which a user initially authenticates. Upon successful authentication, the account link service must redirect the user back to PingFederate with an OpenToken, which PingFederate uses to create an account link for the user. For subsequent SSO requests, PingFederate uses the account link established in the first SSO request to identify the user. It then creates an OpenToken and sends it to the Authentication Service associated with the application.



Sequence

1. PingFederate receives an assertion under either the SAML 2.0 or WS-Federation protocol.
2. If this is the first time the user has initiated SSO to this SP, PingFederate redirects the browser to the Application Server's Account Link Service, where the user must authenticate. Upon successful authentication, an `OpenToken` is returned to PingFederate, and an account link is established for this user within PingFederate. This account link is used on subsequent SSO transactions.
3. PingFederate retrieves the local user ID from its account link data store. PingFederate's OpenToken Adapter generates an `OpenToken` based on the assertion and account link, and then redirects the user's browser to the Web application's SSO Authentication Service, passing the `OpenToken` in the redirect.

Copyright © 2025 Ping Identity Corporation

4. The Authentication Service extracts the contents of the `OpenToken`, establishes a session for the user, and redirects the user's browser to the Target Resource (the `resumePath` URL sent as a query parameter).

In an Account Linking event, the user's browser is redirected to the configured Link Service in the SP OpenToken Adapter instance.

- Sample code

Sample code

The application should capture the `resumePath` upon a `GET` request to this URL with code similar to the following:

```
<?php
   # Use and instantiate an agent
   use pingidentity\opentoken\agent;
   $myagent = new Agent();

   $resumePath = $_GET['resume'];
   $returnUrl = "https://<SERVER_NAME>:9031/" . $resumePath;

   # After the user has authenticated, an OpenToken needs to be
   # generated with the userId and sent back to the previously
   # captured resumePath.

   $myvalues = array(TOKEN_SUBJECT => $userInfo["userid"],
   AUTHN_CTX_ATTRIBUTE_NAME => $userInfo[AUTHN_CTX_ATTRIBUTE_NAME]);

   $queryParam = $myagent->writeTokenToHTTPResponse($myvalues);
   if ($queryParam)
   {
       if (strpos($returnUrl, "?") == FALSE)
       {
          $returnUrl = $returnUrl . "?";
       }
       $returnUrl = $returnUrl . $queryParam;
   }
     header("Location: " . $returnUrl);
?>
```

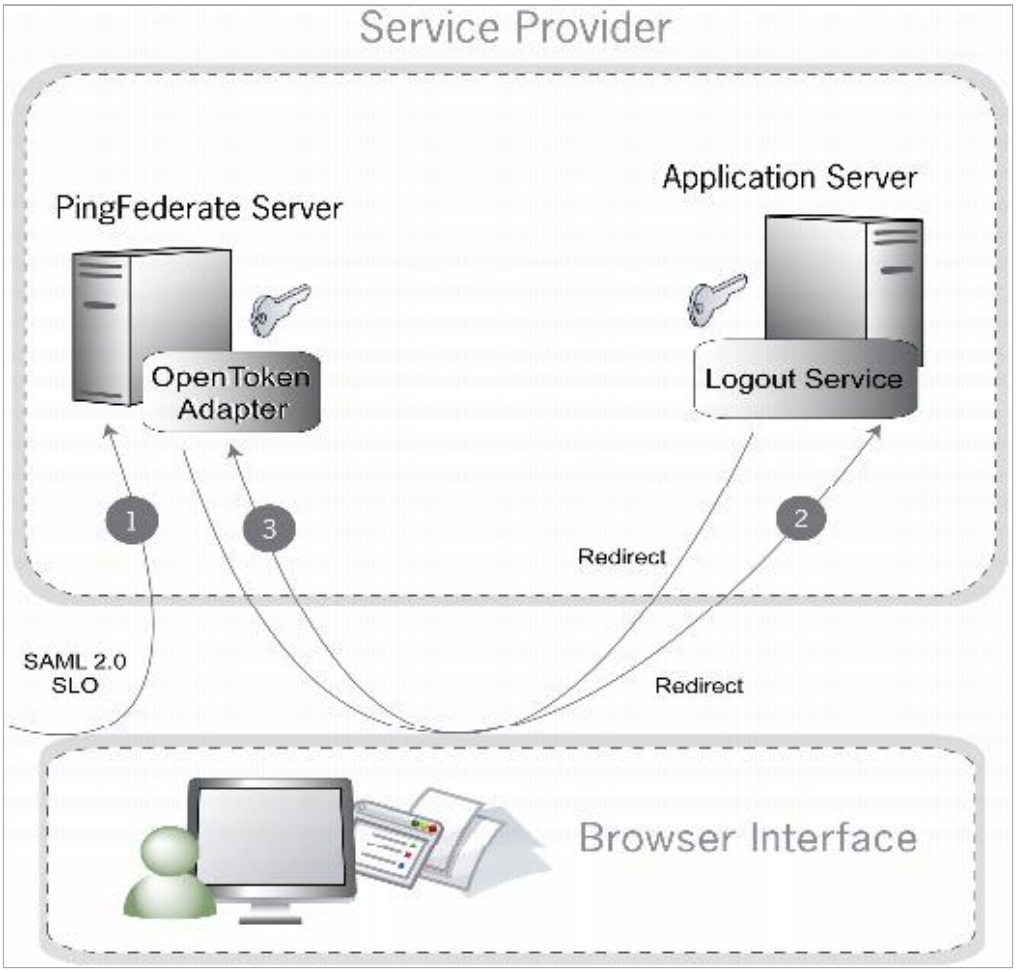
SP single logout integration

When an SP PingFederate server receives a request for SLO, it redirects the user's browser to the Logout Service as configured in the SP OpenToken Adapter instance. As part of the redirect, PingFederate and the OpenToken Adapter include both an OpenToken and a resumePath query parameter.

- The OpenToken includes attributes about the user.
- The resumePath query parameter provides the target application URL.

A user can have multiple sessions. This logout sequence, as shown in the following diagram, will occur for each of the user's sessions controlled by the SP PingFederate server.

Copyright © 2025 Ping Identity Corporation



Sequence

1. PingFederate receives an SLO request under the SAML 2.0 protocol.
2. PingFederate, via the OpenToken Adapter, redirects the browser to the Application Server's Logout Service.
3. The Logout Service returns to PingFederate, indicating that the logout was successful.

The code needed to perform an SP SLO is identical to that required for an IdP SLO. (See Sample code.)

Testing

You can test the PHP Integration Kit using the sample application bundled with this distribution. (See the PHP Sample Application Startup Guide.)

Sample application setup

To see a working demonstration of the PHP Integration Kit, you can deploy the included sample applications.

Copyright © 2025 Ping Identity Corporation

Components

- Identity provider (IdP) sample application
    - Acts as an IdP application in your demonstration environment.
- Service provider (SP) sample application
    - Acts as an SP application in your demonstration environment.
- PingFederate configuration archive
    - This data.zip archive automatically configures PingFederate with OpenToken Adapter instances to work with the two sample applications.
    - The IdP server looks up and sends authentication information to the SP.
    - The SP server forwards this information to the SP sample application to create the local user session.
    - The SP server also sends authentication requests to the IdP on behalf of local users.

Intended audience

The installation and basic usage portions of this guide are intended for PingFederate administrators and web-application architects and developers who want to test the deployment of the PHP Integration Kit or validate end-to-end integration. Some knowledge of the PingFederate administrative console and identity federation using SAML is helpful but not required.

Sections describing the optional use of the PHP source code and advanced alternative deployment scenarios are intended, respectively, for experienced PHP developers and administrators with some expertise using and deploying PingFederate in a production environment.

System requirements

- PingFederate 9.0 or later.
- A web server running PHP 5.4.8 or later.
- The OpenToken Adapter (included).
- A PHP server with the following:
    - The cURL, mcrypt, mhash, and zlib extensions installed.
    - The correct time (recently synchronized).
    - The correct time zone. Set the correct time zone in your `php.ini` file. For example, `date.timezone = America/Halifax;`

      For a list of supported time zones, see Timezones in the PHP documentation.
- The Appropriate Error reporting Level in php.ini file:

```
error_reporting = E_ALL & ~E_DEPRECATED & ~E_STRICT;
```

- The installation script is designed for Apache HTTP server on a Linux platform derived from Red Hat.
- The `data.zip` configuration archive assumes that your PHP server is on the same machine as PingFederate.

Copyright © 2025 Ping Identity Corporation

Configuring PingFederate and the sample applications

To see a working demonstration of the PHP Integration Kit, deploy the configuration archive and sample applications.

About this task

The sample configuration archive configures a single instance of PingFederate with an example integration that uses both the IdP and SP sample applications. It automatically creates two instances of the OpenToken Adapter.

> **Caution**
>
> Deploying the configuration archive will destroy your existing PingFederate configuration. We recommend that you test it on a fresh installation of PingFederate, or back up your current configuration as shown in Exporting an archive in the PingFederate documentation.

Steps

1. Start PingFederate.
2. From the Agentless Integration Kit `.zip` archive, copy `sample/data.zip` to `<pf_install>/pingfederate/server/deploy/drop-in-deployer`.
3. If your PHP server is not hosted on the same machine as PingFederate, adjust the URLs in the OpenToken Adapter instance configurations.
    1. Sign on to the PingFederate administrator console.
    2. On the **Identity Provider > Adapters** screen, click **OTIdPPHP**.
    3. On the **IdP Adapter** screen, click **Show Advanced Fields**.
    4. In the **Authentication Service** and **Logout Service** fields, change the **localhost:9119** portion of the URL to point to your PHP server.
    5. Make the equivalent changes to the URLs in the **Service Provider > Adapters > OTSPPHP** adapter instance configuration.
4. Do one of the following:
    - Deploying on Red Hat Linux.
    - Deploying on Linux or Unix.
    - Deploying on Windows.
    - Deploying on the PHP built-in web server

Deploying on Red Hat Linux

You can deploy the PHP Integration Kit sample applications on Linux Red Hat using the included installation script.

About this task

Copyright © 2025 Ping Identity Corporation

The script copies the sample applications to the Apache HTTP server DocumentRoot, copies configuration files, and changes file permissions and ownership.

Steps

1. On the computer that hosts your Apache HTTP server, sign on as root.
2. Check that the Apache HTTP server is running.
3. Check the `httpd.conf` file to determine the DocumentRoot of the Apache HTTP server. The installation script will prompt you for this information.
4. From the integration `.zip` archive, copy the `sample` directory to the host computer.
5. At a shell command prompt, change the current directory to the sample director.
6. Execute the installation script using the following commands:
    1. `chmod u+x ./install.sh`
    2. `./install.sh`
7. Enter `y` (yes) to accept the detected Apache user.

Deploying on Linux or Unix

You can manually deploy the PHP Integration Kit sample applications on UNIX or other Linux platforms.

Steps

1. From the integration `.zip` archive, copy the `sample` directory to root level of the DocumentRoot of your Apache HTTP server. This makes the directory accessible at https://*hostname*/sample.
2. Move the `sample/config` directory to a directory on the server that is outside the DocumentRoot. This protects the `config` directory from unauthorized access.
3. Update the location of the `agent-config.txt` files. Modify the following files to point to the new location of the `config` directory.
    1. Open `sample/idp/pingidentity/opentoken/helpers/config.php` for editing.
    2. Change the following line based on the new location of your `config` directory:

        ```
        const   AGENT_CONFIG_FILE = "../config/idp/agent-config.txt";
        ```

    3. Repeat the equivalent steps a-b in `sample/sp/pingidentity/opentoken/helpers/config.php` for your SP `agent-config.txt` file.
4. Update the location of the `config.properties` files. Modify the following files to point to the new location of the `config` directory.
    1. Open `sample/idp/Const.php` for editing.
    2. Change the following line based on the new location of your `config` directory:

Copyright © 2025 Ping Identity Corporation

```
define("CONFIG_FILE", "../config/idp/config.properties");
```

3. Repeat the equivalent steps a-b in `sample/sp/Const.php` for your SP `config.properties` file.

5. Change the ownership and permission of the sample application configuration files for the user and/or group that runs the Apache server by doing the following:

```
chown –R <apache_user> <config_dir>/ chmod u+w <config_dir>/
```

6. **If your PingFederate server is hosted on another computer, go to the following URLs and change the PF Host Name value in the SP and IdP sample applications configuration.**
   - **https://*hostname*/sample/idp/ConfigUI.php**
   - **https://*hostname*/sample/sp/ConfigUI.php**

Deploying on Windows

You can manually deploy the PHP Integration Kit sample applications on Windows computers.

Steps

1. **From the integration `.zip` archive, copy the `sample` directory to root level of the DocumentRoot of your Apache HTTP server. This makes the directory accessible at `https://hostname/sample`.**
2. **Move the `sample/config` directory to a directory on the server that is outside the DocumentRoot. This protects the `config` directory from unauthorized access.**
3. **Update the location of the `agent-config.txt` files. Modify the following files to point to the new location of the `config` directory.**
   1. **Open `sample/idp/pingidentity/opentoken/helpers/config.php` for editing.**
   2. **Change the following line based on the new location of your `config` directory:**

```
const   AGENT_CONFIG_FILE = "../config/idp/agent-config.txt";
```

   3. **Repeat the equivalent steps a-b in `sample/sp/pingidentity/opentoken/helpers/config.php` for your SP `agent-config.txt` file.**
4. **Update the location of the `config.properties` files. Modify the following files to point to the new location of the `config` directory.**
   1. **Open `sample/idp/Const.php` for editing.**
   2. **Change the following line based on the new location of your `config` directory:**

```
define("CONFIG_FILE", "../config/idp/config.properties");
```


Copyright © 2025 Ping Identity Corporation

3. Repeat the equivalent steps a-b in `sample/sp/Const.php` for your SP `config.properties` file.

5. Use Windows Task Manager to determine the user that runs the Apache server.
6. Change the security properties for the `config` to give full control to the user that runs the Apache server.
7. If your PingFederate server is hosted on another computer, go to the following URLs and change the PF Host Name value in the SP and IdP sample applications configuration.
    - `https://hostname/sample/idp/ConfigUI.php`
    - `https://hostname/sample/sp/ConfigUI.php`

Deploying on the PHP built-in web server

Steps

1. Check that the PHP path environment variable points to the correct location of PHP.
2. Create a public HTML directory to host the sample applications.
3. From the integration `.zip` archive, copy the `sample` directory to root level of the new public HTML directory. This makes the directory accessible at `https://hostname/sample`.
4. Move the `sample/config` directory to a directory on the server that is outside the public HTML directory. This protects the `config` directory from unauthorized access.
5. Update the location of the `agent-config.txt` files. Modify the following files to point to the new location of the `config` directory.
    1. Open `sample/idp/pingidentity/opentoken/helpers/config.php` for editing.
    2. Change the following line based on the new location of your `config` directory:

    ```
    const   AGENT_CONFIG_FILE = "../config/idp/agent-config.txt";
    ```

    3. Repeat the equivalent steps a-b in `sample/sp/pingidentity/opentoken/helpers/config.php` for your SP `agent-config.txt` file.
6. Update the location of the `config.properties` files. Modify the following files to point to the new location of the `config` directory.
    1. Open `sample/idp/Const.php` for editing.
    2. Change the following line based on the new location of your `config` directory:

    ```
    define("CONFIG_FILE", "../config/idp/config.properties");
    ```

    3. Repeat the equivalent steps a-b in `sample/sp/Const.php` for your SP `config.properties` file.
7. From the public HTML directory, execute the following command to start the PHP built-in Web Server:

`php –s <hostname>:<port>`

Copyright © 2025 Ping Identity Corporation

Using the IdP sample application

The PHP Integration Kit identity provider (IdP) sample application demonstrates IdP-initiated single sign-on (SSO) and single logout (SLO) use cases.

About this task

The IdP sample application simulates the IdP-initiated SSO/SLO scenario in which users authenticate to an IdP locally in order to access a remote SP application. In this scenario, users may be accessing a company portal that provides links to partner applications such as local news and weather, stock market information, and HR and 401(k) benefits.

When you authenticate locally to the IdP sample application, no communication occurs between that application and PingFederate. The user authenticates using the local user store; no SAML use cases are invoked. However, when you click a link to a third-party application, such as your company's 401(k) provider, the IdP initiates an SSO transaction.

Steps

1. Start the PingFederate and Apache HTTP servers or the PHP built-in web server.
2. In a browser, open the sample application:

   `https://hostname/sample/idp/idpmain.php`
3. On the main page, click Login Locally.
4. On the Identity Provider Login page, sign on as any of the listed users with a password of `test`.
5. Click Login.
6. On the Identity Provider screen, try the following:
   1. Optional: To begin an IdP-iniated SSS to the SP sample application, click the Single Sign-On. This starts a user session on the SP and redirects you to the SP sample application. For more information, see Using the SP Sample Application.
   2. Optional: After signing on to the SP sample application and returning to the Identity Provider main page, click Single Sign-Out to initiate a SLO request to the SP. This ends your user session on the SP as well as your local user session.

Using the SP sample application

The PHP Integration Kit service provider (SP) sample application demonstrates SP-initiated single sign-on and single logout use cases.

Steps

1. Start the PingFederate and Apache HTTP servers or the PHP built-in web server.
2. In a browser, open the sample application:

   `https://hostname/sample/sp/spmain.php`

Copyright © 2025 Ping Identity Corporation

3. From the list, select an identity provider (IdP) connection, and then click Single Sign-On to begin an SP-initiated single sign-on (SSO) transaction.
4. On the Identity Provider login screen, sign on as any of the listed users with a password of `test`.

> **Note**
>
> If you have already authenticated with the identity provider, you do not need to authenticate again unless either the ForceAuthn or IsPassive option is checked in the **Advanced Options** section of the application.

5. Click Login.
6. To initiate a single logout (SLO) request, click Single Sign-Out.

Release notes

Changelog

The following is the change history for the PHP Integration Kit.

PHP Integration Kit 2.5.2 – January 2016

- Updated sample application data archive.

PHP Integration Kit 2.5.1 – December 2012

- Updated to address security issue found since the previous release.
- Added support for the 2.5.1 version of the OpenToken Adapter and OpenToken Agent.
- Added Support for PHP 5.4.8.
- Added Namespaces to the PHP OpenToken Agent.

PHP Integration Kit 2.4 – March 2010

- Added token Reply Prevention to the OpenToken IdP Adapter Advanced Settings.

PHP Integration Kit 2.3 – November 2008

- Added POST Transport Method for OpenToken when used by a service Provider.
- Added configuration to specify session cookie vs. persistent cookie.
- Added option to set the "`Secure`" attribute on an OpenToken when cookie is used.
- Correctly handles `null` parameters for SOAP SLO.
- Empty query string (`?`) is not automatically appended to the URL when redirecting to `TargetResource`.
- `TargetResource URL` is URL encoded.

Copyright © 2025 Ping Identity Corporation

PHP Integration Kit 2.2 – June 2008

- Added support for SAML 2.0 `isPassive` and `ForceAuthn`.
- Enforced UTF-8 encoding within the OpenToken adapter.
- Combined the OpenToken adapter and OpenToken Java library jar files into a single adapter file for easier deployment to PingFederate.

PHP Integration Kit 1.2 – December 2007

- Repaired OpenToken decoding to correct bit unpacking issue.
- Added Agent Toolkit API HTML documentation.

PHP Integration Kit 1.1 – December 2007

- Added support for sending and receiving multi-value attributes in the SAML assertion.

PHP Integration Kit 1.0 – November 2007

- Initial release.

Known issues and limitations

The following are known issues or limtiations for the PHP Integration Kit.

Known issues

There are no known issues.

Known limitations

- The Agent Toolkit for PHP supports AES/128/CBC and 3DES/168/CBC.
- The Agent Toolkit for PHP does *not* support AES/256/CBC.
- The Token Name field in the adapter configuration must be unique within the given federation, however is not enforced in the user-interface.
- The AES/256/CBC cipher suite is not supported, due to limitations in the PHP mcrypt libraries.
- All plain-text files have UNIX-style line feeds. This may result in difficulty editing files on a Windows computer.
- Password obfuscation is not supported by the agent.
- The sample application does not support local logout when set to cookie transport mode
- If the SP adapter is set to send extended attributes as cookies, multi-value attributes will fail, because multiple cookies with the same name are not allowed.

Copyright © 2025 Ping Identity Corporation

Download manifest

The following files are included in the PHP Integration Kit `.zip` archive:

- `ReadMeFirst.pdf` – contains links to this online documentation.
- `legal` – a directory that contains this document:
  - `Legal.pdf` – copyright and license information.
- `dist` - a directory that contains libraries needed to run the adapter:
  - `opentoken-adapter-<version>.jar` - OpenToken Adapter JAR file.
  - `pingidentity/opentoken` – directory containing files for the Agent Toolkit for PHP.
    - `agent.php`
    - `/helpers` – directory containing:
      - `config.php`
      - `keyvalueserializer.php`
      - `multistringarray.php`
      - `opentoken.php`
      - `passwordkeygenerator.php`
      - `token.php`
- `sample`
  - `common` – directory that contains common files for the sample applications.
  - `config` – a directory that contains configuration files for the sample applications.
  - `idp` – a directory that contains the IdP sample application.
  - `sp` – a directory that contains the SP sample application.
  - `data.zip` – a PingFederate configuration archive to use with the sample applications.
  - `install.sh` – an installation shell script for the sample applications.

Copyright © 2025 Ping Identity Corporation

PingID Provisioner

The PingID Provisioner allows PingFederate to integrate with PingID for user provisioning.

Features

- Manages users in PingID based on changes in an external data store that is attached to PingFederate
  - Creates, updates, disables, and deletes users
  - Allows you to enable the create, update, disable, and delete capabilities independently
  - Allows you to choose whether to disable or delete users when deprovisioning
  - Allows you to provision disabled users
- Manages users' devices (authentication methods) in PingID
  - Enrolls and manages up to nine devices
  - Allows you to set the management behavior
  - Allows you to set a primary authentication device

Intended audience

This document is intended for PingFederate administrators.

If you need help during the setup process, see the following resources:

- PingFederate documentation:
  - SP connection management
  - Managing digital signing certificates and decryption keys
  - Datastores
  - Configuring outbound provisioning
  - Configuring outbound provisioning settings
- PingID documentation:
  - PingID Administration Guide
  - Automatically update and remove PingID users

System requirements

- PingFederate 9.0 or later
- To allow PingFederate to make outbound connections, you might need to allow the one of the following endpoints in your firewall:
  - North America: idpxnyl3m.pingidentity.com

Copyright © 2025 Ping Identity Corporation

- Europe: idpxnyl3m.pingidentity.eu
- Australia: idpxnyl3m.pingidentity.com.au

User management

The PingID Provisioner synchronizes users from your datastore to PingOne for Enterprise. The following describes the behavior of each provisioning capability.

> **Tip**
>
> You can configure the following capabilities and specify which users to provision when you get to the Creating a provisioning connection part of the setup process.

Synchronizing existing users

PingFederate synchronizes users based on the `username` attribute in PingOne for Enterprise. If a user already exists in your datastore and PingOne for Enterprise, mapping this attribute correctly links the two records together.

For example:

- In PingOne for Enterprise, Janet's `username` is `jsmith`.
- In your datastore, Janet's `sAMAccountName` is `jsmith`.
- On the Attribute Mapping tab of your provisioning connection configuration, map the `username` attribute to `sAMAccountName`.
- When the provisioning connector runs, the datastore user is provisioned with a `username` of `jsmith`. That matches Janet's existing `username` in PingOne for Enterprise, so her information in the datastore is synchronized to her PingOne for Enterprise account.

User provisioning

PingFederate provisions users when any of the following happens:

- A user is added to the datastore group or filter that is targeted by the provisioning connector.
- A user with `disabled` status is added to the datastore group or filter that is targeted by the provisioning connector, and the Provision disabled users provisioning option is enabled. This feature is not available in all provisioning connectors.

> **Tip**
>
> You can define which users PingFederate targets for provisioning on the **Source Location** tab of your provisioning connection configuration.

User updates

PingFederate updates users when a user attribute changes in your datastore.

Copyright © 2025 Ping Identity Corporation

> **Tip**
>
> You can define which attributes PingFederate monitors for changes on the **Attribute Mapping** tab of your provisioning connection configuration.

User deprovisioning

PingFederate deprovisions users when any of the following happens:

- A user is deleted from the user store.
- A user is disabled in the user store.
- A user is removed from the datastore group or filter that is targeted by the provisioning connector.

Device management

As part of the provisioning process, the PingID Provisioner can add and update the authentication devices associated with each user in PingID.

For general information about devices, see PingID authentication for the web and Managing your devices in the PingID documentation.

Nicknames

PingID assigns nicknames to authentication devices. These nicknames appear in the user interface, such as on the device management page. The PingID Provisioner also uses nicknames to identify the devices it manages.

The following are the "managed" nicknames used by the provisioning connector:

- `Email 1`
- `Email 2`
- `Email 3`
- `SMS Number 1`
- `SMS Number 2`
- `SMS Number 3`
- `Voice Number 1`
- `Voice Number 2`
- `Voice Number 3`

The connector can only manage these nine devices.

Copyright © 2025 Ping Identity Corporation

Mapping attributes to nicknames

Each of the managed devices can be populated by a matching user attribute on the Attribute Mapping tab of the channel configuration. The matching attributes are prefixed with "MFA", such as "MFA SMS Number 1".

You can map these attributes in the Creating a provisioning connection part of the setup process.

Device management

The PingID Provisioner can compare the device values (the email address or phone number) in PingID against the values in the datastore. The datastore is always considered the source of truth when updating the nine managed devices, but you can choose how the connector handles other devices it finds. Alternately, you can turn off device management and use PingID as the source of truth.

You can configure the connector's device management behavior using the Manage devices setting in the Creating a provisioning connection part of the setup process.

Manage devices setting

Setting	Description
Do not manage devices	PingFederate doesn't provision or manage any devices in PingID. Users can manage their own devices as shown in Managing your devices in the PingID documentation.
Merge devices	PingFederate provisions devices to PingOne for Enterprise using the nine managed nicknames. When updating a user, if there is a conflict between the datastore and PingID for one of the managed devices, the datastore takes precedence. PingFederate removes and re-creates the device in PingID with the new value. **Note** When updating the user's "primary" device, the user has to choose it as their primary device again. The provisioner doesn't change devices with non-managed nicknames, such as devices added by the user or an administrator.
Overwrite devices	This behaves the same as Merge devices, except the provisioner removes all devices with non-managed nicknames, such as devices added by the user or an administrator.

Primary devices

In PingID, each user can have one "primary" device. The user is prompted to authenticate using this device by default.

When creating a new user, the PingID Provisioner can set a primary device automatically. You can customize this behavior with the Primary Device on Create setting in the Creating a provisioning connection part of the setup process.

When updating existing users, the PingID Provisioner doesn't set or change the primary device.

Copyright © 2025 Ping Identity Corporation

Primary Device on Create

Setting	Description
Do not manage	PingID Provisioner can provision devices, but it doesn't set a primary device.
MFA Email 1	PingID Provisioner sets `Email 1` as the user's primary authentication device.
MFA SMS Number 1	PingID Provisioner sets `SMS Number 1` as the user's primary authentication device.
MFA Voice Number 1	PingID Provisioner sets `Voice Number 1` as the user's primary authentication device.

Setup

Upgrading an existing deployment

If you are upgrading from a previous version of the PingID Provisioner, note your existing service provider (SP) connection configuration and create a new connection.

Steps

1. Back up your current PingFederate configuration as shown in Configuration archive in the PingFederate documentation.
2. In the PingFederate administrative console, open your existing SP connection

   Choose from:

   - For PingFederate 10.1 or later: go to Applications > Integration > SP Connections and select your connection.
   - For PingFederate 10.0 or earlier: go to Identity Provider > SP Connections and select your connection.
3. Note the attribute mappings for your existing SP connection. For help, see Mapping attributes in the PingFederate documentation.
4. Delete your existing SP connection.

   Choose from:

   - For PingFederate 10.1 or later:
     1. Go to Applications > Integration > SP Connections.
     2. For your existing connection, click Select action, and then click Delete. Click Confirm.
   - For PingFederate 10.0 or earlier:
     1. Go to Identity Provider > SP Connections > Manage All.
     2. For your existing connection, click Select action, and then click Delete. Click Save.
5. Complete the steps in Deploying the integration files.

Copyright © 2025 Ping Identity Corporation

6. Complete the steps in Creating a provisioning connection.
    - From Outbound Provisioning > Manage Channels > Channel, on the Attribute Mapping tab, configure the attribute mappings based on your notes.

Deploying the integration files

To get started with the integration, deploy the PingID Provisioner files to your PingFederate directory.

Steps

1. Download the PingID Provisioner `.zip` archive from the Ping Identity Integration Directory.
2. Stop PingFederate.
3. If you're upgrading an existing deployment, delete the `pf-pingid-quickconnection-<version>.jar` file from your `<pf_install>/pingfederate/server/default/deploy` directory.
4. Extract the `.zip` archive and merge the contents of the `dist` directory with your `<pf_install>/pingfederate/server/default/deploy` directory.
5. Enable the PingFederate provisioning engine:
    1. Open your `<pf_install>/pingfederate/bin/run.properties` file.
    2. Change `pf.provisioner.mode` to `STANDALONE`. Save the file.

    > **Note**
    >
    > To configure the `FAILOVER` mode instead, see Deploying provisioning failover in the PingFederate Server Clustering Guide.

6. Start PingFederate.
7. If you operate PingFederate in a cluster, repeat steps 2 - 4 and step 6 for each engine node.

Enabling provisioning in PingFederate

To use PingFederate for provisioning, configure an external datastore.

About this task

Your external datastore acts as the source of data for provisioning. PingFederate also uses an internal datastore to store the state of synchronization between the source datastore and the target datastore.

For more information, see Datastores in the PingFederate documentation.

Steps

1. Configure the datastore for PingFederate to use as the source of user data.

    For instructions, see Datastores in the PingFederate documentation.

Copyright © 2025 Ping Identity Corporation

> **Tip**
>
> When targeting users and groups for provisioning, exclude the user account that you will use to administer users in your connection to PingOne for Enterprise. This prevents the PingFederate provisioning engine from interfering with the account that provisions users and groups.

2. Do one of the following:

   Choose from:

   - For PingFederate 10.1 or later: Go to System > Server > Protocol Settings.
   - For PingFederate 10.0 or earlier: Enable the identity provider (IdP) and outbound provisioning roles:
     1. Go to System > Protocol Settings > Roles & Protocols.
     2. Select Enable Identity Provider IdP Role and Support the Following.
     3. Select Outbound Provisioning. Click Next.

3. On the Outbound Provisioning tab, select the PingFederate internal datastore. Click Save.

   For help, see Configuring outbound provisioning settings in the PingFederate documentation.

Get your PingID settings file

Download your PingID settings file to make it easier to set up your service provider connection.

About this task

For more information, see the PingID Properties File section of the PingID API overview in the PingID API documentation.

Steps

1. Sign on to the PingOne for Enterprise administrative console and open the PingID dashboard.
2. Go to Setup > PingID.
3. On the Client Integration tab, in the Integrate with PingFederate section, click Download, then save the `pingid.properties` file.

> **Note**
>
> You will use this file in PingID Connector settings reference.

Creating a provisioning connection

To allow PingFederate to manage users in PingID, create a service provider (SP) connection.

Steps

1. In the PingFederate administrator console, create a new SP connection:

Copyright © 2025 Ping Identity Corporation

Choose from:

- For PingFederate 10.1 or later: go to Applications > Integration > SP Connections. Click Create Connection.
- For PingFederate 10.0 or earlier: go to Identity Provider > SP Connections. Click Create Connection.

2. Configure the basic connection details with the PingID Provisioner quick connection template.
    1. On the Connection Template tab, select Use a template for this connection.
    2. In the Connection Template list, select PingID Provisioner. Click Next.
    3. On the Connection Type tab select only Outbound Provisioning. Click Next.
    4. On the General Info tab, in the Partner's Entity ID and Connection Name fields, enter a name of your choosing. Click Next.
3. On the Outbound Provisioning tab, configure provisioning with the following details.

    For help, see Configuring outbound provisioning in the PingFederate documentation.

    1. Click Configure Provisioning.
    2. On the Target tab, configure the provisioning connector by referring to PingID Connector settings reference. Click Next.
    3. On the Manage Channels tab, create a channel as shown in Managing channels in the PingFederate documentation. Complete the attribute mappings by referring to Supported attributes reference.
4. On the Activation and Summary tab, above the Summary section, click the toggle to turn on the connection. Click Save.

PingID Connector settings reference

Field descriptions for the PingID Provisioner configuration.

Provisioning connector settings reference

Field	Description
PingID Properties	For PingFederate 10.2 and later. Upload the `pingid.properties` file that you downloaded in Get your PingID settings file.

Copyright © 2025 Ping Identity Corporation

Field	Description
Manage Devices	Determines how the connector manages devices (authentication methods) in PingID. For more detail, see Device management. *Do not manage devices* PingFederate doesn't provision or manage any devices in PingID. *Merge devices* If there is a conflict between the datastore and PingID for one of the managed devices, the datastore takes precedence. **Note** When updating the user's "primary" device, the user has to choose it as their primary device again. The provisioner doesn't change devices with non-managed nicknames, such as devices added by the user or an administrator. *Overwrite devices* This behaves the same as Merge devices, except the provisioner removes all devices with non-managed nicknames, such as devices added by the user or an administrator.
Primary Device on Create	Determines which device (authentication method) the connector sets as the primary when provisioning a new user to PingID. For more detail, see Device management. *Do not manage* PingFederate provisions devices but doesn't set any as the primary device. *MFA Email 1* Sets `Email 1` as the primary device. *MFA SMS Number 1* Sets `Number 1` as the primary device. *MFA Voice Number 1* Sets `Voice Number 1` as the primary device.
Provisioning Options	
User Create	Selected (default) – PingFederate creates users in PingID. Cleared – PingFederate does not create users in PingID.
User Update	Selected (default) – PingFederate updates existing users in PingID. Cleared – PingFederate does not update existing users in PingID.

Copyright © 2025 Ping Identity Corporation

Field	Description
User Disable / Delete	Selected (default) – PingFederate disables or deletes users in PingID. **Note** PingFederate can only re-enable a user if **User Update** is selected. Cleared – PingFederate does not disable or delete users in PingID.
Provision Disabled Users	This option applies when: • The User Create option is selected, and • The provisioning engine targets a user in the datastore that has a "disabled" status. Selected (default) – PingFederate creates the user in PingID with a "disabled" status. Cleared – PingFederate does not create the user in PingID.
Note If any of the **Provision Options** are cleared, PingFederate logs a warning in the user workflow section of `provisioner.log` when the related action fails.	
Remove User Action	This option applies when: • User Disable / Delete is selected, and • a previously-provisioned user no longer meets the condition set on the Source Location screen, or • a user has been disabled or deleted from the data store. Disable (default): PingFederate disables the user in PingID. Delete: PingFederate deletes the user from PingID.

Supported attributes reference

The following attributes can be mapped for user provisioning to PingID.

For general information about attributes, see the AddUser section of PingID User Management API in the PingID API documentation.

Attribute	Description
`Username`	The unique identifier for the user in PingID. A username cannot be updated. This attribute is required.
`Email`	The email for the user (for example, "bjensen@example.com").
`First Name`	The given name of the user, or first name in most Western languages (for example, 'Barbara' given the full name 'Ms. Barbara Jane Jensen, III').

Copyright © 2025 Ping Identity Corporation

Attribute	Description
`Last Name`	The family name of the user, or last name in most Western languages (for example, 'Jensen' given the full name 'Ms. Barbara Jane Jensen, III').
`MFA Email 1`	An email address to use as an authentication device.
`MFA Email 2`	An email address to use as an authentication device.
`MFA Email 3`	An email address to use as an authentication device.
`MFA SMS Number 1`	A phone number to use as an SMS authentication device. For detailed requirements, see SMS and voice authentication in the PingID documentation.
`MFA SMS Number 2`	A phone number to use as an SMS authentication device.
`MFA SMS Number 3`	A phone number to use as an SMS authentication device.
`MFA Voice Number 1`	A phone number to use as a voice authentication device. For detailed requirements, see SMS and voice authentication in the PingID documentation.
`MFA Voice Number 2`	A phone number to use as a voice authentication device.
`MFA Voice Number 3`	A phone number to use as a voice authentication device.

Release notes

Changelog

The following is the change history for the PingID Provisioner.

PingID Provisioner 1.1.1 – November 2021

- Improved security by updating bundled components.

PingID Provisioner 1.1 – March 2021

- Added support for user creation.
- Added support for provisioning disabled users.
- Added support for device management.
- Added the Manage Devices and Primary Devices on Create settings.
- Added the ability to upload the PingID properties file.

Copyright © 2025 Ping Identity Corporation

PingID Provisioner 1.0.1 – May 2018

- Fixed disabling or deleting of users when User Update is false in configurable options.

PingID Provisioner 1.0 – March 2018

- Initial release
- Added support for PingID API v4.9
- Added support for user life cycle management (including updates, disabling users, and deleting users)
- Added configuration options for workflow capabilities (for example, the ability to disable updates)

Known issues and limitations

The following are known issues or limitations with the PingID Provisioner.

Known issues

There are no known issues.

Known limitations

- When an LDAP user is deleted in a targeted group distinguished name (DN), the provisioning connector does not propagate the deletion until a new user is added to the group. This limitation is compounded when the User Create provisioning option is disabled. For solutions, see SaaS provisioner does not remove the user in the Knowledge Base.
- Due to PingFederate limitations, user attributes cannot be cleared once set.
- Due to a limitation in the PingID API, changes made to the user's username are not reflected in the PingOne WebPortal.

Download manifest

The following files are included in the PingID Provisioner `.zip` archive:

- `ReadMeFirst.pdf` – contains links to this online documentation
- `Legal.pdf` – copyright and license information
- `dist` – contains the integration files
    - `deploy/pf-pingid-quickconnection-<version>.jar` – The PingID Provisioner quick connection template.

Copyright © 2025 Ping Identity Corporation

PingID SDK Provisioner

The PingID SDK Provisioner allows PingFederate to manage users in PingID SDK based on changes in an external user datastore.

Features

- Manages users in PingID SDK based on changes in an external data store that is attached to PingFederate
  - Creates, updates, disables, and deletes users
  - Enable the create, update, disable and delete capabilities independently
- Allows you to set a primary authentication method (email, SMS, voice) for new users that are provisioned to PingID SDK
- Manages all email, SMS, and voice pairings
  - Does not remove pairings that already exist in PingID SDK unless the device nickname of the pairing matches that of the attribute name in the connector
- See a list of Supported attributes reference

Components

PingID SDK provisioning connector

- Allows PingFederate to manage users in PingID SDK based on changes in an external user datastore.
- Includes a quick-connection template that pre-populates some configuration settings.

Intended audience

This document is intended for PingFederate administrators. Before you start, you should be familiar with the following:

- PingID SDK Overview in the PingID SDK documentation.
- The following sections of the PingFederate documentation:
  - Identity provider SSO configuration
  - Datastores
  - Managing IdP adapters
  - Managing digital signing certificates and decryption keys
  - SP connection management
  - Configuring outbound provisioning
  - Configuring outbound provisioning settings

Copyright © 2025 Ping Identity Corporation

System requirements

- PingFederate 9.0 or later.
- To allow PingFederate to make outbound connections to the PingID SDK API, you may need to whitelist the following domains in your firewall:
  - North America: sdk.pingid.com
  - Europe: sdk.pingid.eu
  - Australia: sdk.pingid.com.au

Setup

Get information from the PingID SDK

About this task

PingFederate needs connection information and an application ID from PingOne to manage users in the PingID SDK.

Steps

1. Sign on to the PingOne administrator console.
2. Export the configuration file for the PingID SDK:
   1. On the Setup > PingID > Client Integration screen, under Integrate with PingID SDK, click Download.
   2. Save the `pingidsdk.properties` file to a temporary location.
3. Get the application ID that PingOne assigned to PingFederate.
   1. Create an application to represent PingFederate in PingOne by completing the Configuring a new PingID SDK app procedure in the PingID *Administration Guide*.

      > **Important**
      >
      > Under **Alternate Authentication Methods**, enable authentication methods to make the associated attributes available to the provisioning connector.

   2. On the Applications > PingID SDK Applications screen, note the Application ID for the application that you created.

Upgrade an existing connector

About this task

Upgrading from a previous version of the PingID SDK Connector? Create a new connection and copy the configuration from your existing connection.

Copyright © 2025 Ping Identity Corporation

Steps

1. Note the attribute mapping configuration from your existing provisioning connection.
    1. On the PingFederate administrator console, on the Identity Provider screen, in the SP Connections area, click your existing provisioning connection.
    2. On the Outbound Provisioning screen, click Configure Provisioning.
    3. On the Manage Channels screen, click your existing channel configuration.
    4. On the Attribute Mapping screen, note your existing attribute mappings.
2. Disable and delete your existing provisioning connection.
    1. On the Identity Provider screen, in the SP Connections area, click Manage All.
    2. Disable your existing provisioning connection.
    3. Click Select Action, and then click Delete. Click Save.
3. Complete the steps in Deploy the provisioning connector.
4. Complete the steps in Configure provisioning to configure a new SP connection.
    1. On the Outbound Provisioning > Manage Channels > Attribute Mapping screen, configure the attribute mappings based on your notes from step 1.

Deploy the provisioning connector

About this task

This section describes the common steps required to install the PingFederate PingID SDK Connector.

Steps

1. Stop PingFederate.
2. Delete any existing PingID SDK Connector files in the PingFederate directory.
    1. Go to `<pf_install>/pingfederate/server/default/deploy`.
    2. Delete `pf-pingidsdk-quickconnection-<version>.jar`.
3. Deploy the provisioning connector.
    1. Extract the PingID SDK Connector distribution file, and then go to `dist`.
    2. Copy `pf-pingidsdk-quickconnection-<version>.jar` to `<pf_install> /pingfederate/server/default/deploy`.
4. Enable the PingFederate provisioning engine.
    1. In `<pf_install>/pingfederate/bin`, open `run.properties` for editing.
    2. Change `pf.provisioner.mode` to `STANDALONE`. Save the file.

Copyright © 2025 Ping Identity Corporation

> **Note**
>
> To configure the `FAILOVER` mode instead, see Deploying provisioning failover in the PingFederate *Server Clustering Guide*.

5. Start PingFederate.
6. If you operate PingFederate in a cluster, repeat steps 1 - 5 for each instance of PingFederate.

Configure provisioning

About this task

Configure PingFederate to provision users to PingID SDK.

> **Note**
>
> You can follow these steps to create a new SP connection, or you can modify an existing connection.

Steps

1. In the PingFederate administrator console, configure the data store that PingFederate will use as the source of user data. For instructions, see Datastores in the PingFederate documentation.
    - When targeting users and groups for provisioning, exclude the user account that you will use to administer users in your connection to PingID SDK. This prevents the PingFederate provisioning engine from interfering with the account that provisions users and groups.
2. Enable provisioning.
    1. On the System > Protocol Settings > Roles & Protocols screen, select Enable Identity Provider IdP Role and Support the Following.
    2. Select Outbound Provisioning and SAML 2.0. Click Save.
3. Create an SP connection with the PingID SDK quick connection template.
    1. On the Identity Provider screen, in the SP Connections area, click Create new.
    2. On the Connection Template screen, select Use a template for this connection.
    3. In the Connection Template list, select PingID SDK Connector. Click Next.
4. On the Connection Type screen, clear Browser SSO Profiles and select Outbound Provisioning. Click Next.
5. On the General Info screen, click Next.
6. On the Outbound Provisioning screen, configure the provisioning target and channel.

   See Configuring outbound provisioning in the PingFederate documentation.
    1. Click Configure Provisioning.
    2. On the Target screen, complete the Application ID field with the value that you noted in Get information from the PingID SDK.

Copyright © 2025 Ping Identity Corporation

> **Note**
>
> PingFederate verifies the credentials when you activate the channel and SP connection.

3. **Complete the PingID SDK information by doing one of the following:**

   Choose from:

   - **For PingFederate 9.0 and later: On the PingID SDK Properties line, click Choose File. Select the `pingidsdk.properties` file that you saved in Get information from the PingID SDK, and then click Open.**
   - **For PingFederate 8.x: Complete the required fields by copying and pasting the values from the `pingidsdk.properties` file that you saved in Get information from the PingID SDK.**

4. **From the Primary Authentication Method Upon Creation list, select a primary authentication method to set when the connector provisions new users to PingID SDK.**

   > **Note**
   >
   > Users are prompted to authenticate using their primary device. If the user has no mapped attribute value (or an invalid value) for the selected method, the primary device pairing is set to the next valid attribute in this order: `email 1`, `email 2`, `email 3`, `SMS 1`, `SMS 2`, `SMS 3`, `voice 1`, `voice 2`, `voice 3`.

5. **Optional: Under Provisioning Options, enable the provisioning features that you want.**

   See Provisioning options.

6. **In the Remove User Action list, select the deprovisioning action for users in PingID SDK. This is triggered when a user in the datastore is deleted, disabled, or no longer targeted for provisioning. Click Next.**

7. **On the Manage Channels screen, create a channel. Click Done.**

   See Managing channels in the PingFederate documentation.

   > **Tip**
   >
   > To set up synchronization, use the **SP Connection > Configure Channels > Channel > Attribute Mapping** screen to populate the `Username` attribute with a matching attribute from the data store. See Synchronize existing users.

8. **On the Outbound Provisioning screen, click Next.**

7. **On the Activation and Summary screen, above the Summary section, turn on the connection. Click Save.**

Provisioning options

Field Name	Description
User Create	Selected (default): PingFederate creates users. Cleared: PingFederate does not create users.

Copyright © 2025 Ping Identity Corporation

Field Name	Description
User Update	**Selected (default): PingFederate updates existing users.** **Cleared: PingFederate does not update existing users.**
User Disable/Delete	**Selected (default): PingFederate disable or delete users.** **Note** PingFederate can only re-enable a user if **User Update** is selected. **Cleared: PingFederate does not disable or delete users.** **Note** The **Remove User Action** option controls whether PingFederate deletes or disables users.
Provision Disabled Users	**Selected: When a user is flagged as "disabled" in the data store, PingFederate provisions the user with a "disabled" flag.** **Note** Only enable this option if **User Create** is enabled, and **Remove User Action** is set to **Disable**. **Cleared (default): PingFederate does not provision users that are flagged as "disabled" in the data store.**

Note
If any of the above options are cleared, PingFederate logs a warning in the user workflow section of `provisioner.log` when the related action fails.

Supported attributes reference

The following table lists the attributes that can be mapped for user provisioning.

For more details, see the following topics in the PingID SDK API documentation:

- Users API
- Offline devices (SMS) pairing API
- Offline devices (email) pairing API
- Offline devices (voice) pairing API

Copyright © 2025 Ping Identity Corporation

Important

The PingID SDK Connector manages all mapped email, SMS, and voice pairings. A device nickname of `Email 1` and an attribute name of `email1` are considered a pair. If the connector finds a device nickname (not attribute name) of `email1` (created by a previous version of the connector) and a device nickname of `Email 1`, the latter takes priority and the former is removed.

Attribute	Description
Username	PingID SDK's unique identifier for the user. A username cannot be changed. This attribute is required.
First Name	The given name of the user. For example, "Barbara" in the full name "Ms. Barbara Jane Jensen, III".
Last Name	The family name of the user. For example, "Jensen" in the full name "Ms. Barbara Jane Jensen, III".
Email 1	An email address to use for pairing.
Email 2	A second email address to use for pairing.
Email 3	A third email address to use for pairing.
SMS Number 1	A phone number to use for SMS pairing. The phone number must begin with the country code. The PingID SDK server removes all characters except for numbers, so the number doesn't require specific formatting.
SMS Number 2	A second phone number to use for SMS pairing.
SMS Number 3	A third phone number to use for SMS pairing.
Voice Number 1	A phone number to use for voice calls. The phone number must begin with the country code. The PingID SDK server removes all characters except for numbers, so the number doesn't require specific formatting.
Voice Number 2	A second phone number to use for voice calls.
Voice Number 3	A third phone number to use for voice calls.

Synchronize existing users

When the provisioning connector updates users between the data store and PingID SDK, it synchronizes users that share an identical `Username` attribute.

Example scenario:

- In PingID SDK, Janet's `Username` is `jsmith`.

Copyright © 2025 Ping Identity Corporation

- In your data store, Janet's `sAMAccountName` is `jsmith`.
- On the Attribute Mapping screen, you map the `Username` attribute to `sAMAccountName`.
- When the provisioning connector runs, the data store user is provisioned with a `Username` of `jsmith`. That matches Janet's existing `Username` in PingID SDK, so her information in the data store is synchronized to her PingID SDK account.

Release notes

Changelog

PingID SDK Connector 1.2.2 – October 2020

- Fixed an issue that caused the provisioning engine to see `email1` and `Email 1` as different devices. This could cause an error when the provisioning engine tried to add a new device instead of updating the existing device.

PingID SDK Connector 1.2.1 – September 2019

- Improved username validation and encoding to support API changes.

PingID SDK Connector 1.2 – May 2019

- Fixed a library issue that prevented users from being deprovisioned in PingFederate 8.1.4 and earlier.
- Fixed an issue that caused the connector to use the wrong EU hostname.
- Added a PingID SDK configuration file upload option to speed up configuration in PingFederate 9.0 and later.

PingID SDK Connector 1.1 – October 2018

- Added support for three Voice Number user attributes.
- Added support for Primary Authentication Method Upon Creation configuration.
- Improved error handling and reporting when PingID users contain no ID.

PingID SDK Connector 1.0.1 – August 2018

- Fixed an issue when a device is updated. Any unchanged devices on the user account were being removed and re-added to the user.

PingID SDK Connector 1.0 – April 2018

- Initial release
- Added support for PingID SDK v1.0
- Added support for user life cycle management (including creates, updates, disabling users, and deleting users)
- Added configuration options for workflow capabilities (for example, the ability to disable updates)

Copyright © 2025 Ping Identity Corporation

Known issues and limitations

Known issues

There are no known issues.

Known limitations

- Due to a limitation with PingFederate 8.1 and earlier versions, when configuring two SP connections with the same provisioner, the second connection built may be pre-populated with the channel from the first connection. To avoid conflicts, delete this pre-populated channel and create a unique channel for each connection.
- When an LDAP user is deleted in a targeted group distinguished name (DN), the provisioning connector does not propagate the deletion until a new user is added to the group. This limitation is compounded when the User Create provisioning option is disabled. For solutions, see SaaS provisioner does not remove the user in the Knowledge Base.
- Due to PingFederate limitations, user attributes cannot be cleared once set.
- Provisioning options may interact to produce undesired results if configured against recommendations. For example, we recommend that Provision Disabled Users be set to false when Remove User Action is Delete. If it is set to true, the following can occur:
    - A user exists in an active state in the user store and in PingID SDK. The user is disabled in the user store, so the provisioner deletes it from PingID SDK. An attribute of the user is updated in the user store. The provisioner will then re-create the user in PingID SDK in a disabled state.
- The PingID SDK Connector manages all mapped email, SMS, and voice pairings. A device nickname of `Email 1` and an attribute name of `email1` are considered a pair. If the connector finds a device nickname (not attribute name) of `email1` (created by a previous version of the connector) and a device nickname of `Email 1`, the latter takes priority and the former is removed.
- If the selected Primary Authentication Method Upon Creation has no mapped attribute value or an invalid value, the primary device pairing will be set to the next available mapped attribute value. The order is Email (1, 2, 3), SMS (1, 2, 3), Voice (1, 2, 3).

Assigning a user to multiple applications

- When managing multiple PingID SDK Applications, it is possible for a user to have a different active status in each application due to device pairings.
- The PingID SDK create and suspend endpoints are global across all applications managed by your admin account. This means that if a user's status changes in one application, the same action will be taken on all applications managed by your admin account.
    - If you create a user in one application with a device to pair, the user will be created in an `active` state in that application and a `not_active` state in other applications.
    - If there are two applications, each of which has an SP connection set up for it. In one SP connection, the Remove User Action is set to Disable and in the other it is set to Delete. A user in the directory has been targeted by both connections and as such, that user exists in both applications. The user is then disabled in the directory. Provisioner logs will show a delete and a create. The user will end up in a suspended state in both applications.

Copyright © 2025 Ping Identity Corporation

- If there are two applications, each of which has an SP connection set up for it. In one SP connection, the Remove User Action is set to Disable and in the other it is set to Delete. Create a user in application A with an email or other device to pair. Create a user with the same username in the directory without the email or device. The user will then exist in an `active` state in application A and a `not_active` state in application B. Disable the user in the directory. The user will then be suspended in both applications. Enable the user in the directory. The user will then be `not_active` in both applications.

Download manifest

The distribution `.zip` archive for the connector contains the following:

- `ReadMeFirst.pdf` – contains links to this online documentation.
- `/legal`:
  - `Legal.pdf` – copyright and license information.
- `/dist` – contains libraries needed for the connector:
  - `pf-pingidsdk-quickconnection-<version>.jar` – PingFederate PingID SDK Connector

Copyright © 2025 Ping Identity Corporation

Overview of PingOne Integrations

PingOne Integration Kit

The PingOne Integration Kit adds PingOne datastore, password credential validator, and provisioning connector options to PingFederate.

The user guide also provides instructions for creating an OpenID Connect (OIDC) integration with PingOne.

These allow you to integrate PingFederate with PingOne and services such as PingOne MFA in a variety of ways.

PingOne MFA Integration Kit

The PingOne MFA Integration Kit allows PingFederate to use the PingOne MFA service for multi-factor authentication (MFA).

PingOne Risk Integration Kit

The PingOne Risk Integration Kit allows PingFederate to communicate with PingOne Risk service for risk-based authentication.

PingOne Verify Integration Kit

The PingOne Verify Integration Kit allows PingFederate to use the PingOne Verify service for personal identity verification based on government photo ID.

PingOne Davinci Integration Kit

The PingOne Davinci Integration Kit allows PingFederate to use PingOne as an identity provider (IdP).

PingAM Integration Kit

The PingAM Integration Kit allows PingFederate to communicate with PingAM and provide a rich sign-on orchestration experience.

Magic Link Integration Kit

The Magic Link Integration Kit enables PingFederate to send users a one-time link through their verified email address to sign on without a password.

PingOne Protect Integration Kit

The PingOne Protect Integration Kit allows PingFederate to communicate with PingOne Protect for risk-based authentication.

By sending transaction information and an optional device profile to PingOne when a user signs on, PingFederate can get a security risk evaluation for the sign-on event. Including the risk evaluation in your PingFederate authentication policy allows you to dynamically adjust the user's authentication requirements each time they sign on.

Copyright © 2025 Ping Identity Corporation

PingOne Integration Kit

The PingOne Integration Kit adds PingOne datastore, password credential validator, and provisioning connector options to PingFederate. The user guide also provides instructions for creating an OpenID Connect (OIDC) integration with PingOne. These allow you to integrate PingFederate with PingOne and services such as PingOne MFA in a variety of ways.

Components

You can use these components individually or together.

Data store and password credential validator

- The datastore allows PingFederate to retrieve user attributes from PingOne.
- The password credential validator allows PingFederate to validate user credentials against the PingOne data store during sign on.

Provisioning connector

- Allows the PingFederate provisioning engine to manage users and groups in PingOne.
- Allows PingFederate to manage users' SMS, email, and voice devices for PingOne MFA.
- Allows PingFederate to create custom attributes.

The user guide also provides instructions for creating an OpenID Connect integration with PingOne.

Intended audience

This document is intended for PingFederate administrators.

If you need help during the setup process, learn more in the following resources:

- Introduction to PingOne in the PingOne documentation
- Strong Authentication (MFA) in the PingOne MFA documentation
- The following sections of the PingFederate documentation:
  - Datastores
  - Password Credential Validators
  - SP connection management
  - Configuring outbound provisioning
  - Configuring outbound provisioning settings

System requirements

- PingFederate 10.3 or later

Copyright © 2025 Ping Identity Corporation

- To allow PingFederate to make outbound HTTPS connections, you might need to allow the following hostnames in your firewall:
  - https://api.pingone.com, https://api.pingone.asia, or https://api.pingone.eu
  - https://auth.pingone.com, https://auth.pingone.asia, or https://auth.pingone.eu

Setup

Connecting PingFederate to PingOne

To allow PingFederate to communicate with PingOne, create a connection. You can use this connection for all PingFederate components that communicate with PingOne.

About this task

> **Note**
>
> If you already have a connection from PingOne to PingFederate, ensure that the **Identity admin** and **Environment admin** roles are enabled. Skip the steps below.

Steps

1. Create credentials in PingOne:
   1. On the PingOne administrative console, go to Integrations > PingFederate and click the + icon.
   2. On the Create Connection Profile window, enter a connection name, then click Save.
   3. After the New Credential Created window opens, click Copy To Clipboard, then Done.
2. Use the credentials to create a connection in PingFederate:
   1. On the PingFederate administrative console, go to System > External Systems > PingOne Connections and click + Add Connection.
   2. On the Add Connection window, paste the credential information in the Paste The Credential Here field.
   3. In the Connection Name field, enter a unique name and click Save.

Datastore and password credential validator

You can configure PingFederate to use PingOne as a datastore and password credential validator.

Datastore

The datastore enables PingFederate to retrieve user attributes from PingOne.

> **Note**
>
> You can use the datastore without configuring the password credential validator.

Copyright © 2025 Ping Identity Corporation

Learn more in Deploying the datastore and password credential validator files and Configuring the datastore.

Password Credential Validator

The password credential validator enables PingFederate to validate user credentials against PingOne when a user tries to sign on.

> **Note**
>
> To use the password credential validator, you must set up the datastore first.

Learn more in Deploying the datastore and password credential validator files and Configuring the password credential validator.

Deploying the datastore and password credential validator files

To get started with the integration, deploy the datastore (DS) and password credential validator (PCV) files from the PingOne Integration Kit to your PingFederate directory.

Steps

1. Download the PingOne Integration Kit `.zip` archive from the Ping Identity Integration Directory.
2. Stop PingFederate.
3. If you are upgrading an existing deployment, back up your customizations and delete earlier versions of the integration files:
    1. Back up any PingOne Integration Kit files that you customized in `<pf_install>/pingfederate/server/default/conf/` including `pingfederate-messages.properties`.
    2. Delete the following files from `<pf_install>/pingfederate/server/default/deploy`:
        - `pf-pingone-datastore-<version>.jar`
        - `pf-pingone-pcv-<version>.jar`
        - `pf-p14c-datastore-<version>.jar`
        - `pf-p14c-pcv-<version>.jar`
        - `pf-pingone-quickconnection-<version>.jar`
        - `pf-p14c-quickconnection-<version>.jar`
        - `pf-pingoneuserstore-connector-<version>.jar`
4. From the `.zip` archive, copy the contents of `dist` to `<pf_install>/pingfederate/server/default`.
5. If you backed up any customized files, modify the new files with your customizations.

Copyright © 2025 Ping Identity Corporation

6. Merge the contents of `pingone-auth-messages.properties` into `<pf_install>/pingfederate/server/default/conf/pingfederate-messages.properties`.
    1. If a key does not exist in `pingfederate-messages.properties`, add the line from `pingone-auth-messages.properties`.
    2. If a key already exists in `pingfederate-messages.properties`, update it from `pingone-auth-messages.properties`.
7. Start PingFederate.
8. If you operate PingFederate in a cluster, repeat steps 2-7 for each engine node.

Configuring the datastore

To retrieve user attributes from PingOne, add PingOne as a datastore in PingFederate.

About this task

Learn more in Datastores in the PingFederate documentation.

Steps

1. On the PingFederate administrative console, create a new datastore.
    1. Go to System > Data & Credential Stores.
    2. Click Add New Data Store.
2. On the Data Store Type tab, set the basic datastore attributes.
    1. In the Name field, enter a name for the datastore.
    2. In the Type list, select PingOne Data Store.
    3. Click Next.
3. On the Configure Data Store Instance screen, under Attribute Selector, define local attributes and map them to the values of PingOne attributes.
    1. Click Add a new row to 'Attribute Selector'.
    2. In the Local Attribute field, enter a name of your choice for an attribute.
    3. In the PingOne Attribute field, map your local attribute to the value of the matching PingOne for Customers attribute by entering `JSON Pointer syntax`. See JSON Pointer syntax reference.
    4. On the row that you added, click Update.
    5. To add more attributes, repeat steps a-d.
4. Complete the connection to PingOne.
    1. In the PingOne Envrionment list, select your environment.
    2. Click Next.

Copyright © 2025 Ping Identity Corporation

5. Configure the other fields as necessary.

   Learn more in Datastore configuration reference.

6. On the Actions screen, confirm that PingFederate can connect to PingOne.

   1. Click Test Connection.

   2. Resolve any connection issues.

   3. Click Next.

7. On the Summary screen, click Save.

JSON Pointer syntax reference

JavaScript Object Notation (JSON) Pointer defines a syntax for identifying a specific value within a JSON payload. Using the sample payload and JSON Pointer examples below, identify the attributes that you want to use to populate your attribute contract.

You can find a complete technical description of JSON Pointer syntax in JavaScript Object Notation (JSON) Pointer on ietf.org.

Learn more in the following sections of the PingOne API documentation:

- Learn more about the PingOne JSON payload structure in READ One User
- You can find descriptions of the user properties in Users (User Properties section)

Example JSON payload

Copyright © 2025 Ping Identity Corporation

```
{
    "_links": {
        "self": {
            "href": "https://api.pingone.com/v1/environments/5da98f13-ad62-4234-ba04-2b3e2a05c8ca/users/
c3042000-188f-4bc7-a269-d0a1602cf5baf"
        },
        "environment": {
            "href": "https://api.pingone.com/v1/environments/5da98f13-ad62-4234-ba04-2b3e2a05c8ca"
        },
        "population": {
            "href": "https://api.pingone.com/v1/environments/5da98f13-ad62-4234-ba04-2b3e2a05c8ca/populations/
a701eb22-849d-4168-b756-6c20fc087c2f"
        },
        "password": {
            "href": "https://api.pingone.com/v1/environments/5da98f13-ad62-4234-ba04-2b3e2a05c8ca/users/
c3042000-188f-4bc7-a269-d0a1602cf5baf/password"
        },
        "password.reset": {
            "href": "https://api.pingone.com/v1/environments/5da98f13-ad62-4234-ba04-2b3e2a05c8ca/users/
c3042000-188f-4bc7-a269-d0a1602cf5baf/password"
        },
        "password.set": {
            "href": "https://api.pingone.com/v1/environments/5da98f13-ad62-4234-ba04-2b3e2a05c8ca/users/
c3042000-188f-4bc7-a269-d0a1602cf5baf/password"
        },
        "password.validate": {
            "href": "https://api.pingone.com/v1/environments/5da98f13-ad62-4234-ba04-2b3e2a05c8ca/users/
c3042000-188f-4bc7-a269-d0a1602cf5baf/password"
        },
        "password.recover": {
            "href": "https://api.pingone.com/v1/environments/5da98f13-ad62-4234-ba04-2b3e2a05c8ca/users/
c3042000-188f-4bc7-a269-d0a1602cf5baf/password"
        },
        "account.sendVerificationCode": {
            "href": "https://api.pingone.com/v1/environments/5da98f13-ad62-4234-ba04-2b3e2a05c8ca/users/
c3042000-188f-4bc7-a269-d0a1602cf5baf"
        }
    },
    "id": "c3042000-188f-4bc7-a269-d0a1602cf5baf",
    "environment": {
        "id": "5da98f13-ad62-4234-ba04-2b3e2a05c8ca"
    },
    "population": {
        "id": "a701o5ad-849d-4168-b756-6c20fc087c2f"
    },
    "createdAt": "2020-01-08T20:18:31.264Z",
    "email": "jsmith@example.com",
    "enabled": true,
    "lifecycle": {
        "status": "ACCOUNT_OK"
    },
    "mfaEnabled": false,
    "name": {
        "given": "Janet",
        "family": "Smith"
    },
    "nickname": "Janet",
    "updatedAt": "2020-01-08T20:59:56.871Z",
    "username": "jsmith"
}
```


Copyright © 2025 Ping Identity Corporation

JSON Pointer syntax:

Description	JSON Pointer	Example value
First name	/name/given	Janet
Last name	/name/family	Smith
Population ID	/population/id	a701o5ad-849d-4168-b756-6c20fc087c2f
User ID	/id	c3042000-188f-4bc7-a269-d0a1602cf5baf

Datastore configuration reference

Learn more about the fields available for datastore configuration.

Field	Description
API Request Timeout	The amount of time, in milliseconds, that PingFederate waits for a response to a request, or to establish a connection with the Google Verified Access API. The default value is `10000`. **Note** To disable the timeout, enter a value of `0`.
Retry Request	Retry a user data retrieval request if the API fails due to rate-limiting and you have Retry Error Codes configured.
Maximum Retries Limit	The maximum number of retries that the adapter attempts if the API fails and you have Retry Error Codes configured.
Retry Error Codes	A comma-separated list of error code values for API re-attempts, like `429` or `503`.
Enable User Group Info Retrieval	If selected, the adapter makes a request to retrieve user group info during datastore lookup. The `memberOfGroupIDs` and `memberOfGroupNames` core contract attributes will be populated with retrieved values.

Copyright © 2025 Ping Identity Corporation

Field	Description
Enable Device Info Retrieval	If selected, the adapter makes a request to retrieve device info during datastore lookup. The `devices.phone` and `devices.email` core contract attributes will be populated with retrieved values.
Proxy Settings	Defines proxy settings for outbound HTTP requests. The default value is `System Defaults`.
Custom Proxy Host	The proxy server host name to use when Proxy Settings is set to custom. This field is blank by default.
Custom Proxy Port	The proxy server port to use when Proxy Settings is set to custom. This field is blank by default.

Configuring the password credential validator

To allow PingFederate to validate user credentials against PingOne when a user tries to sign on, add the password credential validator (PCV).

About this task

Learn more in Password Credential Validators in the PingFederate documentation.

Steps

1. On the PingFederate administrative console, create a new datastore.
    1. Go to System > Data & Credential Stores > Password Credential Validators.
    2. Click Create New Instance.
2. On the Type tab, set the basic PCV attributes.
    1. In the Instance Name field, enter a name for the PCV.
    2. In the Instance ID field, enter a unique identifier for the PCV.
    3. In the Type list, select PingOne Password Credential Validator.
    4. Click Next.
3. On the Instance Configuration tab, configure the password credential validator, then click Next.
    1. Optional: In the Authentication Error Overrides section, configure error overrides as shown in Customizing authentication error messages.
    2. In the PingOne Datastore list, select the data store that you configured in Configuring the datastore.
    3. Configure the other fields as necessary.

Copyright © 2025 Ping Identity Corporation

Learn more in Password credential validator configuration reference.

4. On the Extended Contract tab, add any attributes that you want to include in the contract, then click Next.
5. On the Summary tab, check your configuration and click Save.

Customizing authentication error messages

When a user fails to sign on, PingFederate parses the response from the datastore and shows a corresponding error message on the login page. In the password credential validator configuration, you can customize which error message is triggered based on the response from the datastore.

About this task

By adding rules that check the datastore response for specific text, you can trigger one of the default error messages or a custom error message.

The default error messages are defined in `<pf_install>/pingfederate/server/default/conf/language-packs/pingfederate-messages.properties`. This file is part of the PingFederate localization framework, and can be adapted to other languages. Learn more in Localizing messages for end users in the PingFederate documentation.

Steps

1. Under Authentication Error Overrides, click Add a new row to 'Authentication Error Overrides'.
2. In the Match Expression field, enter the text, including wildcards (`*`), that you want to trigger the error message.

   For example, `*disabled*` triggers an error message when the word "disabled" appears anywhere in the response from the datastore.
3. Set the error message that you want to show:

   Choose from:

   - To use a default error message, in the Error list, select a default error message.
   - To define a new error message:

     1. In the Error list, select Custom Message Key.
     2. In the Error Message Key field, enter a unique key, such as `pingone.authn.srvr.msg.<your.key.suffix>`.
     3. Open the `<pf_install>/pingfederate/server/default/conf/language-packs/pingfederate-messages.properties` file.
     4. Add a line with the error message key and message.

        Use the following format:

Copyright © 2025 Ping Identity Corporation

```
pingone.authn.srvr.msg.<your.key.suffix>=There was an error. Please try again.
```

5. Save the file.
6. If you support multiple languages, repeat steps c-e for each equivalent file in `language-packs`.
7. Restart PingFederate to reload the message files.

4. Click Update.

Password credential validator configuration reference

Learn more about the fields available for password credential validator configuration.

Field	Description
Case-Sensitive Matching	Allows case-sensitive expression and LDAP error matching.
Display Name Attribute	Indicates the display name to use for personalizing messages in password reset, account unlock, and username recovery flows. The default value is: `name.formatted`
PingID Username Attribute	Indicates the username to use for PingID-based password reset flow. The default value is: `username`
Username Attribute	Indicates the username to use in account recovery flow. The default value is: `username`
Mail Verified Attribute	Indicates whether the user's email address is verified. This attribute is used in password reset, account unlock, and username recovery flows. Acceptable values are `true` or `false`. The default value is: `true`

Copyright © 2025 Ping Identity Corporation

Field	Description
SMS Attribute	Indicates the phone number to use for SMS password reset flow. The default value is: `mobilePhone`
Enable User Group Info Retrieval	If selected, the adapter makes a request to retrieve user group info during credential authentication. It populates the `memberOfGroupIDs` and `memberofGroupNames` core contract attributes with the retrieved values. This checkbox is selected by default.
Enable Device Info Retrieval	If selected, the adapter makes a request to retrieve device info during credential authentication. It populates the `devices.phone` and `devices.email` core contract attributes with the retrieved values. This checkbox is selected by default.
Enable Password Expiry Info Retrieval	If selected, the adapter makes a request to the password policy endpoints to retrieve expiration information during credential authentication. It populates the `passwordExpiryTime` core contract attribute with the retrieved values. This checkbox is selected by default. **Note** If selected, this configuration will prompt the HTML form adapter to show a password expiration warning as described by the *Show Password Expiring Warning* field in Configuring an HTML Form Adapter instance.

Provisioning connector

This provisioning connector allows PingFederate to create, update, disable and delete users in PingOne based on changes in an external user datastore. The provisioning connector also includes configuration options for workflow capabilities, such as the ability to disable user updates.

User and group management

The PingOne Integration Kit synchronizes users and groups from your datastore to PingOne. The behavior of each provisioning capability is described below.

You can configure these capabilities in the Creating a provisioning connection step of the setup process.

Copyright © 2025 Ping Identity Corporation

Synchronizing existing users

PingFederate synchronizes users based on the `Username` attribute in PingOne. If a user already exists in your datastore and PingOne, mapping this attribute correctly links the two records together.

For example:

- In PingOne, Janet's `Username` is `jsmith`.
- In your datastore, Janet's `sAMAccountName` is `jsmith`.
- On the Attribute Mapping tab of your provisioning connection configuration, you map the `Username` attribute to `sAMAccountName`.
- When the provisioning connector runs, the datastore user is provisioned with a `Username` of `jsmith`. That matches Janet's existing `Username` in PingOne, so her information in the datastore is synchronized to her PingOne account.

You can map the matching datastore attribute when you configure your channel in Creating a provisioning connection.

User provisioning

PingFederate provisions users when one of the following happens:

- A user is added to the datastore group or filter that is targeted by the provisioning connector.
- A user with "disabled" status is added to the datastore group or filter that is targeted by the provisioning connector, and the Provision disabled users provisioning option is enabled.

The Source Location tab of your provisioning connection configuration defines which users PingFederate targets for provisioning.

User updates

PingFederate updates users when a user attribute changes in your datastore.

The Attribute Mapping tab of your provisioning connection configuration defines which attributes PingFederate monitors for changes.

Copyright © 2025 Ping Identity Corporation

User deprovisioning (disabling)

PingFederate deprovisions users when one of the following happens:

- A user is deleted from the user store.
- A user is disabled in the user store.
- A user is removed from the datastore group or filter, which is targeted by the provisioning connector.

Synchronizing existing groups

PingFederate synchronizes groups from the datastore to the target service based on the group name.

For example:

- In PingOne, there is a group is named `Accounting`.
- In your datastore, there is a group with a `CN` of `Accounting`.
- When the provisioning connector runs, the two groups are synchronized.

Group provisioning

PingFederate provisions groups when a group is added to the datastore filter that is targeted by the provisioning connector.

The Source Location tab in your provisioning connection configuration defines which groups PingFederate targets for provisioning and monitors for changes.

Group name updates

PingFederate renames groups when they are renamed in the datastore.

Group membership updates

PingFederate updates group memberships when memberships change in the datastore, whether the change is in the group's properties or a user's properties.

Group memberships in the datastore overwrite the group memberships in PingOne.

Copyright © 2025 Ping Identity Corporation

Group deletion

PingFederate deletes groups when any of the following happen:

- The group is deleted in the datastore.
- The group is removed from the datastore group or filter, which is targeted by the provisioning connector.

Group deletions are permanent and cannot be undone.

Authentication method management

The PingOne Connector manages all mapped email and SMS multi-factor authentication (MFA) attributes.

Nicknames

PingOne assigns nicknames to authentication methods (also called devices) to identify them on user-facing screens, such as the device selection screen.

The PingOne Connector uses nicknames when provisioning and synchronizing a user's authentication methods.

The "managed" nicknames used by the provisioning connector are:

- `SMS 1`
  - `SMS 2`
  - `SMS 3`
- `Email 1`
  - `Email 2`
  - `Email 3`
- `Voice 1`
  - `Voice 2`
  - `Voice 3`

Mapping attributes to nicknames

Each device nickname is associated with one attribute on the Attribute Mapping tab of the channel configuration. For example, the `Email 3` nickname holds the value of the `MFA Device Email 3` attribute.

Copyright © 2025 Ping Identity Corporation

You can map these attributes in the Configuring a channel procedure.

Synchronization

When synchronizing a user's authentication methods, the provisioning connector behaves as follows:

Synchronization scenarios

Scenario	Action
A device exists with a managed nickname, but the value does not match the value in the datastore	The provisioner deletes and re-creates the device with the value from the datastore
A value matches between PingOne and the datastore, but the device uses an unmanaged nickname	The provisioner deletes and re-creates the device with the appropriate managed nickname
A device exists with an unmanaged nickname and the value does not match the value in the datastore	The provisioner does not make any changes

Maximum number of authentication methods

Although the provisioning connector supports up to three SMS attributes, three email attributes, and three voice attributes, PingOne accepts a maximum of five authentication methods per user by default. This maximum can be adjusted in the PingOne settings.

Upgrading an existing deployment

If you're upgrading from a previous version of the PingOne Integration Kit, note your existing service provider connection configuration and create a new connection.

Steps

1. In the PingFederate administrative console, go to Applications > Integration > SP Connections and select your existing SP connection.
2. Note the attribute mappings for your existing SP connection.

   Learn more in Mapping attributes in the PingFederate documentation.
3. Delete your existing SP connection.
    1. Go to Applications > Integration > SP Connections.
    2. For your existing connection, click Select action, then click Delete.

Copyright © 2025 Ping Identity Corporation

3. Click Confirm.

4. Complete the steps in Deploying the provisioning connector.

5. Complete the steps in Creating a provisioning connection.

    - On the Outbound Provisioning > Manage Channels > Channel > Attribute Mapping tab, configure the attribute mappings based on your notes.

Deploying the provisioning connector

To get started with the integration, deploy the provisioning connector file from the PingOne Integration Kit to your PingFederate directory.

Steps

1. Download the PingOne Integration Kit `.zip` archive from the Add-ons tab of the PingFederate downloads page.

2. Stop PingFederate.

3. If you are upgrading an existing deployment, back up your customizations and delete earlier versions of the integration files:

    1. Back up any PingOne Integration Kit files that you customized in `<pf_install>/pingfederate/server/default/conf/` including `pingfederate-messages.properties`.

    2. Delete the following files from `<pf_install>/pingfederate/server/default/deploy`:

        - `pf-pingone-datastore-<version>.jar`
        - `pf-pingone-pcv-<version>.jar`
        - `pf-p14c-datastore-<version>.jar`
        - `pf-p14c-pcv-<version>.jar`
        - `pf-pingone-quickconnection-<version>.jar`
        - `pf-p14c-quickconnection-<version>.jar`
        - `pf-pingoneuserstore-connector-<version>.jar`

4. From the `.zip` archive, copy the contents of `dist` to `<pf_install>/pingfederate/server/default/`.

5. If you backed up any customized files, modify the new files with your customizations.

6. Merge the contents of `pingone-auth-messages.properties` into `<pf_install>/pingfederate/server/default/conf/pingfederate-messages.properties`.

    1. If a key does not exist in `pingfederate-messages.properties`, add the line from `pingone-auth-messages.properties`.

    2. If a key already exist in `pingfederate-messages.properties`, update it from `pingone-auth-messages.properties`.

Copyright © 2025 Ping Identity Corporation

7. Enable the PingFederate provisioning engine.
    1. In `<pf_install>/pingfederate/bin`, open `run.properties` for editing.
    2. Change `pf.provisioner.mode` to `STANDALONE`.

    > **Note**
    >
    > To configure the `FAILOVER` mode instead, see Deploying provisioning failover in the PingFederate documentation.

    3. Save the file.
8. Start PingFederate.
9. If you operate PingFederate in a cluster, repeat steps 2-6 and step 8 for each engine node.

Enabling provisioning in PingFederate

To use PingFederate for provisioning, configure an external datastore and enable outbound provisioning.

About this task

Your external datastore acts as the source of data for provisioning. PingFederate also uses an internal datastore to store the state of synchronization between the source datastore and the target datastore.

Learn more in Datastores and Configuring outbound provisioning settings in the PingFederate documentation.

Steps

1. Configure the external datastore that PingFederate will use as the source of user data.

   Learn more in Adding a new datastore in the PingFederate documentation.

   > **Note**
   >
   > When targeting users and groups for provisioning, exclude the user account that you will use to administer users in your connection to PingOne. This prevents the PingFederate provisioning engine from interfering with the account that provisions users and groups.

2. Go to System > Server > Protocol Settings.
3. On the Outbound Provisioning tab, select the PingFederate internal datastore, then click Save.

   Learn more in Configuring outbound provisioning settings in the PingFederate documentation.

Creating a provisioning connection

To allow PingFederate to manage users in PingOne, configure a service provider (SP) connection.

Copyright © 2025 Ping Identity Corporation

Steps

1. In the PingFederate administrative console, go to Applications > Integration > SP Connections and click Create Connection to create a new SP connection:
2. Configure the SP connection with the PingOne quick connection template:
    1. On the Connection Template tab, select Use a template for this connection.
    2. In the Connection Template list, select PingOne Connector, then click Next.
    3. On the Connection Type tab, click Next.
    4. On the General Info tab, click Next.
3. Configure provisioning:

   Learn more in Configuring outbound provisioning in the PingFederate documentation.

    1. On the Outbound Provisioning tab, click Configure Provisioning.
    2. On the Target tab, in the PingOne Environment list, select your environment.
    3. Optional: In the Provisioning Options section, customize the provisioning connector behavior by referring to Provisioning options reference, then click Next.
    4. On the Manage Channels tab, configure a channel as shown in Configuring a channel.
4. On the Outbound Provisioning tab, click Next.
5. On the Activation and Summary tab, above the Summary section, turn on the connection. Click Save.

Provisioning options reference

The PingOne Integration Kit allows you to enable actions independently and customize other provisioning options.

Field Name	Description
Region	Determines the PingOne API that the provisioning engine communicates with. Select the region that appears on Settings > Environment > Properties in PingOne.
User Create	*Selected (default)* PingFederate creates users in PingOne. *Cleared* PingFederate doesn't create users in PingOne.

Copyright © 2025 Ping Identity Corporation

Field Name	Description
User Update	***Selected (default)*** PingFederate updates existing users in PingOne. ***Cleared*** PingFederate does not update existing users in PingOne.
User Disable / Delete	***Selected (default)*** PingFederate disables or deletes users in PingOne. **Note** PingFederate can only re-enable a user if **User Update** is selected. ***Cleared*** PingFederate does not disable or delete users in PingOne.
Provision Disabled Users	This option applies when the User Create option is selected, and the provisioning engine targets a user in the datastore that has a "disabled" status. ***Selected (default)*** PingFederate creates the user in PingOne with a "disabled" status. ***Cleared*** PingFederate does not create the user in PingOne.
Note If any of the above options are cleared, PingFederate logs a warning in the user workflow section of the `provisioner.log` when the related action fails.	
Remove User Action	This option applies when User Disable / Delete is selected, and either: • A previously provisioned user no longer meets the condition set on the Source Location screen. • A user has been disabled or deleted from the datastore. Disable (default) PingFederate disables the user in PingOne. Delete PingFederate deletes the user from PingOne.

Copyright © 2025 Ping Identity Corporation

Field Name	Description
MFA User Device Management	**If you are using the PingOne MFA service, this setting controls how PingFederate manages user devices:** ***Merge with devices in PingOne (default)*** **When a device is added in the datastore, PingFederate adds it to the user's existing devices in PingOne. PingFederate does not remove any devices added in PingOne by other sources.** **When an email or SMS device is updated in the datastore, and the value does not exist in PingOne, PingFederate adds it as a new device.** **Note** PingOne accepts a maximum of five authentication methods per user by default. This maximum can be adjusted in the PingOne settings. If adding a new device hits this limit, PingFederate logs the "User Device Limit Exceeded" error. It does not try to add the device again until the user is updated. ***Overwrite devices in PingOne*** **When a device is added or removed in the datastore, PingFederate overwrites the user's devices in PingOne with those from the datastore.** **If a device in PingOne does not exist in the datastore, PingFederate removes it from PingOne.** **If a device in the datastore does not exist in PingOne, PingFederate adds it in PingOne.**
Default Authentication Method for New Users	**Select a default authentication method to set when the connector provisions new users to PingOne:** **Note** Users are prompted to authenticate using their default device. If the user has no mapped attribute value (or an invalid value) for the selected method, the primary device pairing is set to the next valid attribute in this order: `Email 1`, `Email 2`, `Email 3`, `SMS 1`, `SMS 2`, `SMS 3`, `Voice 1`, `Voice 2`, `Voice 3`. PingFederate attempts to provision all the user's valid MFA devices, which might result in the user not having the selected method as their default device.

Configuring a channel

To complete your PingOne provisioning configuration, map the necessary attributes in your channel configuration.

About this task

Copyright © 2025 Ping Identity Corporation

Learn more about the following steps in Managing channels in the PingFederate documentation.

Steps

1. On the Manage Channels tab, click Create.
2. On the Channel Info tab, in the Channel Name field, enter a unique name, then click Next.
3. On the Source tab, in the Active Data Store list, select the datastore that you configured in Supported attributes reference, then click Next.
4. On the Source Settings tab, click Next.
5. On the Source Location tab, in the Base DN field, enter a base DN that includes the users that you want to provision.
6. In the Users section, complete the Group DN or Filter fields to identify the users that you want to provision, then click Next.
7. On the Attribute Mapping tab, map the required attributes.

   You can find a list of attributes in the Supported attributes reference. Learn more about attribute mapping in Mapping attributes in the PingFederate documentation.

   > **Tip**
   >
   > You can also map custom attributes after adding them. Learn more about adding custom attributes in Adding user attributes and the PingOne Platform API Reference.

   1. On the Username row, click Edit.
   2. On the `Username` Attribute Mapping tab, in the Root Object Class and Attribute lists, select the attribute in your datastore that matches the `Username` attribute in PingOne.

      Learn more in User and group management.
   3. Click Add Attribute, and then click Done.
   4. On the Population ID row, click Edit.
   5. On the Population ID mapping tab, do one of the following:

      Choose from:

      - Map a static population ID:

        1. In the Default Value list, select the PingOne population that you want to target, then click Done.

           > **Note**
           >
           > The **Default Value** list is only available if you entered the correct client credentials from Connecting PingFederate to PingOne. The provisioner will populate the list accordingly.

      - Map a dynamic population ID.

Copyright © 2025 Ping Identity Corporation

> **Tip**
>
> Learn more about getting the ID for a population in Viewing populations in the PingOne documentation.

1. In the Root Object Class and Attribute lists, select the user attribute in your datastore that contains the population ID.
2. Click Add Attribute, and then click Done.

6. If you want to manage users and devices for the PingOne MFA service, map any combination of the `MFA Device Email` and `MFA Device SMS` attributes, and map or assign the `MFA Enabled` attribute.

   Learn more in Supported attributes reference.

7. Click Next.

8. On the Activation and Summary tab, on the Channel Status row, click Active, then click Done.

9. On the Manage Channels tab, click Done.

Supported attributes reference

The following standard attributes can be mapped for user provisioning to PingOne. Custom attributes can also be created and mapped if desired.

> **Note**
>
> For details about authentication device attributes and synchronization behavior, see Authentication method management.

User attributes

Attribute	Description
Username	A unique identifier for the user in PingOne. This attribute is required.
Email	The user's email address, which must be valid (for example, "jsmith@example.com"). This attribute is required.
Population ID	The connector provisions users to this PingOne population. This attribute is required. Select a population name from the Default value list, or map a dynamic population ID. **Note** You must create at least one population before you can create users. An individual user cannot belong to more than one population simultaneously, but users can be moved to a different populations.
Account ID	The user's account ID. This attribute can be an organization-specific identifier.

Copyright © 2025 Ping Identity Corporation

Attribute	Description
Authoritative IdP	The external identity provider for the user. This is used for identity provider discovery. For details, see Identifier first action in the PingOne API documentation.
City	The city or locality component for the user's mailing address.
Country	The country name component. When specified, the value must be in `ISO 3166-1` "alpha-2" code format. For example, the United States and Sweden are "US" and "SE", respectively.
External ID	A string that is an identifier for the resource as defined by the provisioning client. The attribute may simplify the correlation of the user in PingOne with the user's account in another system of record.
First Name	The given name of the user, or first name in most Western languages (for example, 'Barbara' given the full name 'Ms. Barbara Jane Jensen, III').
Force Change Password	Indicates if the user is forced to change their password at next log in. Valid options include: • True • False If no value is provided, this will default to false and the user will not be forced to change their password.
Full Name	The full name, including all middle names, titles, and suffixes as appropriate, formatted for display (for example, 'Ms. Barbara Jane Jensen, III').
Honorific Prefix	The honorific prefix of the user, or title in most Western languages, (for example, 'Ms.' given the full name 'Ms. Barbara Jane Jensen, III').
Honorific Suffix	The honorific suffix of the user, or suffix in most Western languages (for example, 'III' given the full name 'Ms. Barbara Jane Jensen, III').
Job Title	The user's title, such as "Vice President".
Last Name	The family name of the user, or last name in most Western languages (for example, 'Jensen' given the full name 'Ms. Barbara Jane Jensen, III').
Locale	Used to indicate the user's default location for purposes of localizing such items as currency, date time format, or numerical representations. A valid value is a language tag as defined in `RFC 5646` (for example, "en-US").
MFA Device Email 1	An email address that the user has paired with the PingOne MFA service.
MFA Device Email 2	An email address that the user has paired with the PingOne MFA service.
MFA Device Email 3	An email address that the user has paired with the PingOne MFA service.

Copyright © 2025 Ping Identity Corporation

Attribute	Description
MFA Device SMS 1	An SMS phone number that the user has paired with the PingOne MFA service. You must enter a valid phone number in international format, including the leading `+` character. For example, `+14155552671`. **Important** To avoid issues with message delivery, always include the country code. Phone number formats across the globe are constantly expanding and changing. The following list contains a few examples of valid ways to format the phone number: • +1.5125201234 • +15125201234 • +1.512.520.1234 • +1 (512) 520-1234
MFA Device SMS 2	An SMS phone number that the user has paired with the PingOne MFA service. Learn more about formatting requirements in the MFA Device SMS 1 table entry.
MFA Device SMS 3	An SMS phone number that the user has paired with the PingOne MFA service. Learn more about formatting requirements in the MFA Device SMS 1 table entry.
MFA Device Voice 1	A voice phone number that the user has paired with the PingOne MFA service. You must enter a valid phone number in international format, including the leading `+` character. For example, `+14155552671`. **Important** Always include the country code. Phone number formats across the globe are constantly expanding and changing. The following list contains a few examples of valid ways to format the phone number: • +1.5125201234 • +15125201234 • +1.512.520.1234 • +1 (512) 520-1234
MFA Device Voice 2	A voice phone number that the user has paired with the PingOne MFA service. Learn more about formatting requirements in the MFA Device Voice 1 table entry.
MFA Device Voice 3	A voice phone number that the user has paired with the PingOne MFA service. Learn more about formatting requirements in the MFA Device Voice 1 table entry.
MFA Enabled	When the value is `TRUE`, the user is allowed to use the PingOne MFA service.
Middle Name	The middle name of the user (for example, 'Jane' given the full name 'Ms. Barbara Jane Jensen, III').

Copyright © 2025 Ping Identity Corporation

Attribute	Description
Mobile Phone	The mobile phone number for the user. This value must consist of a leading plus sign, 1 to 3-digit country code, dot separator, 4 to 14-digit phone number, and optional 1 to 8-digit extension (for example, '+1.3034682900x1234').
Nickname	The casual way to address the user, such as "Bob" instead of "Robert".
Password	The user's password. Passwords must match the Password Policy configured in PingOne.
Preferred Language	Indicates the user's preferred written or spoken languages and is generally used for selecting a localized user interface as defined in `RFC 7231` (for example, en-US).
Primary Phone	The primary phone number for the user. This value must consist of a leading plus sign, 1 to 3-digit country code, dot separator, 4 to 14-digit phone number, and optional 1 to 8-digit extension (for example, '+1.3034682900x1234').
Profile Image	The user's photo URL. A URI that is a uniform resource locator (as defined in Section 1.1.3 of `RFC 3986`) that points to a resource location representing the user's image. If provided, the resource URL must point directly to the image (for example, a GIF, JPEG, or PNG) rather than a web page containing an image. It must also have a scheme of http or https.
State / Region	The state or region component for the user's mailing address.
Street Address	The full street address component, which may include house number, street name, P.O. box, and multi-line extended street address information.
Timezone	The user's time zone. This must be in IANA Time Zone database format as defined in `RFC 6557` (for example, "America/Los_Angeles").
User Type	Used to identify the relationship between the organization and the user. Typical values used might be "Contractor", "Employee", "Intern", "Temp", "External", and "Unknown", but any value may be used.
ZIP Code	The ZIP or postal code component for the user's mailing address.

Passing binary attributes to PingOne

PingFederate doesn't currently support formatting the binary data in an attribute to anything other than Base64-encoded before sending it to the connector to then be sent to PingOne, or any other service provider.

Before you begin

PingFederate provides an advanced option allowing administrators to map user attributes by way of an expression language. Learn more about enabling expression use in PingFederate in Enabling and disabling expressions.

About this task

Copyright © 2025 Ping Identity Corporation

For example, if you have an attribute mapping that maps the Account ID field to the `objectSid` attribute in Active Directory, then PingFederate will Base64-encode that binary data into a string.

Steps

1. On the Attribute Mapping tab, for the appropriate attribute click Edit.
2. On the Specify Attribute Mapping tab, enter the appropriate expression in the Expression field.

   Example:

   You can use the following OGNL expression to encode the `objectSid` attribute in SID format:

   ```
   #sidBinary = @java.util.Base64@getDecoder().decode(#this.get("objectSid").toString()),

   #sidHex = @String@format("%056x", new java.math.BigInteger(1, #sidBinary)),

   #SID = #sidHex.substring(16),
   #i = new Long(0),
   #subAuths=#SID.split("(?<=\\G.{8})"),
   #result="S-1-5",
   #subAuths.{
     #BE = #this.substring(6,8) + #this.substring(4,6) + #this.substring(2,4) + #this.substring(0,2),
     #DEC = #i.parseLong(#BE,16),
     #result = #result + "-" + #DEC
   },
   #result
   ```

3. Click Save.

Creating an OpenID Connect connection

You can create an OpenID Connect (OIDC) connection with PingFederate as the relying party (RP) and PingOne as the OpenID Provider (OP). This provides PingOne users single sign-on (SSO) access to applications that are connected to PingFederate.

About this task

> **Important**
>
> These steps are independent of the datastore, PCV, and provisioning components of the integration kit.

To set up an OIDC connection, complete the steps in the Create an OpenID Connect IdP connection procedure from the PingFederate documentation. Make the following modifications to the procedure:

Steps

1. When you reach step 4 in the Create an OpenID Connect IdP connection procedure, perform the following steps on the General Info tab:

   1. In the Issuer field, enter the following:

      `https://auth.pingone.com/<client_id>/as`

Copyright © 2025 Ping Identity Corporation

Use the Client ID that you noted in Connecting PingFederate to PingOne. For example:

```
https://auth.pingone.com/abcd1e8f-6a34-4bcd-e134-668f563a4412/as
```

2. In the Client ID and Client Secret fields, enter the values that you noted in Connecting PingFederate to PingOne.
3. Click Load Metadata.

> **Tip**
>
> This automatically populates the **Authorization Endpoint**,**Token Endpoint**, **User Info Endpoint**, and **JWKS URL** values in step 2d.

2. When you reach step 10 in the Create an OpenID Connect IdP connection procedure, perform the following steps on the OpenID Provider Info tab:
   1. In the Scopes field, enter `openid` and any other OIDC scopes that you need.

      Separate scopes with a space.

      > **Note**
      >
      > Learn more about scopes in Resource scopes in the PingOne documentation and READ All Scopes (Resource) in the PingOne API documentation.

   2. In the OpenID Connect Login Type list, select Code.
   3. In the Authentication Scheme list, select Basic.
   4. Confirm that the Authorization Endpoint, Token Endpoint, User Info Endpoint, and JWKS URL fields were populated automatically from the Load Metadata action in step 1c.

Troubleshooting

Enabling debug logging

To help with troubleshooting or monitoring, you can turn on activity logging for components of the integration kit.

About this task

These steps are optional.

For general information about logging, see Enabling debug messages and console logging in the PingFederate documentation.

Steps

1. Open the `<pf_install>/pingfederate/server/default/conf/log4j2.xml` file for editing.

Copyright © 2025 Ping Identity Corporation

2. If you want to log activity for PingFederate and all adapters, do the following:

    1. Find the following section:

    ```
    <AsyncRoot level="INFO" includeLocation="false">
            <!-- <AppenderRef ref="CONSOLE" /> -->
            <AppenderRef ref="FILE" />
    </AsyncRoot>
    ```

    2. Change `INFO` to `DEBUG`.

    ```
    <AsyncRoot level="DEBUG" includeLocation="false">
            <!-- <AppenderRef ref="CONSOLE" /> -->
            <AppenderRef ref="FILE" />
    </AsyncRoot>
    ```

    3. If you want to see the adapter activity in the console, remove the comment tags.

    ```
    <AsyncRoot level="INFO" includeLocation="false">
            <AppenderRef ref="CONSOLE" />
            <AppenderRef ref="FILE" />
    </AsyncRoot>
    ```

3. If you want to log activity just for the datastore, add the following line:

```
<Logger name="com.pingidentity.plugins.datastore.p14c" level="DEBUG"/>
```

4. If you want to log activity just for the password credential validator, add the following line:

```
<Logger name="com.pingidentity.plugins.pcvs.p14c" level="DEBUG"/>
```

5. Save the file.

Release notes

Changelog

The following is the change history for the PingOne Integration Kit. Updated September 2024.

PingOne Integration Kit 2.7 - September 2024

- Fixed an issue that could cause a delay in retrieving an updated user schema.
- Added the ability to skip retrieving user group and device details during datastore lookup.

Copyright © 2025 Ping Identity Corporation

PingOne Integration Kit 2.6.1 - May 2024

- Added support for the Australia region to the PingOne Connector.

PingOne Integration Kit 2.6 - May 2024

- Added the ability to exclude retrieving user group, device details, and password expiration information during password credential validation.

PingOne Integration Kit 2.5.1 - March 2024

- Fixed an issue that caused values retrieved from the PingOne datastore for custom multi-value attributes to be formatted incorrectly.
- Fixed an issue that caused disabled user accounts to receive an account recovery email if the account recovery flow was triggered.

PingOne Integration Kit 2.5 - February 2022

- Fixed an issue that caused an error when trying to update the `population ID` attribute.

PingOne Integration Kit 2.4.1 - October 2021

- Changed North America region to North America (US) in the PingOne Connector.
- Added support for the North America (Canada) region to the PingOne Connector.

PingOne Integration Kit 2.4 – September 2021

- Added support for group provisioning.
- Added the ability to select which device will be the default device on user creation.
- Added the ability to select voice as an option for offline device pairing.
- Fixed an issue that prevented all of a user's authentication methods from being provisioned if any of them were invalid.
- Fixed an issue that allowed duplicate local attributes to be defined when configuring an adapter.
- Fixed an issue that could cause an attribute containing an array of objects to be returned in the incorrect format.
- Fixed an issue that caused password validation to fail intermittently when the user's access token expires.

PingOne Integration Kit 2.3 – August 2021

- Added support for custom string attributes for provisioning to the PingOne Datastore.
- Fixed an error that prevented a user with an expired password from changing their password.

PingOne Integration Kit 2.2.2 – June 2021

- Added the provisioning connection status to the PingOne connection summary page.

Copyright © 2025 Ping Identity Corporation

- Fixed an issue that, after upgrading the PingOne Datastore component, caused an error when using the administrative API to bulk import an earlier version of the PingOne Datastore.

PingOne Integration Kit 2.2.1 – May 2021

- Fixed an issue that could cause an error in the sign-on flow when the user entered an invalid password.

PingOne Integration Kit 2.2 – March 2021

- Added support for the PingOne platform connection that was introduced in PingFederate 10.2.
- Added support for the identifier first action in PingOne. You can now map an Authoritative IdP value when configuring the provisioning channel.
- Added the ability to select PingOne populations from a pre-populated list when configuring the provisioning channel.
- Added support for authentication method nicknames. The provisioning connector can now use nicknames to identify and update authentication methods.
- Added the ability to get a user's group membership information from PingOne.
- Improved the error message that appears when a user tries to change their password too frequently.
- Improved validation rules and error messages for some configuration fields.

PingOne Integration Kit 2.1 – November 2020

- Added a PingOne region list to provisioning options. This allows the provisioning engine to target the PingOne API in your region.

PingOne Integration Kit 2.0 – September 2020

- Changed name to PingOne Integration Kit in support of new and future PingOne services.
- Added support for managing users' SMS and email devices for PingOne MFA.
- Improved logging abilities.

PingOne Integration Kit 1.2.1 – September 2019

- Fixed an issue that caused the adapter to fail when a key-value attribute was mapped with JSON Pointer syntax.
- Improved the input validation for the Custom Proxy Host field in the password credential validator configuration.
- Improved the error message that a user sees when they try to change their password too frequently.

PingOne Integration Kit 1.2 – May 2019

- Added the ability to override the system-default proxy settings in the data store configuration.
- Added the ability to customize authentication error messages in the password credential validator configuration.
- Improved the error messages that appear when a user tries to set a password that doesn't meet the password policy.
- Fixed an issue that prevented users with empty attributes from being provisioned to PingOne.

Copyright © 2025 Ping Identity Corporation

- Removed support for obsolete scopes from the provisioning connector and data store configuration screen.

PingOne Integration Kit 1.1 – February 2019

- Added the ability to use PingOne as a data store and password credential validator.

PingOne Integration Kit 1.0 – September 2018

- Initial release with support for provisioning to PingOne.

Known issues and limitations

The following are known issues or limitations with the PingOne Integration Kit.

Known issues

Provisioning connector

- After upgrading to version 2.2 of the PingOne Connector, using the administrative API to bulk import an earlier version of the PingOne Connector causes an error. Re-create the connection manually.

Known limitations

Provisioning connector

- When an LDAP user is deleted in a targeted Group DN, the provisioning connector does not propagate the deletion until a new user is added to the group. This limitation is compounded when the User Create provisioning option is disabled. Learn more about solutions in SaaS provisioning connector does not remove the user in the Knowledge Base.
- Due to limitations in PingFederate, user attributes cannot be cleared after they are set.
- PingOne Authentication Policies configuration does not apply for the PCV credential because the PCV management uses user management API.

Download manifest

The following files are included in the PingOne Integration Kit `.zip` archive:

- `Legal.pdf` – Copyright and license information
- `dist/pingfederate/server/default` – Contains the integration files
  - `deploy` – Contains the Java libraries
    - `pf-pingone-datastore-<version>.jar` – PingOne data store.
    - `pf-pingone-pcv-<version>.jar` – PingOne PCV.
    - `pf-pingone-quickconnection-<version>.jar` – PingOne provisioning connector.

Copyright © 2025 Ping Identity Corporation

- `conf` – contains the HTML template that presents the PingOne sign-on form
  - `language-packs` – Contains files with customizable user-facing messages
    - `pingone-auth-messages.properties` – a variable file that provides messages for the PCV.

PingOne MFA Integration Kit

The PingOne MFA Integration Kit allows PingFederate to use the PingOne MFA service for multi-factor authentication (MFA).

Features

- Supports MFA using the following authentication methods:
  - Passkeys (FIDO2)
  - One-time passcodes (OTPs) through SMS, voice call, WhatsApp, and email
  - Time-based One-Time Password (TOTP) authenticator apps, such as Google Authenticator
  - Biometrics (FIDO2)
  - Security keys (FIDO2 and U2F)
  - Push notifications through mobile apps built with the PingOne Mobile SDK
  - Authentication codes
- Adds authentication methods for users in PingOne:
  - Can automatically and silently add SMS, voice, and email authentication methods when a new user signs on
  - Allows users to manually add additional SMS, voice, email, TOTP, and FIDO2 authentication methods or remove existing ones
  - Allows users to select a default authentication method
  - Prompts users to set up MFA if they don't have any existing authentication methods

> **Tip**
>
> The PingOne MFA IdP Adapter can add and remove authentication methods, set a default method, and update a method's nickname.
> If you want to synchronize authentication methods and other user attributes, use the PingOne Connector provided in the PingOne Integration Kit.

- Supports the following transaction flows:
  - Multi-factor authentication
  - Client-initiated backchannel (CIBA) authentication (used for transaction approvals)
- Usable with the PingFederate authentication API

Copyright © 2025 Ping Identity Corporation

- Usable with the JavaScript Widget for the PingFederate Authentication API

Components

PingOne MFA IdP Adapter

- Allows PingFederate to interact with PingOne MFA to trigger an MFA challenge during sign on.

Templates

- Allow the adapter to request OTPs from the user and show the status of the authentication request.
- Allow you to modify the appearance of pages shown to the user during authentication.

Language packs

- Allow you to customize or localize the messages returned by the PingFederate authentication API and shown on the templates during authentication. Learn more in Localizing messages for end users in the PingFederate documentation.

PingOne MFA CIBA Authenticator

- Allows PingFederate to use PingOne MFA to prompt users to approve or deny Client Initiated Backchannel Authentication (CIBA) requests. This authenticator can be used independently from the PingOne MFA IdP Adapter.

Intended audience

This document is intended for PingFederate administrators.

If you need help during the setup process, learn more in the following resources:

- PingOne MFA in the PingOne MFA documentation
- The following sections of the PingFederate documentation:
  - Managing IdP adapters
  - Authentication policies

System requirements

- PingFederate 11.3 or later
- To allow PingFederate to make outbound HTTPS connections, you might need to allow the following host names in your firewall:
  - https://api.pingone.com, https://api.pingone.asia, or https://api.pingone.eu
  - https://auth.pingone.com, https://auth.pingone.asia, or https://auth.pingone.eu
- PingOne requirements:
  - A PingOne MFA license

Copyright © 2025 Ping Identity Corporation

> **Tip**
>
> If you don't have a license, you can Try Ping for Free.

- A PingOne user account
  - MFA must be enabled for the user
  - The user must have at least one authentication method added

> **Tip**
>
> You can find help managing users and devices in PingOne MFA in Getting Started with PingOne MFA.
> To enable user provisioning and offline device pairing from PingFederate, set up the PingOne Integration Kit.

- A mobile app built with the PingOne Mobile SDK, required only for the following:
  - Pairing new devices
  - Push authentication challenges for MFA
  - Push authentication challenges for CIBA requests

Release Notes

New features and improvements in PingOne MFA Integration Kit.

PingOne MFA Integration Kit 2.7 (April 2025)

Enforce Device Selection deprecation

Info P14C-68248

Removed the Enforce Device Selection field because it duplicated the behavior of the Method Selection field in the PingOne MFA policy.

> **Important**
>
> Before upgrading to PingOne MFA Integration Kit 2.7, check your MFA policy to ensure that the **Method Selection** field is configured to reflect the desired device selection behavior.

Temporarily hide Resend OTP link with a PingOne notification cooldown period

New PP14C-66035

Added support for PingOne notification policies assigned to the PingOne MFA policy. During authentication and device management flows, the Resend OTP link is hidden during the notification cooldown period set in the notification policy.

Learn more about PingOne notification policies in Notification policies in the PingOne documentation.

Copyright © 2025 Ping Identity Corporation

Use country prefix selector to add an SMS or voice device

New P14C-66232

Added a country prefix selector to the Device Selection page to expedite adding an SMS or voice device.

> **Important**
>
> Before you can use the country prefix selector, you must move the new `CountriesWithCountryCode.json` file from the `pf-pingone-mfa-adapter-2.7.zip/pf-pingone-mfa-integration-kit/dist/pingfederate/server/default/conf/template/assets/data` folder to the `<PF_HOME>/server/default/conf/template/assets/data` folder in your PingFederate deployment. Learn more in Download manifest.
> If the `/data` folder doesn't exist in your PingFederate deployment, make sure to create it before moving the JSON file.

Complete authentication in the PingID app with PingOne number matching

New P14C-68470

Added support for the PingOne number matching feature. During the authentication flow, if number matching is enabled in the PingOne MFA policy, the PingOne MFA IdP Adapter displays a number that the user must select in the PingID mobile app to complete authentication.

> **Note**
>
> Number matching must be configured in the PingOne MFA policy, and can be used only with the PingID mobile app

Learn more about number matching in Configuring an MFA policy for strong authentication in the PingOne documentation.

Extended support for PingOne MFA Integration Kit 2.4 features to the pf-authn-js-widget

Improved P14C-63433

Improved the JavaScript Widget for the PingFederate Authentication API by adding the following features from the PingOne MFA Integration Kit 2.4 release:

- Rename device during pairing
- Bypass MFA for device management operations
- View remaining OTP attempts in HTML templates and authentication API responses

Added a default device indicator to the `DEVICE_SELECTION_REQUIRED` response

Improved P14C-68246

Improved the ability to view whether a device is the default device within the `DEVICE_SELECTION_REQUIRED` response. By default, the first device in the list is always marked as the default device.

Additionally, you can now set a default device regardless of what you've set the Method Selection to in the PingOne MFA policy. Previously, you could only do so if Method Selection was set to User selected default.

Copyright © 2025 Ping Identity Corporation

Added a locked device indicator to the `DEVICE_SELECTION_REQUIRED` response

Improved | P14C-68247

Improved the ability to view whether a device is locked within the `DEVICE_SELECTION_REQUIRED` response.

Fixed error handling for exceeding the PingOne notification policy email quota

Fixed | P14C-55310 and P14C-59860

- Fixed an issue that caused mishandling of the `EMAIL_LIMIT_REACHED` error in PingOne notification policies.
- Improved error messaging and added the ability to change devices if the user has other devices configured.

Fixed amount of detail included in `Incorrect challenge` error message

Fixed | P14C-55379

Improved the level of detail included in the `Incorrect challenge` error message provided by the PingOne MFA Integration Kit to enhance the troubleshooting experience.

Fixed an issue that caused improper Method Selection syncing

Fixed | P14C-64168

Fixed an issue that caused the Method Selection configuration to sync improperly between the PingOne MFA IdP Adapter and the PingOne MFA policy.

Fixed an issue with absent predefined values in the pairing flow

Fixed | P14C-66982

Fixed an issue that caused values predefined in the HTML adapter contract fulfillment not to show up when pairing devices in a clustered PingFederate environment. Trying to continue the pairing flow would cause a `NullPointerException` error.

This issue was applicable only when Allow only predefined values for phone or email devices was selected in the adapter configuration.

Fixed an OpenJDK thread locking issue

Fixed | P14C-68611

Fixed an issue that sometimes caused deadlocks when the PingOne MFA IdP Adapter was collecting an `appSecret` from PingOne.

Fixed conflicting `pingone.mfa.status` in chained MFA adapter flows

Fixed | P14C-68883

Copyright © 2025 Ping Identity Corporation

Fixed an issue that caused the second adapter in a chained PingOne MFA IdP Adapter flow to reuse the `pingone.mfa.status` of the first adapter when the user had no devices.

Fixed ability to reselect the original device on the Change Device page

Fixed P14C-69393

Fixed an issue that caused an unwanted error message after re-selecting a device on the Change Device page. This issue was applicable after the daily one-time passcode (OTP) limit had been reached for that device.

The device is now hidden in the selection options available on the Change Device page to prevent users from trying it more than once.

Fixed prompt parameter mishandling related to signed request objects

Fixed P14C-69930

Fixed an issue that caused the PingOne MFA IdP Adapter to mishandle the prompt parameter if was received in a signed request object.

PingOne MFA Integration Kit 2.6 (January 2025)

Enforce Device Selection will be deprecated in the next release

Info P14C-66231

The Enforce Device Selection field will be deprecated in the next release of the PingOne MFA Integration Kit because it duplicates the behavior of the Method Selection field in the PingOne MFA policy.

Enforce Device Selection is still usable in version 2.6, but make sure to set the Method Selection configuration in the MFA policy to Always Display Devices. Otherwise, for example, the user might receive multiple OTPs.

> **Note**
>
> Before upgrading to the next version, check your MFA policy to make sure that **Method Selection** is configured to reflect the desired behavior.

Insert an authentication policy into the CIBA flow

New P14C-67740

To insert an authentication policy into the CIBA flow based on runtime parameters, pass the extended contract parameter pingone-mfa-acr into an empty PingOne authentication policy that's used in the CIBA authentication policy.

> **Note**
>
> A non-empty PingOne authentication policy always takes precedence over **pingone-mfa-acr**, so make sure that the PingOne authentication policy is empty.

Copyright © 2025 Ping Identity Corporation

Improved `429` error handling

Improved P14C-64515

Added clearer error messaging for rate-limiting `429` errors.

Include number of maximum allowed devices in API response

Improved P14C-64509

All API responses that include the `devices` field now include a new field, `maxAllowedDevices`, which indicates the maximum number of devices that can be paired for a user.

Added support for the `prompt` parameter for OIDC flows

Improved P14C-64927

Updated the adapter to pass the `prompt` parameter to PingOne.

> **Note**
>
> As a prerequisite, you must make sure that PingFederate is tracking the `prompt` parameter in the authentication policy. You can find information about how to track the `prompt` parameter in Adding PingOne MFA to your authentication policy.

Improved error messaging for flow time-outs

Improved P14C-58274

Replaced the generic error message for flow time-outs. The new message makes it clearer that a flow time-out caused the error, not an invalid OTP.

Improved locked device experience

Improved P14C-55624

If the default device becomes locked, you can now return to the Device Selection page and select a different device to use for authentication.

Enhanced MFA policy field configuration

Improved P14C-66230

The MFA policy for registration field in the adapter configuration is now a list instead of a textbox. This expedites selecting the desired policy and prevents typos from invalidating the configuration.

Added extra validation when pairing a user's first device

Security P14C-59819

Copyright © 2025 Ping Identity Corporation

Added validation to the device pairing flow when pairing a user's first device to confirm that no other devices were added during the session.

Updated a method to double-check if pairing process was initiated

Security P14C-63038

Updated a method to double-check if the pairing process was initiated to address potential security concerns.

Added tighter restrictions on cookie paths

Security P14C-53794

Added tighter restrictions on cookie paths to address potential security concerns.

Updated third-party libraries

Security P14C-61311, P14C-66921, and P14C-62241

Updated third-party libraries and removed an unused package to address potential security concerns.

Fixed an issue with configuring incorrect proxy settings

Fixed P14C-57761

Fixed an issue that caused the PingOne MFA IdP Adapter to get stuck if you saved the adapter configuration with incorrect proxy settings. For example, the wrong host or IP address.

Fixed an issue with page presentation when `bypassmfa` is set to `false`

Fixed P14C-65478

Fixed an issue that caused the Device Pairing and OTP Required pages to present an error message after clicking Cancel, if `bypassmfa` was set to `false`. Now:

- If you click Cancel on the Device Pairing page, the adapter presents the Device Selection page.
- If you click Cancel on the OTP Required page, the adapter presents the Device Pairing page.

Fixed error messaging for expired OTPs

Fixed P14C-66403

Fixed an issue that caused expired OTP error responses to display an irrelevant error message.

Fixed error messaging for exceeding the one-time device user quota

Fixed P14C-66494

Copyright © 2025 Ping Identity Corporation

Fixed an issue that caused two different error messages to display after reaching the daily user quota for authentication attempts. The quota is set in the PingOne notification policy.

Fixed exception after reaching the notification limit for multiple device types

Fixed P14C-66815

Fixed an issue that caused an exception when authenticating after reaching the SSO notification limit for multiple device types. Also fixed error messaging differences between the device types.

PingOne MFA Integration Kit 2.5 (September 2024)

Added support for the PingOne FIDO Device Aggregation feature

New P14C-57629

Added support for the PingOne FIDO Device Aggregation feature. FIDO Device Aggregation is part of the FIDO policy in PingOne. When this feature is enabled and a user has multiple FIDO2 devices, the user will see only one passkey device on the device selection screen. This device is an aggregation of all the user's FIDO2 devices. During authentication, the authenticator suggests the best-suited FIDO device to the user.

View PingOne mobile application name in authentication API response

New P14C-61861

You can now view the PingOne mobile application name in the PingFederate authentication API response.

View a OTP's lifetime in the authentication API

New P14C-62073

The PingFederate authentication API response now includes a new field (called `otpLifetime`) for the `OTP_REQUIRED` status. The `otpLifetime` field shows how long the OTP will remain valid.

Include more information in the PingOne logs for adapter authentication attempts

Improved P14C-44523

The PingOne MFA IdP adapter now forwards application and device information for authentication attempts made through the adapter to PingOne. To view this information in the PingOne admin console:

1. Go to Directory > Users and click on a user.
2. Click the Services tab, select Authentication, and go to the Sessions section.

Use dynamic linking to give a unique identifier to a FIDO device pairing attempt

Improved P14C-62023

Copyright © 2025 Ping Identity Corporation

Added the ability to give a unique identifier to a FIDO device pairing attempt. Learn more on dynamic linking in the "Use dynamic linking to give a unique identifier to a FIDO device authentication attempt" release note entry in PingOne MFA Integration Kit 2.4 (August 2024).

Upgraded TLS support

Security P14C-57813

The PingOne MFA IdP adapter now supports only TLS 1.2 and later.

Fixed double `AUTHENTICATION_REQUIRED` response

Fixed P14C-51451

Fixed an issue that caused the `AUTHENTICATION_REQUIRED` response to be returned twice for some specific scenarios in the PingFederate authentication API.

Fixed an authentication API response for `selectDevice` when OTP limit is reached

Fixed P14C-56520

Fixed an issue that caused the authentication API to provide an incorrect error response when a user switched from a mobile device to an OTP-based device and reached the OTP limit (as defined in notification policies in the PingOne admin console). OTP-based devices include:

- SMS
- Voice
- Email
- WhatsApp

Fixed an issue with locked OTP-based devices in HTML templates

Fixed P14C-59939

Fixed an issue that caused the HTML template to allow users to keep entering OTPs (which wouldn't actually be submitted) after they became locked out from a device. Now, when a user is locked out, the adapter displays a new screen to inform the user that the device is locked and present a Change Device button.

Fixed issues on the device selection screen

Fixed P14C-64209

Fixed an issue that caused the device selection screen to unmask the device's information improperly when the user was returning to the device selection screen from the Add method screen and the `bypassmfa` configuration was set to `true`.

Copyright © 2025 Ping Identity Corporation

PingOne MFA Integration Kit 2.4.1 (August 2024)

Fixed pairing of test-devices in PingFederate clustered mode

Fixed STAGING-23420

Fixed an issue that prevented test-devices from being paired in PingFederate clustered mode.

Fixed MFA requirement when the MFA policy doesn't permit the existing device

Fixed P14C-64779

Fixed an issue that caused the adapter to require the user to complete MFA to add an MFA device when the user's existing MFA device was not allowed in the MFA policy.

PingOne MFA Integration Kit 2.4 (August 2024)

Use basic velocity template variables in the PingOne MFA templates

New P14C-57293

Made basic variables that are available in most PingFederate HTML templates available in the PingOne MFA templates.

Pair and authenticate a test device

New P14C-58742

Added the ability to pair a test device when adding a multi-factor authentication method and authenticating with it. A test device causes an OTP to be returned directly in the `OTP_REQUIRED` response.

> **Note**
>
> Test devices are supported only when initiating OpenID Connect (OIDC) flow in the authentication API.

To pair and authenticate with a test device, you must:

1. Add the `pi.testDevice` parameter to the OIDC request with a value of `allow`.
2. Sign the request object with your OIDC client credentials.
3. When submitting the device target in the authentication API request, add the `testMode` field to the request with a value of `true`.

   > **Note**
   >
   > This applies to the following action models: `submitEmailTarget`, `submitSmsTarget`, or `submitVoiceTarget`. Learn more in Models, objects, and error codes.

Use dynamic linking to give a unique identifier to a FIDO authentication attempt

New P14C-58753

Copyright © 2025 Ping Identity Corporation

Added the ability to use a custom challenge when authenticating with FIDO (webAuthn) devices. This enables you to attach meaningful information to the authentication of a FIDO device.

To provide a custom challenge for FIDO authentication, you must:

1. Add the `pi.webAuthn.challenge` parameter to the OIDC request with the custom challenge as the value.
2. Sign the request object with your OIDC client credentials.

Rename device during pairing

Improved P14C-52773

Added the ability to give a device a unique nickname during device pairing:

- Use the PingOne administrative console to configure this setting in the PingOne MFA policy.
- After you configure the ability to rename devices, users will be presented with a new screen before authentication ends. The user can either enter a nickname and click Done to complete the process, or click Skip if they do not want to give the device a nickname.
- If you are using the authentication API, a new state (`UPDATE_NICKNAME`) and two new actions (`updateDeviceNickname` and `skipUpdateDeviceNickname`) are available. Learn more in Models, objects, and error codes.

View remaining OTP attempts in HTML templates and authentication API responses

Improved P14C-57444

Added the ability to view the number of one-time passcode (OTP) attempts remaining after entering an invalid OTP.

- In authentication API responses, the `attemptsRemaining` field displays this information.
- In HTML templates that require an OTP, the following error message appears:

```
This passcode is invalid or has expired. You have  <number_of_attempts>  attempts remaining.
```

Bypass MFA for device management operations

Improved P14C-60088

Added the ability to bypass MFA when performing device management operations. Be cautious with using this attribute if you only have one adapter in the authentication policy. This results in bypassing MFA in the authentication flow entirely, and can lead to a security breach.

Additionally, the Bypass MFA For Device Pairing Attribute field is now the Bypass MFA For Device Management Attribute field.

Learn more in the PingOne MFA IdP Adapter settings reference.

Overwrite only specific authentication methods

Improved P14C-62072

Copyright © 2025 Ping Identity Corporation

Added the ability to overwrite only the devices that share a device type with a newly provided device if the adapter identifies new values for SMS, voice or email devices via Update Authentication Methods.

Additionally, the Overwrite Authentication Methods checkbox is now the Overwrite Authentication Methods Configurations list.

There are three Overwrite Authentication Methods Configurations settings:

- None (default)
- All (SMS, Voice, and Email)
- Specific Methods

Learn more in the PingOne MFA IdP Adapter settings reference.

Fixed default method persistence

Fixed P14C-55013

Fixed an issue that caused Overwrite Authentication Methods (now Overwrite Authentication Methods Configurations) to change the default device designation. This was applicable when a new device of the same type as the default device was provided, and the default device was overwritten.

Fixed empty device nickname issue

Fixed P14C-58407

Fixed an issue that caused devices to save with an empty nickname instead of reverting to the default device name. This was applicable in configurations where Allow Users to Manage Authentication Methods was selected, if a user clicked Edit Name but cleared the field.

Fixed an issue with FIDO usernameless authentication flow ignoring the PingOne authentication policy

Fixed P14C-60584

Fixed an issue that caused the adapter to always use the default multi-factor authentication (MFA) policy in FIDO usernameless authentication flow instead of the PingOne MFA policy configured in the PingOne Authentication Policy field.

Fixed device registration limit issue with MFA bypass in the authentication API

Fixed P14C-61122

Fixed an issue that caused users who had already exceeded the device registration limit to proceed several steps into device registration flow before the flow failed instead of presenting the `MAXIMUM_ALLOWED_METHODS_LIMIT` error message at the beginning of the flow. This issue was relevant to configurations that had the Bypass MFA for Device Pairing Attribute checkbox (now Bypass MFA For Device Management Attribute) selected.

Copyright © 2025 Ping Identity Corporation

PingOne MFA Integration Kit 2.3.1 (March 21, 2024)

Security issue - skipMFA action

Security

A security vulnerability related to the `skipMFA` action has been fixed in this version.

Exception reported on CIBA transaction

Fixed STAGING-20713

When using PingFederate 11.3.1, there were cases where a push notification triggered by the CIBA authenticator was not received. This issue has been fixed.

PingOne MFA Integration Kit 2.3 (January 2, 2024)

Control default to first behavior at policy level instead of environment level

Info

In version 2.2.1 and earlier of the PingOne MFA Integration Kit, the first device in the list was set as the default device. In 2.3 and later, default device behavior depends on what Method Selection in the PingOne MFA policy is set to.

To replicate default to first behavior in 2.3 and later, set Method Selection to User Selected Default.

Language customization for notifications

Improved

- To customize the language for notifications, you no longer have to provide `pi.template` as a JSON object to change the locale when defining the contract for the adapter. You can just specify the locale using `pi.template.locale`. See Transaction approval setup.
- If you have not specified a locale with `pi.template` or `pi.template.locale`, notifications will use the default browser's language. This requires that the relevant language be enabled in both PingOne and PingFederate.

Use of dynamic variables in pairing flows

Improved STAGING-19855

Previously, the values provided in the `pi.template` object for notification customization were used only in authentication flows. Now, customization (including dynamic variables) is applied also to pairing flows, one-time device flows, and change device flows.

User-provided nicknames for devices

New

Copyright © 2025 Ping Identity Corporation

When authenticating, users who are authorized to manage their devices can now edit the nicknames used for their various authentication devices. This capability is also supported in the PingFederate authentication API (action is called `updateDeviceNickname`).

Use of custom proxies in MFA adapter

Fixed STAGING-18848

Previously, if you defined a custom proxy for an adapter, the host specified with Custom proxy host was not used for OAuth tokens (the `token` endpoint). These requests used the system default proxy instead, causing the request to fail. This issue has been fixed.

Event tracking IDs in PingOne Audit log

Improved STAGING-19417

In the PingOne Audit log, events triggered by the MFA adapter now include IDs that can be used to locate events in the PingFederate log. The `trackingid` field from PingFederate is represented in the PingOne Audit log as `sessionId`. If you are using version 11.3 or higher of PingFederate, then the `transactionid` field from PingFederate is also included in the PingOne Audit entry, appearing there as `transactionId`.

Limiting email and SMS pairing info to directory values

New STAGING-19722

By default, when pairing an email or SMS device, users can enter the email address or phone number. However, the adapter configuration now includes an option called Allow only predefined values for phone or email devices. This option allows you to limit the values to the email addresses and phone numbers stored for the user. If you enable this option, the relevant email address or phone number is already filled in when the user tries to add a device, and the user cannot modify the address/phone number.

PingOne MFA Integration Kit 2.2.1 (September 2023)

Specify a PingOne registration policy

New P14C-51600

Added the ability to specify a PingOne MFA policy to use for device pairing. For more information, see the PingOne Registration Policy field.

> **Note**
>
> Make sure to use a registration policy that is compatible with your authentication policy or policies.

Previously, the adapter would always use the default MFA policy for PingOne pair device API calls. Because there wasn't a configuration option for the MFA policy used in pair device API calls, you couldn't pair the type of device that you specified in your PingOne Authentication Policy if it wasn't also included in the default MFA policy. The ability to configure a registration policy fixes this issue.

Copyright © 2025 Ping Identity Corporation

Configure usernameless authentication flow using a FIDO2 device

New P14C-53908

Added support for FIDO2 to the FIDO usernameless authentication flow.

Fixed security vulnerability

Security P14C-53455

Fixed a security vulnerability that allowed for the creation of new devices.

Fixed persistent Device Selection screen

Fixed P14C-53963

Fixed an issue that caused the pf-authn-js-widget to display the Device Selection screen even when the flow status wasn't `device_selection_required`.

Fixed passkey icon in Safari browsers

Fixed P14C-54482

Fixed an issue that prevented the passkey icon from displaying correctly in Safari browsers.

PingOne MFA Integration Kit 2.2 (July 2023)

Use passkeys for second-factor authentication

New

Added the ability to use passkeys as a second-factor authentication method.

Remove authentication methods

Improved

Added support for removing authentication methods. If the Allow Users To Manage Additional Authentication Methods check box is selected, users can now remove authentication methods when they sign on.

Fixed CIBA Authenticator compatibility with PingFederate 11.3

Fixed

Fixed an issue that prevented the PingOne MFA CIBA Authenticator from being compatible with PingFederate 11.3.

Copyright © 2025 Ping Identity Corporation

PingOne MFA Integration Kit 2.1 (May 2023)

PingID SDK adapter migration

New

Added support for the PingID SDK adapter migration scenario. Supported flows:

- Device authorization
- Authentication
- Registration
- CIBA authentication

Parameter attributes

New

Added support for the following dynamic parameter attributes:

- `pingIdSdkAdapterContext`
- `pingIdSdkSkipSuccessScreens`
- `pingIdSdkSkipErrorScreens`
- `pingIdSdkSkipTimeoutScreens`

FIDO usernameless authentication flow

New

Added the ability for the adapter to dynamically skip FIDO usernameless authentication flow.

PingOne authentication

Improved

Added support for additional methods to dynamically provide the PingOne authentication policy.

PingOne MFA Integration Kit 2.0 (November 2022)

FIDO usernameless authentication flow

New

Added support for FIDO usernameless authentication flow.

Copyright © 2025 Ping Identity Corporation

JIT device updates

New

Added support for JIT overwrite of PingOne MFA devices.

OTP flow

Fixed

Fixed error message and response to handle retry failure due to unsuccessful OTP attempts and display the appropriate cool down period.

Authentication API

Fixed

Fixed a defect in the authentication API to prevent selection of different devices when the adapter is configured so.

SSL CRL

Fixed

Fixed a defect with SSL CRL lookup when the adapter instance is configured to use system defaults with PingFederate server configured behind proxy.

PingOne MFA Integration Kit 1.9 (July 2022)

Stateless one-time device flow

New

Added support for statless one-time device OTP flow.

PingOne MFA Integration Kit 1.8 (June 2022)

Use authentication codes in authentication API flows

New

Added support for authentication API flows that use authentication codes. For information about flows, see Authentication API Support.

Log browser and device information

New

Added support for audit logging of browser and selected device information

Copyright © 2025 Ping Identity Corporation

Passwordless login authentication API flows

Improved

Improved support for passwordless login authentication API flows.

FIDO device pairing flow

Fixed

Fixed an issue with the FIDO device pairing flow.

Failover mode

Fixed

Fixed an issue that caused failover mode to not work.

PingOne MFA Integration Kit 1.7 (April 2022)

Force users to select a device during sign on

New

Added support for enforcing device selection during authentication. See the Enforce Device Selection setting in PingOne MFA IdP Adapter settings reference.

PingOne MFA Integration Kit 1.6.1 (February 2022)

User Not Found Failure Mode

Fixed

Fixed an issue that caused an error when User Not Found Failure Mode was set to Bypass Authentication.

Template flow fails when a user sets a default device

Fixed

Fixed an issue that caused the template flow to fail when a user sets a default device.

Previous releases

The following is the change history for the PingOne MFA Integration Kit.

PingOne MFA Integration Kit 1.6 — January 2022

- Added support for passwordless login using FIDO browser management.
- Added the ability to display additional platform information when a user pairs a new biometric device.

Copyright © 2025 Ping Identity Corporation

- Added the ability to provision users and add authentication methods separately.
- Fixed an issue that caused an error for users on iOS or Safari when attempting authentication or pair using FIDO.
- Fixed an MFA vulnerability. See security bulletin SECADV029.

PingOne MFA Integration Kit 1.5.2 — November 2021

- Fixed an issue that prevented pairing security keys and biometrics when the authentication API or JavaScript Widget was hosted on an external domain.

PingOne MFA Integration Kit 1.5.1 — September 2021

- Fixed an issue that caused an error when the PingOne MFA IdP Adapter was used with other adapters in a password reset flow.
- Fixed an issue that could cause an error when pairing devices for users identified with a PingOne user ID.

PingOne MFA Integration Kit 1.5 — August 2021

- Added support for the PingOne device integrity check. For details, see Authentication method management.

> **Note**
>
> If you're upgrading from PingOne MFA Integration Kit 1.4.1 or earlier and want to use the device integrity check feature, update the **API Request Timeout** in your adapter configuration. To provide time for the device integrity check, the default has been increased from 5000ms to 12000ms.

- Improved the user experience for adding additional authentication methods.

PingOne MFA Integration Kit 1.4.1 — June 2021

- Fixed an issue that prevented the device pairing flow from working when the Mobile App option was disabled in the PingOne policy.

PingOne MFA Integration Kit 1.4 — June 2021

- Added the ability to authenticate using voice.
- Added the ability to override the notification template variant that PingOne shows for transaction approval flows.
- Added the ability for a single CIBA authenticator instance to work with multiple PingOne applications. The authenticator now checks for an application identifier in the CIBA request.
- Improved the adapter's Authentication method management features:
  - Added the ability for users to manually add a wide variety of authentication methods. Enable this feature with the Allow Users to Add Additional Authentication Methods setting.
  - Added the ability to prompt users to set up MFA if they have no existing authentication methods. You can also allow users to skip the MFA setup. Enable this feature with the Prompt Users to Set Up MFA and Allow Users to Skip MFA Setup settings.

Copyright © 2025 Ping Identity Corporation

    - Added the ability for users to select a default authentication method. When a default is selected, the adapter skips the selection screen.

- Deprecated the Application Client Secret field for PingFederate 10.2 and later.

PingOne MFA Integration Kit 1.3.2 — June 2021

- Fixed an issue that, after upgrading the adapter, caused an error when using the administrative API to bulk import an earlier version of the adapter.

PingOne MFA Integration Kit 1.3.1 — April 2021

- Fixed an issue that caused an error when no port was specified in the PingFederate base URL.

PingOne MFA Integration Kit 1.3 — March 2021

- Added the ability to authenticate using timed one-time passcodes (TOTP) with mobile devices.
- Added the ability to authenticate using FIDO2-bound biometrics and U2F security keys.
- Added support for single logout from PingOne MFA when the user signs off in PingFederate.
- Added support for the new account lockout error in PingOne MFA when a user fails multiple consecutive MFA attempts.
- Added support for pre-populating adapter settings based on the selected PingOne environment. Available in PingFederate 10.2 or later.
- Improved support for device nicknames, making them available for all authentication methods, on all templates, and in the JavaScript Widget for the PingFederate Authentication API.
- Fixed an issue that caused an error when a user initiated mobile push for account recovery.
- Fixed an issue that caused an error when setting the API Request Timeout value too low.
- Fixed an issue that caused the API to return validation errors when upgrading the adapter.

PingOne MFA Integration Kit 1.2 — February 2021

- Added the Provision Users and Authentication Methods setting, and related fields, to allow the adapter to provision new users to PingOne MFA and automatically add valid authentication methods for the user.
- Added the Update Authentication Methods setting to allow the adapter to register new SMS and Email authentication methods in PingOne MFA.

PingOne MFA Integration Kit 1.1 — December 2020

- Added support for the platform connection to PingOne introduced in PingFederate 10.2.
- Added support for the PingOne MFA transaction approval flow.
- Added support for the PingOne MFA mobile device authorization flow.
- Added the ability to authenticate using third-party TOTP authenticators, such as Google Authenticator.

Copyright © 2025 Ping Identity Corporation

- Added support for client-initiated back-channel authentication (CIBA). For information about CIBA, see Improving the Customer Experience with CIBA on the Ping Identity blog.
- Added support for the JavaScript Widget for the PingFederate Authentication API.
- Added error handling for the following scenarios:
  - The device ID is invalid
  - The OTP format is invalid
  - MFA is disabled for the user
  - The user is not found
  - The user has no devices paired and automatic pairing is not enabled
  - The user has no devices paired, automatic pairing is enabled, but the user is signing on via the web
- Added an error page that supports customizable messages using a language pack file.
- Improved API endpoint selection by replacing the PingOne API fields with a region list.
- Fixed an issue that caused the Authenticating page to refresh periodically when the adapter polled PingOne for updates.
- Fixed an issue that prevented users from changing their selected MFA device when the PingOne authentication policy used the Being a member of any of these populations or User Attributes requirements.

PingOne MFA Integration Kit 1.0 — September 2020

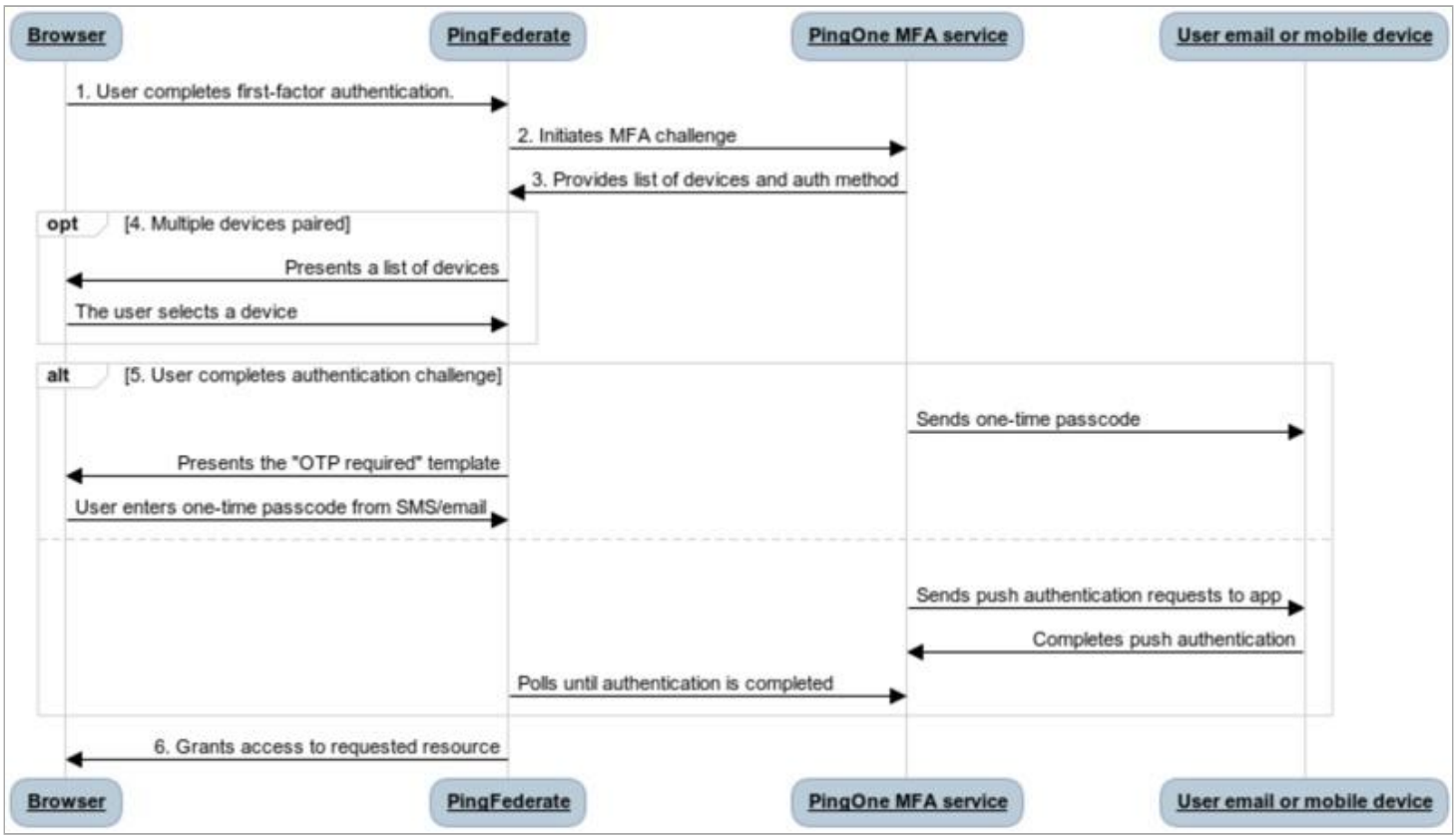
- Initial release.
- Added the ability to authenticate using SMS, email, and push.
- Added the ability to automatically pair SMS and email authentication methods.
- Added the ability to control which message pages (templates) are shown to the user.
- Added the ability to control how the adapter handles sign-on attempts when errors occur.
- Added the ability to override the PingOne policy received in the requested authentication context.
- Added the ability to test the connection to PingOne MFA.
- Added support for the PingFederate authentication API
- Added settings for API connection and request timeouts.
- Added settings to override the PingFederate system-default proxy settings.

Overview of the SSO flow

With the PingOne MFA Integration Kit, PingFederate includes PingOne MFA in the sign-on flow.

Copyright © 2025 Ping Identity Corporation

The following figure illustrates a single sign-on (SSO) scenario in which PingFederate authenticates users to an service provider (SP) application using PingOne MFA.



Description

1. The user initiates SSO with PingFederate and completes the first-factor authentication step, such as an HTML Form Adapter instance.
2. PingFederate contacts PingOne MFA to initiate the multi-factor authentication (MFA) challenge.
3. PingOne MFA indicates whether the authentication method is a one-time passcode (OTP) or push notification. If the user has more than one device paired, PingOne MFA also provides a list of devices.
4. If the user has multiple devices, PingFederate shows the user a list of devices. The user selects a device.
5. Depending on the authentication method, one of the following occurs:
    - For OTP authentication, PingOne MFA sends the OTP to the user by email or SMS. In the browser, PingFederate shows a form requesting the OTP. The user enters the OTP in the form.
    - For push notification, PingOne MFA sends a push notification to the user's mobile app. PingFederate polls the application programming interface (API) until PingOne MFA provides the authentication result.
6. If the user authenticated successfully, PingFederate provides access to the requested resource.

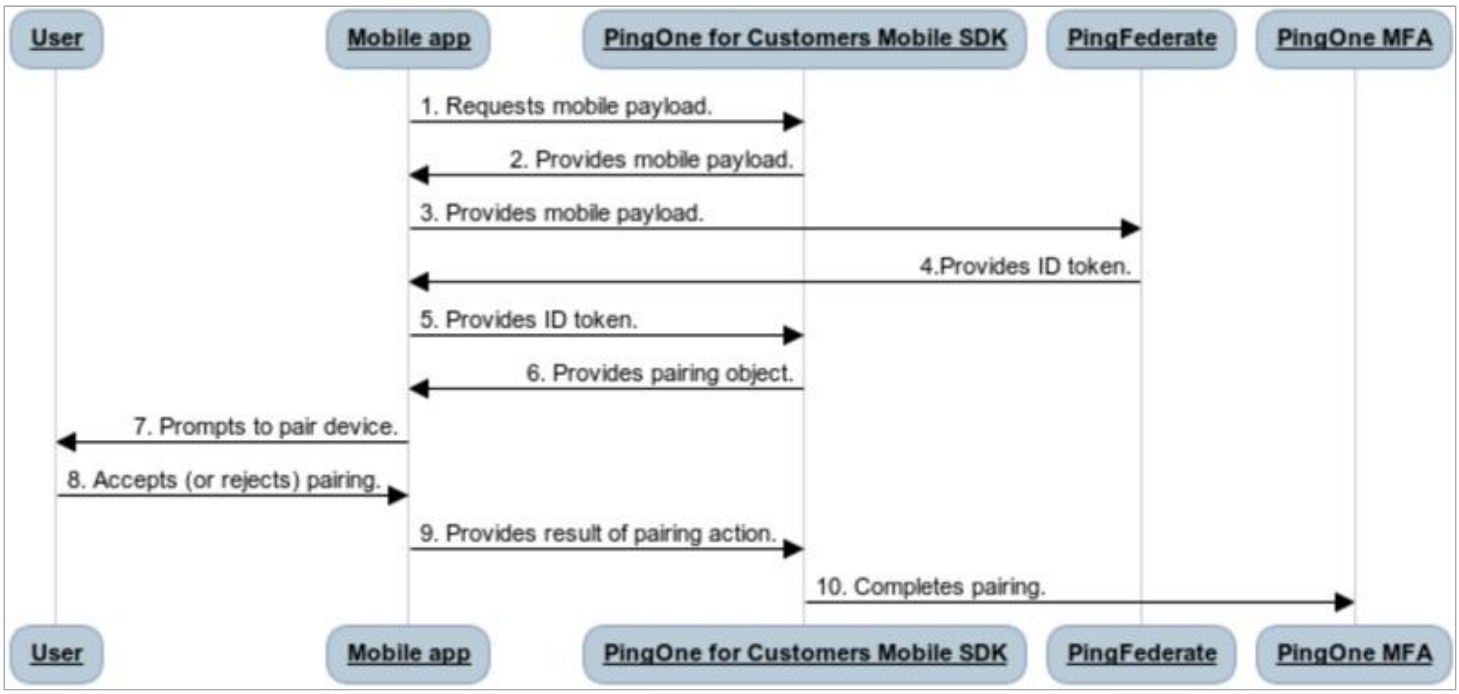
    Otherwise, it shows the user an optional page with the reason authentication failed.

Copyright © 2025 Ping Identity Corporation

Overview of the automatic device pairing flow

With the PingOne MFA Integration Kit, users can pair their devices to PingOne MFA using the mobile app.

The following figure illustrates an enrollment scenario. This flow is triggered any time a user attempts to authenticate with an unpaired device and automatic pairing is enabled in PingOne.



Description

1. The user is identified on the customer's mobile app, usually with a unique user identifier, for example, a username. The app requests a mobile payload from the PingOne Mobile SDK.
2. The PingOne Mobile SDK returns a mobile payload to the app.
3. The app sends an authentication request to PingFederate and provides the mobile payload.

   The PingFederate authentication policy stores the mobile payload as a tracked HTTP parameter.
4. If the user has SMS, email, or another device paired, the PingOne policy is triggered. Typically, the user authenticates with the existing device and the PingOne MFA IdP Adapter responds with an ID token.

   If the user does not have an existing device available, a PingOne setting determines whether the user is blocked or if multi-factor authentication (MFA) is bypassed. If MFA is bypassed, the automatic pairing flow continues.
5. The app passes the ID token to the PingOne Mobile SDK.
6. The PingOne Mobile SDK returns a pairing object to the app that allows it pair or ignore the device.
7. The app prompts the user to pair the device.
8. The user accepts or rejects the device pairing action.
9. Based on the user's choice, the app notifies the PingOne Mobile SDK.

Copyright © 2025 Ping Identity Corporation

10. The PingOne Mobile SDK completes the transaction accordingly by communicating directly with PingOne.

Authentication method management

The PingOne MFA IdP Adapter supports a variety of methods for managing authentication methods for PingOne users.

> **Tip**
>
> The PingOne MFA IdP Adapter can add and remove authentication methods and set a default method, but it cannot update a method's nickname.
> If you want to synchronize authentication methods and other user attributes, use the PingOne Connector provided in the PingOne Integration Kit.

Automatic authentication method provisioning

The PingOne MFA IdP Adapter supports automatic pairing for SMS, voice, and email authentication methods, which use one-time passcodes (OTPs). The adapter creates the user and associated authentication methods from attributes in the PingFederate authentication policy.

For details, see Enabling user and authentication method provisioning.

Prompting users to set up their first authentication method

The PingOne MFA IdP Adapter allows you to prompt users to set up their first multi-factor authentication (MFA) method.

For details, see Enabling the MFA setup prompt.

Allowing users to manage authentication methods

When the Allow Users To Manage Additional Authentication Methods check box is selected in the PingOne MFA IdP Adapter configuration, users can add or remove a new authentication method when they sign on.

> **Note**
>
> This option is only available to users who authenticate with an existing authentication method.

Users can pair any PingOne authentication method, including:

- Authenticator apps, such as Google Authenticator
- SMS message
- Voice call
- Email
- Mobile app built with the PingOne SDK
- FIDO2 biometrics
- Security key

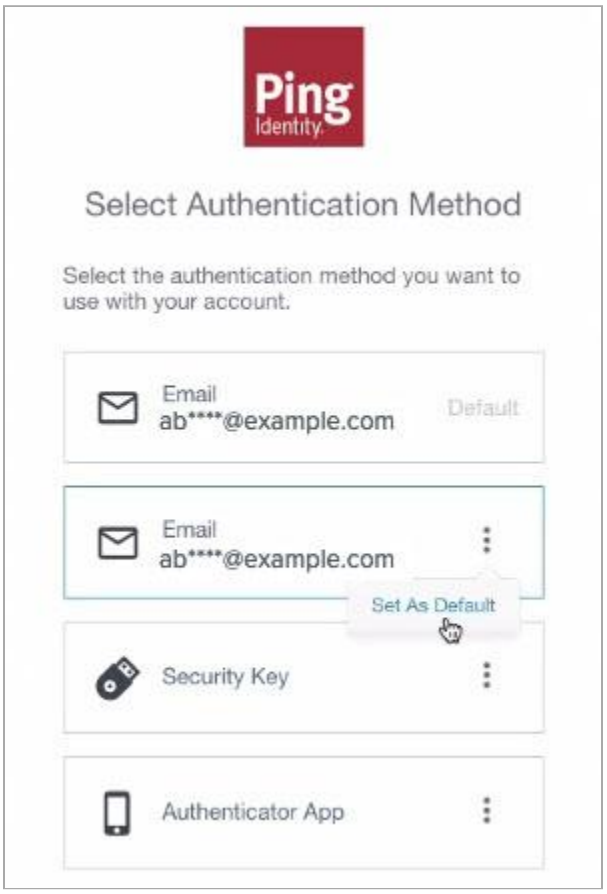
They can also remove any of these authentication methods.

Copyright © 2025 Ping Identity Corporation

Alternately, users can manage their authentication methods directly through the PingOne MFA self-service URL. Learn more in Self service and Managing authentication methods in the PingOne documentation.

Default authentication methods

On the authentication method selection page, users can set a default authentication method.



When a default method is set, the PingOne MFA IdP Adapter skips the selection screen. Users can select a different authentication method if they don't want to continue with the default.

This capability is available when the `User-selected default` option is turned on in PingOne. Learn more details and limitations in the Set a default authentication method section of Managing authentication methods in the PingOne documentation.

Device integrity checks

When you create a native OIDC application as described in Creating a web or native OIDC application in PingOne, you can turn on a device integrity check. The device integrity check identifies jailbroken iOS devices and rooted Android devices. When it's enabled, users can't pair or authenticate with compromised devices.

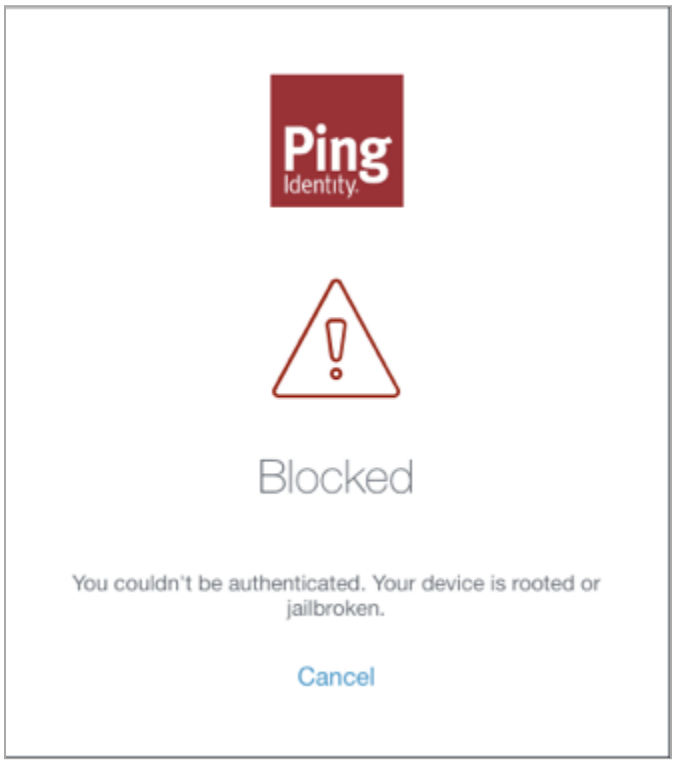
When a user authenticates, PingOne reports whether the device passed the integrity check.

When a device fails, one of the following happens:

- If you're using the PingFederate authentication API, your application receives an `MFA_FAILED` status with the code `DEVICE_INTEGRITY_FAILED`.

Copyright © 2025 Ping Identity Corporation

- If you're using the PingFederate web interface, the PingOne MFA IdP Adapter shows an error page to the user.



You don't need to change anything in the PingOne MFA IdP Adapter configuration to support device integrity checks. Just enable them in your PingOne native OIDC application settings. To do so:

1. In your native OIDC application, go to the Mobile tab and click the Pencil icon.
2. In the Settings section, go to Device Integrity Check and click On.

Setup

Deploying the integration files

To get started with the integration, deploy the PingOne MFA Integration Kit files to your PingFederate directory.

Steps

1. Download the PingOne MFA Integration Kit `.zip` archive from the Ping Identity Integration Directory.
2. Stop PingFederate.
3. If you're upgrading an existing deployment, back up your customizations and delete earlier versions of the integration files:
    1. Back up any PingOne MFA Integration Kit files that you customized in the `<pf_install>/pingfederate/server/default/conf/` directory.
    2. Delete the `pf-pingone-mfa-adapter-<version>.jar` file from your `<pf_install>/pingfederate/server/default/deploy` directory.

Copyright © 2025 Ping Identity Corporation

4. Extract the `.zip` archive and merge the contents of the `dist` directory with your `<pf_install>/pingfederate` directory.
5. If there is more than one version of the `pf-authn-api-sdk-<version>.jar` file in your `<pf_install>/pingfederate/server/default/lib` directory, delete all but the latest version of the file.
6. If you backed up any customized files, modify the new files with your customizations.
7. Start PingFederate.
8. If you operate PingFederate in a cluster, repeat steps 2-8 for each engine node.

Creating an organization and environment in PingOne

The PingOne platform uses an organization-based model to define tenant accounts and their related entities. Create an organization to contain your PingOne MFA environments, application connections, user populations, and settings.

Steps

1. If you don't have a PingOne MFA license, sign up for a free trial.
    1. Go to Try Ping for Free on the Ping Identity site.
    2. Complete the sign up form. Click Submit.
    3. Verify your email using the link emailed to you.
    4. Go to the unique login link provided when you verified your email address.
    5. Enter your email address and password. Click Sign On.
2. On the PingOne console, add an environment to contain your PingOne MFA connections and settings.
    1. Click Add Environment.
    2. On the Create Environment page, in the What kind of environment do you want to create? section, select Build your own solution.
    3. Select the PingOne MFA and PingFederate tiles. Click Next.
    4. Select your deployment preferences. Click Next.
    5. In the Environment Name field, enter a name of your choosing.
    6. Optional: In the Description field, enter a description.
    7. From the Region list, select the region closest to you for the best performance.
    8. From the License list, select your PingOne MFA trial license.
    9. Click Finish.
3. On the Your Environments page, open the PingOne MFA console by clicking the PingOne MFA icon for your new environment.

Copyright © 2025 Ping Identity Corporation

Creating a web or native OIDC application in PingOne

Create an application in PingOne to allow PingFederate to access PingOne authentication flows and policy decisions. This application represents the user-facing sign-on experience.

About this task

You can find more information about application types in Applications in the PingOne documentation.

Steps

1. On the PingOne MFA console, go to Applications > Applications and click the + icon.
2. Create the application profile by entering the following:

   Application name

   A unique identifier for the application.

   Description (optional)

   A brief characterization of the application.

   Icon (optional)

   A graphic representation of the application. Use a file up to 1MB in JPG, JPEG, GIF, or PNG format.
3. In the Application Type section, select Native or OIDC Web App, then click Save.

   > **Note**
   >
   > If you will be Creating a CIBA authentication policy in PingOne, make sure to select **Native**.

   OIDC applications support the following authentication methods:

   - SMS
   - Email
   - Third-party authentication apps
   - Push notifications

   Native OIDC applications provide the following:

   - Device integrity checks to prevent users from pairing or authenticating with compromised devices.
   - Client Initiated Backchannel Authentication (CIBA) support.
4. On the Configuration tab, in the General section, note the application credentials.

   You'll use these in Configuring an adapter instance.
5. On the Configuration tab, click the Pencil icon and enter the following information:

   1. In the Response Type section, select only Token and ID Token.

Copyright © 2025 Ping Identity Corporation

2. In the Grant Type section, select only Implicit.
3. Click Save.

6. On the Resources tab, click the Pencil icon and configure the following:
    1. Select the checkboxes next to the scopes that you want to use to add them to the Selected Scopes tab, then click Save.

       Scopes are not relevant for this integration.
7. Click the Attribute Mappings tab, then click the Pencil icon and configure the following:
    1. Map user attributes to attributes in your application, then click Save.
8. If you're creating a native OIDC application for mobile device or CIBA authentication, enter your Android and iOS configuration details:
    1. On the Mobile tab, click the Pencil icon.
    2. Click Configure for Android and enter your app details.

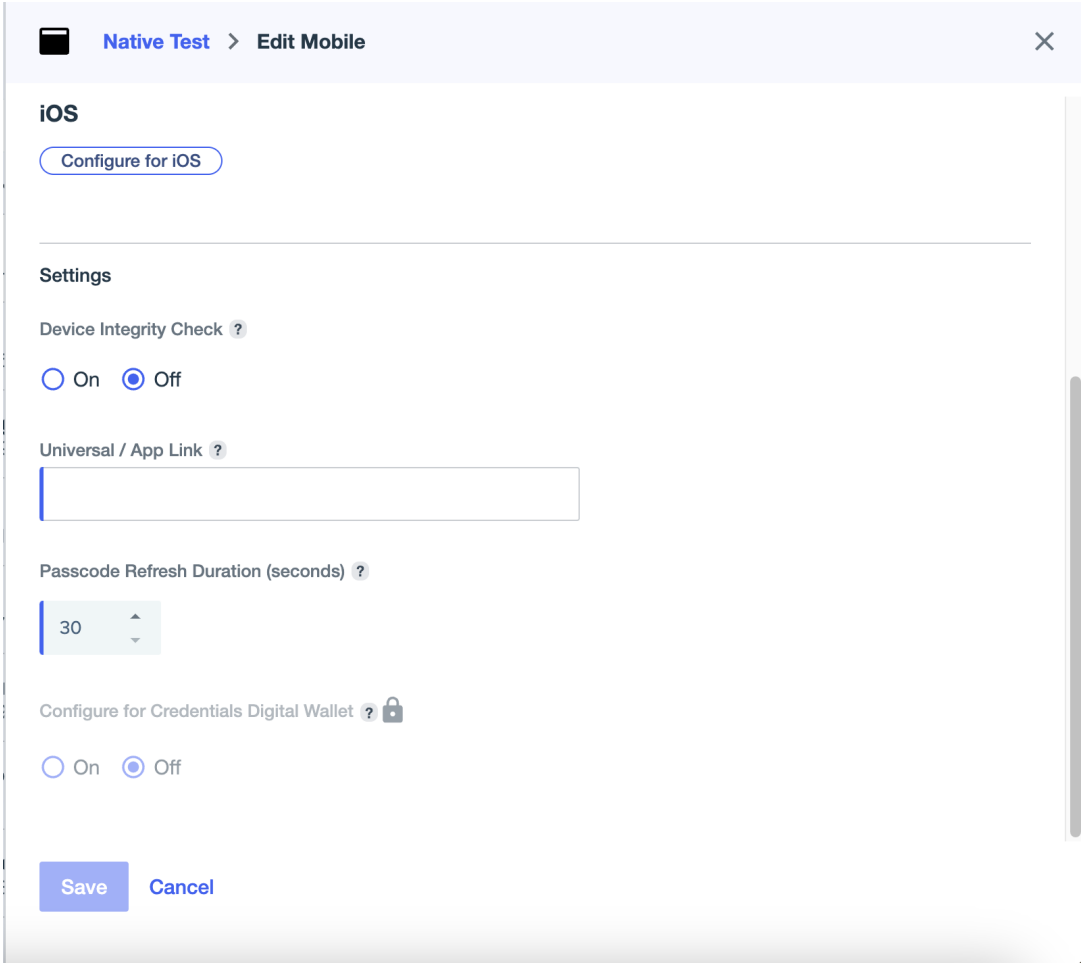
       You can find help in the following section of the Ping Identity GitHub site:

        - PingOne Mobile SDK - Android
    3. Click Configure for iOS and enter your app details.

       You can find help in the following section of the Ping Identity GitHub site:

        - PingOne Mobile SDK - iOS
    4. (Optional) Turn on the Device Integrity Check.

Copyright © 2025 Ping Identity Corporation



Learn more in Device Integrity Checks.

5. Click Save.

9. Click the toggle at the top of the application to activate it.

Connecting PingFederate to PingOne

Create a connection between PingFederate and PingOne. You can use this connection for all PingFederate components that communicate with PingOne.

> **Note**
>
> If you already have a connection from PingOne to PingFederate, ensure that the **Identity admin** and **Environment admin** roles are enabled. Skip the steps below.

Steps

1. Create credentials in PingOne:
    1. On the PingOne administrative console, go to Integrations > PingFederate and click the + icon.
    2. On the Create Connection Profile window, enter a connection name, then click Save.

Copyright © 2025 Ping Identity Corporation

3. After the New Credential Created window opens, click Copy To Clipboard, then Done.

2. Use the credentials to create a connection in PingFederate:

    1. On the PingFederate administrative console, go to System > External Systems > PingOne Connections and click + Add Connection.

    2. On the Add Connection window, paste the credential information in the Paste The Credential Here field.

    3. In the Connection Name field, enter a unique name and click Save.

Use cases

You can use the PingOne MFA Integration Kit for a variety of solutions.

As you follow the steps in MFA setup or CIBA setup, consider these guides to specific use cases.

Enabling user and authentication method provisioning

The PingOne MFA IdP Adapter supports automatic provisioning for users and for some user authentication methods.

About this task

When a user signs on, the adapter gets their username and contact information from attributes in the PingFederate authentication policy. The adapter can use this information to create a user (if necessary) and configure authentication methods for the user in PingOne. This process, called just-in-time provisioning (JIT), happens automatically.

Important

You can only use this technique to provision the following authentication methods:

- SMS
- voice
- email

Note

To enable JIT provisioning, follow this guide while you complete the steps in MFA setup.

Steps

1. Complete the following steps in your adapter configuration:

    1. Select the Provision Users checkbox.

    2. Optional: To add authentication methods for the new user based on the user's SMS Attribute, Voice Attribute, and Email Attribute values, select the Provision Authentication Methods checkbox.

Copyright © 2025 Ping Identity Corporation

> **Note**
>
> To automatically add new authentication methods for existing users, make sure that the **Update Authentication Methods** checkbox is selected.
> If **Update Authentication Methods** is selected, you can select an **Overwrite Authentication Methods Configurations** setting to determine whether the adapter replaces existing SMS, voice, or email authentication methods or just adds the new methods when the adapter identifies that there are new authentication method values.

3. To use a different identifier than a username, enter the name of the attributes that will contain the identifier in the Username Attribute field.
    - If you identify users based on their PingOne username, leave this field blank. New users are named based on the "incoming user ID" set for the adapter in your PingFederate authentication policy.
    - If you identify users based on their PingOne user ID, enter the name of an authentication policy attribute. New users are named based on the attribute instead of the "incoming user ID."
4. If you selected Provision Authentication Methods or Update Authentication Methods, enter the name of the attributes that will contain the user's contact information in the SMS Attribute, Voice Attribute, and Email Attribute fields if the attributes differ from the default values.

    The default values are `SMS` for SMS Attribute, `voice` for Voice Attribute, and `email` for Email Attribute.
5. If you selected Provision Authentication Methods, select a Default Authentication Method for Provisioned Users.

    The default selection is `SMS`.

Learn more about any of the above fields in the PingOne MFA IdP Adapter settings reference. For help configuring an adapter instance, see Configuring an adapter instance.

2. When Adding PingOne MFA to your authentication policy, configure the policy path so that the attributes you named are populated before the PingOne MFA IdP Adapter is triggered.

    For example, you could use the HTML Form Adapter to provide a sign-on form and get the user's phone number from your datastore.

Enabling the MFA setup prompt

The PingOne MFA IdP Adapter allows you to prompt users to set up their first multi-factor authentication (MFA) method.

About this task

To prompt a user to set up their first MFA device:

Steps

1. When Configuring an adapter instance:
    1. Select the Prompt Users to Set Up MFA check box.
    2. If you want to make MFA optional, select the Allow Users to Skip MFA Setup check box.

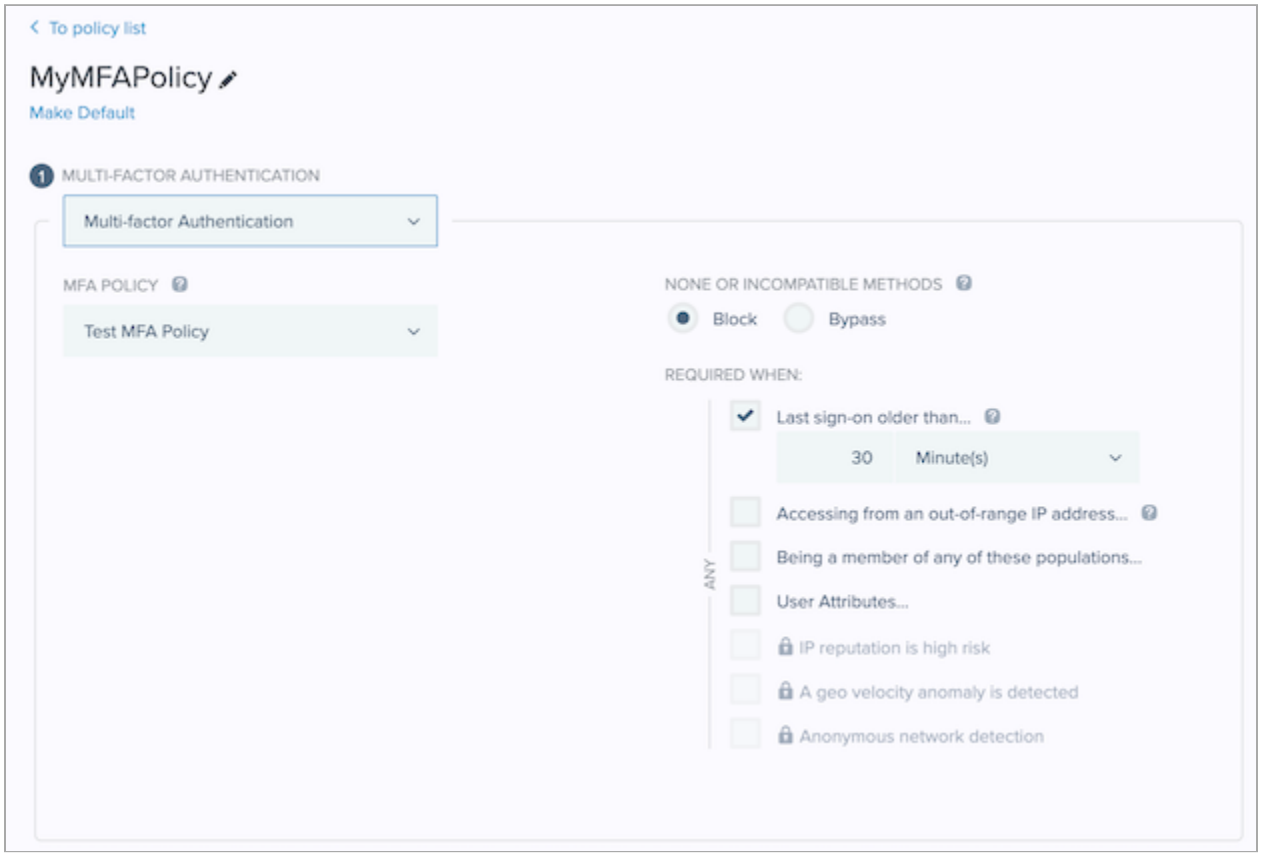
Copyright © 2025 Ping Identity Corporation

This lets the user sign on without adding an authentication method.

2. When Creating an MFA authentication policy in PingOne, in the None Or Incompatible Methods section, select a default behavior for cases where the user does not have a valid authentication method set up:

   Choose from:

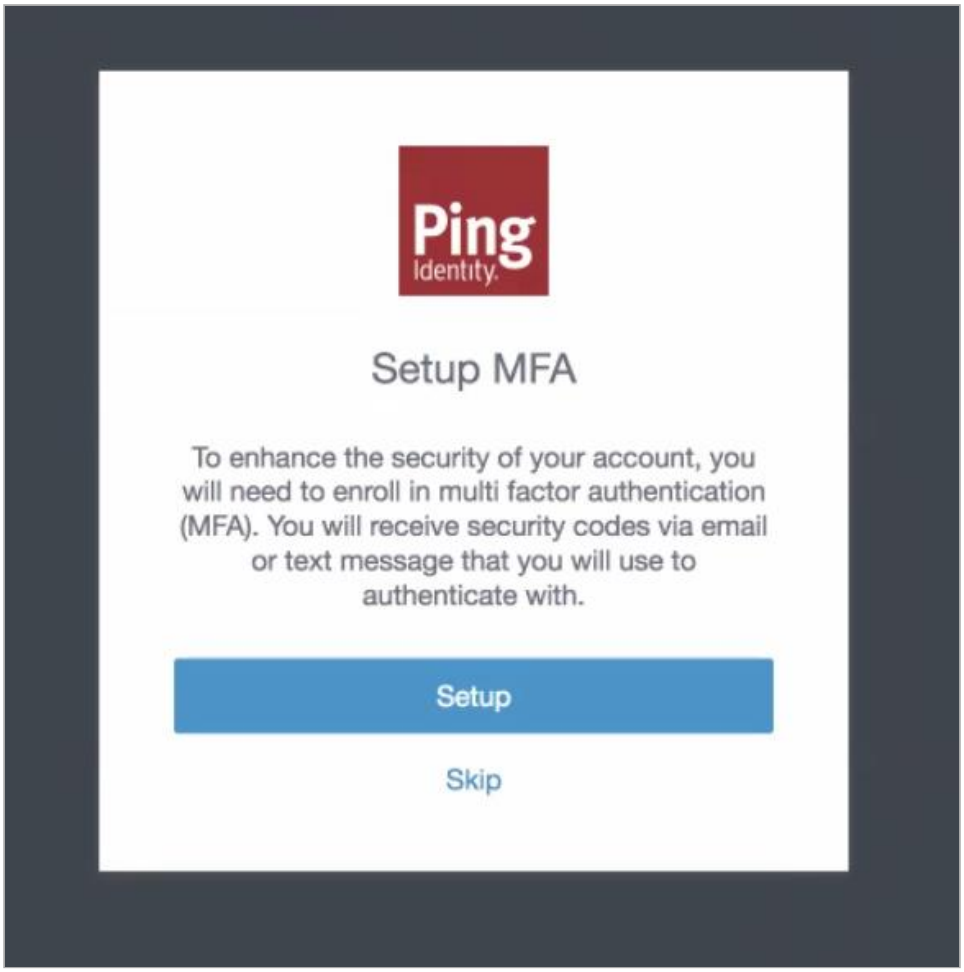
   - Block: If the user doesn't have a valid authentication method set up, MFA fails.
   - Bypass: If the user doesn't have a valid authentication method set up, they continue to the next step as if they completed MFA successfully.



Result

After a user completes first-factor authentication, the adapter gets a list of the user's second-factor authentication methods from PingOne. For users that don't have any existing authentication methods, the MFA setup prompt appears.

Copyright © 2025 Ping Identity Corporation



The user can click Setup, then select the type of authentication method they want to add. You can include an optional Skip button if you don't want to force users to set up MFA.

Next steps

To use the MFA setup prompt in an authentication policy, make sure to add the P1 MFA IdP adapter to your authentication policy and configure an MFA step.

Transaction approval setup

Transaction approval requests allow you to require authentication for certain actions, such as sending a money transfer or accessing a high-security resource. The PingOne MFA IdP Adapter allows PingFederate to trigger transaction approval requests using PingOne MFA.

> **Note**
>
> To enable this feature, complete the steps in MFA setup, and then use this as a guide to pass transaction approval parameters to the adapter.
> You can use the same adapter instance for both the multi-factor authentication (MFA) and transaction approval flows, but you'll need a separate notification template variant for each flow. Create the variants in PingOne.

Copyright © 2025 Ping Identity Corporation

Overview

For all flow types, the PingOne MFA IdP Adapter passes a set of parameters to PingOne. These parameters indicate which notification template PingOne should use, and provide the values for the variables used in that template.

The only difference between the MFA and transaction approval flow is the attributes that the adapter receives:

- For MFA flows, these parameters are provided automatically.
- For transaction approval flows, you must pass the parameters into the PingOne MFA IdP Adapter.

The following sections describe the PingOne notification templates, the parameters that the adapter collects, and the sources that it collects those parameters from.

> **Note**
>
> Variants for PingOne notification templates are only available through the PingOne API at this time.

PingOne notification templates

A PingOne notification template controls the presentation of the transaction approval request. These templates combine customizable text and variables to form the message that the user sees. Learn more in Notification templates in the PingOne documentation.

We recommend that you create a variant of the default PingOne "transaction" template. For example, if you wanted to trigger an authentication challenge for money transfers, you could create the template as follows:

Template type

Transaction

Template variant

Money transfer

Subject

Confirm your money transfer

Body

Enter `${otp}` to confirm your transfer of `${sum} ${currency}` to `${recipient}`.

With the variable values populated, the example template might read as follows:

```
Confirm your money transfer
Enter 454879 to confirm your transfer of $40 USD to jsmith@example.com.
```


Copyright © 2025 Ping Identity Corporation

The `${otp}` variable is required for the SMS and Email versions of the transaction template, but all other variables and text are completely customizable. The only requirement is that PingOne must receive a value for each variable that you include.

Transaction approval parameters

For all flow types (including MFA, transaction approval, and CIBA), the PingOne MFA IdP Adapter passes a set of parameters to PingOne. These parameters indicate which notification template to use, and provide the values for the variables used in those templates.

- For MFA flows, these parameters are provided automatically.
- For transaction approval flows, you must pass the parameters into the PingOne MFA IdP Adapter.

The parameters are structured as follows:

pi.template.name

PingOne uses this parameter to determine which notification template to use.

To trigger PingOne to use the "transaction" notification template, this value must be `transaction`.

pi.template.variant

PingOne uses this parameter to determine which variant of the notification template to use.

For example, if you created a "Money transfer" variant of the transaction template, this value would be `money_transfer`.

pi.template.locale

Use `pi.template.locale` to specify the language to use for notifications, for example, `fr-CA`.

pi.template.variables.<variable_name>

PingOne uses these parameters to populate variables (other than `${otp}`) in the transaction template. How you use these parameters depends on the design of your transaction template.

For example, if you make a `pi.template.variables.recipient` parameter available to the adapter, PingOne will use its value to replace `${recipient}` in your transaction template.

pi.clientContext.<variable_name>

When the transaction is initiated by a mobile app with the embedded mobile SDK, PingOne makes these parameters available to the mobile app. How you use these parameters depends on the design of your mobile app.

For example, you could use a `pi.clientContext.alert.message` parameter to provide the mobile app with text for an alert.

After the PingOne MFA IdP Adapter collects values for these parameters, it creates a JSON object with the following structure:

Copyright © 2025 Ping Identity Corporation

```
"pi.template": {
  "name": "transaction",
  "variant": "money_transfer",
  "variables": {
    "sum": "1,000,000",
    "currency": "USD",
    "recipient": "Charlie Parker"
  }
},
"pi.clientContext": {
  "alert.message": "Confirm your money transfer"
}
```

This JSON object is included in a request that the adapter makes to PingOne.

Parameter sources

The adapter collects values from the following sources:

- Chained attributes passed through the PingFederate authentication policy
- Tracked parameters in the PingFederate authentication policy
- Signed object requests made as part of the OpenID Connect flow from the client application to PingFederate

You can provide parameters to the PingOne MFA IdP Adapter in any grouping you want:

- A complete set of parameters (as shown in the JSON object in the previous section)
- Several groups of parameters
- Individual parameters

The adapter collects parameters from all sources and merges them into a single complete JSON object to send to PingOne.

If more than one source provides conflicting parameters, each of the following sources takes precedence over the ones below it:

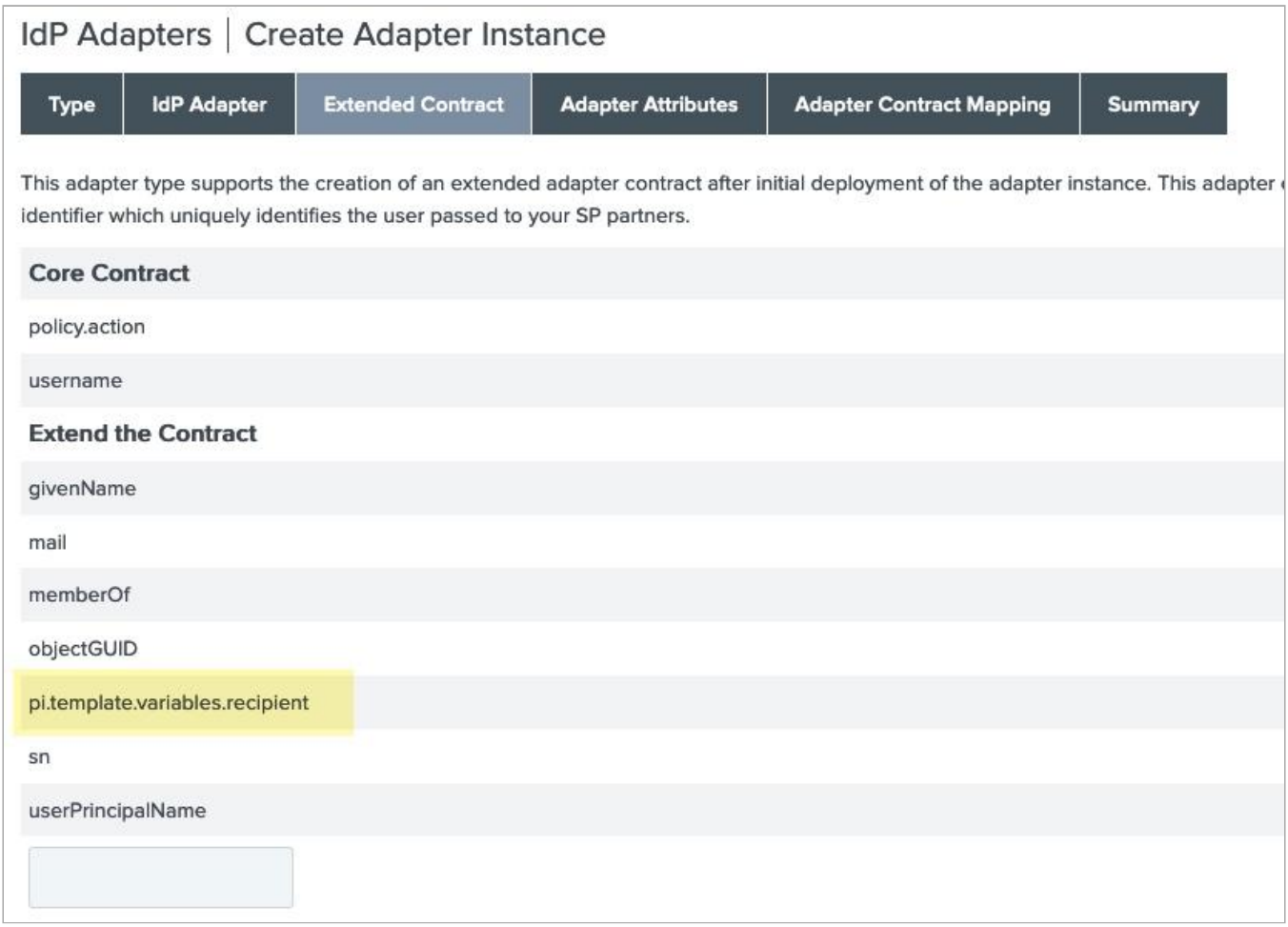
1. Chained attributes
2. Signed object requests
3. Tracked parameters

Providing parameter values using chained attributes

You can provide some or all parameters as chained attributes in the PingFederate authentication policy.

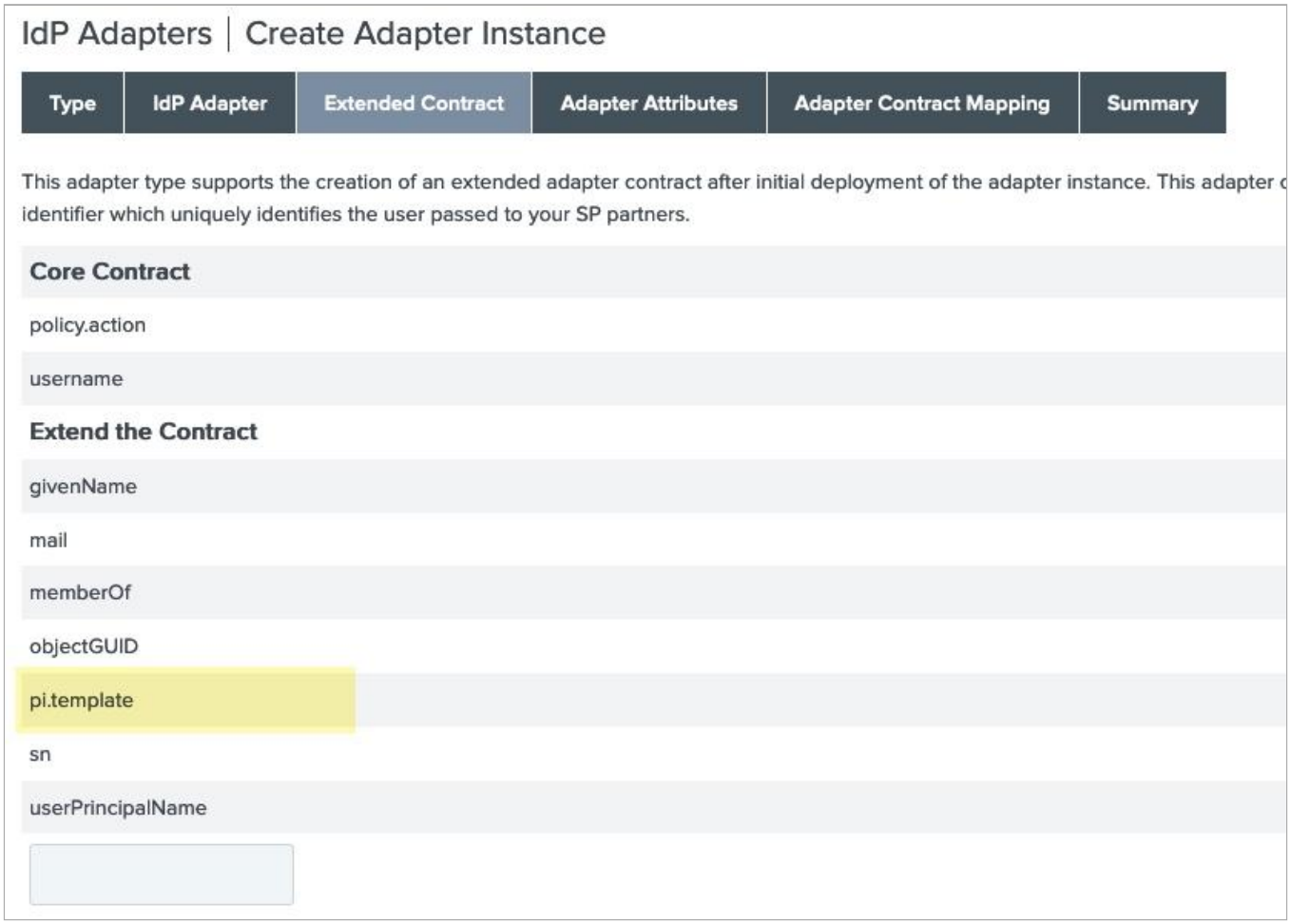
Copyright © 2025 Ping Identity Corporation

For example, you can extend the HTML Form Adapter contract to provide a single parameter, such as `pi.template.variables.recipient`. Your adapter contract mapping determines the value, such as `Charlie Parker`.



Alternately, you could extend the HTML Form Adapter contract to provide the `pi.template` parameter with a collection of values in it as JSON object.

Copyright © 2025 Ping Identity Corporation



Your adapter contract mapping determines the value. In this example, you could populate `pi.template` with a JSON object, such as:

```
{
  "name": "transaction",
  "variant": "money_transfer",
  "variables": {
    "sum": "1,000,000",
    "currency": "USD",
    "recipient": "Charlie Parker"
  }
}
```

> **Note**
>
> You do not have to map the chained attribute into the PingOne MFA IdP Adapter in your authentication policy.

Copyright © 2025 Ping Identity Corporation

Providing values in a signed request object

In an OpenID Connect (OIDC) request from your client to PingFederate, you can provide some or all values in the body of the signed request object.

In the following example, the client has provided a complete set of parameters:

```
https://pf_host:pf_port/as/authorization.oauth2?client_id=ac_oic_client
  &response_type=code
  &scope=openid
  &state=1553ba67-3f92-4768-a3c3-669d0a562d75

&request=eyJraWQiOiJrMSIsImFsZyI6IlJTMjU2In0.eyJpc3MiOiJhY19vaWNfY2xpZW50IiwiYXVkIjoiaHR0cHM6Ly9sb2NhbGhvc3Q6OTAzMSIs
VR7a_FaM2_1eUkjoMwLQWGWfWb0Ao-4tpDFqWPFWcdY5Su8G494tl4iQirz1DzX2yjExWO_uBkFj0y3CmDad1mBK7kWfxhkpuUHwdn74tYLcd0g1uWGbV
xwCZL6Q
  &redirect_uri=https://pf_host:pf_port/OAuthPlayground/authorization_code/oidc/callback
```

The decoded body of the signed `request` object above is:

```
{
  "aud": "https://auth.pingone.com/{envId}/as",
  "iss": "{applicationId}",
  "pi.template": {
    "name": "transaction",
    "variant": "money_transfer",
    "variables": {
      "sum": "1,000,000",
      "currency": "USD",
      "recipient": "Charlie Parker"
    }
  },
  "pi.clientContext": {
    "alert.message": "Confirm your money transfer"
  }
}
```

In this example, the adapter has everything required to send the request to PingOne. It will still collect any available parameters from other sources and merge them into the JSON object provided by the client.

> **Note**
>
> The adapter processes parameter values that are formatted as JSON objects, shown in the example above, or JSON strings.

Providing parameter values using query parameters

In an HTTP request to PingFederate, you can provide some or all parameters using request parameters.

Copyright © 2025 Ping Identity Corporation

In the following example, only the `pi.template.variables.sum` and `pi.clientContext.alert.message` parameters are provided. The adapter will look for the `pi.template.name`, `pi.template.variant`, and other parameters from other sources.

```
https://pf_host:pf_port/idp/startSSO.ping?PartnerSpId=SP_connection_ID
  &pi.template={
    "variables": {
      "sum": "1,000,000"
    }
  }&pi.clientContext={
    "alert.message": "Confirm your money transfer"
  }
```

You can use a mix of parameters that contain a single value and parameters that contain JSON objects:

```
https://pf_host:pf_port/idp/startSSO.ping?PartnerSpId=SP_connection_ID
  &pi.template={
    "name": "transaction",
    "variables": {
      "sum": "1,000,000",
        "recipient": "Charlie Parker"
    }
  }&pi.clientContext={
    "alert.color": "red"
  }
&pi.template.variant=money_transfer
&pi.template.variables.currency=USD
&pi.clientContext.alert.message=Confirm%20your%20money%20transfer
```

Using FIDO device with PingOne custom domain configuration

The PingOne MFA IdP Adapter supports FIDO device pairing and authentication with FIDO services. You must register the paired devices with the PingFederate domain.

FIDO devices that are paired directly using the PingOne self-service do not work with the PingFederate authentication flow unless you register the devices with either Custom Domain or Other configured as the Relying Party ID and make them visible to the PingFederate domain.

When you choose either of these options as the Relying Party ID, you can configure the PingFederate domain to run on its own subdomain of the custom domain by changing the PingFederate base URL or using a virtual host name. Configuring the PingFederate domain to run as its own subdomain enables you to use FIDO devices interchangeably.

- You can find information on how to set up a custom domain in PingOne in Setting up a custom domain.
- You can find information on changing the PingFederate base URL or using a virtual host name in Specifying federation information or Virtual host names in the PingFederate documentation.

Enabling passwordless login using FIDO browser management

The PingOne MFA IdP Adapter supports passwordless login by using a cookie to store FIDO device information for an elevated user experience.

Copyright © 2025 Ping Identity Corporation

About this task

The PingOne MFA IdP Adapter can create a cookie when a FIDO-based authentication is successful. The cookie supports a passwordless experience by remembering the FIDO device and auto-prompting the user for biometrics for all future logins. The passwordless capabilities of the PingOne MFA IdP Adapter are designed to be used with the Identifier First Adapter and HTML Form Adapter.

> **Note**
>
> It's assumed that users are limited to FIDO biometrics and QR code as supported authentication methods.

Steps

1. **When** Configuring an adapter instance**, do the following:**
    1. **Select the Enable Cookie Based Tracking check box.**
    2. **Enter an attribute in the Use Password Config Attribute field.**
    3. **Enter an attribute in the Bypass MFA for Device Pairing Attribute**
    4. **Select the Enforce Device Selection check box.**
2. **When** Adding PingOne MFA to your authentication policy **do the following:**
    1. **Configure the attribute you defined in the Use Password Config Attribute field with value `true` from IDFirst adapter.**

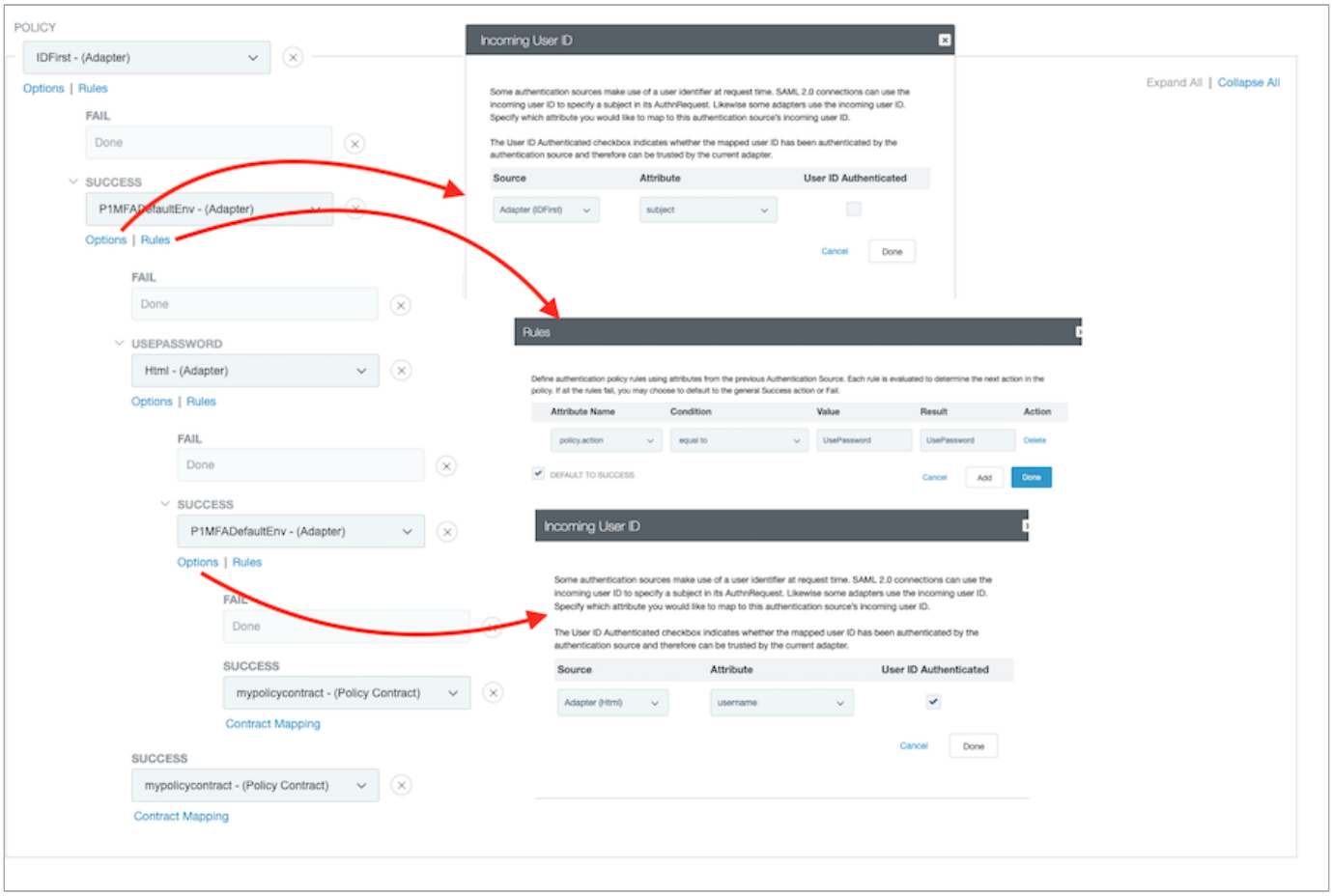
        Result:

        This policy configuration should send the value `false` for the last PingOne MFA IdP Adapter because the Use Password policy action is not applicable for the last adapter in the policy.
    2. **Configure the attribute you defined in Bypass MFA for Device Pairing Attribute received with value `true` from the HTML form adapter.**

        Result:

        This configuration allows the second PingOne MFA IdP Adapter to bypass the authentication requirement upon new pairing.

Copyright © 2025 Ping Identity Corporation



Result

The time the user attempts to login, the cookie will be received. The PingOne MFA IdP Adapter will know the user ID to initiate authentication flow with PingOne and automatically select the device ID to trigger passwordless authentication.

FIDO browser management use cases

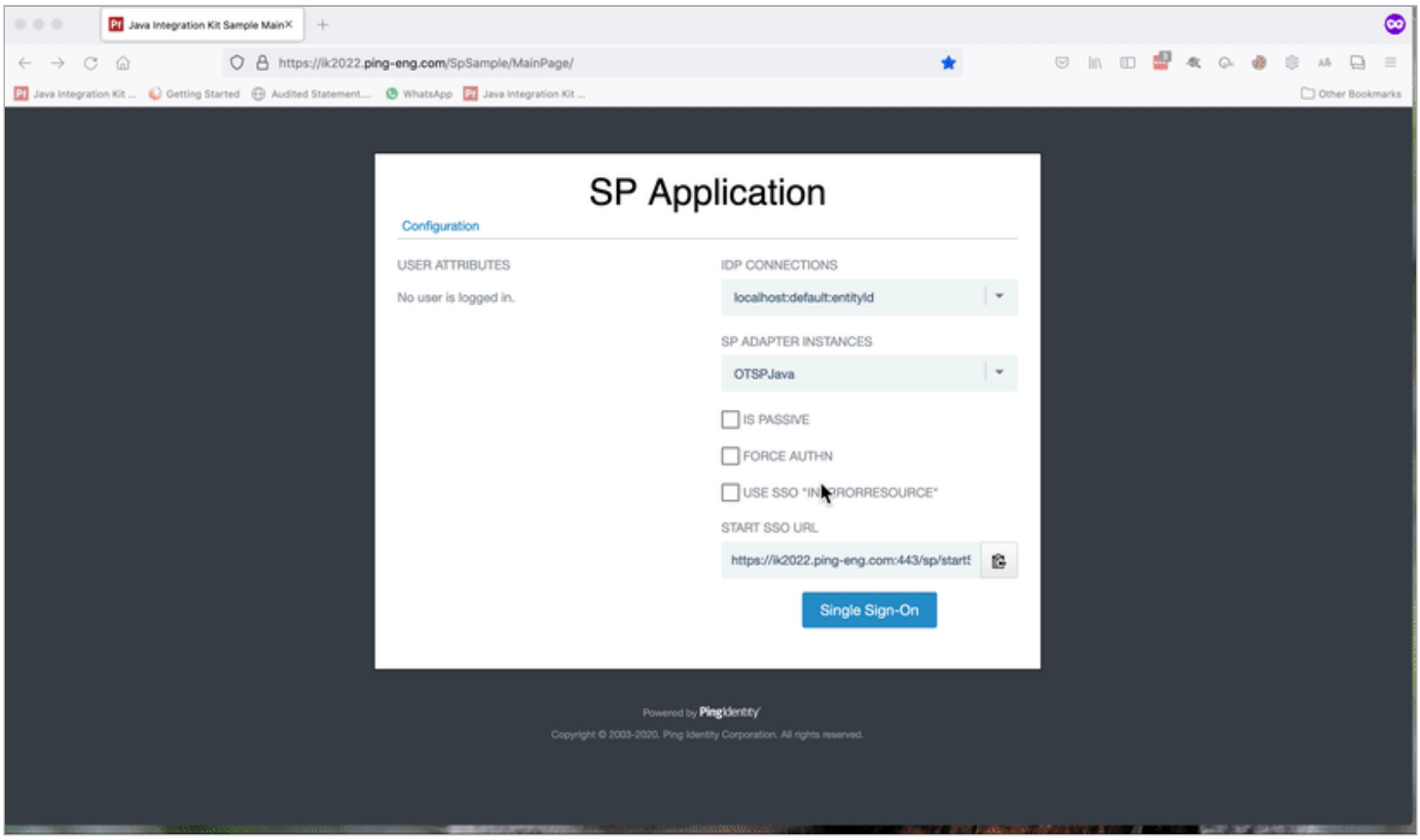
The following process flows describe the possible use cases experienced when using FIDO browser management.

Registering a new FIDO device

1. **The user initiates an authentication flow on a FIDO-supported device.**
2. **The user enters their username (supported by the identifier first adapter).**
3. **If no devices have been paired, the user is prompted to authenticate through the HTML form. If the user has previously paired a device, the user is prompted to select from a list of devices or might fallback to the HTML form adapter for their first factor authentication.**
4. **After the user has authenticated, they are given the option to pair biometrics for the device. See Configuration Step 3 for optional configuration of the Device Selection screen.**

Copyright © 2025 Ping Identity Corporation

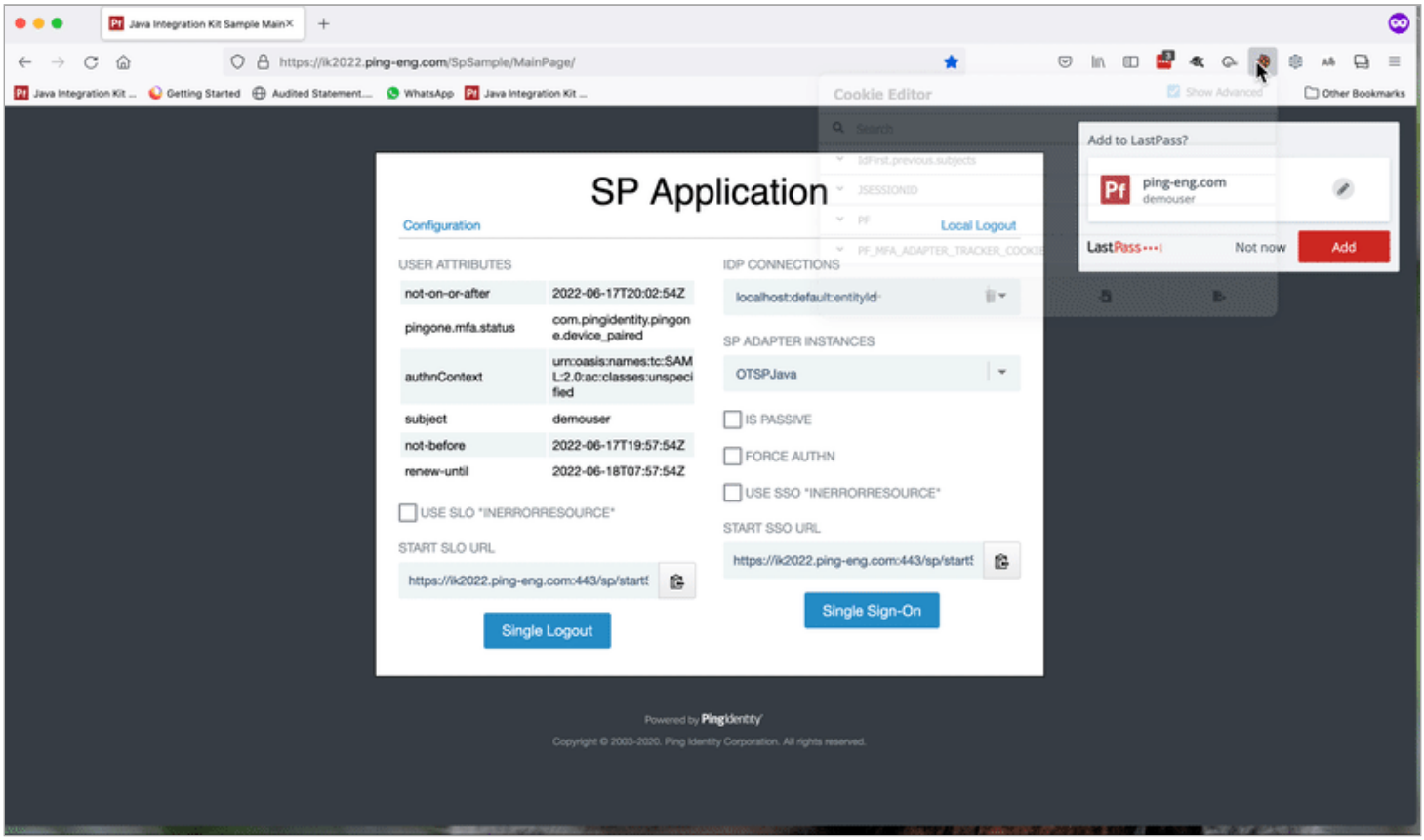
5. The user pairs their biometrics through FIDO and completes the login flow. A cookie with the device ID is placed on the browser with an expiration of 20 years.



Authenticating a paired FIDO device

1. The user initiates an authentication flow on a paired, FIDO-supported device by entering their username into a form supported by the identifier first adapter.
2. A cookie is observed and the user is prompted to authenticate with biometrics for their current device.
3. The user authenticates with biometrics and logs in.

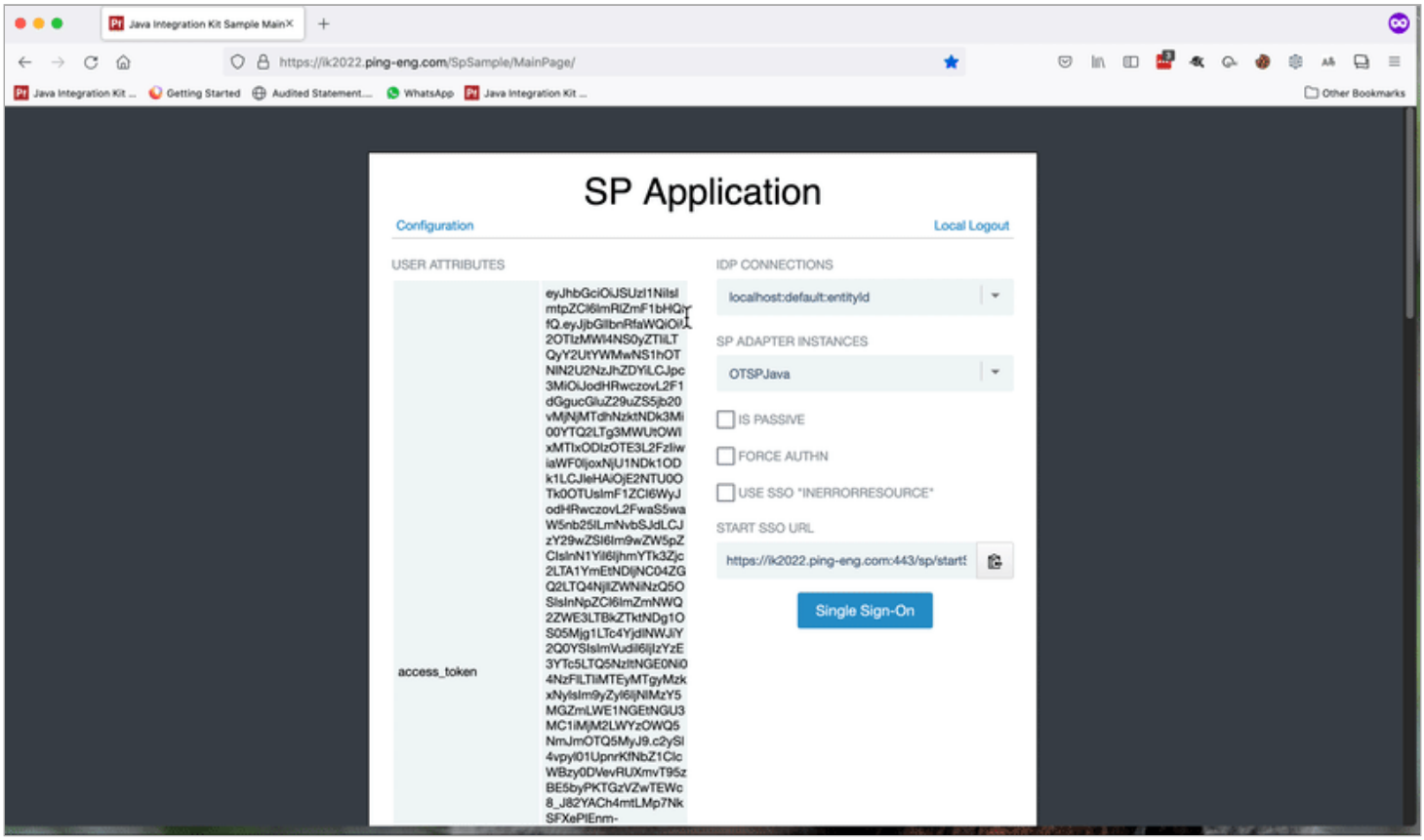
Copyright © 2025 Ping Identity Corporation



The cookie is deleted for a paired FIDO device

1. **The user initiates an authentication flow on a FIDO-supported device.**
2. **The user enters their username supported by the identifier first adapter.**
3. **The user is prompted to select a device or fallback to the HTML form.**
4. **The user selects their current device from the list and authenticates with biometrics, completing login.**
5. **A cookie is reset on the browser with an expiration of 20 years.**

Copyright © 2025 Ping Identity Corporation



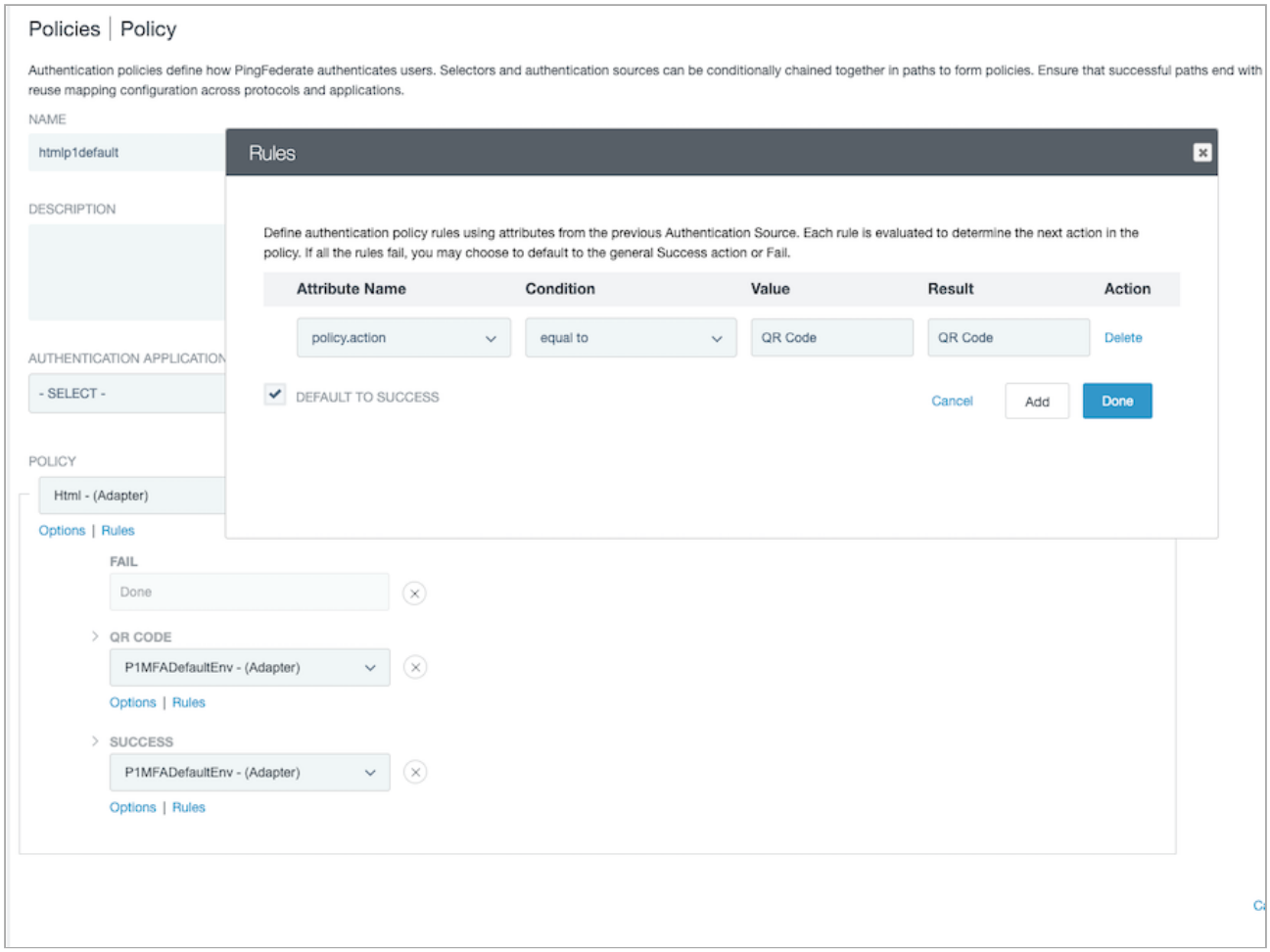
Support for authentication code flow

The PingOne MFA Integration Kit supports authentication code flow if a local identity profile (LIP) has been configured. You can find more information about LIPs in Configuring local identity profiles.

Process flow

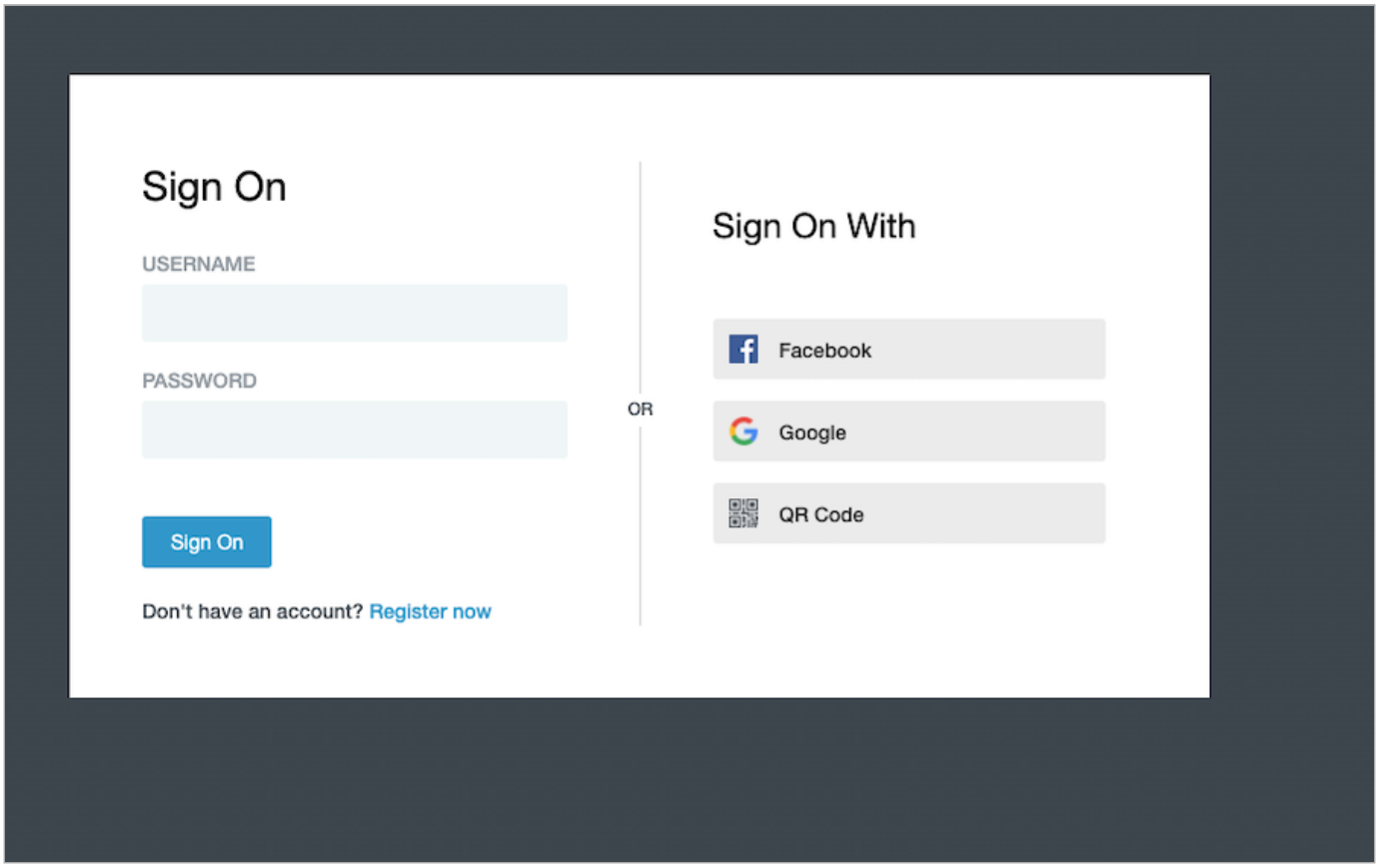
1. The PingOne MFA adapter invokes authentication code flow if the adapter is invoked using a `policy.action` attribute containing the keyword "QR"(case-insensitive). For example, the `policy.action` attribute could be "QR code."

Copyright © 2025 Ping Identity Corporation



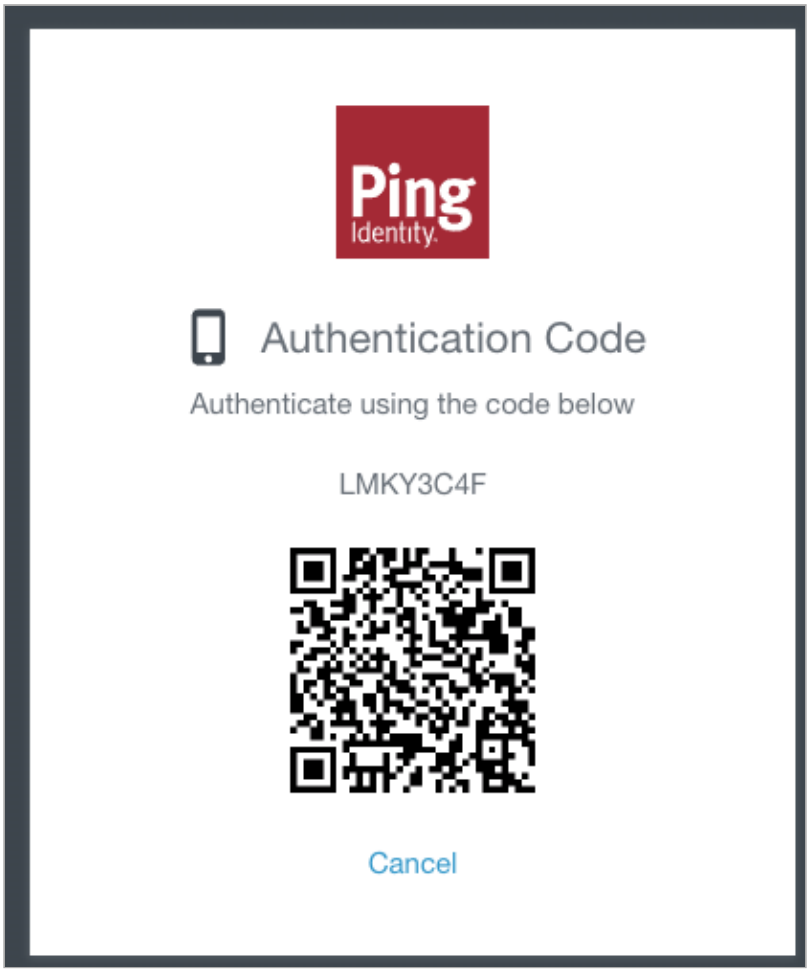
This allows PingFederate to use the built-in QR code image for the Sign-On page.

Copyright © 2025 Ping Identity Corporation



2. When the PingOne MFA adapter is invoked in this mode, the adapter automatically takes the user through authentication code-based authentication flow for the configured application ID.

Copyright © 2025 Ping Identity Corporation



3. This template polls continuously to check for status. If the authentication code expires before user action, a new authentication code is requested automatically and displayed.
4. If approved and authentication flow is successful, the adapter returns the userID and username in the core contract `authentication.code.flow.userid` and `username` respectively along with `pingone.mfa.status` set to the value `com.pingidentity.pingone.mobile_login_authentication_code`.
5. If approval is denied (when user approval is required), the adapter fails the authentication flow.

Complex authentication code request

The authentication code-based flow can be invoked to pass additional attributes described in MFA Authentication Code.

Copyright © 2025 Ping Identity Corporation

```
{
   "application": {
       "id": "{{mobileapplicationId}}"
   },
     "clientContext": {
       "header" : "Authentication process",
       "body": "Do you want to approve this transaction?"
 },
   "lifeTime": {
       "duration": 5,
       "timeUnit": "MINUTES"
   },
   "userApproval": "REQUIRED"
}
```

The adapter expects predefined attributes and when found maps the values automatically to corresponding request attribute as described below.

Incoming Attribute Name	Create Authentication Code API Attribute
`p1mfa.authenticationcode.clientContext.header`	`clientContext.header`
`p1mfa.authenticationcode.clientContext.body`	`clientContext.body`
`p1mfa.authenticationcode.lifeTime.duration`	`lifeTime.duration`
`p1mfa.authenticationcode.lifeTime.timeUnit`	`lifeTime.timeUnit`
`p1mfa.authenticationcode.userApproval`	`userApproval`

These attributes can be provided through:

- **Chained attributes**
- **Signed request object claims**
- **Tracked parameters with the adapter using the values in the same order**

Successful user approval will transition to `MFA_COMPLETED` state. User denial will transition to `MFA_FAILED` state.

> **Note**
>
> The bypass authentication setting when configured with the **Service Unavailable Failure Mode** field does not apply for authentication code flow. The runtime fails hard ifservice becomes unreachable for this flow.

Copyright © 2025 Ping Identity Corporation

Authentication API flow

When authN API flow is invoked for `useAlternativeAuthenticationSource` action for QR code, the adapter responds with a `AUTHENTICATION_CODE_RESPONSE_REQUIRED` state with an authentication code and several other properties as seen below.


```
{
  "id": "XC5Jt",
  "pluginTypeId": "j_AGR1E__nc2USbRhZPMQQ",
  "status": "AUTHENTICATION_CODE_RESPONSE_REQUIRED",
  "authenticationCodeId": "20842db3-b6d6-4d0b-90aa-63c9d9c81ff9",
  "code": "JQQ0GYX8",
  "uri": "pingonesdkauthentication_code=JQQ0GYX8",
  "userApproval": "REQUIRED",
  "expiresAt": "2022-05-17T22:35:04.354Z",
  "updatedAt": "2022-05-17T22:25:04.349Z",
  "createdAt": "2022-05-17T22:25:04.349Z",
  "application": {
    "id": "264748d1-4e4a-477d-b673-b4f44db45ccf"
  },
  "clientContext": {
    "header": "Approve request",
    "body": "1. Do you want to approve this?"
  },
  "lifeTime": {
    "duration": 10,
    "timeUnit": "MINUTES"
  },
  "requestStatus": "UNCLAIMED",
  "_links": {
    "cancelAuthentication": {
      "href": "https://localhost.ping-eng.com:9031/pf-ws/authn/flows/XC5Jt"
    },
    "self": {
      "href": "https://localhost.ping-eng.com:9031/pf-ws/authn/flows/XC5Jt"
    },
    "poll": {
      "href": "https://localhost.ping-eng.com:9031/pf-ws/authn/flows/XC5Jt"
    }
  }
}
```


Support for stateless one-time device OTP flow

The PingOne MFA Integration Kit supports one-time device OTP authentication flow to initiate and complete an MFA action for SMS, voice and email targets without requiring those devices to be previously paired.

> **Note**
>
> The PingOne authentication policy configuration has no impact on this flow.

Copyright © 2025 Ping Identity Corporation

Process Flow

1. The PingOne MFA adapter invokes this flow if a chained attribute with name `support-one-time-device-otp` is received with value as `TRUE` along with valid target device information.
2. The target device for email, voice, and SMS types are provided through `one-time-device-email`, `one-time-device-voice` and `one-time-device-sms` respectively. The value can be a single-valued string or multi-valued attribute value from the previous adapter.

Expected Attribute Name	Expeted Value	Description
`support-one-time-device-otp`	Boolean: True or False	True is required for consideration of this flow.
`one-time-device-email`	Single-valued string or mutli-valued string for multiple values.	The OTP will be sent to this email address.
`one-time-device-sms`	Single-valued string or mutli-valued string for multiple values. **Note** The phone number must be in the E.164 standard format.	The OTP will be sent to this phone number through SMS.
`one-time-device-voice`	Single-valued string or mutli-valued string for multiple values. **Note** The phone number must be in the E.164 standard format.	The OTP will be delivered to this phone through voice call.

3. If more than one target is specified, users are transitioned first to the device selection step, and to OTP authentication flow after device selection. If the flow is invoked with just one specific target device, the user goes directly through the OTP authentication flow for that device.

Support for FIDO usernameless authentication flow

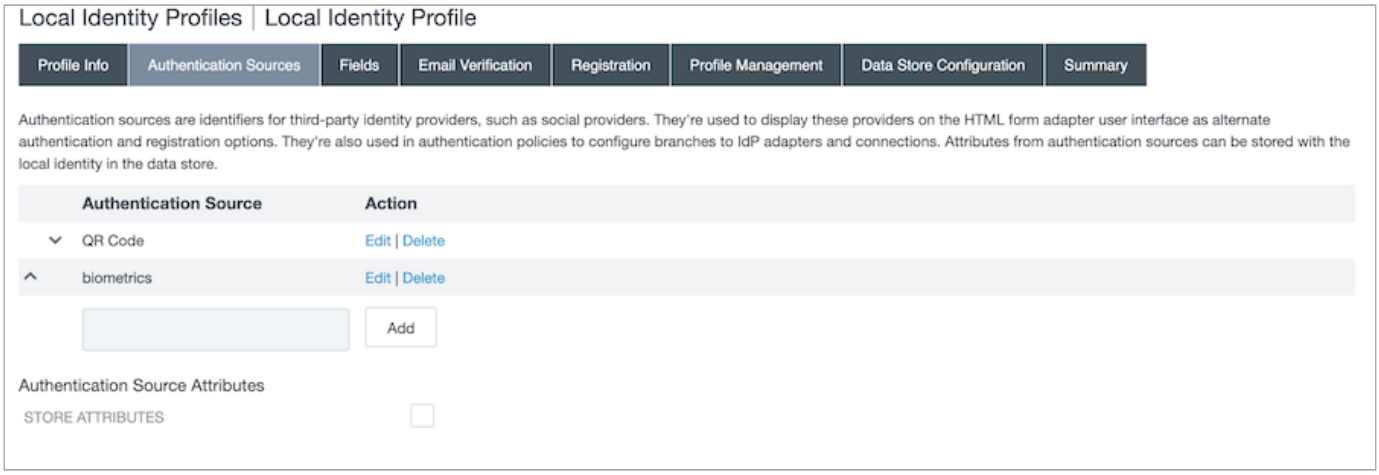
The PingOne MFA Integration Kit supports Fast IDentity Online (FIDO) usernameless authentication code flow if a local identity profile (LIP) has been configured. For more information on LIPs, see Configuring local identity profiles.

Copyright © 2025 Ping Identity Corporation

Process flow

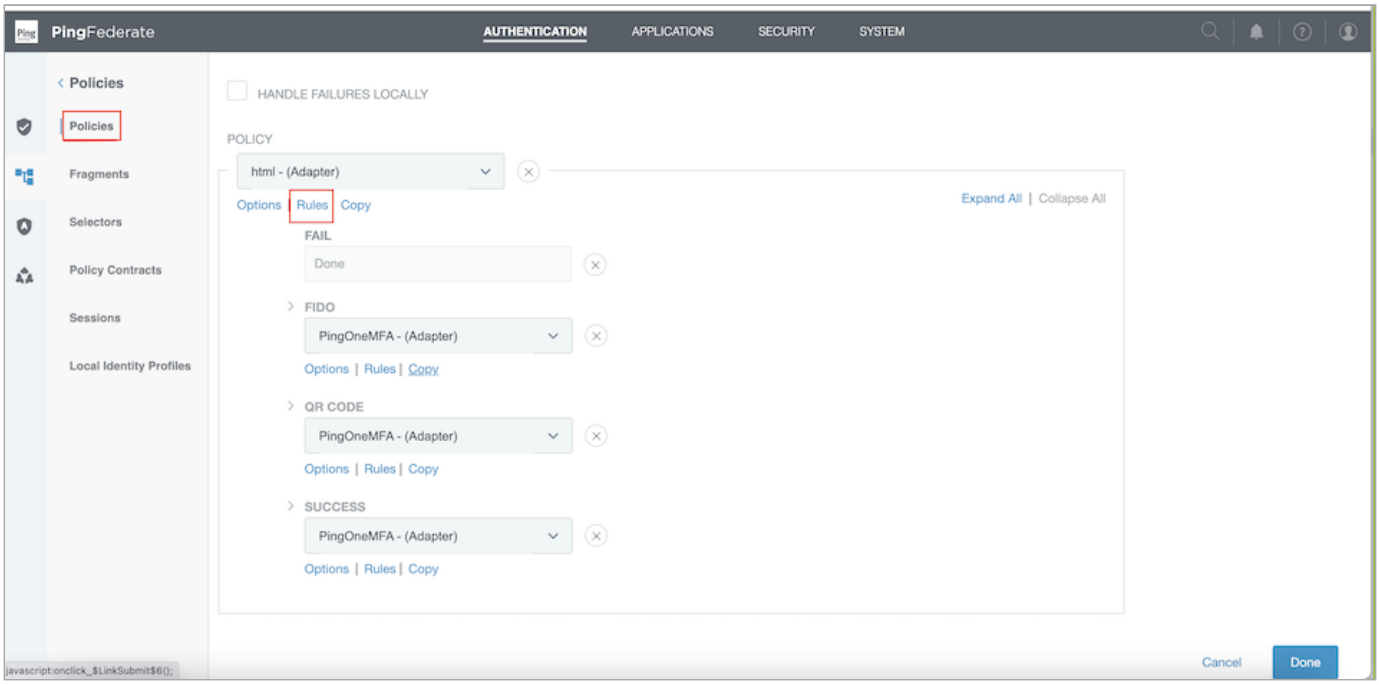
1. The PingOne MFA adapter invokes FIDO usernameless authentication flow if the adapter is invoked using a `policy.action` attribute containing one of the following keywords:
    - `biometrics`
    - `TouchID`
    - `FaceID`
    - `FIDO`

    On the Authentication Sources tab of the Local Identity Profile page in the PingFederate administrative console, add the keyword that you want to use in the Authentication Source section. The following screen captures use the biometrics keyword as an example.



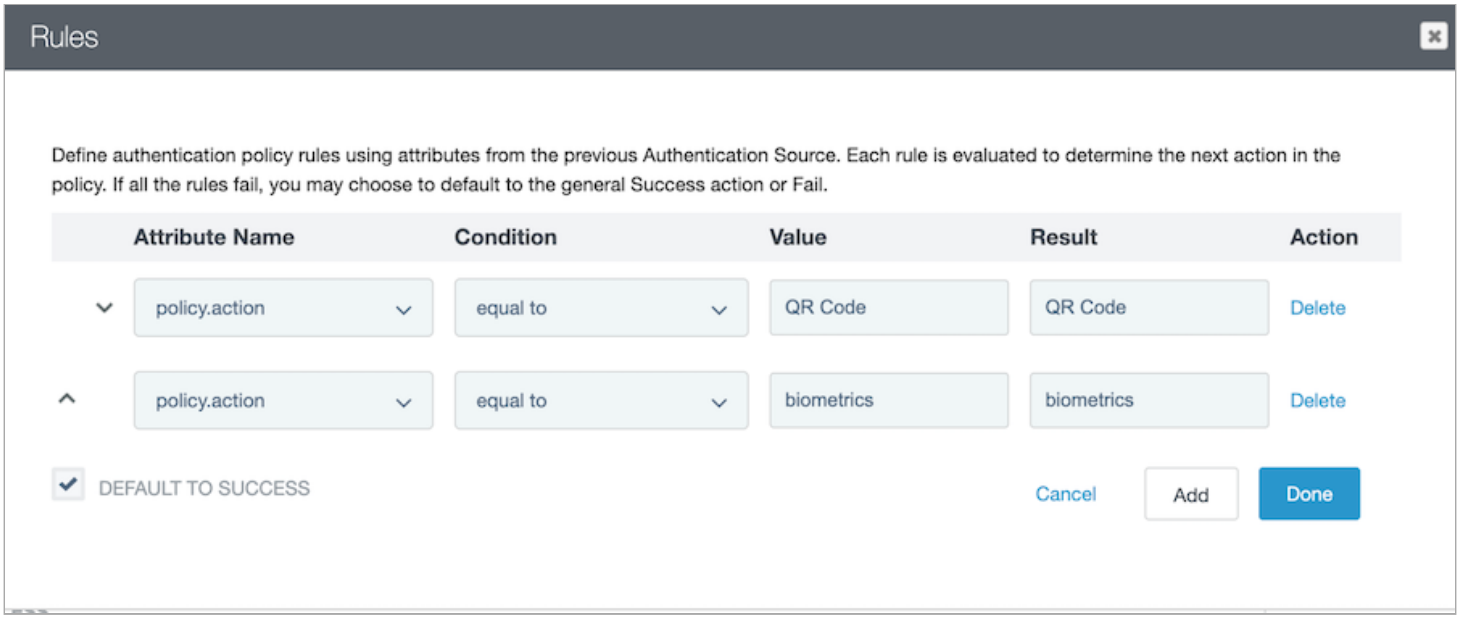
2. In the PingFederate administrative console, go to Authentication > Policies and open the authentication policy that you want to use. In the Policy section, click Rules underneath the html - (Adapter).

Copyright © 2025 Ping Identity Corporation



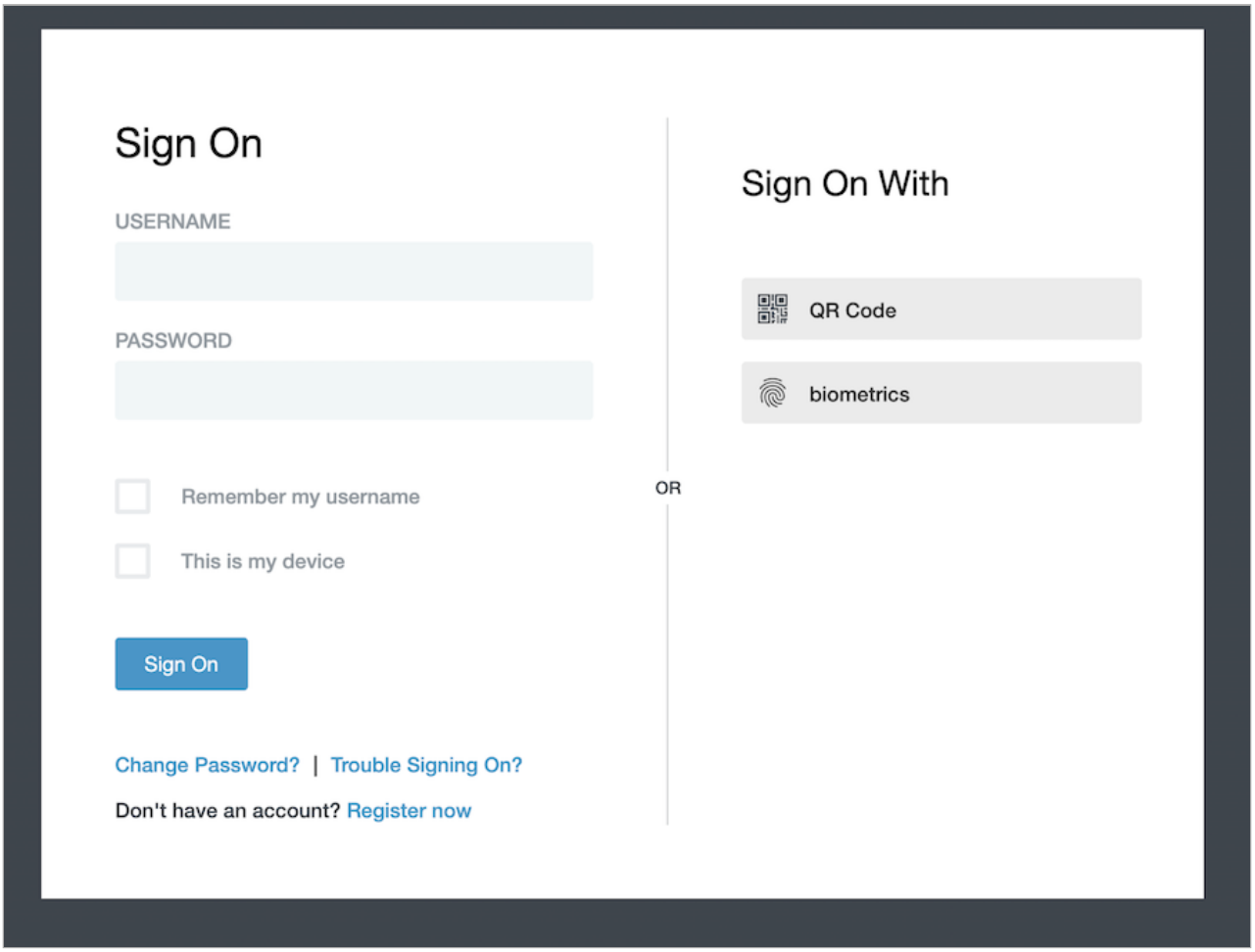
In the Rules window, configure the policy.action attribute for the keyword that you selected in step 1. In this example, `biometrics`. Set the Condition as `equal to` and input your selected keyword as both the Value and Result.

> **Note**
>
> The adapter can be made to skip FIDO usernameless authentication flow and follow regular authentication flow by passing the `ignore-fido-policy-action` attribute through chained attributes with the value **True**. This forces the adapter to skip FIDO policy action and follow its regular authentication flow path.



The keywords that you configured now display as options in the Sign On window.

Copyright © 2025 Ping Identity Corporation



Authentication API flow

When authN API flow is invoked for the `useAlternativeAuthenticationSource` action for biometrics, the adapter responds with the `BIOMETRIC_DEVICE_AUTHENTICATION_INFO_REQUIRED` state.

The application must provide its host name through `origin` to continue with biometric device authentication. This invokes biometric authentication flow transitioning to the `ASSERTION_REQUIRED` state requiring data from client to complete biometric authentication.

Copyright © 2025 Ping Identity Corporation

```
{
  "id": "NqMl7",
  "pluginTypeId": "j_AGR1E__nc2USbRhZPMQQ",
  "status": "ASSERTION_REQUIRED",
  "devices": [],
  "publicKeyCredentialRequestOptions": {
    "challenge": [
      69,
      43,
      -101,
      -53,
      -9,
      63,
      115,
      -91,
      -128,
      -112,
      53,
      -68,
      -95,
      -69,
      -10,
      -41,
      -33,
      -24,
      -91,
      -21,
      -70,
      2,
      -17,
      54,
      86,
      92,
      14,
      110,
      59,
      -122,
      -45,
      -61
    ],
    "timeout": 120000,
    "rpId": "localhost.ping-eng.com",
    "userVerification": "required",
    "allowCredentials": []
  },
  "manualPairingPermitted": false,
  "manualPairing": false,
  "userSelectedDefault": false,
  "changeDevicePermitted": false,
  "_links": {
    "cancelAuthentication": {
      "href": "https://localhost.ping-eng.com:9031/pf-ws/authn/flows/NqMl7"
    },
    "self": {
      "href": "https://localhost.ping-eng.com:9031/pf-ws/authn/flows/NqMl7"
    },
```


Copyright © 2025 Ping Identity Corporation

```
      "checkAssertion": {
        "href": "https://localhost.ping-eng.com:9031/pf-ws/authn/flows/NqMl7"
      }
    }
  }
```

Core contract attributes

At the end of successful authentication flow, the following core attributes are filled based on information found in a successful PingOne check assertion API response:

`username`

(existing core attribute) - username

`usernameless.flow.platform`

(new core attribute) - platform information

`usernameless.flow.userid`

(new core attribute) - user ID

For example, for response below, the `username`, `usernameless.flow.platform` and `usernameless.flow.userid` attribute values are `demouser`, `MAC`, and `78fe678a-5ec0-49b3-bcd7-742523cb216a`, respectively.

Copyright © 2025 Ping Identity Corporation

```
{
  "_links" : {
    "self" : {
      "href" : "https://auth.pingone.com/23c17a79-4972-4a46-871e-9b1121823917/deviceAuthentications/
0326f203-950e-418c-9cdb-829c5a4a2586"
    }
  },
  "_embedded" : {
    "devices" : [ {
      "id" : "00528f83-ae17-459a-838e-3f1226705a3a",
      "type" : "PLATFORM",
      "status" : "ACTIVE",
      "rp" : {
        "id" : "localhost.ping-eng.com",
        "name" : "PingFederate"
      },
      "fidoRegistrationArtifacts" : {
        "attestationType" : "SELF"
      },
      "platform" : "MAC",
      "_embedded" : {
        "webAuthnCredentials" : {
          "credentialId" : "3mP-Uv1u_9i9cM025m9SKIqfMIz-TcDgRBJR_BP8xA2BxeN6Hmgzs-
s8DKYZIXbDgioqrfoLCftUp0kzeFrvHT6hAGHoe2OFoLsrx_vuLsmmREyHqoZGWVsUxFEjNqo1eEhSpsTx5YnO54GQ4H6hMRAuutyWNnlj",
          "userHandle" : "plAY2tGM7kS7coi2+0DPU6xk1A63DDr2TUuYBGVfCRE=",
          "credentialPublicKey" :
"pQECAyYgASFYIBI_xOdmnL_OEcewdD2abF30WCmt8KJlix_8WNPIuysPIlggNxYEbOiR8NlgIEkKvrbSaJKsokXmKxS_Nl3JwT_ctfE=",
          "signCount" : 0
        },
        "user" : {
          "id" : "78fe678a-5ec0-49b3-bcd7-742523cb216a",
          "environment" : {
            "id" : "23c17a79-4972-4a46-871e-9b1121823917"
          },
          "username" : "demouser",
          "email" : "demouser@pingidentity.com",
          "population" : {
            "id" : "50756a1a-1b5d-49e2-b316-b3511fdc7a9f"
          },
          "name" : {
            "given" : "demouser"
          }
        }
      }
    } ]
  },
  "id" : "0326f203-950e-418c-9cdb-829c5a4a2586",
  "environment" : {
    "id" : "23c17a79-4972-4a46-871e-9b1121823917"
  },
  "status" : "COMPLETED",
  "policy" : {
    "id" : "bc8d02b5-ba91-0fe1-195c-4cc8ea5a77a3"
  },
  "selectedDevice" : {
    "id" : "00528f83-ae17-459a-838e-3f1226705a3a"
  },
  "user" : {
    "id" : "78fe678a-5ec0-49b3-bcd7-742523cb216a"
  },
```


Copyright © 2025 Ping Identity Corporation

```
    "rp" : {
      "id" : "localhost.ping-eng.com"
    },
    "publicKeyCredentialRequestOptions" : "{\"challenge\":
[119,-85,-127,71,6,46,47,8,-58,-9,95,-94,77,-82,103,-98,87,11,-90,1,-69,-59,-33,78,-39,63,-55,-10,-101,77,-84,-126],
\"timeout\":120000,\"rpId\":\"localhost.ping-eng.com\",\"allowCredentials\":[],\"userVerification\":\"required\"}",
    "authenticators" : [ "mfa", "user" ],
    "createdAt" : "2022-09-10T00:12:57.489Z",
    "updatedAt" : "2022-09-10T00:13:02.418Z"
  }
```


MFA setup

When a user signs on with PingFederate, the PingOne MFA IdP Adapter allows PingFederate to initiate multi-factor authentication (MFA) challenges using PingOne MFA.

About this task

> **Important**
>
> The MFA setup section is optional and independent from the CIBA setup section.

Steps

1. Complete the steps in Creating a CIBA authentication policy in PingOne.
2. Complete the steps in Configuring a CIBA authenticator instance.
3. Complete the steps in Adding PingOne MFA to your authentication policy.

Creating an MFA authentication policy in PingOne

Create an MFA-only authentication policy in PingOne MFA to handle user authentication requests.

About this task

The PingOne MFA IdP Adapter only supports multifactor authentication (MFA) flows in which Multi-factor Authentication is the only step configured in the PingOne authentication policy.

> **Important**
>
> To use a default policy in PingOne for MFA, you must make sure that **Multi-factor Authentication** is the only step in the policy.

Steps

1. In the PingOne MFA console, go to Authentication > Authentication and click + Add Policy.
2. Enter a policy name of your choosing and note it.

> **Note**
>
> You will use this policy name in Configuring an adapter instance.

Copyright © 2025 Ping Identity Corporation

3. In the Step Type list, select Multi-factor Authentication.
4. In the MFA Policy list, select an MFA policy to specify which authentication methods a user can use to authenticate themselves.
5. In the None Or Incompatible Methods section, select a default behavior for cases where the user does not have a valid authentication method set up:

   Choose from:

   - Block: If the user doesn't have a valid authentication method set up, MFA fails.
   - Bypass: If the user doesn't have a valid authentication method set up, they continue to the next step as if they completed MFA successfully.
6. (Optional) In the Required When section, configure authentication triggers.
7. Click Save.

Next steps

1. Add the policy to the web or native OIDC application that you created in Creating a web or native OIDC application in PingOne:
   1. In the PingOne MFA console, go to Applications > Applications and expand your application.
   2. On the Policies tab, click + Add Policies or click the Pencil icon, then select the checkbox for the policy that you created.
   3. Click Save.

Configuring an adapter instance

Configure the PingOne MFA IdP Adapter to determine how PingFederate communicates with PingOne MFA.

Steps

1. In the PingFederate administrative console, go to Authentication > Integration > IdP Adapters and click Create New Instance.
2. On the Type tab, set the basic adapter instance attributes.
   1. In the Instance Name field, enter a name for the adapter instance.
   2. In the Instance ID field, enter a unique identifier for the adapter instance.
   3. In the Type list, select PingOne MFA IdP Adapter. Click Next.
3. On the IdP Adapter tab, configure the adapter instance by referring to PingOne MFA IdP Adapter settings reference. Click Next.
4. On the Actions tab, test your connection to PingOne MFA. Resolve any issues that are reported, and then click Next.
5. On the Extended Contract tab, add any attributes that you want to include in the contract. Click Next.

Copyright © 2025 Ping Identity Corporation

6. On the Adapter Attributes tab, set pseudonym and masking options as shown in Set pseudonym and masking options in the PingFederate documentation. Click Next.
7. On the Adapter Contract Mapping tab, configure the contract fulfillment details for the adapter as shown in Define the IdP adapter contract in the PingFederate documentation. Click Next.
8. On the Summary tab, check your configuration, then click Save.

PingOne MFA IdP Adapter settings reference

Field descriptions for the PingOne MFA IdP Adapter configuration page.

Field	Description
PingOne Environment	Select the PingOne connection that you created in Connecting PingFederate to PingOne. This field is blank by default.
PingOne Population	If a user does not already exist in PingOne, the adapter provisions the user to this PingOne population. Applies only when Provision Users is selected. This list is populated after you select the PingOne Environment. This field is blank by default.
Application	The PingOne application that you created in Creating a web or native OIDC application in PingOne. This list is populated after you select the PingOne Environment and PingOne Population. This field is blank by default.

Copyright © 2025 Ping Identity Corporation

Field	Description
PingOne Authentication Policy	**The policy name that you chose in Creating an MFA authentication policy in PingOne. This overrides any policy named in the requested authentication context.** **You can enter multiple policy names by separating them with a space. For example, `employees contractors`. These must all be "MFA-only" policies, as described in Creating an MFA authentication policy in PingOne.** **The adapter maps this value to the `acr_values` parameter of the PingOne OIDC request.** **When this field is blank, the adapter does the following:** • **Looks for the presence of the `pingone-mfa-acr` attribute in chained attributes followed by signed request claims object followed by tracked parameters, and if found uses the corresponding as received.** • **If the `pingone-mfa-acr` attribute is not provided in any of the above mentioned dynamic parameters, the adapter will attempt to use the value of the `RequestedAuthnCtx parameter` (for SAML SSO flows) or the `acr_values` parameter (for OAuth flows) in the request.** **This field is blank by default.**
MFA policy for registration	**Select the PingOne MFA policy that you want to use for device pairing.** **Important** Your registration policy can use an MFA policy with different requirements than the MFA policy or policies that the **PingOne Authentication Policy** uses, so make sure to set up MFA policies that are compatible with each other. For example, if you use an MFA policy for authentication that requires offline devices and an MFA policy for registration that requires FIDO-compliant mobile devices, you wouldn't be able to pair any of the device types required for authentication through your registration policy. If this were the case, you could still set up the device types that you require for authentication through device provisioning. However, you would not be able to use any of your paired devices for authentication.

Copyright © 2025 Ping Identity Corporation

Field	Description
Test Username	The PingOne username that the adapter uses to test the PingOne MFA connection on the Actions tab. Enter the username for a user that has a paired device and MFA enabled in PingOne. This field is blank by default.
Notification Template Variant Override	Overrides the notification template variant that the adapter sends to PingOne for authentication and transaction approval flows. The adapter ignores any `pi.template.variant` attribute in the PingFederate authentication policy and uses this value instead. Enter the name of the template variant. For example, if you have a PingOne "transaction" template variant called "money-transfer", enter `money-transfer`. This field is blank by default.
HTML Template Prefix	Identifies the set of HTML templates that the adapter uses to show the authentication status or request a one-time password. If you customize the template file names in the `/server/default/conf/template` directory, enter the new prefix here. You can find a description of the template files in Download manifest. The default value is `pingone-mfa`.
Messages Files	Identifies the customizable language-pack file that the adapter uses to show messages on the templates. If you customize the `pingone-mfa-messages.properties` file name in the `/server/default/conf/language-packs` directory, enter the new name here. The default value is `pingone-mfa-messages`.
Prompt Users to Set Up MFA	Determines whether users with no authentication methods are prompted to add one. Select this if you are Enabling the MFA setup prompt. This checkbox is cleared by default.
Allow Users to Skip MFA Setup	Determines whether the MFA setup prompt includes a Skip option. Consider selecting this if you are Enabling the MFA setup prompt. This allows the user to sign on without setting up MFA. This checkbox is cleared by default.

Copyright © 2025 Ping Identity Corporation

Field	Description
Allow Users to Manage Additional Authentication Methods	Determines whether users can add another authentication method or remove an existing one during sign-on. The user must sign on with their existing authentication method first. This checkbox is cleared by default.
Provision Users	If a user does not already exist in PingOne, the adapter provisions the user to PingOne.
Provision Authentication Methods	Determines whether the adapter adds authentication methods based on the user's SMS Attribute, Voice Attribute, WhatsApp Attribute, and Email Attribute values.
Update Authentication Methods	This setting allows the adapter to automatically add new authentication methods for existing users. Consider selecting this if you are Enabling user and authentication method provisioning. During sign on, the adapter compares the user's authentication methods in PingOne to the user's SMS Attribute, Voice Attribute, WhatsApp Attribute, and Email Attribute values in the PingFederate authentication policy. If any new values are available for the user, the adapter adds them as authentication methods in PingOne. **Note** The PingOne MFA IdP Adapter only adds authentication methods to PingOne. If you want to synchronize authentication methods and other user attributes, use the PingOne Connector provided in the PingOne Integration Kit. Learn more in Known issues and limitations. This checkbox is selected by default.

Copyright © 2025 Ping Identity Corporation

Field	Description
Overwrite Authentication Methods Configurations	If the adapter identifies new values for SMS, voice, WhatsApp, or email devices, this setting determines whether the adapter replaces the existing methods with the new ones or just adds the new authentication methods. The following options are available: ***None (default)*** When selected, the user's existing SMS, voice, WhatsApp, and email authentication methods are not overwritten. New devices are added. ***All (SMS, Voice, WhatsApp, and Email)*** When selected, all the user's existing SMS, voice, WhatsApp, and email authentication methods that do not exist in the input are overwritten. ***Specific Methods*** When selected, the only devices that are overwritten are those of the same device type as a newly provided device. All other devices remain the same. Applies only when Update Authentication Methods is selected.
Allow only predefined values for phone or email devices	This option allows you to limit the values to the email addresses and phone numbers stored for the user. If you enable the option, the relevant email address or phone number is already filled in when the user tries to add a device, and the user cannot modify the address or phone number.
Enable Cookie Based Tracking	When selected, the adapter tracks a previously authenticated FIDO security key or platform device in a cookie so that a user is not prompted again.
Use Password Config Attribute	Adds a Use Password button to the device selection screen. The Use Password policy action automatically exits the user from the current flow when the user has no device to perform MFA and is not authenticated prior to reaching the adapter.

Copyright © 2025 Ping Identity Corporation

Field	Description
Bypass MFA For Device Management Attribute	**Enter the name of an attribute in this field. The adapter checks for this attribute in the authentication policy. If the value is `true`, MFA is not required to perform device management operations.** **Note** The `set default` operation never requires MFA. **Warning** Do not set the value of this attribute to `true` if you only have one adapter in the authentication policy. This results in bypassing MFA in the authentication flow entirely, and can lead to a security breach. We recommend only using this attribute if you have two consecutive adapters in an authentication policy. In this configuration, the first adapter should either not contain this attribute at all, or the value of the attribute should be set to `false`. Only the second adapter should contain this attribute with a value of `true`.
Username Attribute	**Determines the username for users provisioned to PingOne.** **This is used when Enabling user and authentication method provisioning.** **If you identify users based on their PingOne username, leave this field blank. New users are named based on the "incoming user ID" set for the adapter in your PingFederate authentication policy.** **If you identify users based on their PingOne user ID, enter the name of an authentication policy attribute. New users are named based on the attribute instead of the "incoming user ID".** **Tip** This relates to step 5 in Adding PingOne MFA to your authentication policy. **Applies only when Provision Users is selected.** **This field is blank by default.**

Copyright © 2025 Ping Identity Corporation

Field	Description
SMS Attribute	This is used when Enabling user and authentication method provisioning or when Update Authentication Methods is selected. Enter an attribute name in this field. The default value is `sms`. When provisioning users or updating a user's authentication methods, the adapter checks for phone numbers in each attribute that begin with this prefix. The adapter adds these phone numbers as SMS authentication methods in PingOne MFA, up to the maximum number of methods. For example, in the PingFederate authentication policy, a user has three phone numbers in the following attributes: • `sms-1` • `sms-2` • `sms-3` By entering `sms` in this field, the adapter adds all three phone numbers as SMS authentication methods in PingOne MFA. If your attribute value source maps multiple values to the same attribute, multiple authentication methods will be created.
Voice Attribute	This is used when Enabling user and authentication method provisioning. When provisioning users or updating a user's authentication methods, the adapter checks for phone numbers in each attribute that begin with this prefix. The adapter adds these phone numbers as voice authentication methods in PingOne MFA. Follows the same general behavior as the SMS Attribute field. Applies only when Update Authentication Methods is selected. The default value is `voice`.

Copyright © 2025 Ping Identity Corporation

Field	Description
Email Attribute	This is used when Enabling user and authentication method provisioning. When provisioning users or updating a user's authentication methods, the adapter checks for email addresses in each attribute that begin with this prefix. The adapter adds these email addresses as authentication methods in PingOne MFA. Follows the same general behavior as the SMS Attribute field. Applies only when Update Authentication Methods is selected. The default value is `email`.
WhatsApp Attribute	This is used when Enabling user and authentication method provisioning. When provisioning users or updating a user's authentication methods, the adapter checks for phone numbers in each attribute that begin with this prefix. The adapter adds these phone numbers as WhatsApp authentication methods in PingOne MFA. Follows the same general behavior as the SMS Attribute field. Applies only when Update Authentication Methods is selected. The default value is `whatsapp`.
Application ID for Authentication Code Flow	The application ID to use for authentication code based flow.
Default Authentication Method for Provisioned Users	This is used when Enabling user and authentication method provisioning. When provisioning a new user to PingOne, the adapter sets the user's default authentication method based on this setting. For example, when set to SMS, the adapter checks for attributes according to the SMS Attribute field. The first matching attribute, such as `sms-1`, becomes the user's default authentication method. Applies only when Update Authentication Methods is selected. **Note** You can find a related entry in Known issues and limitations. The default selection is `SMS`.

Copyright © 2025 Ping Identity Corporation

Field	Description
User Not Found Failure Mode	When a user error occurs in PingOne, this setting determines whether the adapter blocks the user's sign-on attempt. User errors include the following: • User is disabled • User does not exist • The PingOne MFA service is disabled for the user The default selection is `Block user`.
Service Unavailable Failure Mode	When PingOne doesn't respond, this setting determines how the adapter handles the user's sign-on attempt: ***Bypass Authentication*** The PingOne MFA IdP Adapter bypasses the user's sign-on attempt. ***Block User*** The PingOne MFA IdP Adapter blocks the user's sign-on attempt. **Note** If you select **Block User**, ensure that the **Show Error Screens** checkbox is selected. Learn more about **Show Error Screens** in one of the following table entries. The default selection is `Bypass Authentication`.
Change Authentication Method	Determines whether the adapter shows a "back" button that allows a user to select a different authentication method during a sign-on session. If your PingOne authentication policy uses the Being a member of any of these populations or User Attributes requirements, set this to Deny. The features are not compatible. This setting has no effect when the adapter is used through the PingFederate authentication API. The default selection is `Allow`.
Show Success Screens	Determines whether the adapter shows a success page when the MFA step is successful. This checkbox is selected by default.
Show Error Screens	Determines whether the adapter shows an error page when the MFA step generates an error. This checkbox is selected by default.

Copyright © 2025 Ping Identity Corporation

Field	Description
Show Timeout Screens	Determines whether the adapter shows a "timed out" page when the MFA step times out. This checkbox is selected by default.
Enable Audit Log	When selected, the adapter logs end-user browser details along with selected authentication method info such as device type, correlation ID, and device nickname in PingFederate's audit log.
API Request Timeout	The amount of time in milliseconds that PingFederate allows when establishing a connection with PingOne MFA or waiting for a response to a request. A value of `0` disables the timeout. **Tip** If you don't want to use the device integrity check, you can decrease this to 5000ms. You can find more information about the device integrity check in Authentication method management. The default value is `12000`.
Proxy Settings	Defines proxy settings for outbound HTTP requests. The default value is System Defaults.
Custom Proxy Host	The proxy server host name to use when Proxy Settings is set to Custom. This field is blank by default.
Custom Proxy Port	The proxy server port to use when Proxy Settings is set to Custom. This field is blank by default.

Adding PingOne MFA to your authentication policy

By modifying your PingFederate authentication policy to include the PingOne MFA IdP Adapter, you can challenge users to complete a multi-factor authentication (MFA) step.

About this task

These steps are designed to help you add to an existing authentication policy. You can find more information about configuring authentication policies in Authentication policies in the PingFederate documentation.

Steps

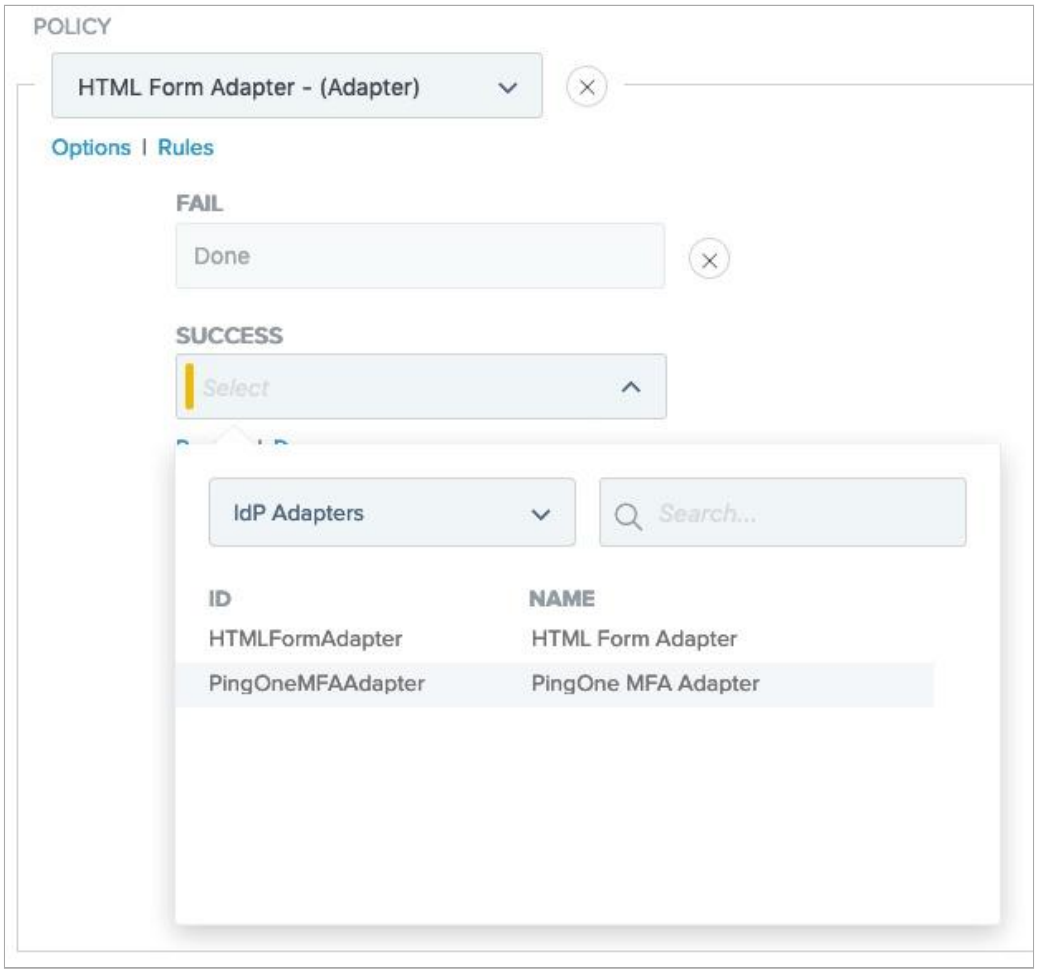
1. In the PingFederate administrative console, go to Authentication > Policies > Policies.
2. Select the IdP Authentication Policies checkbox.

Copyright © 2025 Ping Identity Corporation

3. Open an existing authentication policy, or click Add Policy.

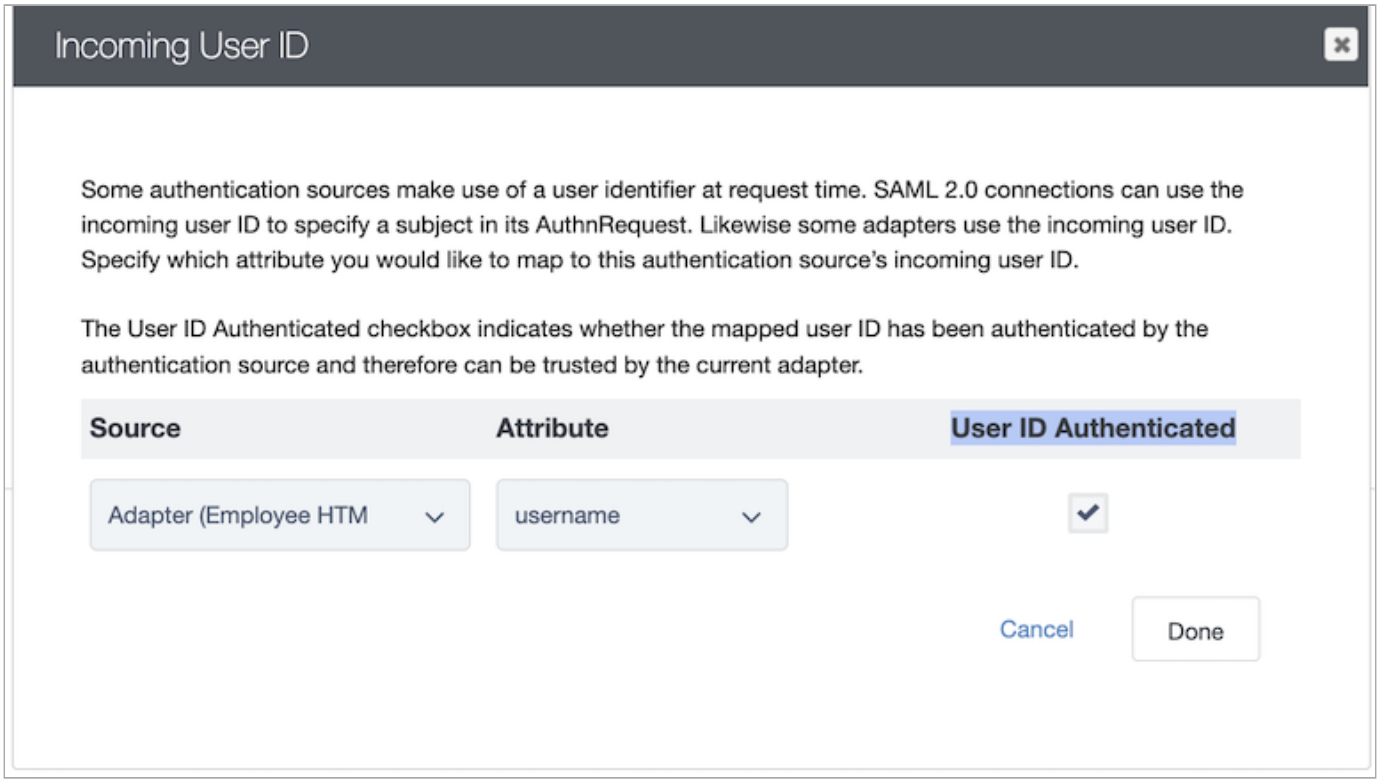
   Learn more in Defining authentication policies in the PingFederate documentation.

4. In the Policy area, in the Select list, select a PingOne MFA IdP Adapter instance.



5. Map the PingOne user ID or username into the PingOne MFA IdP Adapter instance.

Copyright © 2025 Ping Identity Corporation



1. **Under the PingOne MFA IdP Adapter instance, click Options.**
2. **In the Options modal, in the Source list, select a previous authentication source that collects the PingOne user ID or username.**

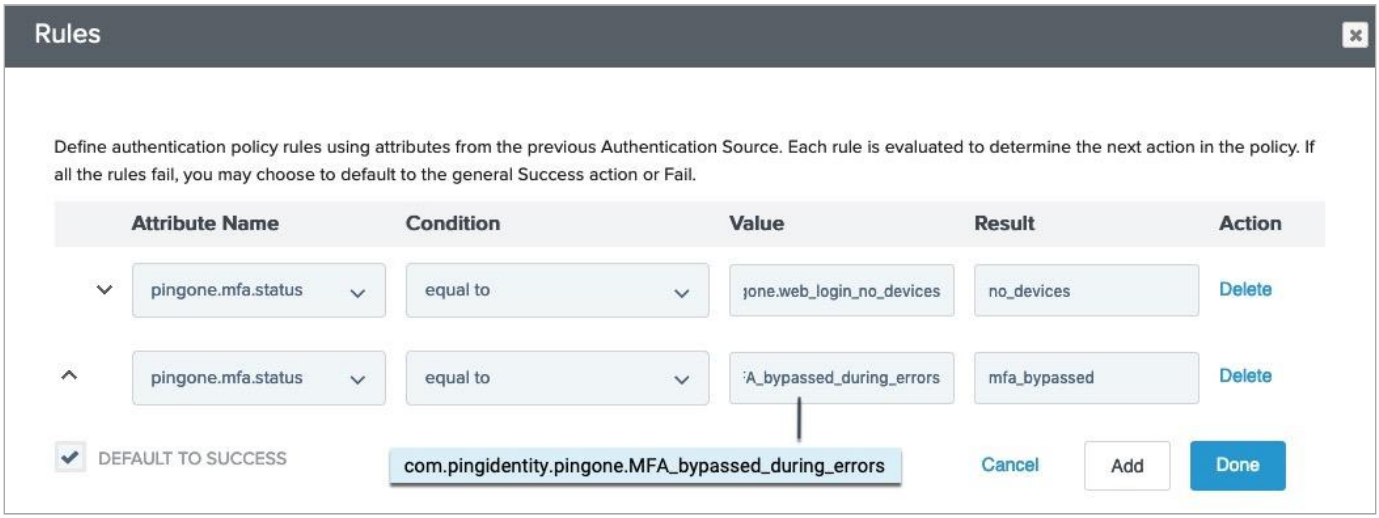
   > **Note**
   >
   > If you left the **Username Attribute** field blank in your PingOne MFA IdP Adapter configuration, the adapter also uses this value as the username when provisioning new users to PingOne.

3. **In the Attribute list, select the user ID. Click Done.**
4. **(Optional) Select the User ID Authenticated checkbox.**

   > **Note**
   >
   > The **User ID Authenticated** checkbox indicates whether the mapped user ID has been authenticated by the authentication source and therefore can be trusted by the current adapter. Device management options are limited if the user isn't authenticated.

6. **(Optional) Define policy paths based on the `pingone.mfa.status` or `pingone.mfa.status.reason` attributes.**

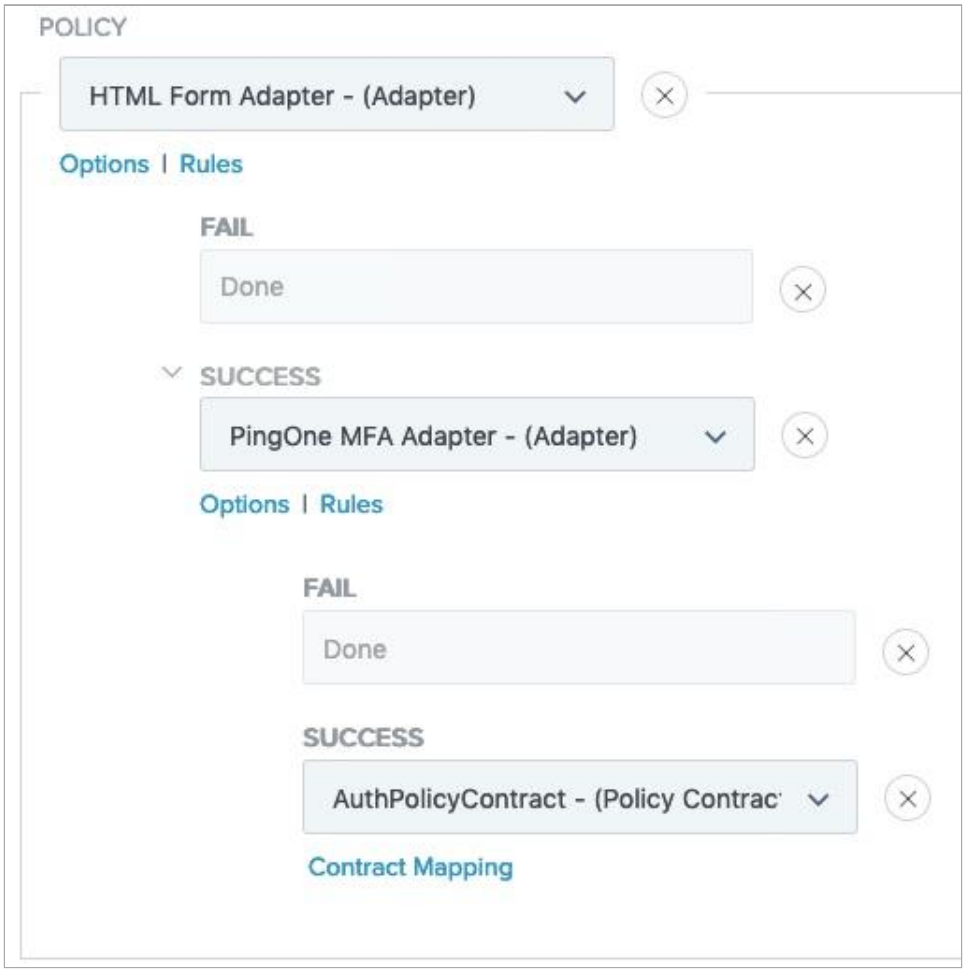
Copyright © 2025 Ping Identity Corporation



1. Under the PingOne MFA IdP Adapter instance, click Rules.
2. In the Rules modal, in the Attribute Name list, select pingone.mfa.status or pingone.mfa.status.reason.
3. In the Condition list, select equal to.
4. In the Value field, enter a value from PingOne MFA status attributes reference.
5. In the Result field, enter a name. This appears as a new policy path that branches from the authentication source.
6. If you want to add more authentication paths, click Add and repeat steps a-e.
7. Click Done.

7. Configure each of the authentication paths.

Copyright © 2025 Ping Identity Corporation



8. Click Done.
9. Configure tracked HTTP parameters:
    1. On the Tracked HTTP Parameters tab, in the Parameter Name field, enter the name of the parameter that you want to track.

       For example:

        - To enable automatic device pairing, enter `mobilePayoad`.
        - To configure the PingOne MFA IdP Adapter to support the prompt parameter, enter `prompt`.
    2. Click Add.
    3. Click Save.
    4. Repeat the previous steps for any other parameters that you want to add.

Copyright © 2025 Ping Identity Corporation

PingOne MFA status attributes reference

PingOne MFA provides two attributes that can be useful in your PingFederate authentication policy.

`pingone.mfa.status` attribute

This attribute describes the status of the current multifactor authentication request.

`pingone.mfa.status` *values*

Value	Description
`com.pingidentity.pingone.web_login_no_devices`	Set when MFA is bypassed for any reason, such as when a user without any trusted devices signs on from the web and the system is configured to bypass authentication for users without a trusted device.
`com.pingidentity.pingone.web_login_sms`	This status is returned on successful SMS authentication from a web sign on flow.
`com.pingidentity.pingone.web_login_email`	This status is returned on successful email authentication from a web sign on flow.
`com.pingidentity.pingone.web_login_mobile`	This status is returned on successful SDK mobile app authentication from a web sign on flow.
`com.pingidentity.pingone.web_login_fido`	This status is returned on successful biometric/security key authentication from a web sign on flow.
`com.pingidentity.pingone.web_login_otp`	This status is returned on successful third-party "timed one-time passcode" (TOTP) authenticator authentication from a web sign on flow.
`com.pingidentity.pingone.mobile_login_sms`	This status is returned on successful SMS authentication when the user signs on from an untrusted mobile device.
`com.pingidentity.pingone.mobile_login_email`	This status is returned when a user signs on with an untrusted mobile device and is authenticated using their trusted email device.
`com.pingidentity.pingone.mobile_login_mobile`	This status is returned on successful mobile authentication when the user signs on from an untrusted mobile device.
`com.pingidentity.pingone.mobile_login_otp`	This status is returned when a user signs on with an untrusted mobile device and is authenticated using their trusted third-party OTP authenticator.

Copyright © 2025 Ping Identity Corporation

Value	Description
`com.pingidentity.pingone.mobile_login_fido`	This status is returned on successful biometric/security key authentication when the user signs on from an untrusted mobile device.
`com.pingidentity.pingone.MFA_bypassed_during_errors`	A user signs on when the system is configured to bypass authentication if there are network problems or PingOne is unreachable.
`com.pingidentity.pingone.device_paired`	Set when the end-to-end pairing process is completed. That is, the ID token from PingOne has been consumed by the PingOne mobile SDK and the user approved the pairing.
`com.pingidentity.pingone.device_not_paired`	Set when the device is paired through the front channel. The device is reported as not paired because the ID token is returned as part of the attribute contract and cannot be passed to the PingOne mobile SDK until after authentication. Also set as part of the API flow when the device was not paired. That is, the user was denied.
`com.pingidentity.pingone.pairing_error`	Set when there is an error during the pairing flow.
`com.pingidentity.pingone.mfa_bypassed_skip_mfa`	A user with no authentication methods was prompted to set up MFA, but they clicked Skip.

`pingone.mfa.status.reason` attribute

When the `pingone.mfa.status` attribute is `device_paired`, this attribute describes the paired device.

`pingone.mfa.status.reason` *values*

Value	Description
`mobile_login_sms`	The user is using SMS to authenticate.
`mobile_login_email`	The user is using email to authenticate.
`mobile_login_mobile`	The user is using the mobile app to authenticate.
`mobile_login_otp`	The user is using a third-party OTP authenticator to authenticate.
`mobile_login_fido`	The user is using a biometric/security key authenticator to authenticate.

Copyright © 2025 Ping Identity Corporation

> **Note**
>
> `pingid.sdk.status` and `pingid.sdk.status.reason` have been added to support PingID SDK adapter migration scenario. For more information, see PingID SDK adapter core contract attributes.

CIBA setup

When a client application sends a Client Initiated Backchannel Authentication (CIBA) request to PingFederate, the PingOne MFA CIBA Authenticator allows PingFederate to initiate an out-of-band authentication request using PingOne MFA.

About this task

By completing the steps below, you can show CIBA prompts on the mobile device that a user has paired with their PingOne MFA account.

For general information about CIBA, see Client Initiated Backchannel Authentication (CIBA) in the PingFederate documentation. Note that the CIBA Authenticator mentioned on that page is different from the PingOne MFA CIBA Authenticator.

For detailed information about the following steps, see CIBA request flow and CIBA prompt customizations.

> **Important**
>
> The CIBA setup section is optional and independent from the MFA setup section.

Steps

1. In PingOne, complete the steps in Creating a CIBA authentication policy in PingOne.
2. In PingFederate, complete the steps in Configuring a CIBA authenticator instance.
3. In PingFederate, create a CIBA request policy as shown in Managing CIBA request policies in the PingFederate documentation.

CIBA request flow

When a client makes a Client Initiated Backchannel Authentication (CIBA) request to PingFederate, PingFederate can modify and add to the request information before sending it to PingOne MFA.

The following steps are involved in conveying the CIBA request information from the client application to the request prompt that the user sees on their mobile device.

1. The client application sends a CIBA request to PingFederate that includes some request context attributes.
2. The CIBA request policy collects attributes from the client request and other attribute sources that you configure, such as a data store.
3. The PingOne MFA CIBA Authenticator parses the client request attributes. The authenticator also processes any custom attribute definitions, as described in CIBA prompt customizations.
4. The authenticator sends the attributes to PingOne along with information to identify which PingOne notification template to use for the user prompt.

Copyright © 2025 Ping Identity Corporation

5. PingOne uses the attributes from the PingOne MFA CIBA Authenticator to populate the variables in a customizable notification template.
6. PingOne sends the notification to the mobile app, which is built with the PingOne Mobile SDK.
7. The mobile app show the request prompt to the user to approve or deny.

CIBA prompt customizations

Client-initiated backchannel authentication (CIBA) involves prompting a user to accept or reject a request. You can customize the request prompt using pre-defined attributes or map complex custom attributes.

Simple prompt customizations

For simple, centralized customizations, you can modify the notification template that PingOne uses to generate the request prompt. Notification template customizations allow you to use the following:

- Static text
- A pre-defined set of attributes from the client request context

Simple prompt customizations require changes to the PingOne notification template only.

- Location: PingOne console > Experiences > Notifications.

You can find more descriptions and examples in Simple CIBA prompt customizations.

Advanced prompt customizations

If you need more advanced customizations in the request prompt, you can define custom attributes in PingFederate and pass them to PingOne.

The PingOne MFA CIBA Authenticator allows you to use the following:

- Static text
- Modified attributes from the client request context
- Contract attributes from the PingFederate CIBA request policy
- Messages from the PingFederate language-pack localization files
- Apache Velocity Template Language code to manipulate any of the above

Copyright © 2025 Ping Identity Corporation

Advanced prompt customizations require coordinated changes to all the following:

- The PingOne notification template
  - Location: PingOne console > Experiences > Notifications.
- The PingOne Template Variables table of the PingOne MFA CIBA Authenticator instance
  - Location: PingFederate console > Authentication > OAuth > CIBA Authenticators > Your PingOne MFA CIBA Authenticator instance > Instance Configuration.
- The Extended Contract tab of the PingOne MFA CIBA Authenticator instance
  - Location: PingFederate console > Authentication > OAuth > CIBA Authenticators > Your PingOne MFA CIBA Authenticator instance > Instance Configuration.
- The PingFederate CIBA request policy
  - Location: PingFederate console > Applications > OAuth > CIBA Request Policies > Your CIBA request policy > Contract Fulfillment.

You can find more descriptions and examples in Advanced CIBA prompt customizations.

Simple CIBA prompt customizations

Simple prompt customizations only require changes to the PingOne notification template.

The PingOne notification template combines customizable text and variables to form the message that the user sees. Learn more in Notifications and Editing a notification in the PingOne documentation.

Request context attributes

The PingOne MFA CIBA Authenticator processes the request context attributes and makes the following variables available to use in your PingOne notification template:

- `${locale}` – The user's locale which represents the user's country and language.
- `${requestedScope}` – The descriptive scope name requested by the client, such as "Email address access". Multiple scopes are separated with a new line (`\n`).
- `${requestingApplicationId}` – ID of the requesting client.
- `${requestingApplicationName}` – Name of the requesting client.
- `${requestingApplicationLogoUrl}` – Logo image URL of the requesting client.
- `${authUserBindingMessage}` – A human-readable message or number intended to be displayed on both the consumption device and authenticating device to provide a visual cue of the interaction to the user.

Copyright © 2025 Ping Identity Corporation

> **Note**
>
> You can use these variables directly in the PingOne notification template. You do not need to add them to the **PingOne Template Variables** table or **Extended Contract** tab in the PingOne MFA CIBA Authenticator configuration.

Example

If you wanted to trigger an authentication challenge for customers working with a bank teller, you could customize the PingOne notification template with the following plain text and variables:

Subject

`${requestingApplicationName}` is requesting approval

Body

Message from `${requestingApplicationName}`: `${authUserBindingMessage}`

The resulting user prompt is:

```
TrustNow Banking is requesting your approval

Message from TrustNow Banking:
Please touch "Approve" to verify your identity to the teller. Transaction code: AFXG.
```

Advanced CIBA prompt customizations

Advanced prompt customizations allow for complex and programmatic attribute mappings. These customizations can require coordinated changes in several locations.

Overview

Advanced prompt customizations are made possible using the PingOne Template Variables table on the Instance Configuration tab of the PingOne MFA CIBA Authenticator configuration. The two columns behave as follows:

PingOne Template Variable Name

Allows you to define an attribute name.

When the authenticator passes this attribute to PingOne, it becomes available to use as a dynamic variable in the PingOne notification template. For example, the attribute `purchaseTotal` becomes accessible in PingOne as the dynamic variable `${purchaseTotal}`.

PingOne Template Variable Value

Allows you to combine or manipulate a variety of inputs to create a dynamic value for the attribute.

You can use Apache Velocity Template Language code in this field. For more information on working with Velocity Template Language, see Velocity Template Language in the Apache documentation.

Copyright © 2025 Ping Identity Corporation

This field is the core of the advanced prompt customizations. The sections below describe the various inputs that you can use in this field.

You will need to coordinate these changes in the following locations:

- PingOne notification templates
  - Use the variables that you define in the PingOne Template Variable Name field.
- PingOne CIBA MFA Authenticator Extended Contract tab
  - Add any CIBA request policy attributes that you use in the PingOne Template Variable Value field.
- PingFederate CIBA request policy
  - Modify the Contract Fulfillment tab to provide the request policy attributes that you use in the PingOne Template Variable Value field.

The sections below describe the types of inputs and techniques that you can use in the PingOne Template Variable Value field. The examples show the coordinating changes that you need to make.

Contract attributes from the CIBA request policy

If your CIBA request policy includes valuable contract attributes that you want to include in the CIBA prompt, you can include the attributes in the PingOne Template Variable Value field.

To make an attribute available in the PingOne Template Variable Value field, add it on the Extended Contract tab of the authenticator configuration.

If your CIBA request policy does not already include an attribute that you want to use, configure it on the Contract Fulfillment tab of your CIBA request policy.

For example, if you wanted to combine plain text and two request policy attributes to create a purchase summary, you would to the following:

1. In the PingOne notification template, use this variable:

```
${purchaseSummary}
```

2. In the PingOne MFA CIBA Authenticator, in the PingOne Template Variables table:
   1. In the PingOne Template Variable Name field, enter:

```
purchaseSummary
```

   2. In the PingOne Template Variable Name field, enter:

```
The purchase total for $subject is: $sum $currency.
```

3. In the PingOne MFA CIBA Authenticator, on the Extended Contract tab, add:
   - `sum`

Copyright © 2025 Ping Identity Corporation

- `currency`

> **Note**
>
> The `subject` attribute is in the core contract and does not need to be added.

4. In the PingFederate CIBA Request Policy, on the Contract Fulfillment tab, map values for the following contract attributes:
   - `sum`
   - `currency`
   - `subject`

In the resulting prompt, the value becomes:

```
The purchase total for jsmith is: $40 USD.
```

Language pack messages

Use the `$languagePackMessages` attribute to include a message from the language-pack file configured for this authenticator. The authenticator includes the message from the language-pack file that matches the user's locale.

For example, to include a localized warning message in the user prompt, you would do the following:

1. In the PingOne notification template, use this variable:

```
${localizedTransferWarning}
```

2. In the PingOne MFA CIBA Authenticator, in the PingOne Template Variables table:
   1. In the PingOne Template Variable Name field, enter:

```
localizedTransferWarning
```

   2. In the PingOne Template Variable Name field, enter:

```
$languagePackMessages.getMessage("transfer.warning").
```

In the resulting prompt, the value becomes:

```
Advertencia: transferencia de dinero iniciada
```

> **Note**
>
> For language pack messages, you do not need to add anything to the **Extended Contract** tab or modify your CIBA request policy.

Copyright © 2025 Ping Identity Corporation

> **Tip**
>
> Although you can localize messages in PingFederate, the PingOne notification templates has built-in localization that is based on the user's PingOne profile. Also, handling localization in PingOne might provide a more centralized administrative experience.

Modified request context attributes

You can use any attributes contained within the `$oobAuthRequestContext` sent by the client to PingFederate.

The attributes are flattened and available to use in the following format in the PingOne Template Variable Value field:

- **`$locale` – The user's locale which represents the user's country and language.**
- **`$requestedScope` – The descriptive scope name requested by the client, such as "Email address access". Multiple scopes are separated with a new line (`\n`).**
- **`$requestingApplicationId` – ID of the requesting client.**
- **`$requestingApplicationName` – Name of the requesting client.**
- **`$requestingApplicationLogoUrl` – Logo image URL of the requesting client.**
- **`$authUserBindingMessage` – A human-readable message or number intended to be displayed on both the consumption device and authenticating device to provide a visual cue of the interaction to the user.**

For detailed information about what is available in `$oobAuthRequestContext` , see Class OOBAuthRequestContext in the PingFederate SDK documentation (Javadoc).

For example, to separate scopes with commas instead of line breaks, you would do the following:

1. **In the PingOne notification template, use this variable:**

```
${commaDelimitedScopes}
```

2. **In the PingOne MFA CIBA Authenticator, in the PingOne Template Variables table:**
    1. **In the PingOne Template Variable Name field, enter:**

```
commaDelimitedScopes
```

    2. **In the PingOne Template Variable Name field, enter:**

```
#set( $delimiter = "")#foreach($scope in $oobAuthRequestContext.requestedScope.values())
$delimiter$scope#set( $delimiter = ", ")#end.
```

In the resulting prompt, the value becomes:

```
Email address access, Phone number access
```


Copyright © 2025 Ping Identity Corporation

> **Note**
>
> For request context attributes, you do not need to add anything to the **Extended Contract** tab or modify your CIBA request policy.

Multiple PingOne application support

You can use a single instance of the PingOne MFA CIBA Authenticator with more than one PingOne application.

The basic authenticator configuration includes a default PingOne application ID, but the client can dynamically specify a different PingOne application with each request. The CIBA authenticator automatically checks the CIBA Request Policy for the following attributes:

- `applicationId`
- `applicationSecret`

Configure the CIBA client to send these attributes as part of the request to PingFederate. When the CIBA authenticator receives a valid application ID and secret, it overrides the defaults saved in the configuration.

Creating a CIBA authentication policy in PingOne

Create a multi-factor authentication (MFA) authentication policy in PingOne MFA to handle client-initiated backchannel authentication (CIBA) authentication requests.

About this task

> **Tip**
>
> If you already completed the steps in Creating an MFA authentication policy in PingOne, you can skip the steps below and use the same policy for CIBA authentication requests if the MFA policy that you selected in step 4 contains a **Mobile Application** in the **Allowed Authentication Method** section. PingOne ignores other authentication methods for CIBA requests.

To create a PingOne authentication policy that you can use for CIBA authentication requests:

Steps

1. In the PingOne MFA console, go to Authentication > Authentication and click + Add Policy.
2. Enter a unique policy name and note it.

   > **Note**
   >
   > You will use this name in Configuring a CIBA authenticator instance.

3. In the Step Type list, select Multi-factor Authentication.
4. In the MFA Policy list, select an MFA policy that has a Mobile Application configured in the Allowed Authentication Methods section.

Copyright © 2025 Ping Identity Corporation

5. In the None Or Incompatible Methods section, select a default behavior for cases where the user does not have a valid authentication method set up:

   Choose from:

   - Block: If the user doesn't have a valid authentication method set up, MFA fails.
   - Bypass: If the user doesn't have a valid authentication method set up, they continue to the next step as if they completed MFA successfully.

6. (Optional) In the Required When section, configure authentication triggers.

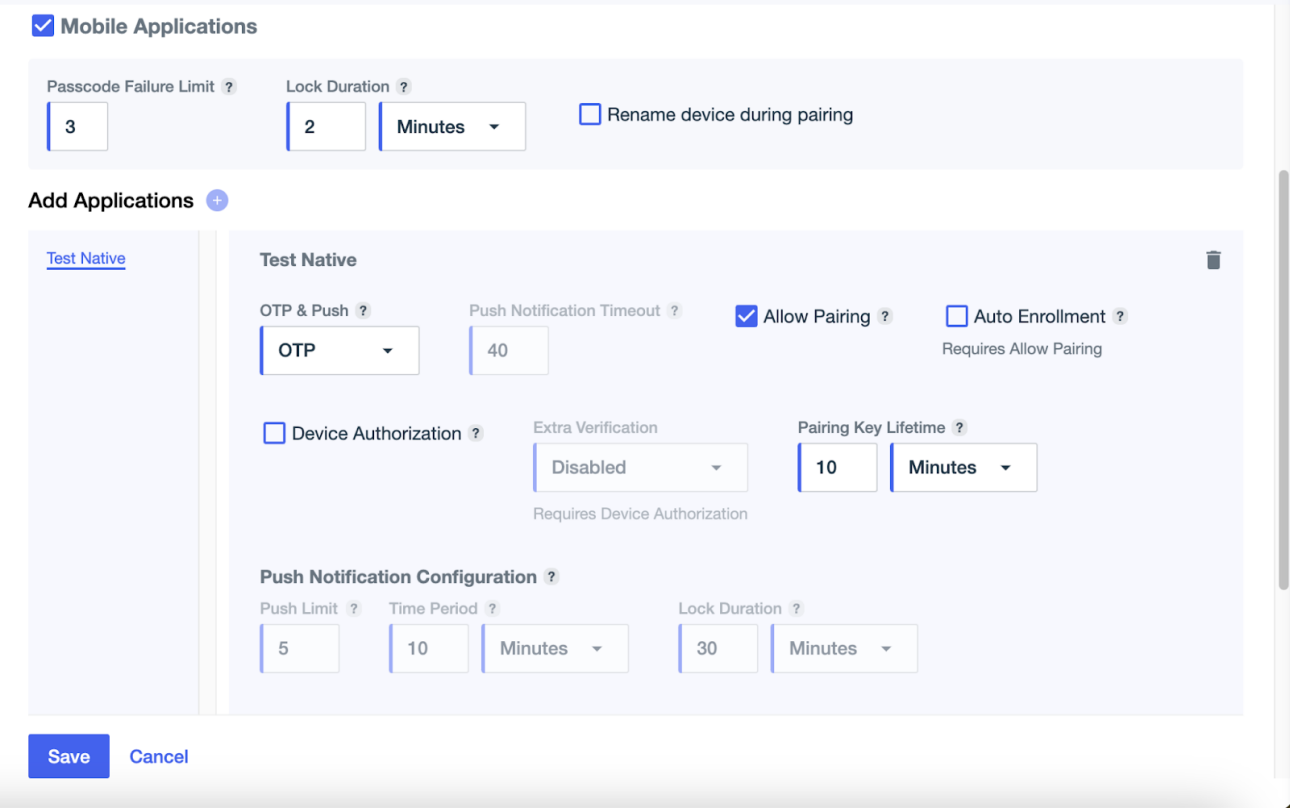
   The None Or Incompatible Methods and Required When sections have no effect on CIBA requests.

7. Click Save.

Next steps

1. Add the policy to your native OIDC application that you created in Creating a web or native OIDC application in PingOne:
   1. In the PingOne MFA console, go to Applications > Applications and expand your application.
   2. On the Policies tab, click + Add Policies or click the Pencil icon, then select the checkbox for the policy that you created.
   3. Click Save.
2. To configure settings for the mobile application attached to the MFA policy you selected in step 4:
   1. In the PingOne MFA console, go to Authentication > MFA, open the corresponding MFA policy, and click the Pencil icon:
   2. In the Mobile Applications section, click the + icon next to Add Applications, then select the native OIDC application that you created in Creating a web or native OIDC application in PingOne.

Copyright © 2025 Ping Identity Corporation



3. Leave the Auto Enrollment and Device Authorization checkboxes cleared.
4. Click Save.

Configuring a CIBA authenticator instance

Configure the PingOne MFA CIBA Authenticator to determine how PingFederate communicates with PingOne MFA for client-initiated backchannel authentication (CIBA) requests.

About this task

These steps are only required if you want to enable CIBA support. Learn more in CIBA request flow.

Steps

1. In the PingFederate administrative console, go to Authentication > OAuth > CIBA Authenticators and click Create New Instance.
2. On the Type tab, set the basic adapter instance attributes.
    1. In the Instance Name field, enter a name for the authenticator instance.
    2. In the Instance ID field, enter a unique identifier for the authenticator instance.
    3. In the Type list, select PingOne MFA CIBA Authenticator. Click Next.
3. (Optional) If you want to use the advanced prompt customizations described in CIBA prompt customizations, on the Instance Configuration tab, in the PingOne Template Variables section, map dynamic values to the variables in your PingOne notification template.

Copyright © 2025 Ping Identity Corporation

You can find more details and examples in Advanced CIBA prompt customizations.

1. Click Add a new row to 'PingOne Template Variables'.
2. In the PingOne Template Variable Name field, enter the name of a variable in your PingOne notification template.
3. In the PingOne Template Variable Value field, define the value based on the examples shown in Advanced CIBA prompt customizations.
4. In the Action column, click Update.
5. To add more attributes, repeat steps a - d.

4. On the Instance Configuration tab, configure the authenticator instance by referring to PingOne MFA CIBA Authenticator settings reference. Click Next.
5. On the Actions tab, test your connection to PingOne MFA. Resolve any issues that are reported, and then click Next.
6. On the Extended Contract tab, add any attributes that you used in the PingOne Template Variable Value fields on the Instance Configuration tab, then click Next.

   The Extended Contract tab acts as an input that defines which attributes are available to use in the PingOne Template Variable Value fields. The `subject` attribute is always included. Learn more in Advanced CIBA prompt customizations.
7. On the Summary tab, check your configuration, then click Save.

PingOne MFA CIBA Authenticator settings reference

Field descriptions for the PingOne MFA CIBA Authenticator configuration tab.

Field	Description
PingOne Environment	Select the PingOne connection that you created in Connecting PingFederate to PingOne. This field is blank by default.
Application	The PingOne application that you created in Creating a web or native OIDC application in PingOne. This list is populated after you select the PingOne Environment and PingOne Population. This field is blank by default.

Copyright © 2025 Ping Identity Corporation

Field	Description
PingOne Authentication Policy	The name of a PingOne authentication policy that is designed for client-initiated backchannel authentication (CIBA). This overrides any policy named in the requested authentication context. You can enter multiple policy names by separating them with a space. For example, `employees contractors`. These must all be "CIBA-only" policies. The adapter maps this value to the `acr_values` parameter of the PingOne OIDC request. When this field is blank, the adapter uses the value of the `RequestedAuthnCtx` parameter (for SAML SSO flows) or `acr_values` parameter (for OAuth flows) in the request. If those parameters are empty, PingOne uses the default authentication policy. This field is blank by default.

Field	Description
Test Username	The PingOne username that the adapter uses to test the PingOne MFA connection on the Actions tab. Enter the username for a user that has a paired device and multi-factor authentication (MFA) enabled in PingOne. This field is blank by default.
PingOne Template Name	Determines the PingOne notification template is used to show the CIBA request to the user. For example, if your CIBA notification template in PingOne is a "transaction" template variant called "ciba-purchases", enter `transaction`. The default value is `transaction`.
PingOne Template Variant	Determines which version of the PingOne notification template used to show the CIBA request to the user. For example, if your CIBA notification template in PingOne is a "transaction" template variant called "ciba-purchases", enter `ciba-purchases`. **Note** Variants for PingOne notification templates are only available through the PingOne API at this time. This field is blank by default.

Copyright © 2025 Ping Identity Corporation

Client Context	This optional field allows you to customize the information sent to PingOne with the CIBA request. While the CIBA prompt customizations allows you to customize the user-facing message, this field allows you to send additional information that can be used by the mobile application for your needs. In this field, you can use the following: ***$oobAuthRequestContext*** Context for the out-of-band authentication/ authorization request. This information is always provided to PingOne. ***$languagePackMessages*** The `language-pack` file configured for this authenticator. This allows you to dynamically customize the language of the message. ***$subject*** The user's PingOne username or user ID. ***$JSONValue*** A JSON utility class that can be used to escape text and convert objects to JSON. Methods include: • `escape(String s)`: Escape quotes, `\`, `/`, `\r`, `\n`, `\b`, `\f`, `\t` and other control characters (U+0000 through U+001F). • `toJSONString(Object value)`: Convert an object to JSON text. ***CIBA request policy contract attributes*** Attributes that you add on the Extended Contract tab are available here using the `$<name>` syntax. For example, `$currency`. The following example shows a variety of context parameters: `{ "requestingApplicationName": "$JSONValue.escape($oobAuthRequestContext.requestingApplication.name)", "requestedScope": ", $JSONValue.toJSONString($oobAuthRequestContext.requestedScope.values())", "amount": "$JSONValue.escape($amount)", "alert.color": "red" }` Lines that appear between `#*` and `*#` are comments. They are not evaluated or sent to PingOne. The default value is a comment explaining the above.

Copyright © 2025 Ping Identity Corporation

Field	Description
Messages Files	Identifies the customizable `language-pack` file that the authenticator uses. If you customize the `language-pack` file names in the `/server/default/conf/template/language-packs` directory, enter the new name here. The default selection is `pingone-mfa-messages`.
API Request Timeout	The amount of time in milliseconds that PingFederate allows when establishing a connection with PingOne MFA or waiting for a response to a request. A value of `0` disables the timeout. The default value is `2000`.
Proxy Settings	Defines proxy settings for outbound HTTP requests. The default value is System Defaults.
Custom Proxy Host	The proxy server host name to use when Proxy Settings is set to Custom. This field is blank by default.
Custom Proxy Port	The proxy server port to use when Proxy Settings is set to Custom. This field is blank by default.

Authentication API Support

You can use the PingFederate authentication API to integrate the PingOne MFA IdP Adapter into your application.

The PingFederate Authentication API provides access to the current state of the authentication flow as a user steps through the PingFederate authentication policy. Learn more in PingFederate authentication API in the PingFederate documentation.

To integrate the PingOne MFA IdP Adapter into your authentication flow, configure your application based on the information in this section.

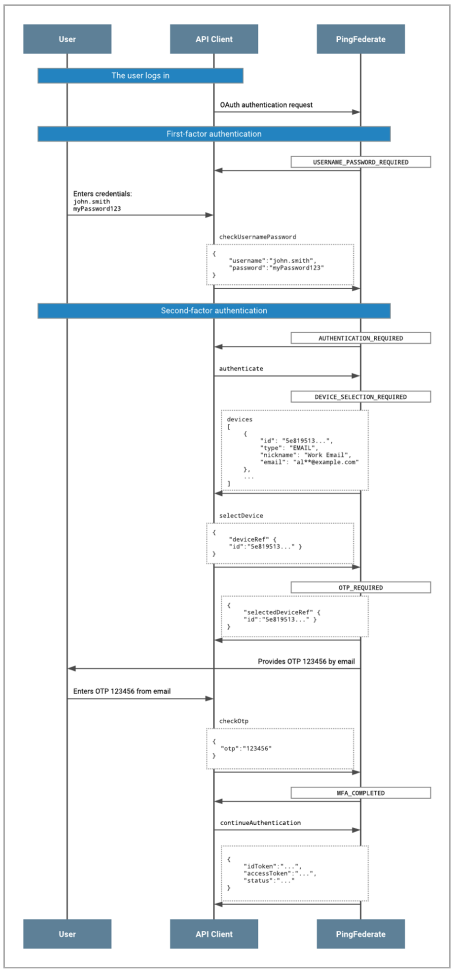
> **Tip**
>
> You can also explore the process using the PingFederate Authentication API Explorer. Learn more in Exploring the Authentication API in the PingFederate documentation.

Authentication flows

When using the PingOne MFA IdP Adapter through the PingFederate authentication application programming interface (API), the following flows are used for multi-factor authentication (MFA) requests. These are initiated in the web browser.

Copyright © 2025 Ping Identity Corporation

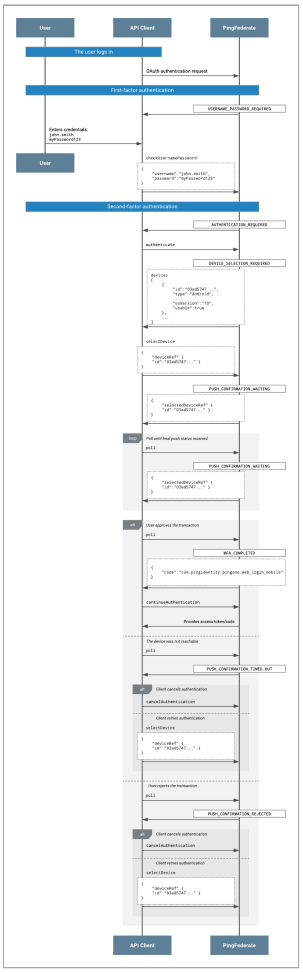
MFA via email



1. The user completes first-factor authentication. Completion of first-factor authentication is a prerequisite before progressing to MFA, when using the PingOne MFA IdP Adapter with the PingFederate Authentication API flow.
2. The status of `AUTHENTICATION_REQUIRED` is returned in the response to the API client.
3. The API client invokes the `authenticate` action.
4. The status of `DEVICE_SELECTION_REQUIRED` is returned with the devices object in the response to the API client.
5. The API client invokes `selectDevice` action and specifies the device ID of the device to use for MFA.
6. The status of `OTP_REQUIRED`, together with the devices and `selectedDeviceRef` object, are returned in the response to the API client. In parallel, the user receives an email containing the one-time passcode (OTP) for authentication.
7. After the user has entered the OTP, the API client invokes the `checkOtp` action, submitting the OTP value to PingFederate.
8. On successful completion of MFA, PingFederate returns the status of `MFA_COMPLETED` to the API client.
9. The API client invokes the `continueAuthentication` action. The API client must call `continueAuthentication` in order to progress in the OIDC flow, and to complete it.
10. PingFederate returns a single sign-on (SSO) ID token and access token to the API client.

Copyright © 2025 Ping Identity Corporation

MFA via the mobile app



1. The user completes first-factor authentication. Completion of first-factor authentication is a prerequisite before progressing to MFA, when using the PingOne MFA IdP Adapter with the PingFederate Authentication API flow.
2. The status of `AUTHENTICATION_REQUIRED` is returned in the response to the API client.
3. The API client invokes the `authenticate` action.
4. The status of `DEVICE_SELECTION_REQUIRED` is returned with the devices object in the response to the API client.
5. The API client invokes `selectDevice` action and specifies the device ID of device to use for multi-factor authentication.
6. The status of `PUSH_CONFIRMATION_WAITING`, together with the devices and `selectedDeviceRef` object, are returned in the response to the API client.
7. The API client invokes the `poll` action, so that PingFederate gets the status of the mobile push. This is repeated until either a successful status is received or a timeout is reached.
8. One of the following alternative statuses is reached:
    - `MFA_COMPLETED`:
        - The user receives a push notification and approves the authentication.

Copyright © 2025 Ping Identity Corporation

- The API client invokes the `continueAuthentication` action. The API client must call `continueAuthentication` in order to progress in the OIDC flow, and to complete it.
- PingFederate returns an access token for SSO, to the API client.

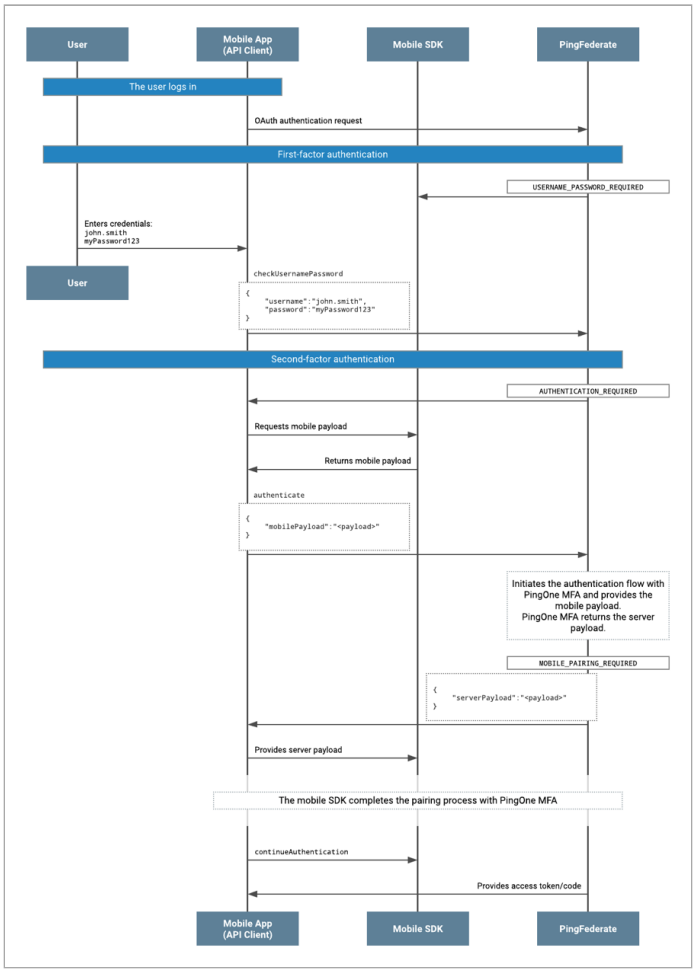
- `PUSH_CONFIRMATION_TIMED_OUT`:
  - The device was not reachable.
  - There are three options available via the API client:
    - Retry by calling `selectDevice` with the `deviceRef` object.
    - Select a different device by calling `selectDevice` with a different `deviceRef` object.
    - Cancel the authentication request by calling `cancelAuthentication`.
- `PUSH_CONFIRMATION_REJECTED`:
  - The user receives a push notification, but denies it.
  - There are three options available via the API client:
    - Retry by calling `selectDevice` with the `deviceRef` object.
    - Select a different device by calling `selectDevice` with a different `deviceRef` object.
    - Cancel the authentication request by calling `cancelAuthentication`.

Device pairing flows

When using the PingOne MFA IdP Adapter through the PingFederate authentication application programming interface (API), the following flows are used for device pairing. These are initiated in the mobile app.

Copyright © 2025 Ping Identity Corporation

Pairing an initial device using automatic pairing



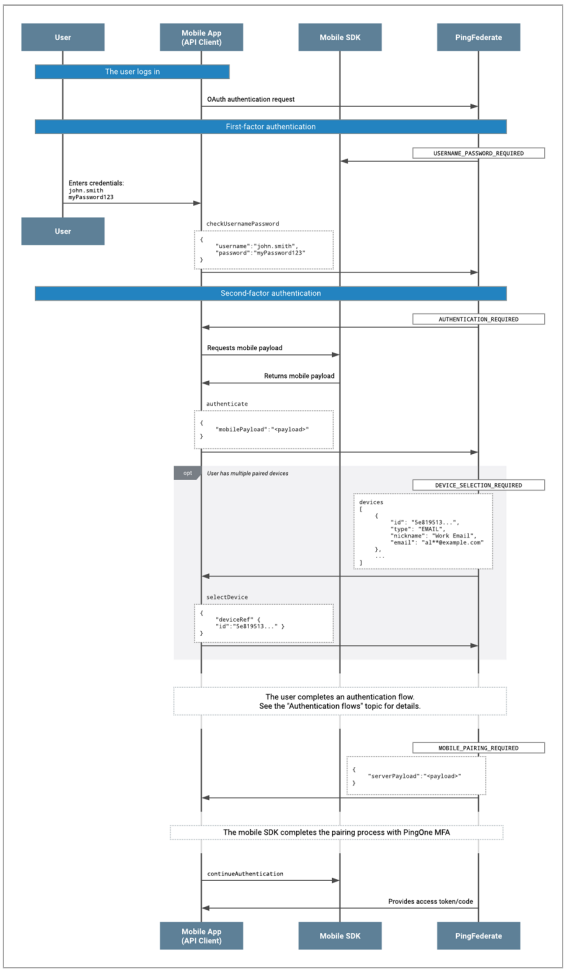
1. The user completes first-factor authentication. Completion of first-factor authentication is a prerequisite before progressing to multi-factor authentication (MFA), when using the PingOne MFA IdP Adapter with the PingFederate Authentication API flow.
2. The status of `AUTHENTICATION_REQUIRED` is returned in the response to the Mobile app (API client).
3. The Mobile app (API client) gets a mobile payload from the mobile SDK.
4. The Mobile app (API client) invokes the `authenticate` action, using the mobile payload.
5. The status of `MOBILE_PAIRING_REQUIRED` together with the `serverPayload` are returned in the response to the Mobile app (API client).
6. The Mobile app (API client) passes the `serverPayload` to the mobile SDK, in order to continue with the pairing process.
7. Once pairing is done, the Mobile app (API client) invokes the `continueAuthentication` action. The Mobile app (API client) must call `continueAuthentication` in order to progress in the OpenID Connect (OIDC) flow, and to complete it.

Copyright © 2025 Ping Identity Corporation

8. PingFederate returns an access token to the Mobile app (API client).

    - Even if the pairing is not successful, it is possible for the Mobile app (API client) to send the `continueAuthentication` action. In this case, the contract attribute `pingone.mfa.status` will have the value `com.pingidentity.pingone.device_not_paired`, rather than the value `com.pingidentity.pingone.device_paired`.
    - In the event of an error occuring during device pairing, the adapter will return a success status, and `pingone.mfa.status` will have the value `com.pingidentity.pingone.pairing_error`.

Pairing an additional device using automatic pairing



1. The user completes first-factor authentication. Completion of first-factor authentication is a prerequisite before progressing to MFA, when using the PingOne MFA IdP Adapter with the PingFederate Authentication API flow.
2. The status of `AUTHENTICATION_REQUIRED` is returned in the response to the Mobile app (API client). The returned device is the user's primary device.
3. The Mobile app (API client) gets a mobile payload from the mobile SDK.
4. The Mobile app (API client) invokes the `authenticate` action, using the mobile payload.

Copyright © 2025 Ping Identity Corporation

5. The response status would be different since users may have one or more devices already paired:

    1. The status of `PUSH_CONFIRMATION_WAITING` is returned if the mobile device is the only device that is paired. Push notification is sent to the paired mobile.

       The Mobile app (API client) invokes the `poll` action, so that PingFederate gets the status of the mobile push. This is repeated until the user approves or denies the push authentication request.

    2. The status of `OTP_REQUIRED` is returned if the only device that is paired is SMS or time-based one-time passcode (TOTP) authenticator or email.

       The Mobile app (API client) invokes the `checkOtp` action submitting the OTP value to PingFederate.

    3. The status of `DEVICE_SELECTION_REQUIRED` is returned with the `devices` object in the response to the API client if the user has more than one device paired.

       The Mobile app (API client) invokes the `selectDevice` action with the `deviceRef` object. This in turn can get the `PUSH_CONFIRMATION_WAITING` or `DEVICE_SELECTION_REQUIRED` status as mentioned above.

Even if the pairing is not successful, it is possible for the Mobile app (API client) to send the `continueAuthentication` action. In this case, the contract attribute `pingone.mfa.status` will have the value `com.pingidentity.pingone.device_not_paired`, rather than the value `com.pingidentity.pingone.device_paired`.

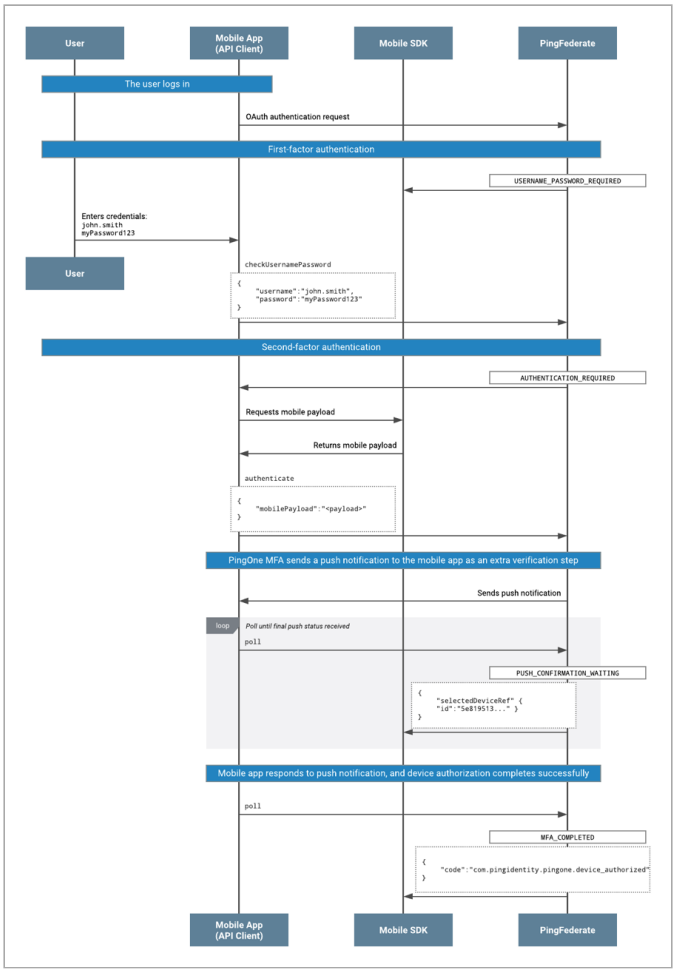
In the event of an error occurring during device pairing, the adapter will return a success status, and `com.pingidentity.pingone.status` will have the value `com.pingidentity.pingone.pairing_error`.

Authorization flow

When using the PingOne MFA IdP Adapter through the PingFederate authentication application programming interface (API), the following flow is used for requesting authorization using a push notification to the user's paired mobile app.

Copyright © 2025 Ping Identity Corporation

Authorization via the mobile app



1. The user completes first-factor authentication. Completion of first-factor authentication is a prerequisite before progressing to multi-factor authentication (MFA), when using the PingOne MFA IdP Adapter with the PingFederate Authentication API flow.
2. The status of `AUTHENTICATION_REQUIRED` is returned in the response to the Mobile app (API client).
3. The Mobile app (API client) gets a mobile payload from the mobile SDK.
4. The Mobile app (API client) invokes the `authenticate` action, using the mobile payload.
5. The status of `PUSH_CONFIRMATION_WAITING` together with the `selectedDeviceRef` object are returned in the response to the Mobile app (API client).
6. The Mobile app (API client) invokes the `poll` action, so that PingFederate gets the status of the mobile push. This is repeated until either a successful status is received or a timeout is reached.
7. The status of `MFA_COMPLETED` together with the `device_authorized` code are returned in the response to the Mobile app (API client).
8. The Mobile app (API client) invokes the `continueAuthentication` action. The Mobile app (API client) must call `continueAuthentication` in order to progress in the OIDC flow, and to complete it.

Copyright © 2025 Ping Identity Corporation

9. PingFederate returns an access token to the Mobile app (API client).

Models, objects, and error codes

When using the PingOne MFA IdP Adapter through the PingFederate authentication application programming interface (API), the adapter uses the following state models, action models, objects, and error codes.

State models

Status	Response Model	Action	Description
`AUTHENTICATION_REQUIRED`	• `user` : The user's username.	• `authenticate` • `cancelAuthentication`	Indicates that authentication is required. This state is the first state returned from the adapter.

Copyright © 2025 Ping Identity Corporation

`ASSERTION_REQUIRED`	• `devices`: The list of all devices associated with the user. • `user`: The user's username. • `selectedDeviceRef`: The device identifier. • `publicKeyCredentialRequestOptions`: The `PublicKeyCredentialRequestOptions` object containing data needed to generate an assertion. For help, see Web Authentication in the W3C documentation. ◦ `challenge`: string ◦ `timeout`: integer (milliseconds) ◦ `rpId`: Relying party identifier string. ◦ `userVerification`: string ◦ `allowCredentials` ▪ `type`: string ▪ `id`: array of bytes • `manualPairing`: Indicates the user has initiated the multi-factor pairing process. • `userSelectedDefault`: Indicates whether multi-factor authentication (MFA) settings has 'user-selected default device' or 'prompt user for device selection'. • `changeDevicePermitted` (boolean): Indicates whether the user can switch devices used for authentication. • `maxAllowedDevices`: The maximum number of devices that can be paired for a user.	• `cancelAuthentication` • `checkAssertion` • `selectDevice` • `setDefaultDevice` • `setupMfa` • `removeDevice` • `updateDeviceNickname`	Indicates that authentication is required. This state is the first state returned from the adapter.

Copyright © 2025 Ping Identity Corporation

Status	Response Model	Action	Description
`DEVICE_SELECTION_REQUIRED`	• `devices` : The list of all devices associated with the user. • `user` : The user's username. • `manualPairing` : Indicates the user has initiated the multi-factor pairing process. • `userSelectedDefault` : Indicates whether MFA settings has 'user-selected default device' or 'prompt user for device selection'. • `changeDevicePermitted` (boolean): Indicates whether the user can switch devices used for authentication. • `maxAllowedDevices` : The maximum number of devices that can be paired for a user.	• `setupMfa` • `cancelAuthentication` • `selectDevice` • `removeDevice` • `updateDeviceNickname`	Indicates that device selection is required, and the user has more than one device.

Copyright © 2025 Ping Identity Corporation

Status	Response Model	Action	Description
`OTP_REQUIRED`	• `selectedDeviceRef`: The device identifier. • `devices`: The list of all devices associated with the user. • `user`: The user's username. • `manualPairing`: Indicates the user has initiated the multi-factor pairing process. • `userSelectedDefault`: Indicates whether MFA settings has 'user-selected default device' or 'prompt user for device selection'. • `changeDevicePermitted` (boolean): Indicates whether the user can switch devices used for authentication. • `maxAllowedDevices`: The maximum number of devices that can be paired for a user.	• `checkOtp` • `cancelAuthentication` • `selectDevice` • `setDefaultDevice` • `setupMfa` • `removeDevice` • `updateDeviceNickname`	Indicates that one-time passcode (OTP) is required. This state is returned when the user is prompted to provide an OTP. The OTP is one of the following: • Sent to the user in an SMS, voice call, or email. • The OTP that is displayed in the user's authenticator app.

Copyright © 2025 Ping Identity Corporation

Status	Response Model	Action	Description
`PUSH_CONFIRMATION_WAITING`	• `selectedDeviceRef` : The device identifier. • `devices` : The list of all devices associated with the user. • `user` : The user's username. • `manualPairing` : Indicates the user has initiated the multi-factor pairing process. • `userSelectedDefault` : Indicates whether MFA settings has 'user-selected default device' or 'prompt user for device selection'. • `changeDevicePermitted` (boolean): Indicates whether the user can switch devices used for authentication. • `maxAllowedDevices` : The maximum number of devices that can be paired for a user.	• `checkOtp` • `poll` • `cancelAuthentication` • `selectDevice` • `setDefaultDevice` • `setupMfa` • `removeDevice` • `updateDeviceNickname`	**Indicates that a push was sent to the user.** **Note** To get the final push confirmation state, the API client can either call the poll, or call `GET` .

Copyright © 2025 Ping Identity Corporation

Status	Response Model	Action	Description
`PUSH_CONFIRMATION_TIMED_OUT`	• `selectedDeviceRef` : The device identifier. • `devices` : The list of all devices associated with the user. • `user` : The user's username. • `manualPairing` : Indicates the user has initiated the multi-factor pairing process. • `userSelectedDefault` : Indicates whether MFA settings has 'user-selected default device' or 'prompt user for device selection'. • `changeDevicePermitted` (boolean): Indicates whether the user can switch devices used for authentication. • `maxAllowedDevices` : The maximum number of devices that can be paired for a user.	• `checkOtp` • `cancelAuthentication` • `selectDevice` • `setDefaultDevice` • `setupMfa` • `removeDevice` • `updateDeviceNickname`	Indicates a push timeout state.

Copyright © 2025 Ping Identity Corporation

Status	Response Model	Action	Description
`PUSH_CONFIRMATION_REJECTED`	• `selectedDeviceRef`: The device identifier. • `devices`: The list of all devices associated with the user. • `user`: The user's username. • `manualPairing`: Indicates the user has initiated the multi-factor pairing process. • `userSelectedDefault`: Indicates whether MFA settings has 'user-selected default device' or 'prompt user for device selection'. • `Reason`: The push confirmation rejection reason. Possible values: ◦ `DENIED_BY_USER` • `changeDevicePermitted` (boolean): Indicates whether the user can switch devices used for authentication. • `maxAllowedDevices`: The maximum number of devices that can be paired for a user.	• `cancelAuthentication` • `selectDevice` • `setDefaultDevice` • `setupMfa` • `removeDevice` • `updateDeviceNickname`	Indicates that the user has rejected the push.
`MFA_COMPLETED`	• `code`: The success code. Learn more in PingOne MFA status attributes reference.	• `continueAuthentication`	Indicates a successful MFA. The API client must call `continueAuthentication` in order to progress in the flow, and to complete it.

Copyright © 2025 Ping Identity Corporation

Status	Response Model	Action	Description
`MFA_FAILED`	• `code` : The error code. • `message` : The developer-facing error message. • `userMessage` : The user-facing error message. See `MFA_FAILED` codes in the Error Codes section.	• `cancelAuthentication`	Indicates a dead end. The API client can proceed in the flow by calling `cancelAuthentication` . The adapter will return a FAILURE status.
`MOBILE_PAIRING_REQUIRED`	• `serverPayload` : String The PingOne ID token.	• `continueAuthentication`	Indicates that mobile pairing is required.
`MFA_SETUP_REQUIRED`	This state has no model.	• `setupMfa` • `skipMfa` • `cancelAuthentication`	The user must set up a device for multi-factor authentication.

Copyright © 2025 Ping Identity Corporation

Status	Response Model	Action	Description
`DEVICE_PAIRING_METHOD_REQUIRED`	• `devicePairingMethods`: The list of available device pairing methods for multi-factor authentication. ◦ `DevicePairingMethod`: A multi-factor authentication device pairing method. ▪ `deviceType`: The type of the device. ▪ `applicationName`: The name of the native application associated with this device, if the device is a mobile application. For other device types, this attribute is omitted. ▪ `applicationId`: The id of the native application associated with this device, if the device is a mobile application. For other device types, this attribute is omitted.	• `selectDevicePairingMethod` • `cancelDevicePairing` • `cancelAuthentication`	The user must select a device type for multi-factor authentication to proceed.

Copyright © 2025 Ping Identity Corporation

Status	Response Model	Action	Description
`MFA_DEVICE_PAIRING_METHOD_FAILED`	• `code` : The error code. • `message` : The developer-facing error message. • `userMessage` : The user-facing error message.	• `cancelDevicePairing` • `cancelAuthentication`	Device integrity validation failed. The device integrity check determined that the device is jailbroken (iOS) or rooted (Android).
`EMAIL_PAIRING_TARGET_REQUIRED`	This state has no model.	• `submitEmailTarget` • `cancelDevicePairing` • `cancelAuthentication`	The user must submit an email address for multi-factor authentication.
`SMS_PAIRING_TARGET_REQUIRED`	This state has no model.	• `submitSmsTarget` • `cancelDevicePairing` • `cancelAuthentication`	The user must submit a phone number for multi-factor authentication.
`VOICE_PAIRING_TARGET_REQUIRED`	This state has no model.	• `submitVoiceTarget` • `cancelDevicePairing` • `cancelAuthentication`	The user must submit a phone number for multi-factor authentication.
`MOBILE_ACTIVATION_REQUIRED`	• `applicationName` : The name of the native application associated with this device. • `pairingKey` : The pairing key used to pair the mobile device.	• `poll` • `cancelDevicePairing` • `cancelAuthentication`	The user must activate their mobile device to complete the device pairing.
`TOTP_ACTIVATION_REQUIRED`	• `pairingKey` : The pairing key used to pair the mobile device. • `keyUri` : The URI format of the QR code.	• `activateTotpDevice` • `cancelDevicePairing` • `cancelAuthentication`	The user must submit their authenticator activation code to complete the device pairing.
`EMAIL_ACTIVATION_REQUIRED`	• `email` : The user's specified email address for multi-factor authentication.	• `activateEmailDevice` • `resendOtp` • `cancelDevicePairing` • `cancelAuthentication`	The user must submit their email activation code to complete the device pairing.

Copyright © 2025 Ping Identity Corporation

Status	Response Model	Action	Description
`SMS_ACTIVATION_REQUIRED`	• `phone` : The user's specified phone number for multi-factor authentication.	• `activateSmsDevice` • `resendOtp` • `cancelDevicePairing` • `cancelAuthentication`	The user must submit their SMS activation code to complete the device pairing.
`VOICE_ACTIVATION_REQUIRED`	• `phone` : The user's specified phone number for multi-factor authentication.	• `activateVoiceDevice` • `resendOtp` • `cancelDevicePairing` • `cancelAuthentication`	The user must submit their voice activation code to complete the device pairing.
`PLATFORM_ACTIVATION_REQUIRED`	• `rp` : The relying party identifier. • `publicKeyCredentialCreationOptions` : A JSON serialization of the client data returned for registering a FIDO2 device.	• `activatePlatformDevice` • `cancelDevicePairing` • `cancelAuthentication`	The user must pair their biometrics with the browser to complete the device pairing.
`SECURITY_KEY_ACTIVATION_REQUIRED`	• `rp` : The relying party identifier. • `publicKeyCredentialCreationOptions` : A JSON serialization of the client data returned for registering a FIDO2 device.	• `activateSecurityKeyDevice` • `cancelDevicePairing` • `cancelAuthentication`	The user must pair their security key with the browser to complete the device pairing.
`AUTHENTICATION_CODE_RESPONSE_REQUIRED`	• Authentication code request data object.	• `poll` • `cancelAuthentication`	The user must complete MFA using an authentication code.

Copyright © 2025 Ping Identity Corporation

Status	Response Model	Action	Description
`ONE_TIME_DEVICE_OTP_METHOD_TYPE_INPUT_REQUIRED`	• `devices` : The list of all One-Time Device OTP devices object. • `changeDevicePermitted` (boolean): Indicates whether the user can switch device used for authentication. • `maxAllowedDevices` : The maximum number of devices that can be paired for a user.	• `selectOneTimeDeviceMethod` • `cancelAuthentication`	The user must select the one-time device from the list to proceed ahead with authentication flow.
`ONE_TIME_DEVICE_OTP_INPUT_REQUIRED`	• Selected One-Time Device Info: object • `devices` : The list of all One-Time Device OTP devices object • `changeDevicePermitted` (boolean): Indicates whether the user can switch device used for authentication • `maxAllowedDevices` : The maximum number of devices that can be paired for a user.	• `checkOtp` • `selectOneTimeDeviceMethod` • `resendOtp` • `cancelAuthentication`	Indicates that OTP is required. This state is returned when the user is prompted to provide an OTP. The OTP is one of the following: Sent to the user in an SMS, voice call, or email.
`BIOMETRIC_DEVICE_AUTHENTICATION_INFO_REQUIRED`	This state has no model.	• `continueBiometricDeviceAuthentication` • `cancelAuthentication`	The user must provide the server name where the fetch originates.
`UPDATE_NICKNAME`	This state has no model.	• `updateDeviceNickname` • `skipUpdateDeviceNickname`	The user can either add a device nickname or skip adding a device nickname.

Copyright © 2025 Ping Identity Corporation

Action models

Action	Request Model	Errors	Description
`authenticate`	• `mobilePayload` : The mobile payload. The payload is a small data package created by the PingOne Mobile SDK component that identifies the device, which is used as part of the device's authorization.	• Error: `REQUEST_FAILED` • Error: `VALIDATION_ERROR` ErrorDetail: `INVALID_MOBILE_PAYLOAD`	Starts an authentication flow. The mobile payload is required in cases of mobile app access. The presence or absence of the mobile payload determines whether the flow is a mobile or web authentication, respectively. You cannot switch midway between mobile and web authentication flows.
`selectDevice`	• `deviceRef` : The device identifier. (required) • `mobilePayload` : The mobile payload. The payload is a small data package created by the PingOne Mobile SDK component that identifies the device, which is used as part of the device's authorization.	• Error: `REQUEST_FAILED` ErrorDetail: `OTP_RESEND_LIMIT` • Error: `REQUEST_FAILED` ErrorDetail: `PUSH_FAILED` • Error: `VALIDATION_ERROR` ErrorDetail: `INVALID_DEVICE` • Error: `VALIDATION_ERROR` ErrorDetail: `INVALID_MOBILE_PAYLOAD` • Error: `REQUEST_FAILED` ErrorDetail: `TOTP_ATTEMPTS_LIMIT`	Starts an authentication with the specified deviceId. For example: `{ "deviceRef": { "id": "<device ID>" } }` If there is already an authentication in progress, this authentication will be canceled. This action is available only when the user has at least one device, and can also be used as an authentication retry. A mobile payload is required if a user requests access from an untrusted mobile app, and needs to select of one of the user's trusted devices to approve or deny access.

Copyright © 2025 Ping Identity Corporation

Action	Request Model	Errors	Description
`setDefaultDevice`	• `deviceRef` : The device identifier. (required)	• Error: `REQUEST_FAILED` ErrorDetail: `OTP_RESEND_LIMIT` • Error: `REQUEST_FAILED` ErrorDetail: `PUSH_FAILED` • Error: `VALIDATION_ERROR` ErrorDetail: `INVALID_DEVICE` • Error: `VALIDATION_ERROR` ErrorDetail: `INVALID_MOBILE_PAYLOAD` • Error: `REQUEST_FAILED` ErrorDetail: `TOTP_ATTEMPTS_LIMIT`	Sets the user's default authentication device. For example: `{ "deviceRef": { "id": "<device ID>" } }` If there is already an authentication in progress, this authentication will be canceled. This action is available only when the user has at least one device, and can also be used as an authentication retry. A mobile payload is required if a user requests access from an untrusted mobile app, and needs to select of one of the user's trusted devices to approve or deny access.
`checkOtp`	• `otp` : The OTP (one-time passcode) submitted by the user.	• Error: `VALIDATION_ERROR` ErrorDetail: `INVALID_OTP` • Error: `REQUEST_FAILED` ErrorDetail: `OTP_ATTEMPTS_LIMIT` • Error: `REQUEST_FAILED` ErrorDetail: `TOTP_ATTEMPTS_LIMIT`	Validates the provided OTP.
`poll`	This action has no model.	• Error: `REQUEST_FAILED` • Error: `VALIDATION_ERROR`	This action returns the authentication code request status.
`cancelAuthentication`	This action has no model.	This action has no errors.	This action cancels the current authentication step.

Copyright © 2025 Ping Identity Corporation

Action	Request Model	Errors	Description
`continueAuthentication`	This action has no model.	This action has no errors.	This action continues the current authentication flow.
`setupMfa`	This action has no model.	This action has no errors.	Initiates the multi-factor authentication pairing process.
`skipMfa`	This action has no model.	• Error: `INVALID_REQUEST`	Skip the multi-factor authentication pairing process.
`selectDevicePairingMethod`	• `devicePairingMethod`: DevicePairingMethod	• Error: `VALIDATION_ERROR` Error detail: `INVALID_DEVICE_PAIRING_METHOD`	Select a device pairing method to pair as a new multi-factor authentication device.
`cancelDevicePairing`	This action has no model.	This action has no errors.	Cancel the multi-factor authentication pairing process.
`submitEmailTarget`	• `email`: The user's specified email address for multi-factor authentication. • `testMode`: Determines whether this multi-factor authentication device is configured as a test device.	• Error: `VALIDATION_ERROR` Error detail: `INVALID_EMAIL`	Pair an email address for use as a new multi-factor authentication device.
`submitSmsTarget`	• `phone`: The user's phone number for multi-factor authentication. • `testMode`: Determines whether this multi-factor authentication device is configured as a test device.	• Error: `VALIDATION_ERROR` Error detail: `INVALID_PHONE`	Pair a phone number for use as a new multi-factor authentication device.

Copyright © 2025 Ping Identity Corporation

Action	Request Model	Errors	Description
`submitVoiceTarget`	• `phone` : The user's phone number for multi-factor authentication. • `testMode` : Determines whether this multi-factor authentication device is configured as a test device.	• Error: `VALIDATION_ERROR` Error detail: `INVALID_PHONE`	Pair a phone number for use as a new multi-factor authentication device.
`activateEmailDevice`	• `otp` : The OTP (one-time passcode) submitted by the user.	• Error: `VALIDATION_ERROR` Error detail: `INVALID_OTP` • Error: `REQUEST_FAILED` Error detail: `OTP_ATTEMPTS_LIMIT`	Submit the email activation code to complete the device pairing.
`activateSmsDevice`	• `otp` : The OTP (one-time passcode) submitted by the user.	• Error: `VALIDATION_ERROR` Error detail: `INVALID_OTP` • Error: `REQUEST_FAILED` Error detail: `OTP_ATTEMPTS_LIMIT`	Submit the SMS activation code to complete the device pairing.
`activateVoiceDevice`	• `otp` : The OTP (one-time passcode) submitted by the user.	• Error: `VALIDATION_ERROR` Error detail: `INVALID_OTP` • Error: `REQUEST_FAILED` Error detail: `OTP_ATTEMPTS_LIMIT`	Submit the voice activation code to complete the device pairing.

Copyright © 2025 Ping Identity Corporation

Action	Request Model	Errors	Description
`activateTotpDevice`	• `otp` : The OTP (one-time passcode) submitted by the user.	• Error: `VALIDATION_ERROR` Error detail: `INVALID_OTP` • Error: `REQUEST_FAILED` Error detail: `OTP_ATTEMPTS_LIMIT`	Activate an authenticator for use as a new multi-factor authentication device. Submit the authenticator activation code to complete the device pairing.
`activatePlatformDevice`	• `origin` : A string that specifies the server name where the fetch originates, providing the URI scheme and hostname. • `attestation` : The attestation generated by the browser as a response to a user action, such as a fingerprint or click on the security key.	• Error: `VALIDATION_ERROR` Error detail: `INVALID_DATA`	
`activateSecurityKeyDevice`	• `origin` : A string that specifies the server name where the fetch originates, providing the URI scheme and hostname. • `attestation` : The attestation generated by the browser as a response to a user action, such as a fingerprint or click on the security key.	• Error: `VALIDATION_ERROR` Error detail: `INVALID_DATA`	
`selectOneTimeDeviceMethod`	• `One-Time Device Info` : object	• Error: `REQUEST_FAILED` • Error: `VALIDATION_ERROR`	Starts the One-Time Device OTP authentication flow for the selected device.

Copyright © 2025 Ping Identity Corporation

Action	Request Model	Errors	Description
`resendOtp`	This action has no model.	• Error: `REQUEST_FAILED` • Error: `VALIDATION_ERROR`	Resend the OTP to the previously selected device.
`continueBiometricDeviceAuthentication`	• `origin`: A string that specifies the server name where the fetch originates.	• Error: `REQUEST_FAILED` • Error: `VALIDATION_ERROR`	Continue with biometric device authentication.
`updateDeviceNickname`	• `id (string)`: Unique identifier of a trusted device in the PingOne MFA server (required). • `nickname (string)`: The device's nickname (required).	• Error: `RESOURCE_NOT_FOUND` • Error: `REQUEST_FAILED`	Give a unique nickname to a device.
`skipUpdateDeviceNickname`	This action has no model.	This action has no errors.	Skip updating the device nickname during the device pairing flow.
`removeDevice`	• `deviceRef`: The device identifier (required). • `id (string)`: Unique identifier of a trusted device in the PingOne MFA server (required).	• Error: `VALIDATION_ERROR` ErrorDetail: `FIELD_REQUIRED`, `INVALID_DEVICE`, or `UNRECOGNIZED_FIELD_NAME`. • Error: `INVALID_REQUEST`	Remove a device. See the `selectDevice` table entry for an example `deviceRef`.

Objects

Device object

Parameter name	Type	Description
`id`	String	The unique identifier of a trusted device in the PingOne MFA server.

Copyright © 2025 Ping Identity Corporation

Parameter name	Type	Description
`type`	String	The model of the device. For example, `iPhone 5S`. This parameter is empty for OTP devices (SMS, voice, email, TOTP).
`name`	String	The name of the device. For example, `iPhone 5S`. This parameter is empty for OTP devices (SMS, voice, email, TOTP).
`nickname`	String	The device's nickname.
`applicationId`	String	The ID of the customer mobile application.
`osVersion`	String	The device's operating system version.
`applicationVersion`	String	The device's application version.
`target`	String	The device's masked email address/phone number.
`usable`	boolean	Indicates whether the device is usable.

One-time device OTP device object

`type`	String	The type of One-Time Device. Possible values - SMS, voice, email.
`target`	String	The device's masked email address/ phone number.
`id`	String	Random ID to identify the device within the response.

User object

Parameter name	Type	Description
`username`	String	The user's PingOne user ID or username that was mapped into the adapter.

Resource reference (ResourceRef) object

Parameter name	Type	Description
`id`	String	The resource's identifier.

Copyright © 2025 Ping Identity Corporation

Device pairing method (DevicePairingMethod) object

Parameter name	Type	Description
`deviceType`	String	The type of the device.
`applicationName`	String	The name of the native application associated with this device, if the device is a mobile application. For other device types, this attribute is omitted.
`applicationId`	String	The ID of the native application associated with this device, if the device is a mobile application. For other device types, this attribute is omitted.
`relyingPartyId`	String	Specifies the server name/relying party that is initiating the request. This is required to initiate pairing requests of security key/biometrics devices. If not provided, the PingFederate domain is used.
`relyingPartyName`	String	Identifies the server name/relying party that is initiating the request. This is required to initiate pairing requests of security key/biometrics devices. If not provided, the string `PingFederate` is used.
`userAgent`	String	User agent of device initiating the pairing request. This is applicable only for platform/biometrics device pairing requests. It is passed to PingOne so it can accurately capture additional platform information of the device.

Relying party (RelyingParty) object

Parameter name	Type	Description
`id`	String	The relying party identifier.
`name`	String	The relying party's human-readable display name.

Authentication code request data object

Parameter Name	Type	Description
`authenticationCodeId`	String	A string that specifices the code request ID.

Copyright © 2025 Ping Identity Corporation

Parameter Name	Type	Description
`code`	String	A string that specifies a universal link such as https://www.example.com/pingonesdk?authentication_code={{generated token}}). This property can also set a link to a schema application such as pingoneapp://pingonesdk?authentication_code={{generated token}}). If a universal link or schema application link is not set, the response does not include the pingonesdk?authentication_code={{generated token}} portion of the URL.
`userApproval`	String	An enumeration that specifies whether the mobile device must verify that the user approves the authentication with the scanned code. Options are `REQUIRED` and `NOT_REQUIRED`.
`expiresAt`	String	A date that specifies the expiration time of authentication code.
`updatedAt`	String	A date that specifies when the resource was last updated.
`createdAt`	String	A date that specifies when the resource was created.
`application`	Application	Mobile application info.
`clientContext`	ClientContext	Provides relevant information to the mobile application that can be shown to application users.
`lifeTime`	LifeTime	Specifies the length of time for this authentication code to be valid.
`requestStatus`	String	A string that specifies the status of the authentication code. Options are `UNCLAIMED` or `CLAIMED`.

Copyright © 2025 Ping Identity Corporation

Parameter Name	Type	Description
`Applicationid`	String	A string that specifies the ID of a mobile application for which the code is valid; only the specified mobile application can scan the QR code.
`header`	String	Client information header.
`body`	String	Client information body
`Lifetime duration`	Integer	Specifies the length of time for this authentication code to be valid.
`timeUnit`	String	A string that specifies the time unit for the `lifeTime.duration` property. Options are `SECONDS` and `MINUTES`.

> **Note**
>
> There may be values such as SMS message, email configuration type and others, that you want the client to pass to PingFederate for step-up authentication and transaction approval. However, these values cannot be sent using the PingFederate Authentication API, due to security reasons. See Transaction approval setup for details on the available parameters and the instructions for sending them to PingFederate.

Error codes

An error code is returned if the call flow state has not reached a dead end, and the user can still authenticate with a device. In cases where a flow reaches a dead end, the `MFA_FAILED` state is returned with a corresponding code.

Top level error codes

Error code	Message	HTTP status
`VALIDATION_ERROR`	One or more validation errors occurred.	`400`
`REQUEST_FAILED`	The request could not be completed. There was an issue processing the request.	`400`
`RESOURCE_NOT_FOUND`	The requested resource was not found.	`404`

Detail level error codes

Error code	Message	userMessageKey	Parent code
`INVALID_OTP`	An invalid or expired passcode was provided.	`authn.api.invalid.otp`	`VALIDATION_ERROR`

Copyright © 2025 Ping Identity Corporation

Error code	Message	userMessageKey	Parent code
`OTP_EXPIRED`	The passcode has expired.	`authn.api.otp.expired`	`REQUEST_FAILED`
`INVALID_DEVICE`	An invalid device was provided.		`VALIDATION_ERROR`
`OTP_ATTEMPTS_LIMIT`	The user performed too many unsuccessful passcode attempts.	`authn.api.otp.attempts.limit`	`REQUEST_FAILED`
`TOTP_ATTEMPTS_LIMIT`	The user performed too many unsuccessful time-based one-time passcode (TOTP) passcode attempts.	`authn.api.totp.attempts.limit`	`REQUEST_FAILED`
`OTP_RESEND_LIMIT`	The user has resent the passcode the maximum number of times.	`authn.api.otp.resend.limit`	`REQUEST_FAILED`
`PUSH_FAILED`	Failed to send the push message.	`authn.api.push.failed`	`REQUEST_FAILED`
`INVALID_MOBILE_PAYLOAD`	An invalid mobile payload was provided.		`VALIDATION_ERROR`
`INVALID_DEVICE_PAIRING_METHOD`	An invalid device pairing method was provided.		`VALIDATION_ERROR`
`INVALID_DATA_DEVICE_PAIRING_METHOD`	The request could not be completed. One or more validation errors were in the request.		`VALIDATION_ERROR`
`INVALID_EMAIL`	An invalid email address was provided.	`pingone.mfa.email.pairing.invalid.email`	`VALIDATION_ERROR`
`INVALID_PHONE`	An invalid phone number was provided.	`pingone.mfa.sms.pairing.invalid.phone` or `pingone.mfa.voice.pairing.invalid.phone`	`VALIDATION_ERROR`
`MAXIMUM_ALLOWED_METHODS_LIMIT`	Reached maximum number of allowed methods limit.	`pingone.mfa.device.selection.maximum.allowed.methods.limit`	`VALIDATION_ERROR`

MFA_FAILED codes

Error code	Message	userMessageKey
`SERVER_ERROR`	Server error.	`authn.api.server.error`
`SERVICE_UNAVAILABLE`	Service unavailable.	`authn.api.service.unavailable`

Copyright © 2025 Ping Identity Corporation

Error code	Message	userMessageKey
`OTP_RESEND_LIMIT` This error code can also be returned if this is not a dead end.	The user has re-sent the passcode the maximum number of times.	`authn.api.otp.resend.limit`
`PUSH_FAILED`	Failed to send the push message.	`authn.api.push.failed`
`NO_RESPONSE_PASSIVE_PUSH`	Mobile payload is valid but the extra push verification did not arrive.	`authn.api.no.response.passive.push`
`NO_USABLE_DEVICES`	The user has no usable devices for the requested authentication policy.	`authn.api.no.usable.devices`
`MFA_DISABLED`	The user does not have MFA enabled.	`authn.api.mfa.disabled`
`USER_NOT_FOUND`	The user is not enabled in PingOne or does not exist.	`authn.api.user.not.found`
`DEVICE_INTEGRITY_FAILED`	Device integrity validation failed.	`authn.api.device.rooted`

Troubleshooting

Known issues and limitations

The following are known issues or limitations for the PingOne MFA Integration Kit.

Known issues

There are no known issues.

Known limitations

- The PingOne MFA IdP Adapter only supports automatic device enrollment for SMS, voice, and email authentication methods. Users can add other authentication methods directly through the PingOne MFA self-service URL. Learn more in Self service and Managing authentication methods in the PingOne MFA documentation.
- To use a localized version of the adapter messages file, a copy of the core PingFederate messages file must exist with the same language tag. For example, to allow `pingone-mfa-messages_fr.properties` to work, create `pingfederate-messages_fr.properties`.
- As a security measure, if the user initiates a password reset flow and multi-factor authentication (MFA) is not satisfied, the PingOne MFA IdP Adapter fails. For example, this applies when the user clicks the password reset link on the HTML Form Adapter and the PingOne authentication policy dictates that MFA is bypassed for the user.

Copyright © 2025 Ping Identity Corporation

- For the Default Authentication Method Type setting: if a user has existing authentication methods, but no default is set, the adapter does not set a default authentication method. This scenario can occur if the user was created before PingOne supported default authentication methods.
- The PingOne MFA IdP Adapter only adds authentication methods to PingOne. If you want to synchronize authentication methods and other user attributes, use the PingOne Connector provided in the PingOne Integration Kit.

Enabling debug logging

To help with troubleshooting or monitoring, you can turn on activity logging for the adapter.

About this task

These steps are optional. Learn more in Enabling debug messages and console logging in the PingFederate documentation.

Steps

1. Open the `<pf_install>/pingfederate/server/default/conf/log4j2.xml` file for editing.
2. If you want to log activity for PingFederate and all adapters, do the following.
    1. Find the following section.

    ```
    <AsyncRoot level="INFO" includeLocation="false">
            <!-- <AppenderRef ref="CONSOLE" /> -->
            <AppenderRef ref="FILE" />
    </AsyncRoot>
    ```

    2. Change `INFO` to `DEBUG`.

    ```
    <AsyncRoot level="DEBUG" includeLocation="false">
            <!-- <AppenderRef ref="CONSOLE" /> -->
            <AppenderRef ref="FILE" />
    </AsyncRoot>
    ```

    3. If you want to see the adapter activity in the console, remove the comment tags.

    ```
    <AsyncRoot level="INFO" includeLocation="false">
            <AppenderRef ref="CONSOLE" />
            <AppenderRef ref="FILE" />
    </AsyncRoot>
    ```

3. If you want to log activity just for the PingOne MFA IdP Adapter, add the following line.

```
<Logger name="com.pingidentity.adapters.pingone.mfa" level="DEBUG"/>
```

4. If you want to log activity just for the PingOne MFA CIBA Authenticator, add the following line.

Copyright © 2025 Ping Identity Corporation

```
<Logger name="com.pingidentity.oobauth.pingone.mfa" level="DEBUG"/>
```

5. Save the file.

Download manifest

The following files are included in the PingOne MFA Integration Kit `.zip` archive:

- `Legal.pdf` – Copyright and license information
- `dist/pingfederate/server/default` – Contains the integration files
  - `conf` – Contains the HTML template that presents the PingOne sign-on form
    - `language-packs` – Contains files with customizable user-facing messages
      - `authn-api-messages.properties` – A version of the common PingFederate authentication application programming interface (API) variable file that includes adapter-specific messages
      - `pingone-mfa-messages.properties` – A variable file that customizes the messages that appear on the template file
    - `template` – Contains user-facing HTML template files
      - `assets` – Contains functional scripts and files used by the template
        - `css` – Contains CSS files for the templates
          - `end-user/<version>/end-user.css` – A CSS file that customizes the appearance of the template files
          - `pingone-mfa.css` – A CSS file that customizes the appearance of the template files
        - `fonts/end-user/icons` – Contains template icons
        - `images` – Contains template image files
          - `icons` – A collection of icons used in the template file
          - `flags.png` – A collection of flags for internationalization
          - `ping-logo.png` – An image file with company branding
          - `ping-logo.svg` – An image file with company branding
          - `spinner-alt.svg` – An image file used in a spinner animation
          - `spinner.svg` – An image file used in a spinner animation
        - `scripts` – Contains script files used to collect and send information
          - `jquery-<version>.min.js` – A JavaScript file that contains the jQuery library
          - `jquery-qrcode.min.js` – A JavaScript file that generates QR codes

Copyright © 2025 Ping Identity Corporation

- `data` – Contains JSON files used by the templates
    - `CountriesWithCountryCode.json` - A JSON file that enables use of the country prefix selector on the Device Selection page
- `pingone-mfa-assertion-required.html` – A sample template that indicates a request has been sent to a FIDO2 device
- `pingone-mfa-completed.html` – A sample template that indicates the multi-factor authentication (MFA) step completed successfully
- `pingone-mfa-device-pairing-method-failed.html` – A sample template that indicates the device pairing method failed on the server side
- `pingone-mfa-device-pairing.html` – A sample template that allows the user to pair an MFA device
- `pingone-mfa-device-selection.html` – A sample template that allows the user to select the MFA device they want to use
- `pingone-mfa-email-activation.html` – A sample template that prompts the user to activate an email MFA device
- `pingone-mfa-email-pairing.html` – A sample template that prompts the user to enter an email to use for MFA
- `pingone-mfa-failed.html` – A sample template that indicates the MFA step failed
- `pingone-mfa-mobile-activation.html` – A sample template that prompts the user to activate a mobile device
- `pingone-mfa-otp-required.html` – A sample template that prompts the user to enter a one-time passcode
- `pingone-mfa-platform-activation.html` – A sample template that prompts the user to activate a biometrics device
- `pingone-mfa-push-confirmation-rejected.html` – A sample template that indicates an MFA push was rejected
- `pingone-mfa-push-confirmation-timed-out.html` – A sample template that indicates an MFA push timed out
- `pingone-mfa-push-confirmation-waiting.html` – A sample template that indicates the adapter is waiting for a response to an MFA push
- `pingone-mfa-security-key-device-activation.html` – A sample template that prompts the user to activate a security key device
- `pingone-mfa-setup.html` – A sample template that prompts the user to set up an MFA device
- `pingone-mfa-sms-activation.html` – A sample template that prompts the user to activate an SMS MFA device

Copyright © 2025 Ping Identity Corporation

        - `pingone-mfa-sms-pairing.html` – A sample template that prompts the user to enter a phone number to use for MFA
        - `pingone-mfa-totp-device-activation.html` – A sample template that prompts the user to activate a timed one-time password (TOTP) device
        - `pingone-mfa-voice-activation.html` – A sample template that prompts the user to activate a voice MFA device
        - `pingone-mfa-voice-pairing.html` – A sample template that prompts the user to enter a phone number to use for MFA
    - `deploy` – Contains the Java libraries
        - `pf-pingone-mfa-adapter-<version>.jar` – A JAR file that contains the PingOne MFA IdP Adapter
    - `lib/pf-authn-api-sdk-<version>.jar` – A JAR file that contains the PingFederate Authentication API SDK

PingOne Risk Integration Kit

The PingOne Risk Integration Kit allows PingFederate to communicate with PingOne Risk for risk-based authentication.

By sending transaction information and an optional device profile to PingOne when a user signs on, PingFederate can get a security risk evaluation for the sign-on event. Including the risk evaluation in your PingFederate authentication policy allows you to dynamically adjust the user's authentication requirements each time they sign on.

Download

To download the PingOne Risk Integration Kit, go to PingFederate Downloads on the Ping Identity site.

Components

PingOne Risk IdP Adapter

- When a user signs on through PingFederate, the adapter sends the transaction information to PingOne Risk, and retrieves a Result and other information about the user's current and previous transactions.

Template and script files

- When a user signs on through PingFederate and device profiling is enabled, these files create a device profile for the adapter to send to PingOne Risk.

Intended audience

This document is intended for PingFederate administrators.

If you need help during the setup process, learn more in the following resources:

- PingOne Protect on the Ping Identity site
- PingOne Protect in the PingOne documentation

Copyright © 2025 Ping Identity Corporation

- The following sections of the PingFederate documentation:
  - Managing IdP adapters
  - Authentication policies

System requirements

- PingFederate 10.2 or later
- To allow PingFederate to make outbound HTTPS connections, you might need to allow the following host names in your firewall:
  - https://api.pingone.com, https://api.pingone.asia, or https://api.pingone.eu
  - https://auth.pingone.com, https://auth.pingone.asia, or https://auth.pingone.eu
- A PingOne Risk license

  Learn more about creating a trial account in Starting a PingOne Trial.

Overview of PingOne Risk

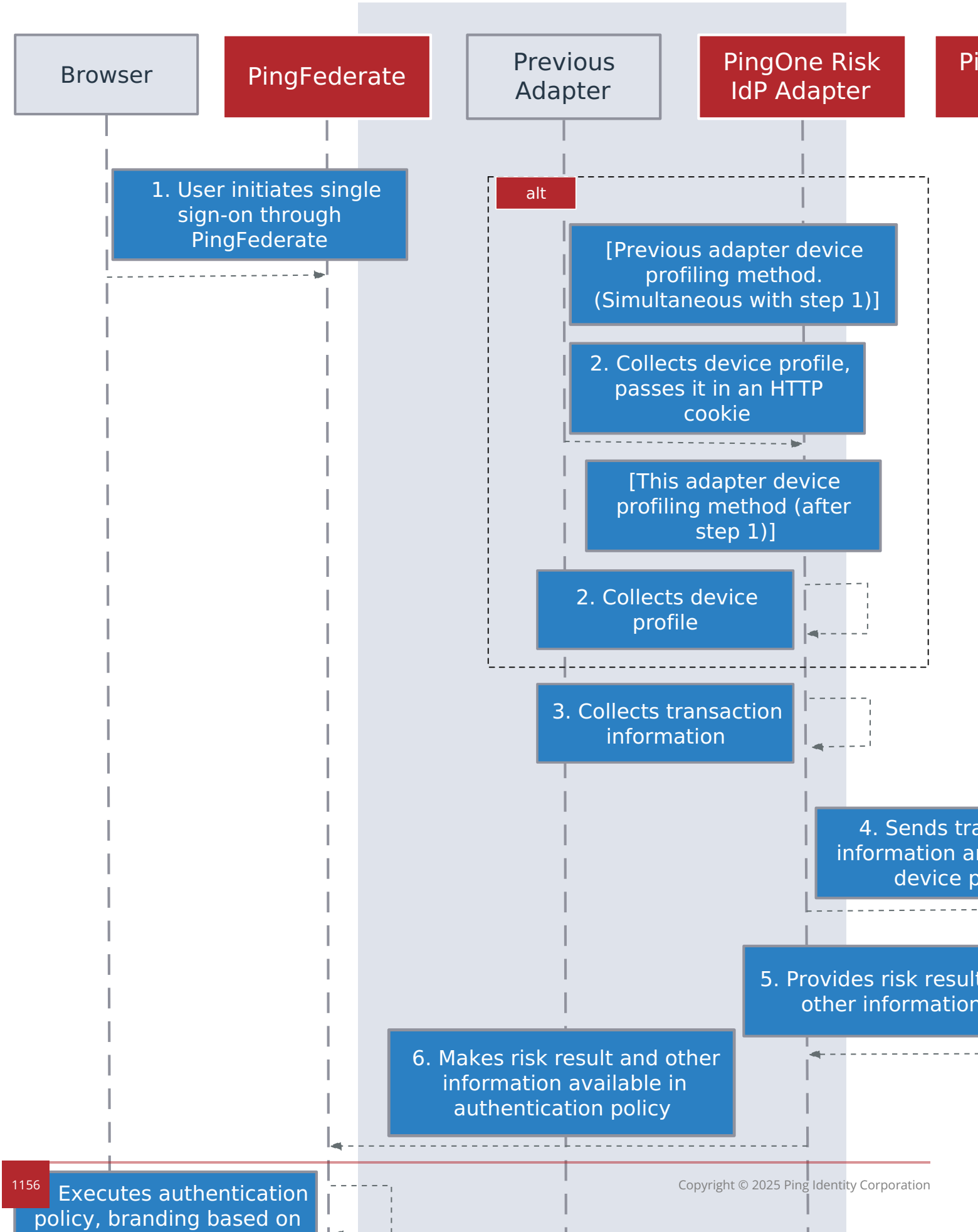
PingOne Risk evaluates the level of security risk for a user sign-on event based on a device profile and transaction information. For more information, see PingOne Protect in the PingOne documentation.

Overview of the SSO flow

With the PingOne Risk Integration Kit, PingFederate includes PingOne Risk in the sign-on flow.

The following figure shows how PingOne Risk is integrated into the sign-on process:

Copyright © 2025 Ping Identity Corporation



Copyright © 2025 Ping Identity Corporation

Description

1. A user initiates the sign-on process by requesting access to a protected resource.
2. When device profiling is enabled, one of the following occurs, depending on the device profiling method:
    - An adapter that is earlier in the authentication flow runs a script that creates a device profile. The script passes the device profile to the PingOne Risk IdP Adapter in a series of HTTP cookies.
    - The PingOne Risk IdP Adapter creates a device profile.
3. The PingOne Risk IdP Adapter collects transaction information, such as the user's IP address.
4. The adapter sends the transaction information and optional device profile to PingOne Risk.
5. PingOne Risk returns a JSON payload with the risk result and other information, such as the IP reputation, to the adapter.
6. The PingOne Risk IdP Adapter makes the risk result and other information available in the PingFederate authentication policy.
7. PingFederate executes the authentication policy, which branches based on the risk result provided by the adapter.
8. PingFederate returns the resource that the user requested.
9. The adapter notifies PingOne Risk whether authentication ultimately succeeded. This helps PingOne Risk evaluate subsequent sign-on attempts.

Device profiling methods

In addition to the basic transaction information, PingOne Risk can include an optional device profile in its risk evaluation.

There are different methods for collecting the device profile depending on your preferences and whether users are authenticating directly through PingFederate or through the PingFederate authentication application programming interface (API).

Complete steps for each device profiling method are available when you reach the Integrating device profiling part of the setup process.

Signals SDK and Fingerprint JS

Signals SDK is the preferred device profiling method and is recommended for use in the PingOne Risk Integration Kit 1.3 and later. This method provides the best risk detection and is required for the newest risk predictors such as New Device.

For backwards compatibility, Fingerprint JS is also supported. Either option can be used for device profiling and can be used with each of the following scenarios.

> **Note**
>
> Some resource files might need to be updated if using Fingerprint JS.

Copyright © 2025 Ping Identity Corporation

Direct authentication mode: Captured by the PingOne Risk IdP Adapter

The PingOne Risk IdP Adapter inserts a page into the sign-on flow that shows a spinner animation while it collects the device profile. The page appears in the sign-on flow in the order set by your PingFederate authentication policy, typically after the first-factor sign-on page.

This method does not require modifications to any other pages, but adds a wait time to the sign-on process while the device profile is collected. The length of the wait depends on your environment.

Direct authentication mode: Captured by a previous adapter

You add a device profiling script to a page that appears earlier in the sign-on flow than the PingOne Risk IdP Adapter. The script creates the device profile and stores it in a series of HTTP cookies. When the PingOne Risk IdP Adapter is triggered by the PingFederate authentication policy, it picks up the device profile from the cookies. The adapter sends the device profile to PingOne Risk along with the transaction information.

By adding the device profiling script to your first-factor authentication page, for example, the device profile is created while the user types in their credentials. This can reduce the perceived wait time during sign on.

You can integrate the device profiling script into any page that meets the following criteria:

- The page appears before the PingOne Risk IdP Adapter in the sign-on flow.
- The page is hosted in the same domain as your PingFederate server. This allows HTTP cookies to pass the transaction information to the PingOne Risk IdP Adapter. You might be able to work around this requirement by consolidating your domains with a reverse proxy server.
- You have access to the page to add the script.

Authentication API mode: Captured by your web application first-factor sign-on page

You add a device profiling script to your web application sign-on page. The script creates the device profile and stores it in a series of HTTP cookies. When the PingOne Risk IdP Adapter is triggered by the PingFederate authentication policy, it picks up the device profile from the cookies. The adapter sends the device profile to PingOne Risk along with the transaction information.

Setup

Deploying the integration files

To get started with the integration, deploy the PingOne Risk Integration Kit files to your PingFederate directory.

Before you begin

Make sure you have completed the following tasks:

1. Create an environment in PingOne. Learn more in Adding an environment.
2. Create a custom risk policy in PingOne Risk. Learn more in Risk policies.
3. Connect PingFederate to PingOne Risk. Learn more in Connecting PingFederate to PingOne.

Copyright © 2025 Ping Identity Corporation

Steps

1. Download the PingOne Risk Integration Kit `.zip` archive from the Ping Identity Integration Directory.
2. Stop PingFederate.
3. If you are upgrading an existing deployment, back up your customizations and delete earlier versions of the integration files:
    1. Back up any PingOne Risk Integration Kit files that you have customized in the `<pf_install>/pingfederate/server/default/conf/` directory.
    2. Delete the `pf-pingone-risk-management-adapter-<version>.jar` file from the `<pf_install>/pingfederate/server/default/deploy` directory.

    > **Note**
    >
    > If you are upgrading and not moving to the Signals SDK, you must update the template HTML and JS files.
4. Extract the `.zip` archive and merge the contents of the `dist` directory with your `<pf_install>/pingfederate` directory.
5. If there is more than one version of the `pf-authn-api-sdk-<version>.jar` file in your `<pf_install>/pingfederate/server/default/lib` directory, delete all but the latest version of the file.
6. If you backed up any customized files, modify the new files with your customizations.
7. Start PingFederate.
8. If you operate PingFederate in a cluster, repeat steps 2-7 for each engine node.

Connecting PingFederate to PingOne

To allow PingFederate to communicate with PingOne, create a connection.

Connecting PingFederate 10.2 or later to PingOne

About this task

Create a connection between PingFederate and PingOne. In PingFederate 10.2 and later, you can use this connection for all PingFederate components that communicate with PingOne.

> **Note**
>
> If you already have a connection from PingOne to PingFederate, ensure that the **Identity admin** and **Environment admin** roles are enabled. Skip the steps below.

Steps

1. Create credentials in PingOne:
    1. On the PingOne administrative console, go to Integrations > PingFederate and click the + icon.

Copyright © 2025 Ping Identity Corporation

2. On the Create Connection Profile window, enter a connection name, then click Save.
3. After the New Credential Created window opens, click Copy To Clipboard, then Done.

2. Use the credentials to create a connection in PingFederate:
    1. On the PingFederate administrative console, go to System > External Systems > PingOne Connections and click + Add Connection.
    2. On the Add Connection window, paste the credential information in the Paste The Credential Here field.
    3. In the Connection Name field, enter a unique name and click Save.

Connecting PingFederate 10.1 or earlier

About this task

In PingOne, create a set of credentials for PingFederate. For PingFederate 10.1 and earlier, you must enter these credentials in each PingFederate component that communicates with PingOne.

Steps

1. On the PingOne administrative console, go to Applications > Applications and click the + icon.
2. On the Add Application window, enter a unique Application Name.
3. In the Application Type list, select Worker, then Save.
4. On the Roles tab, click Grant Roles, then select the Identity Data Administrator check box for the environment.
5. On the Overview or Configuration tab, note the Client ID, Client Secret, and Environment ID.

    You'll use these credentials in Configuring an adapter instance.
6. Click the toggle at the top of the application to activate it.

Integrating device profiling

You can optionally send a device profile to PingOne Risk.

About this task

You can find detailed descriptions of each method in Device profiling methods.

Steps

- To send a device profile to PingOne Risk, do one of the following:

Method	Steps
Direct authentication mode: Captured by the PingOne Risk IdP Adapter	No extra steps are necessary. Continue to Configuring an adapter instance.

Copyright © 2025 Ping Identity Corporation

Method	Steps
Direct authentication mode: Captured by a previous adapter, such as the Identifier First adapter	**Complete the steps in Adding device profiling to a browser-based authentication page.**
Authentication API mode: Captured by your web application first-factor sign-on page	**Complete the steps in Adding device profiling to a browser-based authentication page.**

> **Tip**
>
> Learn more in PingFederate authentication API and Exploring the Authentication API in the PingFederate documentation.

Adding device profiling to a browser-based authentication page

There are two ways for device profiling data to be collected: by the PingOne Risk IdP Adapter, or by a previous adapter such as the HTML Form Adapter.

About this task

Getting the device profile on the HTML Form Adapter page reduces the perceived wait times for the user. Learn more in Device profiling methods.

Adding device profiling to a browser-based authentication page using the PingOne Risk (Signals) SDK with integration kit 1.3.1

About this task

You can adapt these instructions to add device profiling to any page, such as the HTML Form Adapter or your web application. The page must meet the criteria listed in Device profiling methods.

> **Important**
>
> PingOne Risk Integration Kit version 1.3.1 must be deployed before any changes can be made on the HTML side. SDK version 5.2 and later requires adapter version 1.3.1 or later.

Steps

1. **If you are modifying an external web application, copy the following files from the integration `.zip` archive to a location that your page can access.**
    - `pingone-risk-profiling-signals-sdk.js`
    - `pingone-risk-management-embedded.js`
    - `signals-sdk-<version>.js`
2. **Optional: Edit the `pingone-risk-profiling-signals-sdk.js` file and add your PingOne environment ID.**

Copyright © 2025 Ping Identity Corporation

```
function profileDevice(callback) {
    // Initialize the SDK
    // replace <envid> with the PingOne console > Environment > Environment ID value
    onPingOneSignalsReady(function () {
        _pingOneSignals.initSilent({
             envId: "<envid>",
            behavioralDataCollection: false,
            deviceAttributesBlackList: []
        }).then(function ()
```

3. Add the following external script references to the sign-on page:

> **Important**
>
> The scripts must be added in the following order.

```
<script type="text/javascript" src="signals-sdk-<version>.js"></script>
<script type="text/javascript" src="pingone-risk-profiling-signals-sdk.js"></script>
<script type="text/javascript" src="pingone-risk-management-embedded.js"></script>
```

4. Optional: Customize the device profile cookie name prefix to suit your environment.
    1. Open `pingone-risk-management-embedded.js` for editing.
    2. On the following line, change the value to a name of your choosing:

    ```
    var cookieNamePrefix = "pingone.risk.device.profile";
    ```

    3. Save the file.
5. When you complete the steps in Configuring an adapter instance, follow the instructions to set the Device profiling method to Captured by a previous adapter. Update the Cookie Name Prefix field if you customized it above.

Adding device profiling to a browser-based authentication page using the PingOne Risk (Signals) SDK and integration kit 1.3 or earlier

About this task

You can adapt these instructions to add device profiling to any page, such as the HTML Form Adapter or your web application. The page must meet the criteria listed in Device profiling methods.

Steps

1. If you are modifying an external web application, copy the following files from the integration `.zip` archive to a location that your page can access.
    - `pingone-risk-profiling-signals-sdk.js`

Copyright © 2025 Ping Identity Corporation

- `pingone-risk-management-embedded.js`
- `signals-sdk.js`

2. Edit the `pingone-risk-profiling-signals-sdk.js` file and add your PingOne environment ID.

> **Important**
>
> The PingOne Risk SDK won't work if the PingOne environment ID is missing.

```
function profileDevice(callback) {
    // Initialize the SDK
    // replace <envid> with the PingOne console > Environment > Environment ID value
    onPingOneSignalsReady(function () {
        _pingOneSignals.initSilent({
             envId: "<envid>",
            deviceAttributesBlackList: []
        }).then(function ()
```

3. Add the following external script references to the sign-on page:

> **Important**
>
> The scripts must be added in the following order.

```
<script type="text/javascript" src="signals-sdk.js"></script>
<script type="text/javascript" src="pingone-risk-profiling-signals-sdk.js"></script>
<script type="text/javascript" src="pingone-risk-management-embedded.js"></script>
```

4. Optional: Customize the device profile cookie name prefix to suit your environment.
    1. Open `pingone-risk-management-embedded.js` for editing.
    2. On the following line, change the value to a name of your choosing:

    ```
    var cookieNamePrefix = "pingone.risk.device.profile";
    ```

    3. Save the file.
5. When you complete the steps in Configuring an adapter instance, follow the instructions to set the Device profiling method to Captured by a previous adapter. Update the Cookie Name Prefix field if you customized it above.

Adding device profiling to a browser-based authentication page using Fingerprint JS

About this task

Copyright © 2025 Ping Identity Corporation

You can adapt these instructions to add device profiling to any page, such as the HTML Form Adapter or your web application. The page must meet the criteria listed in Device profiling methods.

> **Note**
>
> The PingOne Risk (Signals) SDK is the preferred way to get device profiling and is recommended for use in the PingOne Risk Integration Kit 1.3 and later.

Steps

1. If you are modifying an external web application, copy the following files from the integration `.zip` archive to a location that your page can access.
   - `fingerprint2-<version>.min.js`
   - `pingone-risk-management-profiling.js`
   - `pingone-risk-management-embedded.js`
2. Add the following external script references to the sign-on page.

   > **Important**
   >
   > The scripts must be added in the following order:

   ```
   <script type="text/javascript" src="fingerprint2-<version>.min.js"></script>
   <script type="text/javascript" src="{file-prefix}-profiling.js"></script>
   <script type="text/javascript" src="{file-prefix}-embedded.js"></script>
   ```
3. Edit the `pingone-risk-management-embedded.js` file to use Fingerprint JS according to the comments in the file.
4. Optional: Customize the device profile cookie name prefix to suit your environment.
   1. Open `pingone-risk-management-embedded.js` for editing.
   2. On the following line, change the value to a name of your choosing:

      ```
      var cookieNamePrefix = "pingone.risk.device.profile";
      ```
   3. Save the file.
5. When you complete the steps in Configuring an adapter instance, follow the instructions to set the Device profiling method to Captured by a previous adapter. Update the Cookie Name Prefix field if you customized it above.

Adding device profiling to an authentication API-based application

If you are using the PingFederate authentication application programming interface (API) and want to avoid using HTTP cookies, modify your application to collect the device profile and send it to the authentication API.

Copyright © 2025 Ping Identity Corporation

Adding device profiling to an authentication API-based application using the PingOne Risk (Signals) SDK and integration kit 1.3.1

About this task

Use the following instructions to add device profiling to a web application using the PingOne Risk (Signals) SDK. Learn more about the authentication API in Authentication API in the PingFederate documentation or PingOne Risk Native SDKs in the PingOne Native SDK documentation.

> **Important**
>
> PingOne Risk Integration Kit version 1.3.1 must be deployed before any changes can be made on the HTML side. SDK version 5.2.1 or later requires adapter version 1.3.1 or later.

Steps

1. Implement the Signals SDK by following the steps in PingOne Risk Native SDKs.
2. Configure functions to format the device profile and submit it to the authentication API. Use the following code sample as a guide.

   > **Note**
   >
   > The `pingone-risk-profiling-signals-sdk.js` assumes you are using the web SDK. If using a mobile device and you want to send the `deviceProfile` check the SDK documentation and use accordingly.

Copyright © 2025 Ping Identity Corporation

```
function onCompletion(flowId, deviceProfile) {
        submitDeviceProfile(flowId, deviceProfile);
}

function submitDeviceProfile(flowId, deviceProfile) {
        var myHeaders = new Headers();
        myHeaders.append("X-XSRF-Header", "test");
        myHeaders.append("Content-Type", "application/vnd.pingidentity.submitDeviceProfile+json");

        var raw = JSON.stringify({
        "signalsSdkDeviceProfile": deviceProfile
        });

        var requestOptions = {
        method: 'POST',
        headers: myHeaders,
        body: raw,
        redirect: 'follow'
        };

        fetch("https://pf_host:pf_port/pf-ws/authn/flows/" + flowId, requestOptions)
        .then(response => response.text())
        .then(result => console.log(result))
        .catch(error => console.log('error', error));
}
```

When the authentication API is in the `DEVICE_PROFILE_REQUIRED` state, your application should `submitDeviceProfile` with the value from the `profileDevice(callback)` method.

3. When you complete the steps in Configuring an adapter instance, set the Device profiling method to Captured by previous adapter.

Adding device profiling to an authentication API-based application using the PingOne Risk (Signals) SDK and integration kit 1.3 or earlier

About this task

Use the following instructions to add device profiling to a web application using the PingOne Risk (Signals) SDK. Learn more about the authentication API in Authentication API in the PingFederate documentation or PingOne Risk Native SDKs in the PingOne Native SDK documentation.

Steps

1. Implement the Signals SDK by following the steps in PingOne Risk Native SDKs.
2. Configure functions to format the device profile and submit it to the authentication API. Use the following code sample as a guide.

Copyright © 2025 Ping Identity Corporation

> **Note**
>
> The `pingone-risk-profiling-signals-sdk.js` assumes you are using the web SDK. If using a mobile device and you want to send the `deviceProfile` check the SDK documentation and use accordingly.

```
function onCompletion(flowId, deviceProfile) {
        submitDeviceProfile(flowId, deviceProfile);
}

function submitDeviceProfile(flowId, deviceProfile) {
        var myHeaders = new Headers();
        myHeaders.append("X-XSRF-Header", "test");
        myHeaders.append("Content-Type", "application/vnd.pingidentity.submitDeviceProfile+json");

        var raw = JSON.stringify({
        "signalsSdkDeviceProfile": deviceProfile
        });

        var requestOptions = {
        method: 'POST',
        headers: myHeaders,
        body: raw,
        redirect: 'follow'
        };

        fetch("https://pf_host:pf_port/pf-ws/authn/flows/" + flowId, requestOptions)
        .then(response => response.text())
        .then(result => console.log(result))
        .catch(error => console.log('error', error));
}
```

When the authentication API is in the `DEVICE_PROFILE_REQUIRED` state, your application should `submitDeviceProfile` with the value from the `profileDevice(callback)` method.

3. When you complete the steps in Configuring an adapter instance, set the Device profiling method to Captured by previous adapter.

Adding device profiling to an Authentication API-based application using Fingerprint JS

About this task

Use the following instructions to add device profiling to a web application using Fingerprint JS. Learn more about the authentication API in Authentication API in the PingFederate documentation.

> **Note**
>
> The PingOne Risk (Signals) SDK is the preferred way to get device profiling and is recommended for use in the PingOne Risk Integration Kit 1.3 and later.

Copyright © 2025 Ping Identity Corporation

Steps

1. Copy the following files from the integration `.zip` archive to your web server.
    - `fingerprint2-<version>.min.js`
    - `pingone-risk-management-profiling.js`
2. Add the following external script references to the sign-on page:

> **Important**
>
> The scripts must be added in the following order.

```
<script type="text/javascript" src="fingerprint2-<version>.min.js"></script>
<script type="text/javascript" src="{file-prefix}-profiling.js"></script>
```

3. Configure your application to call the `profileDevice(onCompletion)` function. This creates the device profile.
4. Configure functions to format the device profile and submit it to the authentication API. Use the following code sample as a guide.

```
function onCompletion(flowId, components) {
    var deviceProfile = transformComponentsToDeviceProfile(components);
    submitDeviceProfile(flowId, deviceProfile);
}

function submitDeviceProfile(flowId, deviceProfile) {
    var myHeaders = new Headers();
    myHeaders.append("X-XSRF-Header", "test");
    myHeaders.append("Content-Type", "application/vnd.pingidentity.submitDeviceProfile+json");

    var raw = JSON.stringify({
        "deviceProfile": deviceProfile
    });

    var requestOptions = {
        method: 'POST',
        headers: myHeaders,
        body: raw,
        redirect: 'follow'
    };

    fetch("https://pf_host:pf_port/pf-ws/authn/flows/" + flowId, requestOptions)
        .then(response => response.text())
        .then(result => console.log(result))
        .catch(error => console.log('error', error));
}
```

When the authentication API is in the `DEVICE_PROFILE_REQUIRED` state, your application should `submitDeviceProfile` with a value structured as follows:

Copyright © 2025 Ping Identity Corporation

```
{
   "deviceProfile":{
      "userAgent":"Mozilla/5.0 (Macintosh; Intel Mac OS X 10.14; rv:82.0) Gecko/20100101 Firefox/82.0",
      "language":"en-CA",
      "colorDepth":24,
      "deviceMemory":null,
      "hardwareConcurrency":12,
      "screenResolution":[
         3440,
         1440
      ],
      "availableScreenResolution":[
         3440,
         1417
      ],
      "timezoneOffset":480,
      "timezone":"America/Los_Angeles",
      "sessionStorage":true,
      "localStorage":true,
      "indexedDb":true,
      "addBehaviour":false,
      "openDatabase":false,
      "cpuClass":null,
      "platform":"MacIntel",
      "plugins":[
      ],
      "webgl":[
         "
                [...]
                WEBGL_draw_buffers;WEBGL_lose_context",
         "webgl aliased line width range:[1, 1]",
         "webgl aliased point size range:[1, 8191]",
         [...]
         "webgl fragment shader low int precision rangeMin:24",
         "webgl fragment shader low int precision rangeMax:24"
      ],
      "webglVendorAndRenderer":"ATI Technologies Inc.~AMD Radeon Pro 560X OpenGL Engine",
      "adBlock":false,
      "hasLiedLanguages":false,
      "hasLiedResolution":false,
      "hasLiedOs":false,
      "hasLiedBrowser":false,
      "touchSupport":[
         "0",
         "false",
         "false"
      ],
      "fonts":[
         "Andale Mono",
         "Arial",
         [...]
         "Wingdings",
         "Wingdings 2",
         "Wingdings 3"
      ],
      "audio":"35.7383295930922"
   }
}
```


Copyright © 2025 Ping Identity Corporation

5. When you complete the steps in Configuring an adapter instance, set the Device profiling method to Captured by this adapter.

Configuring an adapter instance

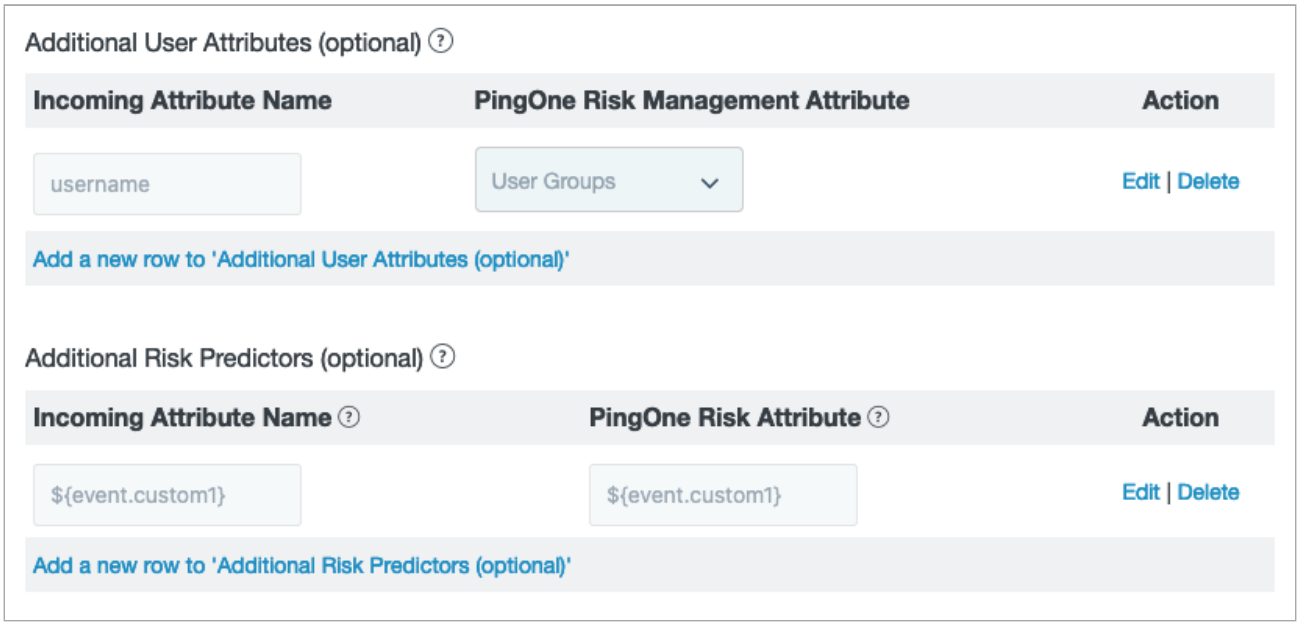
To get started with the integration, deploy the PingOne Risk Integration Kit files to your PingFederate directory.

Steps

1. In the PingFederate administrative console, go to Authentication > Integration > IdP Adapters. Click Create New Instance.
2. On the Type tab, set the basic adapter instance attributes:
    1. In the Instance Name field, enter a name for the adapter instance.
    2. In the Instance ID field, enter a unique identifier for the adapter instance.
    3. In the Type list, select PingOne Risk IdP Adapter. Click Next.
3. On the IdP Adapter tab, in the Additional User Attributes section, configure additional information to send to PingOne Risk.
    1. Click Add a new row to 'Additional User Attributes'.
    2. In the Incoming Attribute Name field, enter the name of an attribute from any authentication source that appears earlier in your PingFederate authentication policy than the PingOne Risk IdP Adapter.
    3. In the PingOne Risk Attribute list, select the PingOne attribute that you want to populate.
    4. In the Action column, click Update.
    5. To add more attributes, repeat Steps a-d.
4. Optional: On the IdP Adapter tab, map any custom risk predictors.

    Learn more in Using custom risk predictors.
    1. Click Add a new row to 'Additional Risk Predictors (optional)'.
    2. In the Incoming Attribute Name field, enter the name of an attribute from any authentication source that appears earlier in your PingFederate authentication policy than the PingOne Risk IdP Adapter.
    3. In the PingOne Risk Attribute list, select the PingOne attribute that will be assessed in the risk policy, for example `$\{event.isManaged}`.
    4. In the Action column, click Update.
    5. To add more attributes, repeat steps a - d.

Copyright © 2025 Ping Identity Corporation



5. Optional: On the IdP Adapter tab, in the PingOne Risk API Response Mappings section, map attributes from PingOne Risk Evaluation API response to the attribute contract.

   These attributes become available in your PingFederate authentication policy.

   1. Click Add a new row to 'PingOne Risk API Response Mappings'.
   2. In the Local Attribute field, enter a name of your choosing for an attribute.
   3. In the PingOne Risk API Attribute Mapping field, enter the JSON Pointer syntax for the source PingOne attribute as shown in JSON pointer syntax reference.

      For example, the JSON pointer `/details/ipAddressReputation/level` will return the IP address repuation level, such as `LOW`.
   4. In the Action column, click Update.
   5. To add more attributes, repeat steps a-d.
6. On the IdP Adapter tab, configure the adapter instance by referring to PingOne Risk IdP adapter settings reference. Click Next.
7. On the Actions tab, test your connection to PingOne Risk. Resolve any issues that are reported, and then click Next.
8. On the Extended Contract tab, add any attributes that you included in the PingOne Risk API Response Mappings section of the IdP Adapter tab. Click Next.
9. On the Adapter Attributes tab, set pseudonym and masking options as shown in Set pseudonym and masking options in the PingFederate documentation. Click Next.
10. On the Adapter Contract Mapping tab, configure the contract fulfillment details for the adapter as shown in Define the IdP adapter contract in the PingFederate documentation. Click Next.
11. On the Summary tab, check and save your configuration. Click Save.

Copyright © 2025 Ping Identity Corporation

JSON pointer syntax reference

JavaScript Object Notation (JSON) Pointer defines a syntax for identifying a specific value within a JSON payload. Using the sample payload and JSON Pointer examples below, identify the attributes that you want to use to populate your attribute contract.

You can find a complete technical description of JSON Pointer syntax in JavaScript Object Notation (JSON) Pointer on ietf.org.

Copyright © 2025 Ping Identity Corporation

Example PingOne Risk JSON payload

Copyright © 2025 Ping Identity Corporation

```
{
  "_links":{
    "self":{
      "href":"https://api.pingone.com/v1/environments/fe561fe9-e374-4de8-be4c-0e38b3b83356/riskEvaluations/
8d482f35-4ae5-4ecc-b9ff-db696a01f39f"
    },
    "environment":{
      "href":"https://api.pingone.com/v1/environments/fe561fe9-e374-4de8-be4c-0e38b3b83356"
    },
    "event":{
      "href":"https://api.pingone.com/v1/environments/fe561fe9-e374-4de8-be4c-0e38b3b83356/riskEvaluations/
8d482f35-4ae5-4ecc-b9ff-db696a01f39f/event"
    }
  },
  "id":"8d482f35-4ae5-4ecc-b9ff-db696a01f39f",
  "environment":{
    "id":"fe561fe9-e374-4de8-be4c-0e38b3b83356"
  },
  "createdAt":"2022-09-16T20:49:04.029Z",
  "updatedAt":"2022-09-16T20:49:04.029Z",
  "event":{
    "completionStatus":"IN_PROGRESS",
    "ip":"209.133.96.106",
    "flow":{
      "type":"AUTHENTICATION"
    },
    "user":{
      "id":"john",
      "name":"John DeMock",
      "type":"EXTERNAL",
      "groups":[
        {
          "name":"dev"
        },
        {
          "name":"sre"
        }
      ]
    },
    "origin":"PF_RISK_IK",
    "customProp":{
      "":{
        "nested":[
          "abcd",
          3.14
        ]
      }
    },
    "myNumber":8.1,
    "isManaged":"No"
  },
  "riskPolicySet":{
    "id":"368a8459-3f46-4301-bc1e-f4214eb4a5f5",
    "name":"IK-2406"
  },
  "result":{
    "value":"Unmanaged device!",
    "level":"HIGH",
    "score":1.8918918918918919,
    "source":"OVERRIDE",
```


Copyright © 2025 Ping Identity Corporation

```
      "type":"VALUE"
    },
    "details":{
      "ipAddressReputation":{
        "score":0,
        "level":"LOW"
      },
      "anonymousNetworkDetected":false,
      "country":"united states",
      "state":"georgia",
      "city":"atlanta",
      "impossibleTravel":false,
      "ipVelocityByUser":{
        "level":"LOW",
        "threshold":{
          "source":"MIN_NOT_REACHED"
        },
        "velocity":{
          "distinctCount":1,
          "during":3600
        },
        "type":"VELOCITY"
      },
      "userLocationAnomaly":{
        "level":"HIGH",
        "reason":"New transaction location",
        "status":"NEW_LOCATION",
        "type":"USER_LOCATION_ANOMALY"
      },
      "userVelocityByIp":{
        "level":"LOW",
        "threshold":{
          "source":"MIN_NOT_REACHED"
        },
        "velocity":{
          "distinctCount":1,
          "during":3600
        },
        "type":"VELOCITY"
      },
      "geoVelocity":{
        "level":"LOW",
        "type":"GEO_VELOCITY"
      },
      "userRiskBehavior":{
        "reason":"Not enough information to assess risk score",
        "status":"NOT_AVAILABLE",
        "type":"USER_RISK_BEHAVIOR"
      },
      "ipRisk":{
        "level":"LOW",
        "type":"IP_REPUTATION"
      },
      "userBasedRiskBehavior":{
        "reason":"Not enough information to assess risk score",
        "status":"IN_TRAINING_PERIOD",
        "type":"USER_RISK_BEHAVIOR"
      },
      "anonymousNetwork":{
        "level":"LOW",
        "type":"ANONYMOUS_NETWORK"
      },
```


Copyright © 2025 Ping Identity Corporation

```
      "myNumber":{
        "level":"HIGH",
        "reason":"Attribute ${event.myNumber} is 8.1.",
        "attribute":"${event.myNumber}",
        "value":8.1,
        "type":"MAP"
      },
      "nested":{
        "reason":"Not enough information to assess risk score",
        "status":"NOT_AVAILABLE",
        "attribute":"${event.customProp.nested}",
        "type":"MAP"
      },
      "isManaged":{
        "level":"HIGH",
        "reason":"Attribute ${event.isManaged} is \"No\".",
        "attribute":"${event.isManaged}",
        "value":"No",
        "type":"MAP"
      }
    }
  }
```

JSON Pointer syntax

Description	JSON Pointer	Example value
Minimum Risk Module Policy Score	`/details/policies/MIN_RISK_MODULE/score`	`501`
Minimum Risk Module Policy Level	`/details/policies/MIN_RISK_MODULE/level`	`MEDIUM_RISK`

> **Note**
>
> To populate an attribute with the entire JSON response, leave the **PingOne Risk API Attribute Mapping** field blank.

PingOne Risk IdP adapter settings reference

Field descriptions for the PingOne Risk IdP Adapter configuration screen.

Standard fields

Field	Value	Description
PingOne Environment	*<PingOne Connection>*	**Your PingOne Environment. Create connections in System > External Systems > PingOne Connections. This field is blank by default.**

Copyright © 2025 Ping Identity Corporation

Field	Value	Description
PingOne Risk Policy	*<Default Risk Policy>*	The risk policy used by PingOne for the risk evaluation. Overrides the environment and global default policy selections. This list is populated when you select a PingOne Environment.

Advanced fields

Field	Value	Description
Include Device Profile		When enabled, PingFederate will include a device profile in the risk evaluation. This option is selected by default.
Device Profiling Method	• Captured by this adapter (default) • Captured by a previous adapter	When Include Device Profile is enabled, this setting determines whether the adapter will collect the device profile itself, or look for a device profile provided by another adapter that appears earlier in the authentication flow. If you manually added the device profiling script to an authentication page, select Captured by a previous adapter. Otherwise, select Captured by this adapter.
Device Profiling Timeout	`5000` (default)	The amount of time in milliseconds that PingFederate waits for the device profiling script to collect device details. Applies only if Device Profiling Method is Captured by this adapter.
Cookie Name Prefix	`pingone.risk.device.profile`	The name prefix of the cookie(s) that contains the device profile captured by a previous adapter. Applies only if Device Profiling Method is Captured by a previous adapter. The default value is `pingone.risk.device.profile`.

Copyright © 2025 Ping Identity Corporation

Field	Value	Description
Failure Mode	• Continue with fallback policy decision • Fail (default)	When PingOne Risk is unavailable or an error occurs, this setting determines whether the user's sign-on attempt should fail or continue with a pre-determined policy decision.
Fallback Policy Decision Value	`MEDIUM` (default)	The risk result (for example, `MEDIUM`, `HIGH`, or unknown) to use in the authentication policy when the PingOne Risk service is unavailable or an error occurs, and Failure Mode is set to Continue with fallback policy decision.
Risk Policy Override		The attribute that contains the Risk policy ID to use when doing evaluations. This field is blank by default.
Device Profiling Script	• Fingerprint JS • Signals SDK (default)	The script used to gather the client device profile.
API Request Timeout	`2000`	The amount of time in milliseconds that PingFederate allows when establishing a connection with PingOne Risk or waiting for a response to a request. A value of `0` disables the timeout. The default value is `2000`.
Proxy Settings		Defines proxy settings for outbound HTTP requests. The default value is System Defaults.
Custom Proxy Host		The proxy server host name to use when Proxy Settings is set to Custom. This field is blank by default.
Custom Proxy Port		The proxy server port to use when Proxy Settings is set to Custom. This field is blank by default.

Copyright © 2025 Ping Identity Corporation

Using custom risk predictors

The PingOne Risk IdP Adapter lets you use attributes from your PingFederate authentication flow as risk predictors in PingOne Risk.

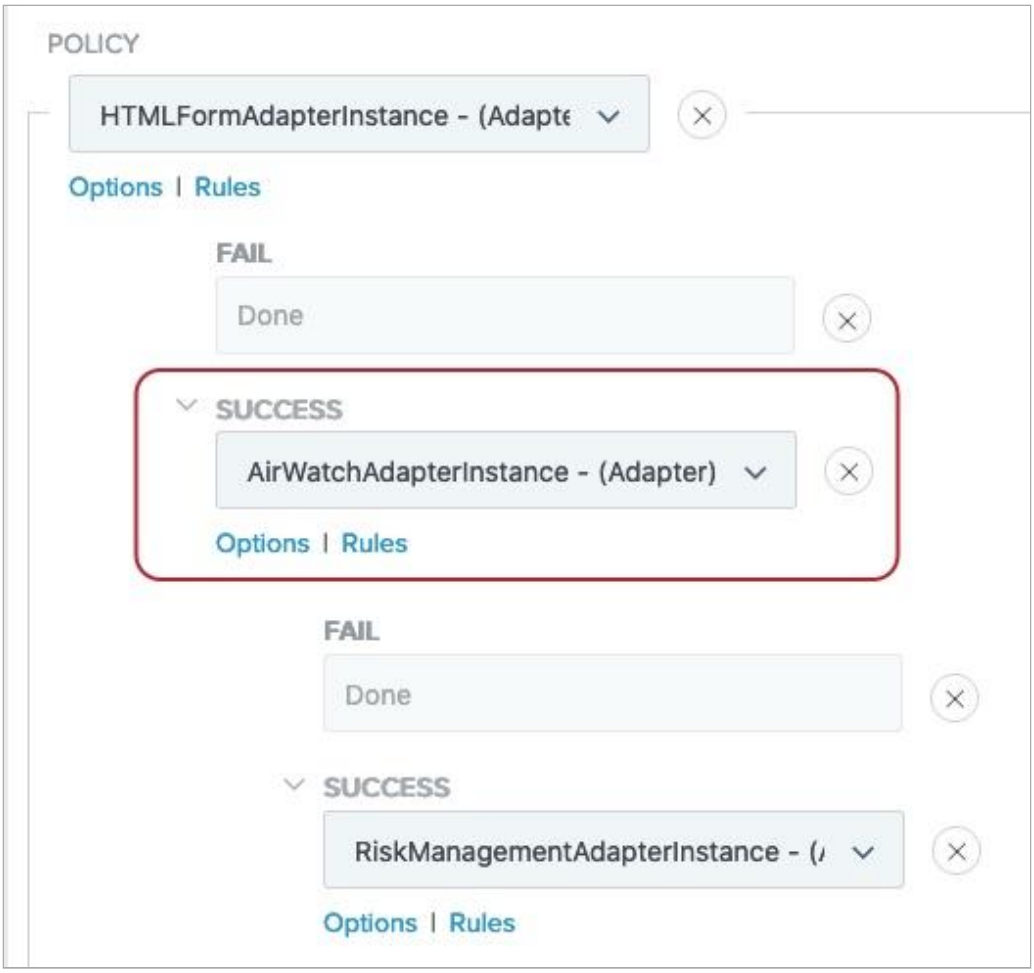
About this task

You can find an overview of risk predictors in Predictors in the PingOne Risk documentation.

The following steps provide an example that shows how to include the device security state from a mobile device management (MDM) service in the PingOne Risk risk evaluation.

Steps

1. Make the predictor available as an attribute in your PingFederate authentication policy.



    1. Add the source of the predictor data to your authentication policy.

       MDM example: Add a mobile device management adapter. On the Extended Contract tab of the configuration for that adapter instance, the attribute that holds the security state of the user's device is called `ComplianceStatus`.

    2. Later in the flow, add the PingOne Risk IdP Adapter that you configured in Configuring an adapter instance.

2. In PingOne Risk, add the risk predictor and include it in your risk policy.

Copyright © 2025 Ping Identity Corporation

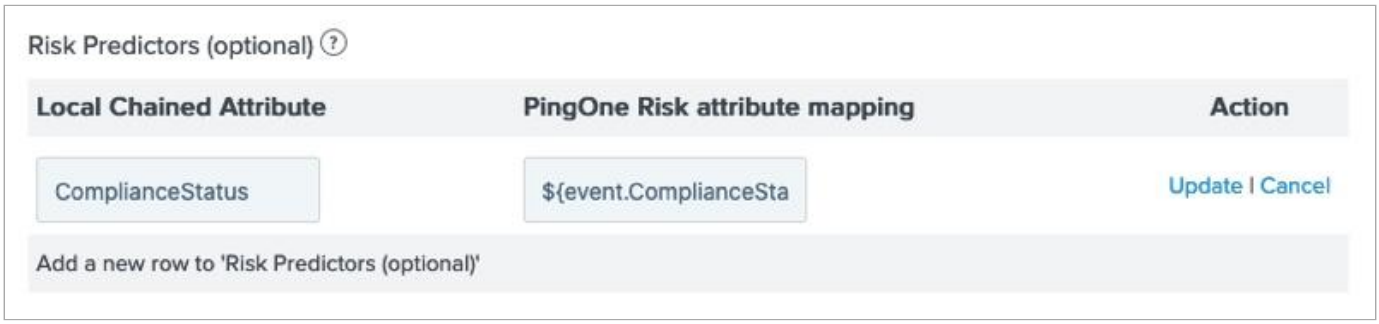
Learn more in Predictors in the PingOne Risk documentation.

MDM example: Add a predictor with the JSON pointer `$\{event.ComplianceStatus}`.

3. In the Risk Predictors table of your PingOne Risk IdP Adapter configuration, map the predictor attribute from your PingFederate authentication policy to the JSON pointer that you defined in PingOne Risk.

   Learn more in Configuring an adapter instance.

   MDM example: Map the PingFederate `ComplianceStatus` attribute to the PingOne Risk predictor in your PingOne Risk IdP Adapter configuration.



   During the authentication flow, the PingOne Risk IdP Adapter gets the predictor attribute from the PingFederate authentication policy and passes it to PingOne Risk. Next, PingOne Risk compares the value to the risk levels that you defined and includes it in the risk evaluation.

Adding risk level results to your authentication policy

By modifying your PingFederate authentication policy to include the Result from PingOne Risk, you can dynamically change authentication requirements based on security risk level.

Before you begin

The steps in this topic assume that an HTML Form Adapter exists for login purposes. Learn more about creating an HTML Form Adapter for login in Configuring an HTML Form Adapter instance.

About this task

These steps are designed to help you add to an existing authentication policy. Learn more about configuring authentication policies in PingFederate authentication API in the PingFederate documentation.

For new deployments, you should allow for a training period. To do this, configure your policy to pass traffic through the PingOne Risk IdP Adapter and continue regardless of the Result result. When you are ready to end the training period, adjust your authentication policy as described here.

> **Important**
>
> When the authentication flow finishes, PingFederate informs PingOne Risk whether the user ultimately succeeded or failed. This is an important consideration when designing your authentication flow.
> For example, a user receives a Result of `HIGH`, but ultimately completes the PingFederate authentication policy successfully. Based on that success, PingOne Risk now considers the user authenticated and lowers the Result to `MEDIUM` or `LOW` on the next attempt.

Copyright © 2025 Ping Identity Corporation

Steps

1. In the PingFederate administrative console, go to the Policies tab.

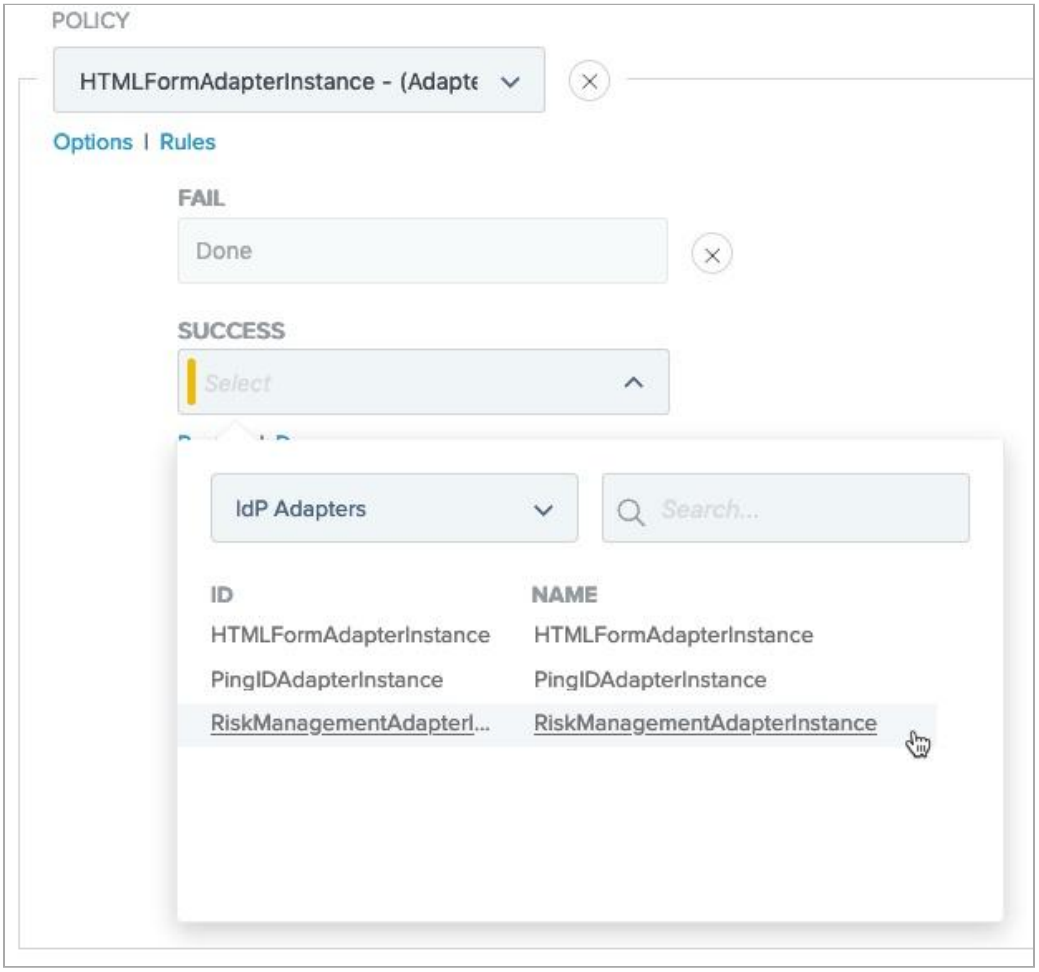
   Choose from:

   - For PingFederate 10.1 or later: go to Authentication > Policies > Policies.
   - For PingFederate 10.0 or earlier: go to Identity Provider > Authentication Policies > Policies.

2. Select the IdP Authentication Policies checkbox.
3. Open an existing authentication policy, or click Add Policy.

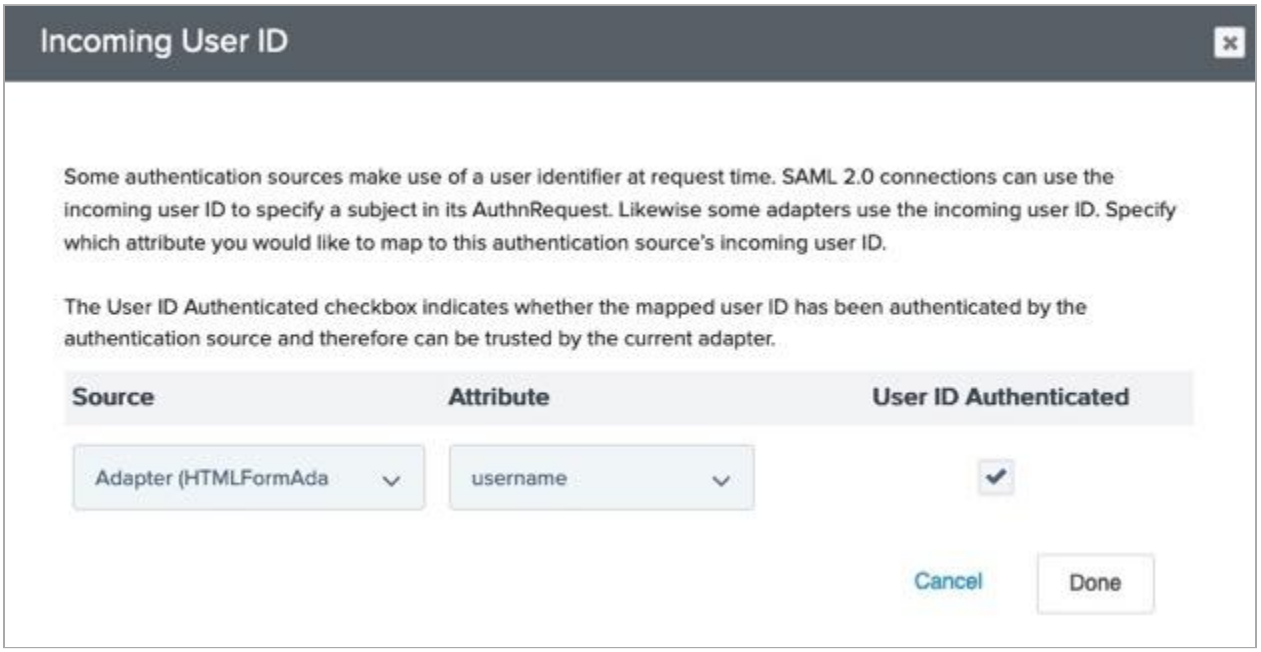
   Learn more in Defining authentication policies in the PingFederate documentation.

4. In the Policy area, in the Select list, select a PingOne Risk IdP Adapter instance.



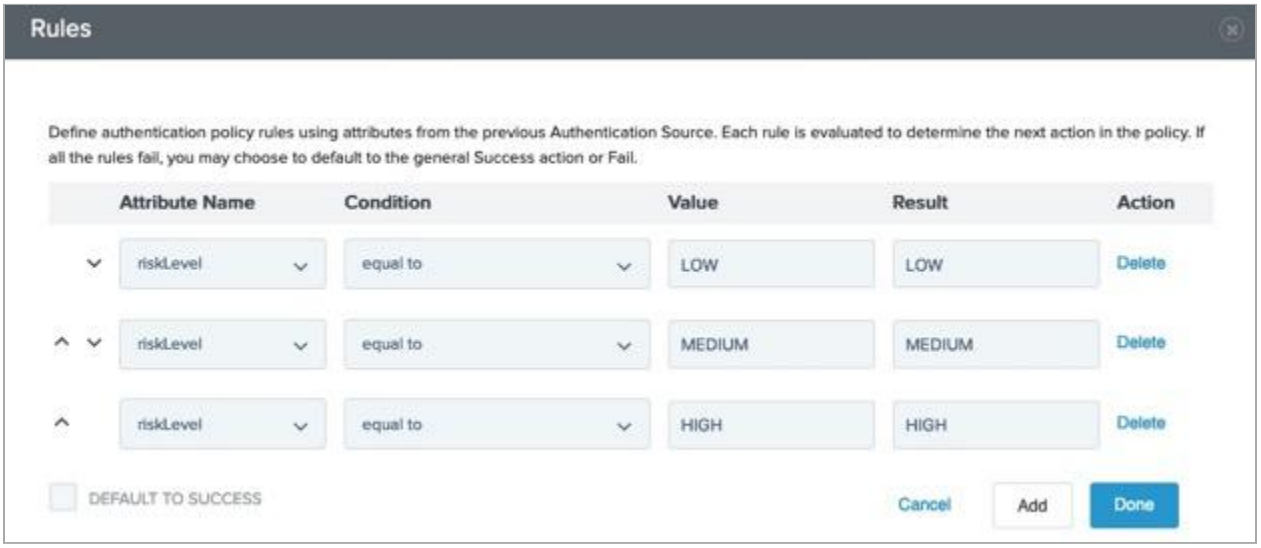
5. Map the user ID into the PingOne Risk IdP Adapter instance.

Copyright © 2025 Ping Identity Corporation



1. Under the PingOne Risk IdP Adapter instance, click Options.
2. In the Options dialog, in the Source list, select a previous authentication source that collects the user ID.
3. In the Attribute list, select the user ID. Click Done.

6. Define policy paths based on risk results.



1. Under the PingOne Risk IdP Adapter instance, click Rules.
2. In the Rules dialog, in the Attribute Name list, select riskLevel or riskValue.
3. In the Condition list, select equal to.
4. In the Value field, if you selected riskLevel, enter `LOW`, `MEDIUM`, or `HIGH`.

   If you selected riskValue, enter one of the risk values that you configured in PingOne.

Copyright © 2025 Ping Identity Corporation

5. In the Result field, enter a name.

   This appears as a new policy path that branches from the authentication source.

6. Optional: To add more policy paths, click Add and repeat steps 6b-6e.

7. Optional: Clear the Default to success check box.

8. Click Done.

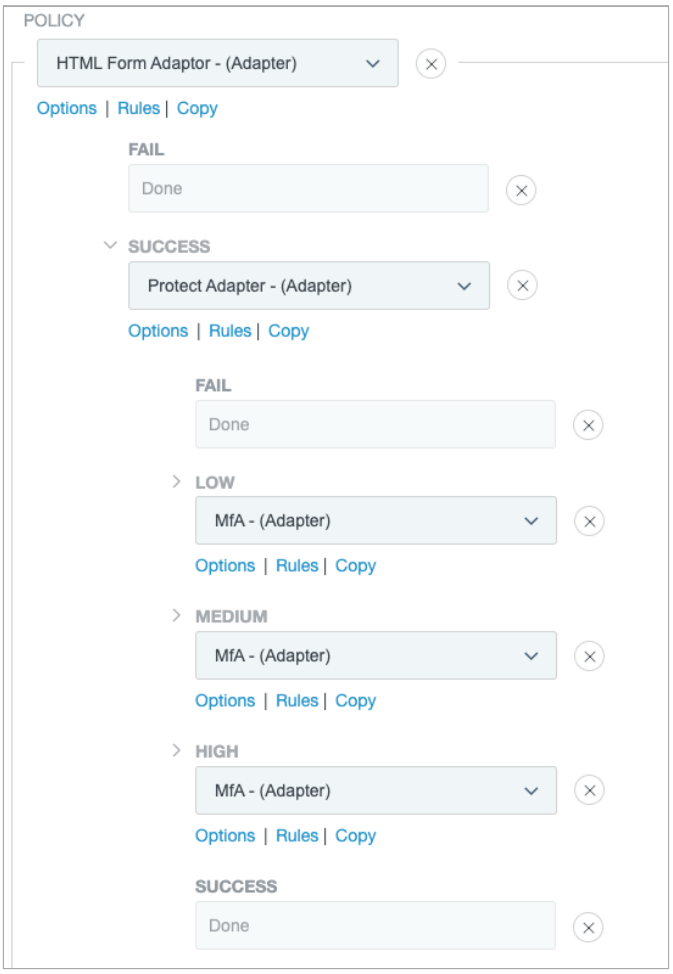
7. Complete the authentication policy:

   1. Configure each of the policy paths.

   2. Optional: To allow users to continue to sign on by satisfying stricter authentication requirements when PingOne Risk is unreachable or returns an error, do one of the following:

      Choose from:

      - In your PingOne Risk IdP Adapter instance, set the Failure mode as shown in PingOne Risk IdP adapter settings reference.
      - In your authentication policy, set the Fail outcome of the PingOne Risk IdP Adapter instance to point to a second authentication factor, as shown in the following image.



8. Click Done.

Copyright © 2025 Ping Identity Corporation

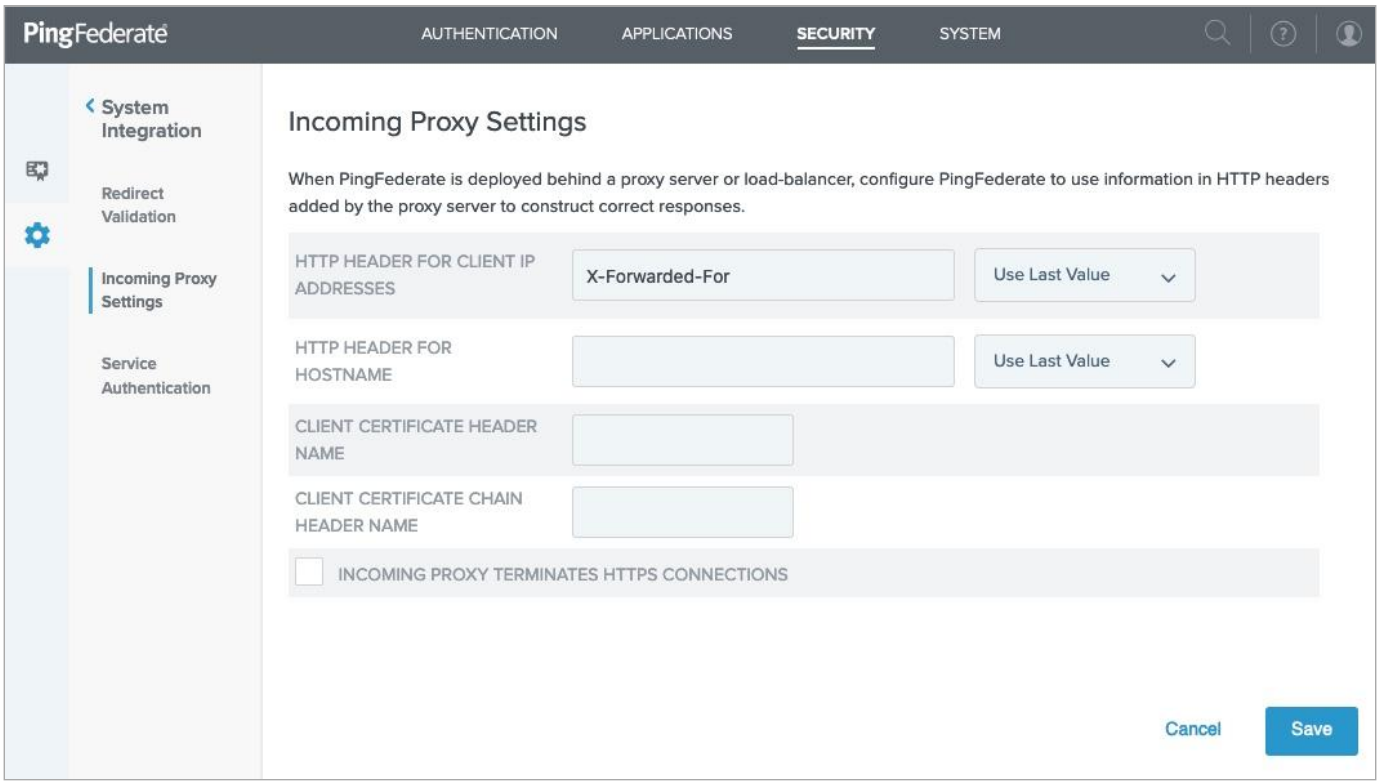
9. In the Policies window, click Save.

Configuring PingFederate to forward IP addresses

The PingOne Risk service needs the client's public-facing (rather than local) IP address for the risk assessment.

Steps

1. On the PingFederate administrative console, go to Security > System Integration > Incoming Proxy Settings.
2. In the HTTP Header for Client IP Addresses field, enter `X-Forwarded-For`.
3. Leave Use Last Value selected.
4. Click Save.



Troubleshooting

Enabling debug logging

To help with troubleshooting or monitoring, you can turn on activity logging for PingFederate, the PingOne Risk IdP Adapter, or both.

About this task

This task is optional. You can use logging for troubleshooting or analytics.

Copyright © 2025 Ping Identity Corporation

Learn more about logging in Enabling debug messages and console logging in the PingFederate documentation.

Steps

1. Open the `<pf_install>/pingfederate/server/default/conf/log4j2.xml` file for editing.
2. To log activity for PingFederate and all adapters:
    1. Find the following section in the file.

```
<AsyncRoot level="INFO" includeLocation="false">
        <!-- <AppenderRef ref="CONSOLE" /> -->
        <AppenderRef ref="FILE" />
</AsyncRoot>
```

    2. Change `INFO` to `DEBUG`.

       The following code snippet shows `DEBUG` in bold for visibility.

```
<AsyncRoot level="DEBUG" includeLocation="false">
        <!-- <AppenderRef ref="CONSOLE" /> -->
        <AppenderRef ref="FILE" />
</AsyncRoot>
```

    3. Optional: To see the adapter activity in the console as well as the log file, remove the comment tags (`<!--` and `-->`) that surround the `CONSOLE` line.

```
<AsyncRoot level="INFO" includeLocation="false">
        <AppenderRef ref="CONSOLE" />
        <AppenderRef ref="FILE" />
</AsyncRoot>
```

3. If you want to log activity relating to the PingOne Risk IdP Adapter, do one of the following:

> **Tip**
>
> You can use this information with a third-party log analysis tool to monitor for important events, such as when a sign-on event has a high-risk Result.

Choose from:

- To log activity for the PingOne Risk IdP Adapter as well as its HTTPS and component activity, add the following line.

```
<Logger name="com.pingidentity.adapters.pingone.risk" level="DEBUG"/>
```

- To log activity for the adapter's HTTPS activity and other components but not the adapter itself, add the following line.

Copyright © 2025 Ping Identity Corporation

```
<Logger name="com.pingidentity.adapters.pingone.risk.shade" level="DEBUG"/>
```

- To log activity for the PingOne Risk IdP Adapter but not its HTTPS or component activity, add the following lines.

```
<Logger name="com.pingidentity.adapters.pingone.risk" level="DEBUG"/>
<Logger name="com.pingidentity.adapters.pingone.risk.shade" level="INFO"/>
```

4. Save the file.

Troubleshooting information

The following information addresses technical situations that you might encounter after setting up the PingOne Risk Integration Kit.

Situation	Information
Include Device Profile is selected in the adapter configuration, but the device profile does not affect the risk result or appear in the response from PingOne Risk.	When this setting is enabled, it is possible for an error to prevent the device profile from reaching PingOne Risk. Because PingOne Risk considers the device profile to be optional, it still successfully returns a risk evaluation to the adapter. The adapter logs a warning in the PingFederate error log about the missing device profile and returns a "success" result to the authentication policy. As a result, the process succeeds but no device profile information is available. To address the problem generating or sending the device profile, review the steps in Integrating device profiling. Make sure you have completed the correct set of steps (authentication page vs. web application) and completed the steps exactly as described.

Release notes

Changelog

The following is the change history for the PingOne Risk Integration Kit.

PingOne Risk Integration Kit 1.3 — September 2022

- Changed the product name to PingOne Risk Integration Kit.
- Added the ability to set a Policy ID as a chained attribute and use it to set the risk policy.

Copyright © 2025 Ping Identity Corporation

- Added Signals SDK Javascript and mechanism to capture device profiling.
- Added support for Risk Score as an attribute returned by the adapter.
- Fixed an issue that caused page expired issues when the target application did not respond during the Risk Timeout.

PingOne Risk Integration Kit 1.2.1 — February 2022

- Fixed an issue that caused an error when users attempted the get risk predictors action.

PingOne Risk Integration Kit 1.2 — September 2021

- Added support for including third-party risk predictors in risk evaluations. Learn more in Using custom risk predictors.
- Improved the PingOne Risk Policy list by clarifying that the default selection uses the default PingOne Risk policy.
- Improved the clarity of the tooltips in the adapter configuration.
- Improved the device profiling script by excluding low-value attributes, such as `fonts`, `touchSupport`, `webgl` and `audio`. This reduces the size of the device profile cookie by up to 86% without impacting the quality of the risk evaluations.

PingOne Risk Integration Kit 1.1 — March 2021

- Added support for pre-populating adapter settings based on the selected PingOne environment. Available in PingFederate 10.2 or later.
- Fixed an issue that caused an error when the adapter received an unexpected attribute type.
- Fixed an issue that skipped device profiling in Internet Explorer 11 when Captured by this device was selected.

PingOne Risk Integration Kit 1.0 — December 2020

- Initial release.
- Added support for getting risk results and transaction information from PingOne Risk.
- Added the ability to send a device profile and transaction information to PingOne Risk.
- Added the ability to add device profiling into any browser-based authentication source.
- Added support for risk models based on user behavior, IP reputation, geovelocity, and anonymous network detection.
- Added the ability to map any PingOne Risk API response attribute to an attribute in the PingFederate authentication policy.
- Added authentication success/failure reporting to PingOne Risk.
- Added the ability to test the connection to PingOne Risk.
- Added support for the JavaScript Widget for the PingFederate Authentication API.
- Added support for the PingFederate authentication API.
- Added settings for API connection and request timeouts.
- Added settings to override the PingFederate system-default proxy settings.

Copyright © 2025 Ping Identity Corporation

Known issues and limitations

The following are known issues or limitations for the PingOne Risk Integration Kit.

Known issues

There are no known issues.

Known limitations

There are no known limitations.

Download manifest

The following files are included in the PingOne Risk Integration Kit `.zip` archive:

- `ReadMeFirst.pdf` – Contains links to this online documentation
- `Legal.pdf` – Copyright and license information
- `dist` – Contains the integration files
  - `deploy` – Contains the Java libraries
    - `pf-pingone-risk-management-adapter-<version>.jar` – JAR file that contains the PingOne Risk IdP Adapter.
  - `conf` – contains the HTML template that presents the PingOne sign-on form.
    - `language-packs` – Contains files with customizable user-facing messages
      - `pingone-risk-management-messages.properties` – a variable file that customizes the messages that appear on the template file.
    - `template` – contains user-facing HTML template files
      - `pingone-risk-management-template.html` – a sign-on redirect page used with the "captured by this adapter" device profiling method. Runs scripts to show a spinner animation and create the device profile.
      - `assets` – contains functional scripts and files used by the template
        - `css` – contains CSS files for the templates.
          - `pingone-risk-management.css` – a CSS file that customizes the appearance of the template files.
          - `end-user/<version>/end-user.css` – a CSS file that customizes the appearance of the template files.
        - `fonts/end-user/icons` – contains template icons

Copyright © 2025 Ping Identity Corporation

- `images` – contains template image files
  - `ping-logo.svg` – an image file with company branding
  - `spinner.svg` – an image file used in a spinner animation
- `scripts` – contains script files used to collect and send information
  - `fingerprint2-<version>.min.js` – a JavaScript script that collects information and creates a device profile. Used when device profiling is enabled.
  - `pingone-risk-management-profiling.js` – a JavaScript script that wraps the device profiling script. Used with both device profiling methods when device profiling is enabled.
  - `pingone-risk-management-embedded.js` – a JavaScript script that can be embedded in any authentication page. Initiates the device profiling script and stores the device profile in a cookie. Used with the "captured by a previous adapter" device profiling method.

- `lib/pf-authn-api-sdk-<version>.jar` – a JAR file that contains the PingFederate Authentication API SDK

PingOne Verify Integration Kit

The PingOne Verify Integration Kit allows PingFederate to use the PingOne Verify service for personal identity verification based on government photo ID.

Features

- Initiate an identity verification process as part of user registration or sign-on.
- Supports mobile web identity verification.
- Supports the PingFederate authentication API.
- Supports the JavaScript Widget for the PingFederate Authentication API.
- Supports version 2 of the PingOne Verify Mobile SDK with QR verification flows.

Components

PingOne Verify IdP Adapter

- Allows PingFederate to use PingOne Verify to trigger an identity verification challenge as part of the PingFederate authentication policy or registration flow.

Templates

- Allow the adapter to show the status of the verification process.
- Allow you to modify the appearance of pages shown to the user during authentication.

Copyright © 2025 Ping Identity Corporation

Language packs

- Allow you to customize or localize the messages returned by the PingFederate authentication application programming interface (API) and shown on the templates during authentication. Learn more in Localizing messages for end users in the PingFederate documentation.

Intended audience

This document is intended for PingFederate administrators.

If you need help during the setup process, learn more in the following resources:

- Introduction to PingOne Verify in the PingOne documentation
- The following sections of the PingFederate documentation:
  - Managing IdP adapters
  - Authentication policies

System requirements

- PingFederate 11.3 or later
- To allow PingFederate to make outbound HTTPS connections, you might need to allow the following host names in your firewall:
  - https://api.pingone.com, https://api.pingone.asia, or https://api.pingone.eu
  - https://auth.pingone.com, https://auth.pingone.asia, or https://auth.pingone.eu

Use cases

The following are common uses for the PingOne Verify Integration Kit.

Account Recovery

For password reset and sign-on flows, a malicious user could try to reset the password of an account other than their own by using their own ID.

To prevent this:

1. Set up a comparison in the PingFederate policy to check if the identity verified matches the user's record.
2. After the PingOne Verify Integration Kit succeeds, confirm that the first and last names of the verified ID match those in the directory for the username that was entered on the sign-on or forgot password page.

Authentication

When a user signs on, you can use PingOne Verify to verify a user's identity instead of, or in combination with, a standard multi-factor authentication (MFA) step.

Copyright © 2025 Ping Identity Corporation

For example, you might use PingOne MFA by default and only trigger PingOne Verify based on:

- Frequency: For example, annually.
- Group: For members of the "contractors" group.
- Account Age: For accounts made in the last two months.
- Past Verification: For users that have never completed the verification process.
- Security Risk: For users with a PingOne Protect risk evaluation of `MEDIUM` or `HIGH`.

You can create these rules in your PingFederate authentication policy.

Registration

When a person registers a new user account, you can use PingOne Verify to verify their identity against information from a government photo ID. For example, an address.

By requiring identity verification, you can reduce the number of accounts created by bots, spammers, and imposters.

Overview of the verification flow

With the PingOne Verify Integration Kit, PingFederate includes PingOne Verify in the sign-on or self-service registration flow.

Description

1. The user initiates single sign-on (SSO) or self-service registration with PingFederate. They complete the first authentication or self-registration step provided by an HTML Form Adapter instance, for example.
2. The PingOne Verify IdP Adapter contacts PingOne Verify to initiate the verification process.

   (Optional) If the user doesn't exist in PingOne, the adapter provisions the user.
3. PingOne Verify provides a verification code and a QR code image URL.
4. The PingOne Verify IdP Adapter presents an HTML page that shows the verification code and QR code.
5. The user follows the prompts to complete the verification process.

   This typically involves scanning their government-issued photo ID and submitting a live face capture.
6. The PingOne Verify IdP Adapter polls PingOne Verify while it waits for the result of the verification process.
7. PingOne Verify provides the adapter with the result of the verification process.
8. (Optional) The PingOne Verify IdP Adapter presents an HTML page that shows the success or error message, depending on the verification result.
9. If the user completed the verification process successfully, PingFederate provides access to the requested resource or completes the registration process.

Setup

Copyright © 2025 Ping Identity Corporation

Deploying the integration files

To get started with the integration, deploy the PingOne Verify Integration Kit files to your PingFederate directory.

Steps

1. Download the PingOne Verify Integration Kit `.zip` archive from the Ping Identity Integration Directory.
2. Stop PingFederate.
3. If you're upgrading an existing deployment, back up your customizations and delete earlier versions of the integration files:
    1. Back up any PingOne Verify Integration Kit files that you customized in the `<pf_install>/pingfederate/server/default/conf/` directory.
    2. Delete the `pf-pingone-verify-adapter-<version>.jar` file from your `<pf_install>/pingfederate/server/default/deploy` directory.
4. Extract the `.zip` archive and merge the contents of the `dist` directory with your `<pf_install>/pingfederate` directory.
5. If there is more than one version of the `pf-authn-api-sdk-<version>.jar` file in your `<pf_install>/pingfederate/server/default/lib` directory, delete all but the latest version of the file.
6. If you backed up any customized files, modify the new files with your customizations.
7. Start PingFederate.
8. If you operate PingFederate in a cluster, repeat steps 2-7 for each engine node.
9. If you upgraded from version 1.1 to version 2.0 of the adapter, update your configuration to support the new core contract attributes:
    1. In your adapter configuration, update your existing PingOne Verify Response Mappings using the updated JSON Pointer syntax reference information.
    2. Update any affected attributes in your authentication policy.

Connecting PingFederate to PingOne

About this task

Create a connection between PingFederate and PingOne. You can use this connection for all PingFederate components that communicate with PingOne.

> **Note**
>
> If you already have a connection from PingOne to PingFederate, ensure that the **Identity admin** and **Environment admin** roles are enabled. Skip the steps below.

Copyright © 2025 Ping Identity Corporation

Steps

1. Create credentials in PingOne:
    1. On the PingOne administrative console, go to Integrations > PingFederate and click the + icon.
    2. On the Create Connection Profile window, enter a connection name, then click Save.
    3. After the New Credential Created window opens, click Copy To Clipboard, then Done.
2. Use the credentials to create a connection in PingFederate:
    1. On the PingFederate administrative console, go to System > External Systems > PingOne Connections and click + Add Connection.
    2. On the Add Connection window, paste the credential information in the Paste The Credential Here field.
    3. In the Connection Name field, enter a unique name and click Save.

Configuring an adapter instance

Configure the PingOne Verify IdP Adapter to determine how PingFederate communicates with PingOne Verify.

Steps

1. In the PingFederate administrative console, go to Authentication > Integration > IdP Adapters and click Create New Instance.
2. On the Type tab, set the basic adapter instance attributes.
    1. In the Instance Name field, enter a name for the adapter instance.
    2. In the Instance ID field, enter a unique identifier for the adapter instance.
    3. In the Type list, select PingOne Verify IdP Adapter. Click Next.
3. (Optional) On the IdP Adapter tab, in the PingOne Verify Response Mappings section, map attributes from the PingOne Verify response to the attribute contract.

    These attributes become available in your PingFederate authentication policy.

    > **Note**
    >
    > Before using this table, check the **Extended Contract** tab to see which attributes are already included in the core contract.

    1. Click Add a new row to 'Optional PingOne Verify Response Mappings'.
    2. In the Local Attribute field, enter a name for the new attribute.
    3. In the PingOne Verify API Mapping field, enter the JSON Pointer syntax for the value of the matching PingOne Verify attribute.

        You can find help in JSON Pointer syntax reference.
    4. In the Action column, click Update.

Copyright © 2025 Ping Identity Corporation

5. To add more attributes, repeat steps a-d.

4. On the IdP Adapter tab, configure the adapter instance by referring to PingOne Verify IdP Adapter settings reference. Click Next.
5. On the Actions tab, test your connection to PingOne Verify. Resolve any issues that are reported, and then click Next.
6. On the Extended Contract tab, click Next.
7. On the Adapter Attributes tab, for subject, select the Pseudonym checkbox. Click Next.
8. On the Adapter Contract Mapping tab, click Next.
9. On the Summary tab, check your configuration, then click Save.

PingOne Verify IdP Adapter settings reference

Field descriptions for the PingOne Verify IdP Adapter configuration screen.

Field	Description
PingOne Environment	Select the PingOne connection you created in Connecting PingFederate to PingOne. This field is blank by default.
PingOne Population	If a user doesn't already exist in PingOne, the adapter provisions the user to this PingOne population. **Note** Applies only when **Provision User** is selected. This list populates after you select the PingOne Environment.
Verify Policy	The name of the verify policy you want to use. This list populates after you select a PingOne environment.
Email Chained Attribute	The name of the incoming chanied attribute that contains the user's email address. The default value is `mail`.
Phone Chained Attribute	The name of the incoming chained attribute that contains the user's phone number. The default value is `mobile`.

Copyright © 2025 Ping Identity Corporation

Field	Description
Reference Image Chained Attribute	The name of the incoming chained attribute that contains the user's reference image. **Note** The **facial comparison only** verify policy requires the reference image as a base64-encoded string. The default value is photo.
Verification URL Delivery Method	Specifies how the verification link is sent to users to begin the identity verification process. When set to User selection, users can choose their preferred delivery method. For example, email or SMS. The default value is User selection.
Redirect URL	The destination URL that users are directed to automatically after completing identity verification successfully.
Redirect Message	A customizable message displayed to users before they're directed to the Redirect URL you specified. **Note** If you configure a **Redirect Message**, you must configure a **Redirect URL** before you can save the configuration.
First Name Chained Attribute	The name of the incoming chained attribute that contains the user's first name. This value is compared against the first name extracted from the government ID submitted during verification to ensure they match. If this field is left empty, record matching isn't performed for this user attribute.
Last Name Chained Attribute	The name of the incoming chained attribute that contains the user's last name. This value is compared against the last name extracted from the government ID submitted during verification to ensure they match. If this field is left empty, record matching isn't performed for this user attribute.

Copyright © 2025 Ping Identity Corporation

Field	Description
Full Name Chained Attribute	**The name of the incoming chained attribute that contains the user's full name.** **Important** Required when verifying against ID types that only support full name extraction (instead of separate first and last name components). **This value is compared against the full name extracted from the government ID submitted during verification to ensure they match.** **If this field is left empty, record matching isn't performed for this user attribute.**
Address Chained Attribute	**The name of the incoming chained attribute that contains the user's address.** **This value is compared against the address extracted from the government ID submitted during verification to ensure they match.** **If this field is left empty, record matching isn't performed for this user attribute.**
Date of Birth Chained Attribute	**The name of the incoming chained attribute that contains the user's date of birth.** **Important** Use the following format: `YYYY-MM-DD`. **This value is compared against the date of birth extracted from the government ID submitted during verification to ensure they match.** **If this field is left empty, record matching isn't performed for this user attribute.**

Field	Description
Test Username	**The PingOne username that the adapter uses to test the PingOne Verify connection on the Actions tab.** **This field is blank by default.**

Copyright © 2025 Ping Identity Corporation

Field	Description
HTML Template Prefix	Identifies the set of HTML templates that the adapter uses to show the authentication status or request a one-time passcode (OTP). If you customize the template file names in the `/server/default/conf/template` directory, enter the new prefix here. You can find a description of the template files in Download manifest. The default value is `pingone-verify`.
Messages Files	Identifies the customizable language-pack file that the adapter uses to show messages on the templates. If you customize the `pingone-verify-messages.properties` file name in the `/server/default/conf/language-packs` directory, enter the new name here. The default value is `pingone-verify-messages`.
Error Message Key Prefix	Identifies the error messages in the language-pack file that the adapter uses to show messages on the templates. If you customize the error message names in `/server/default/conf/language-packs/pingone-verify-messages.properties`, enter the new prefix here. The default value is `pingone.verify.error.`.
Provision User	If a user does not already exist in PingOne, the adapter provisions the user to PingOne. This checkbox is selected by default.
Allow Verification Retries	When ID verification fails, this setting determines whether a user can try again. Your PingOne Verify configuration determines the maximum number of retry attempts. This checkbox is cleared by default.
Reset Verification Status	Require ID verification for every authentication by resetting the user's status. This checkbox is cleared by default.

Copyright © 2025 Ping Identity Corporation

Field	Description
User Not Found Failure Mode	When a user error occurs in PingOne, this setting determines whether the adapter blocks the user's sign-on or registration attempt. User errors include the following: • User is disabled • User does not exist • User cannot be provisioned • The PingOne Verify service is disabled for the user The default selection is `Block user`.
Service Unavailable Failure Mode	When PingOne does not respond, this setting determines whether the adapter blocks the user's sign-on or registration attempt. The default selection is `Bypass authentication`.
Show Success Screens	Determines whether the adapter shows a success page when the verification process is successful. This checkbox is selected by default.
Show Error Screens	Determines whether the adapter shows an error page when the verification process fails and the user cannot retry. This checkbox is selected by default.
Show Timeout Screens	Determines whether the adapter shows a "timed out" page when the verification process times out. This checkbox is selected by default.
State Timeout	The amount of time in seconds that the adapter allows for each state of the process when waiting for a user to complete the verification process. The default value is `1200`. The minimum value is `60`.
API Request Timeout	The amount of time in milliseconds that PingFederate allows when establishing a connection with PingOne Verify or waiting for a response to a request. A value of `0` disables the timeout. The default value is `5000`.
Proxy Settings	Defines proxy settings for outbound HTTP requests. The default value is System Defaults.
Custom Proxy Host	The proxy server host name to use when Proxy Settings is set to Custom. This field is blank by default.

Copyright © 2025 Ping Identity Corporation

Field	Description
Custom Proxy Port	The proxy server port to use when Proxy Settings is set to Custom. This field is blank by default.

JSON Pointer syntax reference

JavaScript Object Notation (JSON) Pointer defines a syntax for identifying a specific value within a JSON payload. Using the sample payload and JSON Pointer examples below, identify the attributes that you want to use to populate your attribute contract.

For a complete technical description of JSON Pointer syntax, see JavaScript Object Notation (JSON) Pointer on ietf.org.

Example PingOne Verify JSON payload

For information on example JSON payloads, see Verification MetaData and Verified Data in the PIngOne Platform API Reference documentation.

JSON Pointer syntax

The following attributes are already included in the core contract, so you do not need to add them to the PingOne Verify Response Mappings table. We have included them here to help you understand the JSON Pointer mapping scheme for newly-added PingOne Verify attributes.

Description	JSON Pointer	Example value
Verification transaction status	`/transactionStatus/status`	`SUCCESS`
User's first name (from their photo ID)	`/_embedded/verifiedData/0/data/firstName`	`Alice`
User's selfie	`/_embedded/verifiedData/0/data/IMAGE`	
User's metadata	`/_embedded/metaData`	
User's state (from their photo ID)	`/_embedded/verifiedData/0/data/addressState`	`CA`

Copyright © 2025 Ping Identity Corporation

Description	JSON Pointer	Example value
User's date of birth (from their photo ID)	`/_embedded/verifiedData/0/data/birthDate`	`1980-01-20`

> **Note**
>
> To populate an attribute with the entire JSON response, leave the attribute mapping field blank.

Authentication setup

By modifying your PingFederate authentication policy to include the PingOne Verify IdP Adapter, you can challenge users to verify their identity as part of the registration process.

This allows a user to access an existing account by verifying their identity based on photo ID.

> **Note**
>
> If you want to include verification as part of the account registration process, see Self-service registration setup.

Adding PingOne Verify to your authentication policy

By modifying your PingFederate authentication policy to include the PingOne Verify IdP Adapter, you can challenge users to verify their identity as part of the authentication process.

About this task

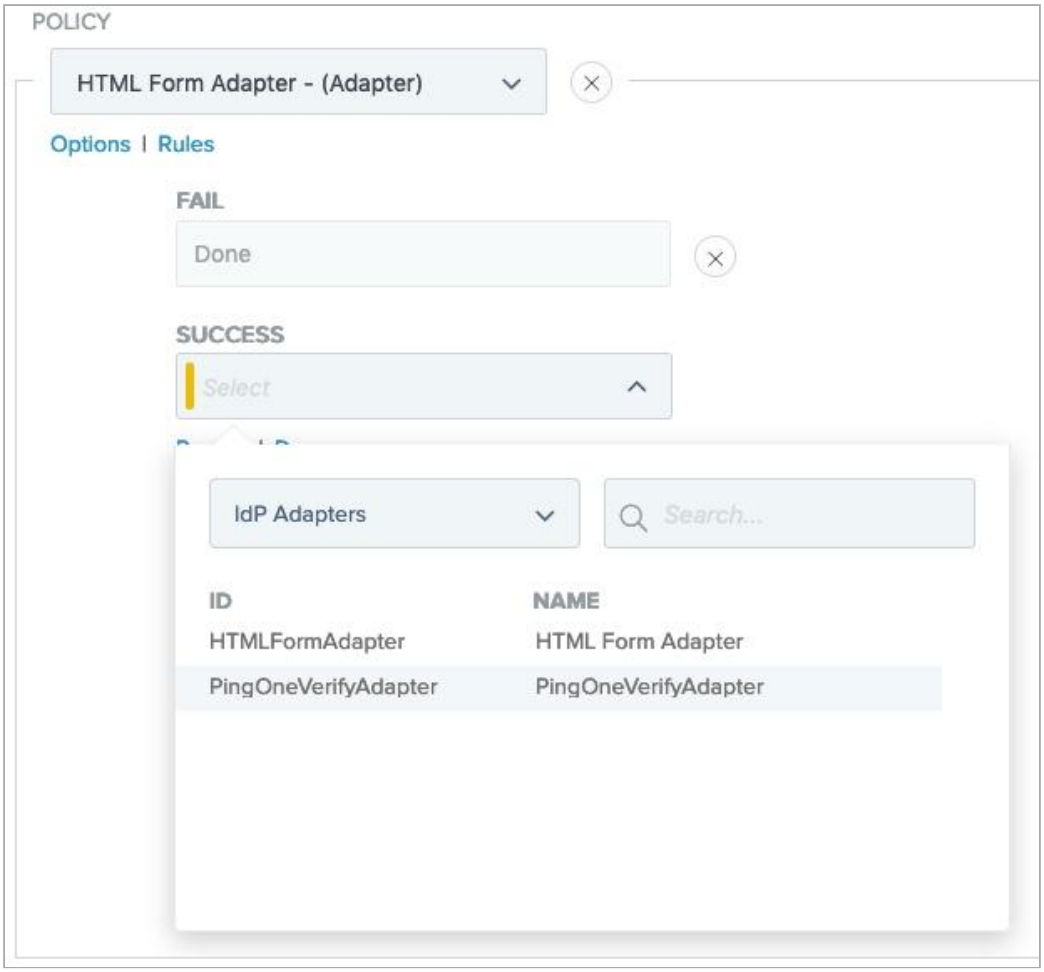
Use these steps to add to an existing authentication policy. You can find general information about configuring authentication policies in Authentication policies in the PingFederate documentation.

Steps

1. In the PingFederate administrative console, go to Authentication > Policies > Policies.
2. On the Policies tab, select the IdP Authentication Policies checkbox.
3. Open an existing authentication policy, or click Add Policy.

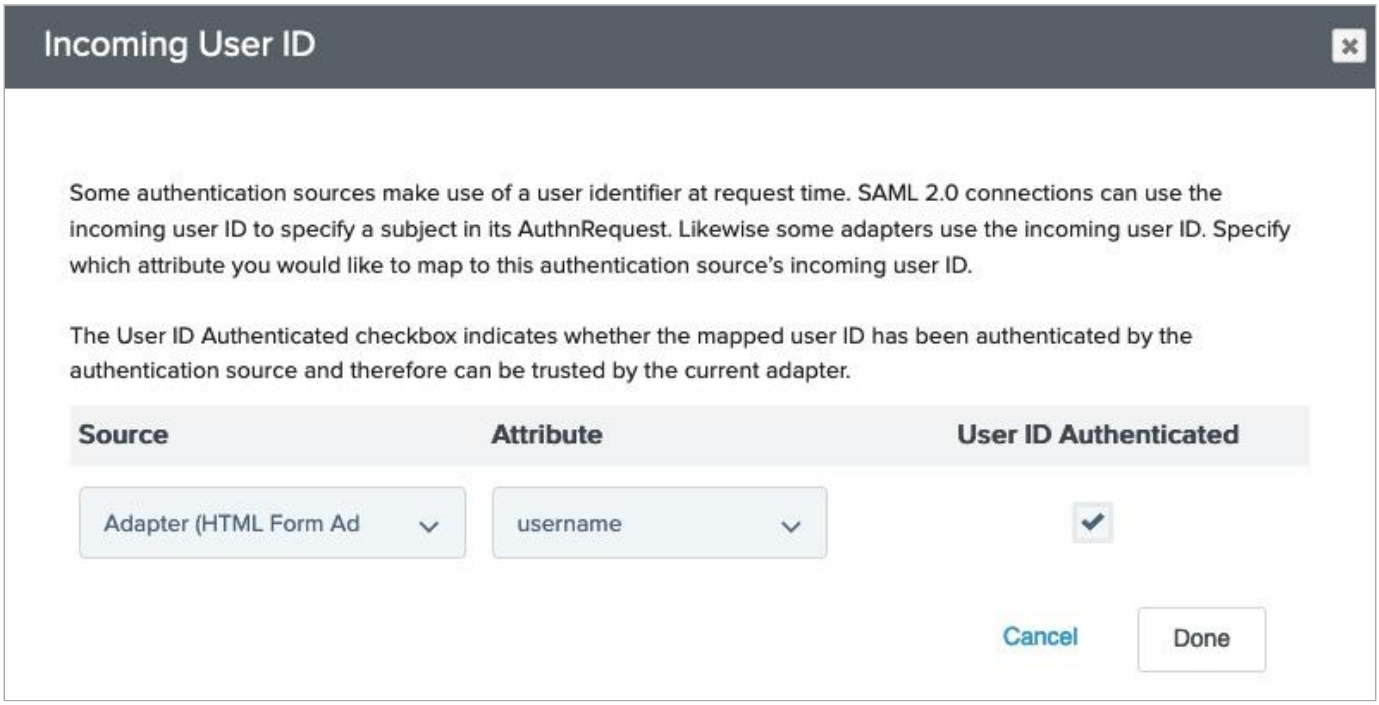
   You can find more information in Defining authentication policies in the PingFederate documentation.
4. In the Policy section, in the Select list, select a PingOne Verify IdP Adapter instance.

Copyright © 2025 Ping Identity Corporation



5. Map the PingOne `user ID` or `username` into the PingOne Verify IdP Adapter instance.

Copyright © 2025 Ping Identity Corporation

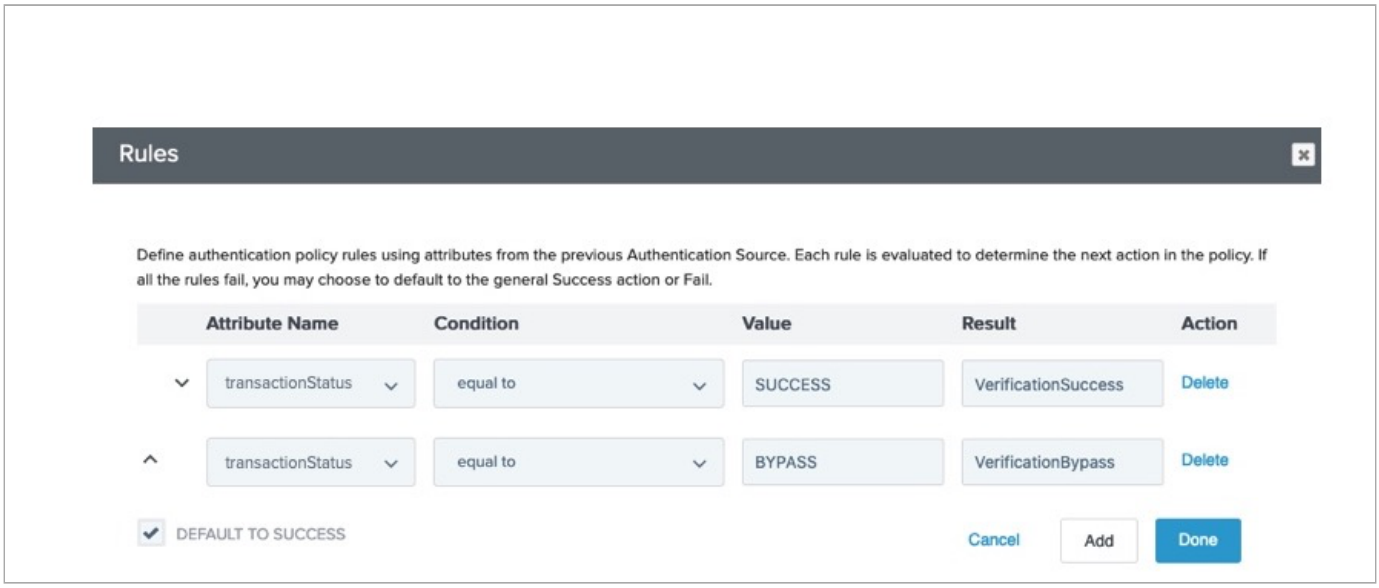


1. Under the PingOne Verify IdP Adapter instance, click Options.
2. On the Options window, in the Source list, select a previous authentication source that collects the PingOne user ID or username.
3. In the Attribute list, select the user ID.
4. Select the User ID Authenticated checkbox.
5. Click Done.

6. (Optional) Define policy paths based on verification results.

> **Tip**
>
> Depending on the failure mode settings in your adapter configuration, the adapter can potentially return a `success` result in the authentication policy even when the user verification process did not succeed. It can be useful to create separate policy paths for a successful adapter result and a successful user verification result.

Copyright © 2025 Ping Identity Corporation



1. Under the PingOne Verify IdP Adapter instance, click Rules.
2. On the Rules window, in the Attribute Name list, select transactionStatus.
3. In the Condition list, select equal to.
4. In the Value field, enter `SUCCESS` or `BYPASS`.

   SUCCESS

   The user successfully verified their identity.

   BYPASS

   An error occurred, and the verification process was not completed, but the adapter is configured to bypass verification and continue the authentication flow.

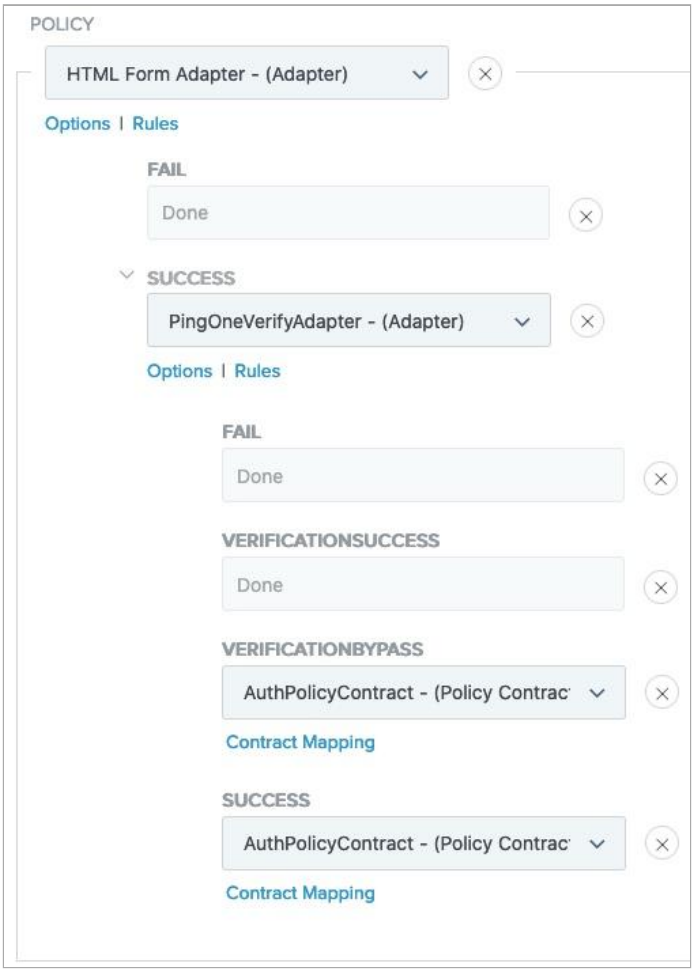
5. In the Result field, enter a name.

   This appears as a new policy path that branches from the authentication source.

6. To add more authentication paths, click Add and repeat steps a-e.
7. Click Done.

7. Configure each of the authentication paths, including Fail, Success, and any paths that you defined in the Rules window.

Copyright © 2025 Ping Identity Corporation



8. Click Done.

9. In the Policies window, click Save.

Self-service registration setup

By modifying your PingFederate user registration flow to include the PingOne Verify IdP Adapter, you can challenge users to verify their identity as part of the registration process.

This allows a user to register a new account by providing their basic information and verifying their identity based on photo ID.

> **Note**
>
> If you want to include verification as part of the authentication process for existing accounts, learn more in Authentication setup.

Requirements for self-service registration setup

- PingFederate 11.3 or later

Copyright © 2025 Ping Identity Corporation

- An existing user registration configuration, including:
  - PingDirectory
  - An authentication policy contract
  - A local identity profile
  - An HTML Form Adapter instance
  - An authentication policy

> **Important**
>
> Before you continue, complete the steps in Setting up self-service registration in the PingFederate documentation. You will modify this initial configuration in the sections below.

Registration form modifications

Configure your HTML Form Adapter to validate the registration form data before allowing users to continue to the identity verification process.

The template path is: `<pf_install>/pingfederate/server/default/conf/template/local.identity.registration.html`. Learn more about modifying templates in Customizable user-facing pages in the PingFederate documentation.

Name fields

Near the name fields, modify the template to tell users to "Enter the name as shown on your ID card". Users can only register with their verified name.

Field validation

If your datastore has length or character requirements for passwords, for example, add validation to the password field that checks against the same criteria.

You can add form validation by modifying the HTML Form Adapter registration template with custom JavaScript.

Learn more about form validation JavaScript in JavaScript Form Validation on w3schools.com.

Creating input and output policy contracts

Before creating a PingOne Verify policy fragment, create policy contracts to define which attributes are provided to and collected from the adapter.

Copyright © 2025 Ping Identity Corporation

Steps

1. Create an input policy contract.

   Learn more in Policy contracts in the PingFederate documentation.

   1. In the PingFederate administrative console, go to Authentication > Policies > Policy Contracts.
   2. Click Create New Contract.
   3. On the Contract Info tab, in the Contract Name field, enter a name, such as `PingOneVerifyInput`. Click Next.
   4. On the Contract Attributes tab, add attributes to provide to the PingOne Verify IdP Adapter, such as `email` and `username`. Click Save.

2. Create an output policy contract.

   1. Click Create New Contract.
   2. On the Contract Info tab, in the Contract Name field, enter a name, such as `PingOneVerifyOutput`. Click Next.
   3. On the Contract Attributes tab, add user attributes to get from the PingOne Verify IdP Adapter, such as `email`, `username`, and `status`. Click Save.

Creating a policy fragment

Create a policy fragment that will allow you to include the PingOne Verify IdP Adapter in your user registration flow.

Steps

1. In the PingFederate administrative console, go to Authentication > Policies > Fragments.
2. Click Add Fragment.
3. In the Name field, enter a name, such as `PingOne Verify Fragment`.
4. In the Inputs list, select your input policy contract, such as PingOneVerifyInput.
5. In the Outputs list, select your input policy contract, such as PingOneVerifyOutput.
6. In the Policy area, select a PingOne Verify IdP Adapter instance.
   1. Under the PingOne Verify IdP Adapter instance, click Options.
   2. On the Options dialog, from the Source list, select Inputs.
   3. In the Attribute list, select the attribute that contains the user identifier, such as username.
   4. Select the User ID Authenticated checkbox.
   5. Click Done.
7. For the Fail path, select Done.
8. For the Success path, select your output policy contract, such as PingOneVerifyOutput.

Copyright © 2025 Ping Identity Corporation

9. Configure the policy contract to include attributes from the input as well as from the PingOne Verify IdP Adapter.
    1. Under the output policy contract, click Contract Mapping.
    2. On the Contract Fulfillment tab, map the output attributes to allow the policy fragment to provide information, such as the transaction status and user identifier, to the authentication policy.

    Table 1. Example contract fulfillment mappings

Source	Value
Inputs	email
Inputs	username
Adapter (PingOne Verify IdP Adapter instance)	transactionStatus
Adapter (PingOne Verify IdP Adapter instance)	subject

    3. Click Done.
10. Click Save.

Modifying your local identity profile

Modify your local identity profile to enable registration using the policy fragment that you created.

About this task

These steps build on the local identity profile that you created as part of Setting up self-service registration in the PingFederate documentation.

Steps

1. In the PingFederate administrative console, go to Authentication > Policies > Local Identity Profiles.
2. Open the local identity profile that you created in your initial self-service registration configuration.
3. Optional: On the Profile Info tab, select the Enable Profile Management checkbox.

   This allows the user to sign on to a profile management page.
4. Optional: Enable email address verification.

   Learn more in Configuring email ownership verification options in the PingFederate documentation.
    1. On the Fields tab, click Create New Field.
    2. In the ID field, enter a name, such as `emailVerified`.
    3. In the Label field, enter a label, such as `Email verification status`.
    4. In the Type list, select Hidden.

Copyright © 2025 Ping Identity Corporation

5. Click Done.
6. On the Email Verification tab, from the Email Address Field, select the email attribute, such as lipEmail.
7. In the Ownership Status Field, select the email verification status attribute, such as emailVerified.
8. On the Data Store Configuration tab, map the new attribute to a data store attribute.

   Learn more in Managing datastore configuration in the PingFederate documentation.

5. On the Registration tab, configure registration to use the policy fragment that you created in Creating a policy fragment.

   Learn more in Configuring registration options in the PingFederate documentation.

   1. In the Username Field list, select the field that you want to use as the username for the authentication session, for example, lipEmail.
   2. In the Registration Workflow list, select the policy fragment that you created in Creating a policy fragment.
   3. For Execute Workflow, select Before Account Creation.
6. On the Profile Management tab, select the Enable Profile Deletion checkbox.
7. Click Save.

Adding PingOne Verify to your user registration flow

By modifying your PingFederate authentication policy to include the PingOne Verify IdP Adapter policy fragment, you can challenge users to verify their identity as part of the self-service registration process.

Steps

1. In the PingFederate administrative console, go to Authentication > Policies > Policies.
2. Select the IdP Authentication Policies checkbox.
3. Open the authentication policy that you created in your initial registration configuration.
4. In the Policy area, define a policy path for user registration.
   1. Under the HTML Form Adapter instance, click Rules.
   2. On the Rules dialog, from the Attribute Name list, select policy.action.
   3. In the Condition list, select equal to.
   4. In the Value field, enter `identity.registration`.
   5. In the Result field, enter a name, such as `Registration`.

      This appears as a new policy path that branches from the PingOne Verify IdP Adapter.
   6. Click Done.
5. After the HTML Form Adapter instance, for the Fail path, select Done.

Copyright © 2025 Ping Identity Corporation

6. After the HTML Form Adapter instance, for the Registration path, select the local identity profile that you created for PingOne Verify.
    1. Click Local Identity Mapping.
    2. On the Inbound Mapping tab, map the registration form fields to the input policy contract attributes.

    Table 1. Example mappings

Inbound Mapping Fulfillment	Value
firstName	regFirstName
email	registrar@example.com

    3. On the Contract Fulfillment tab, map values for the outbound contract.

Outbound Mapping Fulfillment	Source	Value
attr1	Local Identity (Verify Identity Profile)	email
subject	Local Identity (Verify Identity Profile)	firstName

    4. Click Done.
7. After the HTML Form Adapter instance, for the Success path, configure the standard (non-registration) authentication path.
8. Click Done.
9. In the Policies window, click Save.

Authentication API Support

You can use the PingFederate authentication API to integrate the PingOne Verify IdP Adapter into your application.

The PingFederate Authentication API provides access to the current state of the authentication flow as a user steps through the PingFederate authentication policy. Learn more in PingFederate authentication API in the PingFederate documentation.

To integrate the PingOne Verify IdP Adapter into your authentication flow, configure your application based on the information in this section.

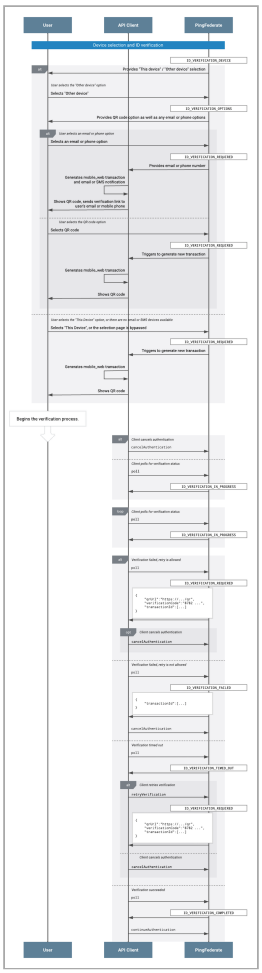
> **Tip**
>
> You can also explore the process using the PingFederate Authentication API Explorer. Learn more in Exploring the Authentication API in the PingFederate documentation.

Copyright © 2025 Ping Identity Corporation

Verification flow

When using the PingOne Verify IdP Adapter through the PingFederate authentication application programming interface (API), the following flow is used for verification requests.

Identity verification



1. The user completes first-factor authentication or begins the registration flow.

   ID verification begins when using the PingOne Verify IdP Adapter is triggered in the PingFederate authentication or registration flow.

2. PingFederate returns a status of `ID_VERIFICATION_DEVICE` or `ID_VERIFICATION_OPTIONS`. The user selects how they want to proceed with the verification process.

> **Note**
>
> `ID_VERIFICATION_OPTIONS` is not always present in the flow. It appears if the user has an email address or mobile number tied to the account. The email address or phone number can come from authentication policy attributes or from a LDAP server.
> Also, if your configuration forces the email or phone option, the adapter skips the selection step.

Copyright © 2025 Ping Identity Corporation

3. PingFederate returns a status of `ID_VERIFICATION_REQUIRED` in the response to the API client and provides a QR code and a verification code.
4. The API client displays the QR code and verification code to the user.
5. The user completes the ID verification process in mobile web.
6. The API client invokes the poll action.
7. PingFederate returns the status of `ID_VERIFICATION_IN_PROGRESS` to the API client.
8. The poll action repeats until one of the following statuses occurs:
    - `ID_VERIFICATION_REQUIRED`
        - Verification failed, but a retry is allowed
        - The user can complete (or retry) the verification process with the provided QR code and verification code
        - There are two options available to the API client:
            1. Continue polling by calling `poll` and wait for the status to change
            2. Cancel the verification request by calling `cancelAuthentication`
    - `ID_VERIFICATION_FAILED`
        - Verification failed, and a retry is not allowed
        - The authentication request is cancelled
    - `ID_VERIFICATION_TIMED_OUT`
        - Verification timed out
        - There are two options available to the API client:
            1. Retry by calling `retryVerification`
            2. Cancel the verification request by calling `cancelAuthentication`
    - `ID_VERIFICATION_COMPLETED`
        - Verification succeeded
        - The API client continues the flow by calling `continueAuthentication`
        - PingFederate returns the transaction status and user data to the API client

State models, action models, and attributes

When using the PingOne Verify IdP Adapter through the PingFederate authentication API, the adapter uses the following state models, action models, and attributes.

Copyright © 2025 Ping Identity Corporation

State models

Status	State Model	Action	Description
`ID_VERIFICATION_REQUIRED`	• `verifyAppName` • `qrUrl` • `verificationCode` • `errorDetails`	• `poll` • `cancelAuthentication`	Indicates that verification is required.
`ID_VERIFICATION_DEVICE`		• `cancelAuthentication` • `deviceAuthentication`	Indicates whether to use this device or another device to start verification.
`ID_VERIFICATION_OPTIONS`		• `retryVerification` • `cancelAuthentication` • `optionsAuthentication`	Indicates whether or not an email or phone number is selected for the notification option.
`ID_VERIFICATION_IN_PROGRESS`		• `poll`	Indicates that ID verification is in progress.
`ID_VERIFICATION_COMPLETED`		• `continueAuthentication`	Indicates that ID verification completed successfully.
`ID_VERIFICATION_FAILED`	• `errorDetails`	• `cancelAuthentication`	Indicates that ID verification failed and a retry is not allowed.
`ID_VERIFICATION_TIMED_OUT`		• `retryVerification` • `cancelAuthentication`	Indicates that ID verification timed out.

Copyright © 2025 Ping Identity Corporation

Action models

Action	Action Model	Error	Description
`poll`	(void)		Gets the current ID transaction status.
`retryVerification`	(void)		Retries verification after a time out or after selecting cancel on any screen.
`deviceAuthentication`	(void)		Select a device to start verification.
`optionsAuthentication`	(void)		Select email, phone, or QR-only for sending the web link to the user.
`cancelAuthentication`			Cancels the current authentication step.

Attribute models

Attribute name	Type	Description
`qrUrl`	String	The URL of the QR code image associated with the verification session. Use this URL to show the QR code to the user. The user scans the QR code with the verification app to begin the verification process. To display the QR code, your app needs to perform a GET request to the URL provided in `qrUrl`. Include an <img> element, and use the result of the GET request as the source. QR URL provided by PingFederate: `"qrUrl":"https://api.pingone.com/v1/idValidations/shortcode/008030285946/qr"` GET request to get the QR code image: `GET "https://api.pingone.com/v1/idValidations/shortcode/008030285946/qr"` GET request response: ` IsIxwcKDcpLDAxNDQ0Hyc5PTgyPC4zNDL...2wBDAQkJCQwL DBgNDRgyIRwhMjIyMjIyMj` HTML element that displays the image in your app: `<img src="data:image/jpegbase64,9j4AAQSkZJRgABAgAAAQABAAD IsIxwcKDcpLDAxNDQ0Hyc5PTgyPC4zNDL...2wBDAQkJCQwL DBgNDRgyIRwhMjIyMjIyMj"/>`
`verificationCode`	String	A code that the user can enter instead of scanning the QR code image.

Copyright © 2025 Ping Identity Corporation

Attribute name	Type	Description
`errorDetails`	List< `CustomAuthnErrorDetail` >	A list of errors.

Troubleshooting

Enabling debug logging

To help with troubleshooting or monitoring, you can turn on activity logging for the adapter.

About this task

These steps are optional. You can find more general information about logging in Enabling debug messages and console logging in the PingFederate documentation.

Steps

1. Open the `<pf_install>/pingfederate/server/default/conf/log4j2.xml` file for editing.
2. If you want to log activity for PingFederate and all adapters, do the following.
    1. Find the following section:

    ```
    <AsyncRoot level="INFO" includeLocation="false">
            <!-- <AppenderRef ref="CONSOLE" /> -->
            <AppenderRef ref="FILE" />
    </AsyncRoot>
    ```

    2. Change `INFO` to `DEBUG`.

    ```
    <AsyncRoot level="DEBUG" includeLocation="false">
            <!-- <AppenderRef ref="CONSOLE" /> -->
            <AppenderRef ref="FILE" />
    </AsyncRoot>
    ```

    3. If you want to see the adapter activity in the console, remove the comment tags.

    ```
    <AsyncRoot level="INFO" includeLocation="false">
            <AppenderRef ref="CONSOLE" />
            <AppenderRef ref="FILE" />
    </AsyncRoot>
    ```

3. If you want to log activity just for the PingOne Verify IdP Adapter, add the following line.

Copyright © 2025 Ping Identity Corporation

```
<Logger name="com.pingidentity.adapters.pingone.verify" level="DEBUG"/>
```

4. Save the file.

Release notes

Changelog

The following is the change history for the PingOne Verify Integration Kit.

PingOne Verify Integration Kit 2.3 - February 2025

- Added support for biographic matching. Configure the expected names of incoming chain attributes to collect the actual user values during verification and send them to PingOne Verify.

  Learn more in the First Name Chained Attribute through Date of Birth Chained Attribute table entries in the PingOne Verify IdP Adapter settings reference. Learn more about biographic matching in Types of verification.

  > **Note**
  >
  > Currently, you must parse the biographic matching metadata to view results and decide how to proceed. Verify policy support for biographic matching results will be added in a future release.

- Added the ability to configure a redirect URL and message. Learn more in the Redirect URL and Redirect Message table entries in the PingOne Verify IdP Adapter settings reference.
- Added the ability to provision a user during the password reset flow if Provision User is selected in the adapter configuration. Learn more about Provision User in the PingOne Verify IdP Adapter settings reference.

  > **Important**
  >
  > Learn more about mitigating risk for password reset flows in the Account recovery use case.

- Removed a deprecated endpoint, `verifyStatus`.
- Fixed an issue that caused a `Null Pointer Exception` error when creating a new adapter instance if the PingOne environment didn't include the PingOne Verify service.

  The PingOne environment must include at least one verify policy because you must select a Verify Policy before you can save the adapter configuration.

- Improved the error messaging when a user fails a verification attempt and renamed the Cancel button Retry for clarification and alignment.

  To use the updated template files, make sure to follow the Deploying the integration files procedure when upgrading to PingOne Verify Integration Kit 2.3.

Copyright © 2025 Ping Identity Corporation

PingOne Verify Integration Kit 2.2.1 – August 2023

- The adapter now syncs its dependencies with PingFederate 11.3.

PingOne Verify Integration Kit 2.2 – July 2023

- Added an adapter configuration option to reset a user's verification status for every sign-on attempt.
- Added support for password reset in the PingOne Verify IdP Adapter. To ensure maximum security, the password reset flow requires full verification for all users, even if they're already verified.

> **Important**
>
> For password reset and sign-on flows, a malicious user might try to reset the password of an account other than their own by using their own ID. To prevent this, set up a comparison in the PingFederate policy to check if the identity verified matches the user's record. After the PingOne Verify Integration Kit succeeds, confirm that the first and last names of the verified ID match those in the directory for the username that was entered on the sign-on or forgot password page.

- Fixed an issue where administrators had to map whether to block or bypass authentication for a user with a disabled verify status in Service Unavailable Failure Mode instead of User Not Found Failure Mode.

PingOne Verify Integration Kit 2.1 – December 2022

- Added support for facial comparison only verify policy in the adapter.
- Added support for the verify policy.
- Added support for the PingOne Verify Mobile SDK version 2 with QR verification flows.

PingOne Verify Integration Kit 2.0 – July 2022

- Added support for web authentication flow.
- Added `gender`, `issuingCountry`, `nationality`, and `weight` as core attributes.
- Add `selfie` and `metadata` as option attributes.
- Removed support for PingOne Verify Mobile SDK.

PingOne Verify Integration Kit 1.1 – July 2021

- Added support for changes to PingOne Verify, including:
  - The adapter now gets the user's `verifyEnabledState` attribute when getting the verification result
  - Updated the list of core contract attributes to support changes to the structure of the `verifiedUserData` attribute
  - Added support for change to the flow behavior when the `verifyDocument` flag is off
  - Updated the core contract to support changes in PingOne Verify:

Copyright © 2025 Ping Identity Corporation

Change description	Version 1.0	Version 1.1
Attribute name change	`verificationStatus`	`transactionStatus`
Attribute name change	`dateOfBirth`	`birthDate`
Attribute name change/merge	`addressLine1`	`addressStreet`
Attribute name change/merge	`addressLine2`	
Attribute name change	`city`	`addressCity`
Attribute name change	`state`	`addressState`
Attribute name change	`postalCode`	`addressZip`
Attribute name change	`documentId`	`idNumber`
New attribute		`idType`

Tip

You can find upgrade instructions in Deploying the integration files.

PingOne Verify Integration Kit 1.0 – January 2021

- Initial release.
- Added the ability to direct users to an identity verification app as part of a sign-on or registration flow.
- Added the ability to control which message pages (templates) are shown to the user.
- Added the ability to customize template messages using a language pack file.
- Added the ability to test the connection to PingOne Verify.
- Added a setting to automatically provision new users to PingOne.
- Added settings to control how the adapter handles sign-on and registration attempts when errors occur.
- Added support for the platform connection to PingOne introduced in PingFederate 10.2.
- Added support for the PingFederate authentication API
- Added support for the JavaScript Widget for the PingFederate Authentication API.
- Added settings for API connection and request timeouts.
- Added settings to override the PingFederate system-default proxy settings.

Copyright © 2025 Ping Identity Corporation

Known issues and limitations

The following are known issues or limitations for the PingOne Verify Integration Kit.

Known issues

There are no known issues.

Known limitations

There are no known limitations.

Download manifest

The following files are included in the PingOne Verify Integration Kit `.zip` archive:

- `ReadMeFirst.pdf` – Contains links to this online documentation
- `Legal.pdf` – Copyright and license information
- `dist/pingfederate/server/default` – Contains the integration files
  - `deploy` – Contains the Java libraries
    - `pf-pingone-verify-adapter-<version>.jar` – JAR file that contains the PingOne Verify IdP Adapter.
  - `conf` – contains the HTML template that presents the PingOne sign-on form.
    - `language-packs` – Contains files with customizable user-facing messages
      - `pingone-verify-messages.properties` – a variable file that customizes the messages that appear on the template file.
    - `template` – contains user-facing HTML template files
      - `pingone-verify-completed.html` – a sample template that indicates the verification step completed successfully.
      - `pingone-verify-failed.html` – a sample template that indicates that the user failed the verification challenge.
      - `pingone-verify-required.html` – a sample template that asks the user to complete the verification challenge.
      - `pingone-verify-timed-out.html` – a sample template that indicates that the verification challenge took too long.
      - `assets` – contains functional scripts and files used by the template
        - `css` – contains CSS files for the templates.
          - `pingone-verify.css` – a CSS file that customizes the appearance of the template files.

Copyright © 2025 Ping Identity Corporation

        - `end-user/<version>/end-user.css` – a CSS file that customizes the appearance of the template files.
    - `fonts/end-user` – contains template fonts and icons
        - `icons` – contains template icons
        - `proxima-nova` – contains template fonts
    - `images` – contains template image files
        - `ping-logo.svg` – an image file with company branding
        - `spinner.svg` – an image file used in a spinner animation
    - `scripts` – contains script files used to collect and send information
        - `jquery-<version>.min.js` – a JavaScript file that contains the jQuery library.
- `lib/pf-authn-api-sdk-<version>.jar` – a JAR file that contains the PingFederate Authentication API SDK

PingOne Verify Integration Kit 1.1

The PingOne Verify Integration Kit allows PingFederate to use the PingOne Verify service for personal identity verification based on government photo ID.

> **Note**
>
> This documentation is for the PingOne Verify Integration Kit 1.1 and earlier. For later versions of the integration kit, see the latest PingOne Verify Integration Kit documentation.

Features

- Initiate an identity verification process as part of user registration or sign-on.
- Supports identity verification through mobile apps
- Supports the PingFederate authentication API.
- Supports the JavaScript Widget for the PingFederate Authentication API.

Components

PingOne Verify IdP Adapter

- Allows PingFederate to use PingOne Verify to trigger an identity verification challenge as part of the PingFederate authentication policy or registration flow.

Templates

- Allow the adapter to show the status of the verification process.
- Allow you to modify the appearance of pages shown to the user during authentication.

Copyright © 2025 Ping Identity Corporation

Language packs

- Allow you to customize or localize the messages returned by the PingFederate authentication application programming interface (API) and shown on the templates during authentication. Learn more in Localizing messages for end users in the PingFederate documentation.

Intended audience

This document is intended for PingFederate administrators.

If you need help during the setup process, learn more in the following resources:

- Introduction to PingOne Verify in the PingOne documentation
- The following sections of the PingFederate documentation:
  - Managing IdP adapters
  - Authentication policies

System requirements

- PingFederate 9.3 or later to include PingOne Verify as part of the authentication flow

  PingFederate 10.2 or later to include PingOne Verify as part of the user registration flow
- To allow PingFederate to make outbound HTTPS connections, you might need to allow the following host names in your firewall:
  - https://api.pingone.com, https://api.pingone.asia, or https://api.pingone.eu
  - https://auth.pingone.com, https://auth.pingone.asia, or https://auth.pingone.eu

Use cases

The following are two common uses for the PingOne Verify Integration Kit.

Authentication

When a user signs on, you can use PingOne Verify to verify a user's identity instead of, or in combination with, a standard multi-factor authentication (MFA) step.

For example, you might use PingOne MFA by default and only trigger PingOne Verify based on:

- Frequency, such as annually
- Group, such as for members of the "contractors" group
- Account age, such as for accounts made in the last two months
- Past verification, such as users that have never completed the verification process
- Security risk, such as for users with a PingOne Protect risk evaluation of `MEDIUM` or `HIGH`

Copyright © 2025 Ping Identity Corporation

You can create these rules in your PingFederate authentication policy.

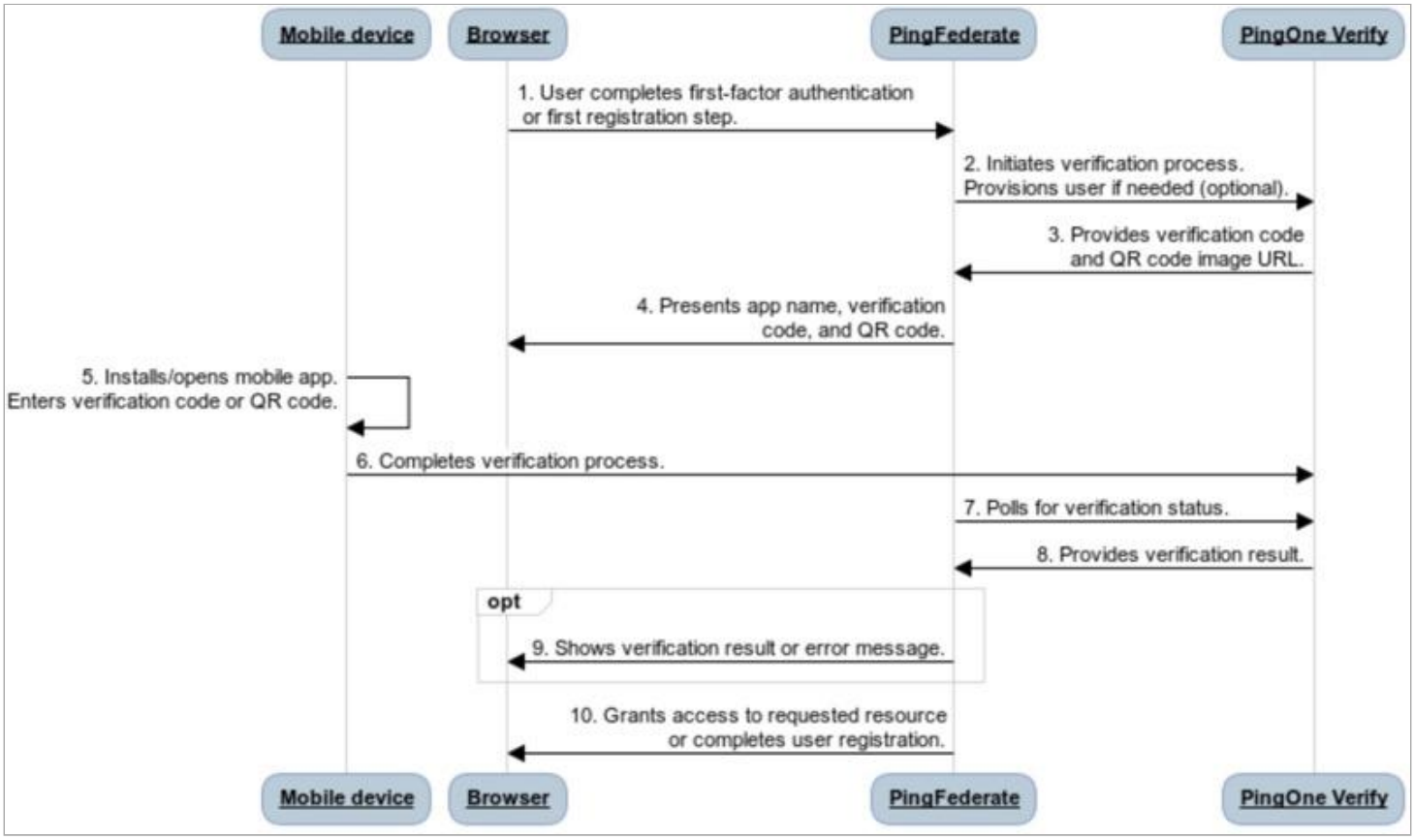
Registration

When a person registers a new user account, you can use PingOne Verify to verify their identity against a piece of government photo ID. By requiring identity verification, you can reduce the number of accounts created by bots, spammers, and imposters.

Optionally, your mobile app can use the reliable information from the person's government photo ID to populate user attributes in your datastore.

Overview of the verification flow

With the PingOne Verify Integration Kit, PingFederate includes PingOne Verify in the sign-on or self-service registration flow.



Description

1. The user initiates single sign-on (SSO) or self-service registration with PingFederate. They complete the first authentication or self-registration step provided by an HTML Form Adapter instance, for example.
2. The PingOne Verify IdP Adapter contacts PingOne Verify to initiate the verification process.

   (Optional) If the user does not exist in PingOne, the adapter provisions the user.
3. PingOne Verify provides a verification code and a QR code image URL.

Copyright © 2025 Ping Identity Corporation

4. The PingOne Verify IdP Adapter presents an HTML page that shows the verification code, QR code, and the name of your PingOne Verify-enabled mobile app.
5. If it is not already installed, the user installs the mobile app from the Android Play Store or Apple App Store.

   After opening the app, the user enters the verification code or QR code shown in the browser.
6. The user follows the app prompts to complete the verification process.

   This typically involves scanning their government-issued photo ID and submitting a live face capture.
7. The PingOne Verify IdP Adapter polls PingOne Verify while it waits for the result of the verification process.
8. PingOne Verify provides the adapter with the result of the verification process.
9. (Optional) The PingOne Verify IdP Adapter presents an HTML page that shows the success or error message, depending on the verification result.
10. If the user completed the verification process successfully, PingFederate provides access to the requested resource or completes the registration process.

Setup

Deploying the integration files

To get started with the integration, deploy the PingOne Verify Integration Kit files to your PingFederate directory.

Steps

1. Download the PingOne Verify Integration Kit `.zip` archive from the Ping Identity Integration Directory.
2. Stop PingFederate.
3. If you're upgrading an existing deployment, back up your customizations and delete earlier versions of the integration files:
    1. Back up any PingOne Verify Integration Kit files that you customized in the `<pf_install>/pingfederate/server/default/conf/` directory.
    2. Delete the `pf-pingone-verify-adapter-<version>.jar` file from your `<pf_install>/pingfederate/server/default/deploy` directory.
4. Extract the `.zip` archive and merge the contents of the `dist` directory with your `<pf_install>/pingfederate` directory.
5. If there is more than one version of the `pf-authn-api-sdk-<version>.jar` file in your `<pf_install>/pingfederate/server/default/lib` directory, delete all but the latest version of the file.
6. If you backed up any customized files, modify the new files with your customizations.
7. Start PingFederate.
8. If you operate PingFederate in a cluster, repeat steps 2-7 for each engine node.

Copyright © 2025 Ping Identity Corporation

9. If you upgraded from version 1.0 to version 1.1 of the adapter, update your configuration to support the new core contract attributes:
    1. In your adapter configuration, update your existing PingOne Verify Response Mappings using the updated JSON Pointer syntax reference information.
    2. Update any affected attributes in your authentication policy.

Connecting PingFederate to PingOne

To allow PingFederate to communicate with PingOne, create a connection.

PingFederate 10.2 or later

About this task

Create a connection between PingFederate and PingOne. In PingFederate 10.2 and later, you can use this connection for all PingFederate components that communicate with PingOne.

> **Note**
>
> If you already have a connection from PingOne to PingFederate, ensure that the **Identity admin** and **Environment admin** roles are enabled. Skip the steps below.

Steps

1. Create credentials in PingOne:
    1. On the PingOne administrative console, go to Integrations > PingFederate and click the + icon.
    2. On the Create Connection Profile window, enter a connection name, then click Save.
    3. After the New Credential Created window opens, click Copy To Clipboard, then Done.
2. Use the credentials to create a connection in PingFederate:
    1. On the PingFederate administrative console, go to System > External Systems > PingOne Connections and click + Add Connection.
    2. On the Add Connection window, paste the credential information in the Paste The Credential Here field.
    3. In the Connection Name field, enter a unique name and click Save.

PingFederate 10.1 or earlier

About this task

In PingOne, create a set of credentials for PingFederate. For PingFederate 10.1 and earlier, you must enter these credentials in each PingFederate component that communicates with PingOne.

Steps

1. On the PingOne administrative console, go to Applications > Applications and click the + icon.
2. On the Add Application window, enter a unique Application Name.

Copyright © 2025 Ping Identity Corporation

3. In the Application Type list, select Worker, then Save.
4. On the Roles tab, click Grant Roles, then select the Identity Data Administrator check box for the environment.
5. On the Overview or Configuration tab, note the Client ID, Client Secret, and Environment ID.

   You'll use these credentials in Configuring an adapter instance.
6. Click the toggle at the top of the application to activate it.

Configuring an adapter instance

Configure the PingOne Verify IdP Adapter to determine how PingFederate communicates with PingOne Verify.

Steps

1. In the PingFederate administrative console, create a new IdP adapter instance:

   Choose from:

   - For PingFederate 10.1 or later: go to Authentication > Integration > IdP Adapters. Click Create New Instance.
   - For PingFederate 10.0 or earlier: go to Identity Provider > Adapters. Click Create New Instance.
2. On the Type tab, set the basic adapter instance attributes.
   1. In the Instance Name field, enter a name for the adapter instance.
   2. In the Instance ID field, enter a unique identifier for the adapter instance.
   3. In the Type list, select PingOne Verify IdP Adapter. Click Next.
3. Optional: On the IdP Adapter screen, in the PingOne Verify Response Mappings section, map attributes from the PingOne Verify response to the attribute contract.

   These attributes become available in your PingFederate authentication policy.

   > **Note**
   >
   > Before using this table, check the **Extended Contract** tab to see which attributes are already included in the core contract.

   1. Click Add a new row to 'Optional PingOne Verify Response Mappings'.
   2. In the Local Attribute field, enter a name for the new attribute.
   3. In the PingOne Verify API Mapping field, enter the JSON Pointer syntax for the value of the matching PingOne Verify attribute.

      Learn more in JSON Pointer syntax reference.
   4. In the Action column, click Update.
   5. To add more attributes, repeat steps a-d.

Copyright © 2025 Ping Identity Corporation

4. On the IdP Adapter tab, configure the adapter instance by referring to PingOne Verify IdP Adapter settings reference. Click Next.
5. On the Actions tab, test your connection to PingOne Verify. Resolve any issues that are reported, and then click Next.
6. On the Extended Contract tab, click Next.
7. On the Adapter Attributes tab, for subject, select the Pseudonym checkbox. Click Next.
8. On the Adapter Contract Mapping tab, click Next.
9. On the Summary tab, check and save your configuration:
    - For PingFederate 10.1 or later: click Save.
    - For PingFederate 10.0 or earlier: click Done. On the Manage IdP Adapter Instances tab, click Save.

PingOne Verify IdP Adapter settings reference

Field descriptions for the PingOne Verify IdP Adapter configuration screen.

Standard fields

Field	Description
PingOne Environment	For PingFederate 10.2 and later. Select the PingOne connection that you created in Connecting PingFederate 10.1 or earlier. This field is blank by default.
PingOne Population	For PingFederate 10.2 and later. If a user does not already exist in PingOne, the adapter provisions the user to this PingOne population. Applies only when Provision User is selected. This list is populated after you select the PingOne Environment. This field is blank by default.
Verify App Name	The mobile app name shown in user-facing instructions: `Download the <app name> app for iOS or Android` **Note** If your app is not hosted on the iOS and Android app stores, you can change the user-facing instructions by modifying the `pingone-verify-messages.properties` file included in this integration kit. This value is for display only and does not need to match a value anywhere else. This field is blank by default.

Copyright © 2025 Ping Identity Corporation

Field	Description
Environment ID	For PingFederate 10.1 and earlier. The environment ID that you noted in Connecting PingFederate 10.1 or earlier. This field is blank by default.
Region	For PingFederate 10.1 and earlier. Determines the PingOne API that the adapter communicates with. Select the region that appears on Settings > Environment > Properties in PingOne.
PingFederate Connection Client ID	For PingFederate 10.1 and earlier. The client ID that you noted in Connecting PingFederate 10.1 or earlier. This is required for automatic device pairing. This field is blank by default.
PingFederate Connection Client Secret	For PingFederate 10.1 and earlier. The client secret that you noted in Connecting PingFederate 10.1 or earlier. This is required for automatic device pairing. This field is blank by default.
Population ID	For PingFederate 10.1 and earlier. If a user does not already exist in PingOne, the adapter provisions the user to this PingOne population. Your population ID appears on Identities > Populations in PingOne. Applies only when Provision User is selected. This field is blank by default.

Advanced fields

Field	Description
Test Username	The PingOne username that the adapter uses to test the PingOne Verify connection on the Actions tab. This field is blank by default.
HTML Template Prefix	Identifies the set of HTML templates that the adapter uses to show the authentication status or request a one-time password. If you customize the template file names in the `/server/default/conf/template` directory, enter the new prefix here. You can find a description of the template files in Download manifest. The default value is `pingone-verify`.

Copyright © 2025 Ping Identity Corporation

Field	Description
Messages Files	Identifies the customizable language-pack file that the adapter uses to show messages on the templates. If you customize the `pingone-verify-messages.properties` file name in the `/server/default/conf/language-packs` directory, enter the new name here. The default value is `pingone-verify-messages`.
Error Message Key Prefix	Identifies the error messages in the language-pack file that the adapter uses to show messages on the templates. If you customize the error message names in `/server/default/conf/language-packs/pingone-verify-messages.properties`, enter the new prefix here. The default value is `pingone.verify.error.`.
Provision User	If a user does not already exist in PingOne, the adapter provisions the user to PingOne. This checkbox is selected by default.
Allow Verification Retries	When ID verification fails, this setting determines whether a user can try again. Your PingOne Verify configuration determines the maximum number of retry attempts. This checkbox is cleared by default.
User Not Found Failure Mode	When a user error occurs in PingOne, this setting determines whether the adapter blocks the user's sign-on or registration attempt. User errors include the following: • User is disabled • User does not exist • User cannot be provisioned • The PingOne Verify service is disabled for the user The default selection is `Block user`.
Service Unavailable Failure Mode	When PingOne does not respond, this setting determines whether the adapter blocks the user's sign-on or registration attempt. The default selection is `Bypass authentication`.
Show Success Screens	Determines whether the adapter shows a success page when the verification process is successful. This checkbox is selected by default.

Copyright © 2025 Ping Identity Corporation

Field	Description
Show Error Screens	Determines whether the adapter shows an error page when the verification process fails and the user cannot retry. This checkbox is selected by default.
Show Timeout Screens	Determines whether the adapter shows a "timed out" page when the verification process times out. This checkbox is selected by default.
State Timeout	The amount of time in seconds that the adapter allows for each state of the process when waiting for a user to complete the verification process. The default value is `300`. The minimum value is `60`.
API Request Timeout	The amount of time in milliseconds that PingFederate allows when establishing a connection with PingOne Verify or waiting for a response to a request. A value of `0` disables the timeout. The default value is `5000`.
Proxy Settings	Defines proxy settings for outbound HTTP requests. The default value is System Defaults.
Custom Proxy Host	The proxy server host name to use when Proxy Settings is set to Custom. This field is blank by default.
Custom Proxy Port	The proxy server port to use when Proxy Settings is set to Custom. This field is blank by default.

JSON Pointer syntax reference

JavaScript Object Notation (JSON) Pointer defines a syntax for identifying a specific value within a JSON payload. Using the sample payload and JSON Pointer examples below, identify the attributes that you want to use to populate your attribute contract.

For a complete technical description of JSON Pointer syntax, see JavaScript Object Notation (JSON) Pointer on ietf.org.

Copyright © 2025 Ping Identity Corporation

Example PingOne Verify JSON payload


```
{
    "_links": {
        "self": {
            "href": "https://api.pingone.com/v1/environments/6yyd1b9c-12c8-4995-tg35-10tt2589e26d/users/
bc8806t0-614t-4h04-a497-6cb626o6fa35/verifyTransactions/f99fbf13-7460-4bf6-9b3b-67fb3acda29c/userData/
f99fbf13-7460-4bf6-9b3b-67fb3acda29c"
        },
        "environment": {
            "href": "https://api.pingone.com/v1/environments/6yyd1b9c-12c8-4995-tg35-10tt2589e26d"
        },
        "user": {
            "href": "https://api.pingone.com/v1/users/bc8806t0-614t-4h04-a497-6cb626o6fa35"
        }
    },
    "id": "f99fbf13-7460-4bf6-9b3b-67fb3acda29c",
    "transactionStatus": {
        "status": "SUCCESS"
    },
    "verifiedUserData": {
        "firstName": "Alice",
        "lastName": "DeMarco",
        "birthDate": "1980-05-15",
        "addressStreet": "123 Example Street, Unit 4",
        "addressCity": "Sacramento",
        "addressState": "CA",
        "country": "United States",
        "addressZip": "95813",
        "idNumber": "1234562",
        "expirationDate": "2024-01-17",
        "issueDate": "2019-01-17",
        "idType": "[PassportPicturePage]"
    }
}
```


JSON Pointer syntax

The following attributes are already included in the core contract, so you do not need to add them to the PingOne Verify Response Mappings table. We have included them here to help you understand the JSON Pointer mapping scheme for newly-added PingOne Verify attributes.

Description	JSON Pointer	Example value
Verification transaction status	`/transactionStatus/status`	`SUCCESS`
User's first name (from their photo ID)	`/verifiedUserData/firstName`	`Alice`

Copyright © 2025 Ping Identity Corporation

Description	JSON Pointer	Example value
User's state (from their photo ID)	`/verifiedUserData/addressState`	`CA`
User's date of birth (from their photo ID)	`/verifiedUserData/birthDate`	`1980-01-20`

> **Note**
>
> To populate an attribute with the entire JSON response, leave the attribute mapping field blank.

Authentication setup

By modifying your PingFederate authentication policy to include the PingOne Verify IdP Adapter, you can challenge users to verify their identity as part of the registration process.

This allows a user to access an existing account by verifying their identity based on photo ID.

> **Note**
>
> If you want to include verification as part of the account registration process, see Self-service registration setup.

Adding PingOne Verify to your authentication policy

By modifying your PingFederate authentication policy to include the PingOne Verify IdP Adapter, you can challenge users to verify their identity as part of the authentication process.

About this task

Use these steps to add to an existing authentication policy. You can find general information about configuring authentication policies in Authentication policies in the PingFederate documentation.

Steps

1. In the PingFederate administrative console, go to the Policies tab.

   Choose from:

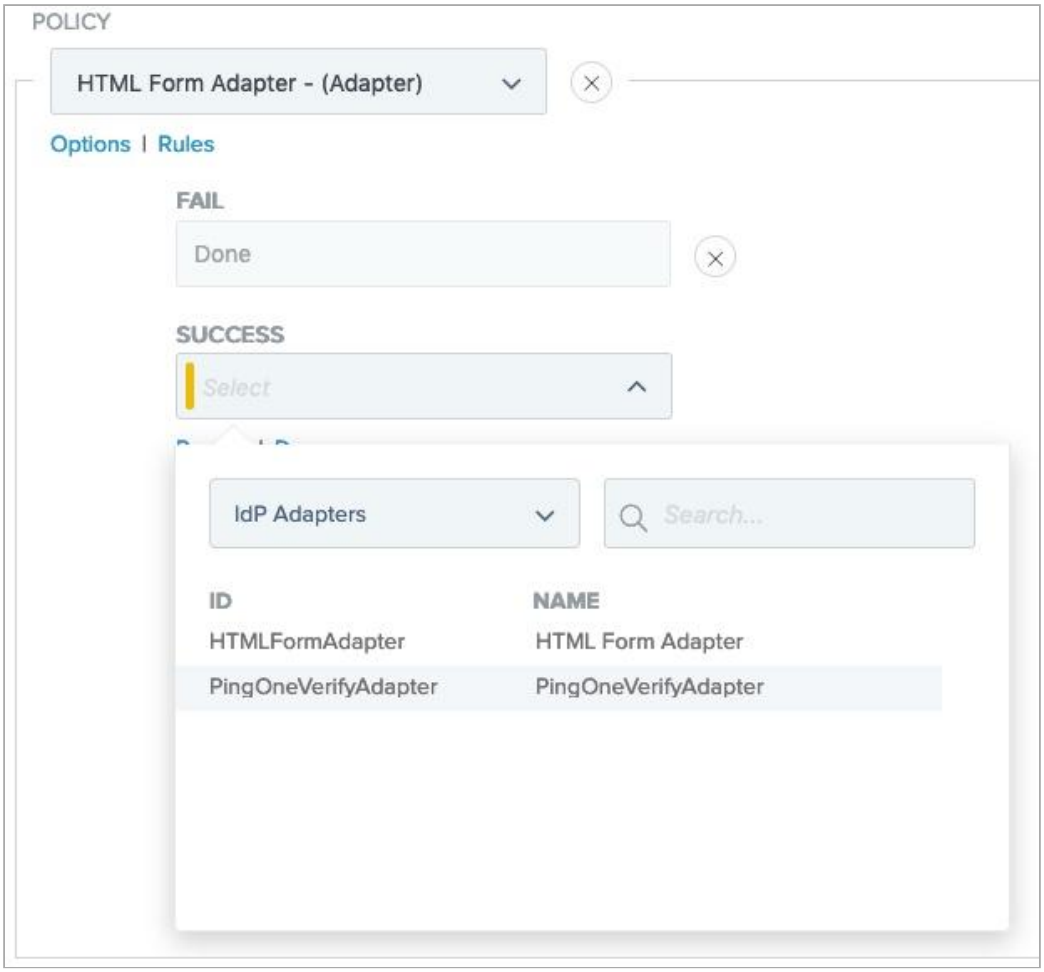
   - For PingFederate 10.1 or later: go to Authentication > Policies > Policies.
   - For PingFederate 10.0 or earlier: go to Identity Provider > Authentication Policies > Policies.

2. Select the IdP Authentication Policies checkbox.
3. Open an existing authentication policy, or click Add Policy.

   For more information, see Defining authentication policies in the PingFederate documentation.

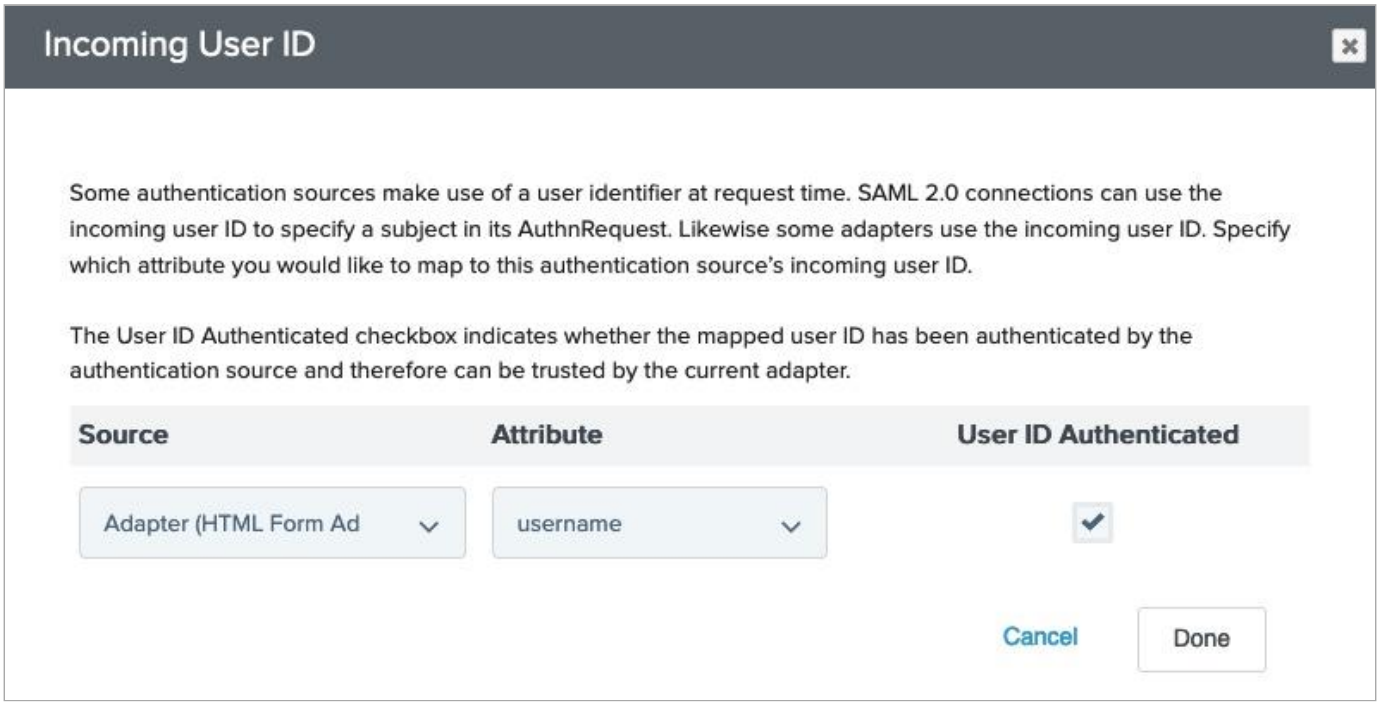
4. In the Policy section, in the Select list, select a PingOne Verify IdP Adapter instance.

Copyright © 2025 Ping Identity Corporation



5. Map the PingOne `user ID` or `username` into the PingOne Verify IdP Adapter instance.

Copyright © 2025 Ping Identity Corporation

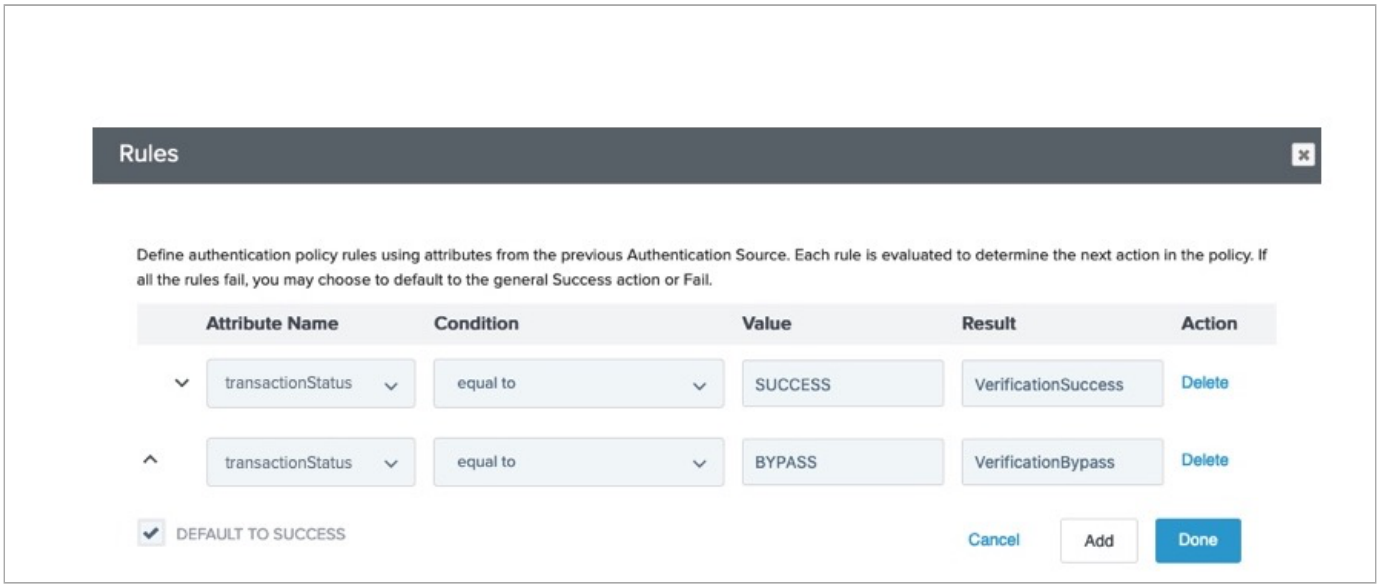


1. Under the PingOne Verify IdP Adapter instance, click Options.
2. On the Options window, in the Source list, select a previous authentication source that collects the PingOne user ID or username.
3. In the Attribute list, select the user ID.
4. For PingFederate 10.2 and later, select the User ID Authenticated checkbox.
5. Click Done.

6. Optional: Define policy paths based on verification results.

> **Tip**
>
> Depending on the failure mode settings in your adapter configuration, the adapter can potentially return a `success` result in the authentication policy even when the user verification process did not succeed. It can be useful to create separate policy paths for a successful adapter result and a successful user verification result.

Copyright © 2025 Ping Identity Corporation



1. Under the PingOne Verify IdP Adapter instance, click Rules.
2. On the Rules window, in the Attribute Name list, select transactionStatus.
3. In the Condition list, select equal to.
4. In the Value field, enter `SUCCESS` or `BYPASS`.

   SUCCESS

   The user successfully verified their identity.

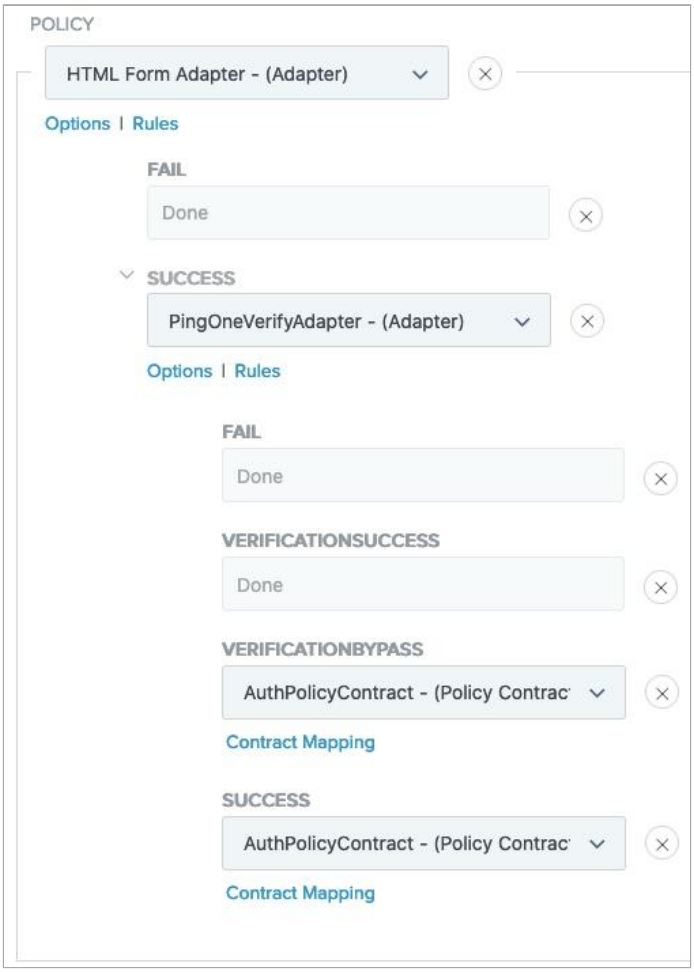
   BYPASS

   An error occurred, and the verification process was not completed, but the adapter is configured to bypass verification and continue the authentication flow.
5. In the Result field, enter a name.

   This appears as a new policy path that branches from the authentication source.
6. To add more authentication paths, click Add and repeat steps a-e.
7. Click Done.

7. Configure each of the authentication paths, including Fail, Success, and any paths that you defined in the Rules window.

Copyright © 2025 Ping Identity Corporation



8. Click Done.

9. In the Policies window, click Save.

Self-service registration setup

By modifying your PingFederate user registration flow to include the PingOne Verify IdP Adapter, you can challenge users to verify their identity as part of the registration process.

This allows a user to register a new account by providing their basic information and verifying their identity based on photo ID.

> **Note**
>
> If you want to include verification as part of the authentication process for existing accounts, learn more in Authentication setup.

Requirements for self-service registration setup

- PingFederate 10.2 or later

Copyright © 2025 Ping Identity Corporation

- An existing user registration configuration, including:
  - PingDirectory
  - An authentication policy contract
  - A local identity profile
  - An HTML Form Adapter instance
  - An authentication policy

> **Important**
>
> Before you continue, complete the steps in Setting up self-service registration in the PingFederate documentation. You will modify this initial configuration in the sections below.

Registration form modifications

Configure your HTML Form Adapter to validate the registration form data before allowing users to continue to the identity verification process.

The template path is: `<pf_install>/pingfederate/server/default/conf/template/local.identity.registration.html`. Learn more about modifying templates in Customizable user-facing pages in the PingFederate documentation.

Name fields

Near the name fields, modify the template to tell users to "Enter the name as shown on your ID card". Users can only register with their verified name.

Field validation

If your datastore has length or character requirements for passwords, for example, add validation to the password field that checks against the same criteria.

You can add form validation by modifying the HTML Form Adapter registration template with custom JavaScript.

Learn more about form validation JavaScript in JavaScript Form Validation on w3schools.com.

Creating input and output policy contracts

Before creating a PingOne Verify policy fragment, create policy contracts to define which attributes are provided to and collected from the adapter.

Copyright © 2025 Ping Identity Corporation

Steps

1. Create an input policy contract.

   You can find more information in Policy contracts in the PingFederate documentation.

   1. In the PingFederate administrative console, go to Authentication > Policies > Policy Contracts.
   2. Click Create New Contract.
   3. On the Contract Info tab, in the Contract Name field, enter a name, such as `PingOneVerifyInput`. Click Next.
   4. On the Contract Attributes tab, add attributes to provide to the PingOne Verify IdP Adapter, such as `email` and `username`. Click Save.

2. Create an output policy contract.

   1. Click Create New Contract.
   2. On the Contract Info tab, in the Contract Name field, enter a name, such as `PingOneVerifyOutput`. Click Next.
   3. On the Contract Attributes tab, add user attributes to get from the PingOne Verify IdP Adapter, such as `email`, `username`, and `status`. Click Save.

Creating a policy fragment

Create a policy fragment that will allow you to include the PingOne Verify IdP Adapter in your user registration flow.

Steps

1. In the PingFederate administrative console, go to Authentication > Policies > Fragments.
2. Click Add Fragment.
3. In the Name field, enter a name, such as `PingOne Verify Fragment`.
4. In the Inputs list, select your input policy contract, such as PingOneVerifyInput.
5. In the Outputs list, select your input policy contract, such as PingOneVerifyOutput.
6. In the Policy area, select a PingOne Verify IdP Adapter instance.
   1. Under the PingOne Verify IdP Adapter instance, click Options.
   2. On the Options dialog, from the Source list, select Inputs.
   3. In the Attribute list, select the attribute that contains the user identifier, such as username.
   4. Select the User ID Authenticated checkbox.
   5. Click Done.
7. For the Fail path, select Done.
8. For the Success path, select your output policy contract, such as PingOneVerifyOutput.

Copyright © 2025 Ping Identity Corporation

9. Configure the policy contract to include attributes from the input as well as from the PingOne Verify IdP Adapter.
    1. Under the output policy contract, click Contract Mapping.
    2. On the Contract Fulfillment tab, map the output attributes to allow the policy fragment to provide information, such as the transaction status and user identifier, to the authentication policy.

    Example contract fulfillment mappings

Source	Value
Inputs	email
Inputs	username
Adapter (PingOne Verify IdP Adapter instance)	transactionStatus
Adapter (PingOne Verify IdP Adapter instance)	subject

    3. Click Done.
10. Click Save.

Modifying your local identity profile

Modify your local identity profile to enable registration using the policy fragment that you created.

About this task

These steps build on the local identity profile that you created as part of Setting up self-service registration in the PingFederate documentation.

Steps

1. In the PingFederate administrator console, go to Authentication > Policies > Local Identity Profiles.
2. Open the local identity profile that you created in your initial self-service registration configuration.
3. Optional: On the Profile Info tab, select the Enable Profile Management checkbox.

   This allows the user to sign on to a profile management page.
4. Optional: Enable email address verification.

   Learn more in Configuring email ownership verification options in the PingFederate documentation.
    1. On the Fields tab, click Create New Field.
    2. In the ID field, enter a name, such as `emailVerified`.
    3. In the Label field, enter a label, such as `Email verification status`.
    4. In the Type list, select Hidden.

Copyright © 2025 Ping Identity Corporation

5. Click Done.
6. On the Email Verification tab, in the Email Address Field, select the email attribute, such as lipEmail.
7. In the Ownership Status Field, select the email verification status attribute, such as emailVerified.
8. On the Data Store Configuration tab, map the new attribute to a data store attribute.

   Learn more in Managing datastore configuration in the PingFederate documentation.

5. On the Registration tab, configure registration to use the policy fragment that you created in Creating a policy fragment.

   Learn more in Configuring registration options in the PingFederate documentation.

   1. In the Username Field list, select the field that you want to use as the username for the authentication session, for example, lipEmail.
   2. In the Registration Workflow list, select the policy fragment that you created in Creating a policy fragment.
   3. For Execute Workflow, select Before Account Creation.
6. On the Profile Management tab, select the Enable Profile Deletion checkbox.
7. Click Save.

Adding PingOne Verify to your user registration flow

By modifying your PingFederate authentication policy to include the PingOne Verify IdP Adapter policy fragment, you can challenge users to verify their identity as part of the self-service registration process.

Steps

1. In the PingFederate administrative console, go to Authentication > Policies > Policies.
2. Select the IdP Authentication Policies checkbox.
3. Open the authentication policy that you created in your initial registration configuration.
4. In the Policy area, define a policy path for user registration.
   1. Under the HTML Form Adapter instance, click Rules.
   2. On the Rules dialog, in the Attribute Name list, select policy.action.
   3. In the Condition list, select equal to.
   4. In the Value field, enter `identity.registration`.
   5. In the Result field, enter a name, such as `Registration`. This appears as a new policy path that branches from the PingOne Verify IdP Adapter.
   6. Click Done.
5. After the HTML Form Adapter instance, for the Fail path, select Done.

Copyright © 2025 Ping Identity Corporation

6. After the HTML Form Adapter instance, for the Registration path, select the local identity profile that you created for PingOne Verify.
    1. Click Local Identity Mapping.
    2. On the Inbound Mapping tab, map the registration form fields to the input policy contract attributes.

Table 1. Example mappings

Inbound Mapping Fulfillment	Value
firstName	regFirstName
email	registrar@example.com

    3. On the Contract Fulfillment tab, map values for the outbound contract.

Table 2. Example mappings

Outbound Mapping Fulfillment	Source	Value
attr1	Local Identity (Verify Identity Profile)	email
subject	Local Identity (Verify Identity Profile)	firstName

    4. Click Done.
7. After the HTML Form Adapter instance, for the Success path, configure the standard (non-registration) authentication path.
8. Click Done.
9. In the Policies window, click Save.

Authentication API Support

You can use the PingFederate authentication API to integrate the PingOne Verify IdP Adapter into your application.

The PingFederate Authentication API provides access to the current state of the authentication flow as a user steps through the PingFederate authentication policy. Learn more in PingFederate authentication API in the PingFederate documentation.

To integrate the PingOne Verify IdP Adapter into your authentication flow, configure your application based on the information in this section.

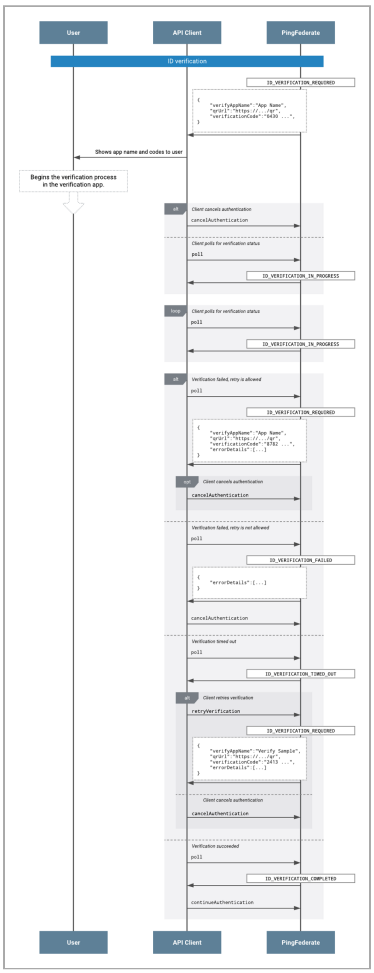
> **Tip**
>
> You can also explore the process using the PingFederate Authentication API Explorer. Learn more in Exploring the Authentication API in the PingFederate documentation.

Copyright © 2025 Ping Identity Corporation

Verification flow

When using the PingOne Verify IdP Adapter through the PingFederate authentication application programming interface (API), the following flow is used for verification requests.

Identity verification



1. The user completes first-factor authentication or begins the registration flow. ID verification begins when using the PingOne Verify IdP Adapter is triggered in the PingFederate authentication or registration flow.
2. PingFederate returns a status of `ID_VERIFICATION_REQUIRED` in the response to the API client and provides a QR code and a verification code.
3. The API client displays the QR code and verification code to the user.
4. The user completes the ID verification process in the verification app.
5. The API client invokes the poll action.
6. PingFederate returns the status of `ID_VERIFICATION_IN_PROGRESS` to the API client.

Copyright © 2025 Ping Identity Corporation

7. The poll action repeats until one of the following statuses occurs:
    - `ID_VERIFICATION_REQUIRED`
        - Verification failed, but a retry is allowed
        - The user can complete (or retry) the verification process with the provided QR code and verification code
        - There are two options available to the API client:
            1. Continue polling by calling `poll` and wait for the status to change
            2. Cancel the verification request by calling `cancelAuthentication`
    - `ID_VERIFICATION_FAILED`
        - Verification failed, and a retry is not allowed
        - The authentication request is cancelled
    - `ID_VERIFICATION_TIMED_OUT`
        - Verification timed out
        - There are two options available to the API client:
            1. Retry by calling `retryVerification`
            2. Cancel the verification request by calling `cancelAuthentication`
    - `ID_VERIFICATION_COMPLETED`
        - Verification succeeded
        - The API client continues the flow by calling `continueAuthentication`
        - PingFederate returns the transaction status and user data to the API client

State models, action models, and attributes

When using the PingOne Verify IdP Adapter through the PingFederate authentication application programming interface (API), the adapter uses the following state models, action models, and attributes.

Table 1. State models

Status	State Model	Action	Description
`ID_VERIFICATION_REQUIRED`	• `verifyAppName` • `qrUrl` • `verificationCode` • `errorDetails`	• `poll` • `cancelAuthentication`	Indicates that verification is required. This state is the first state returned from the adapter.

Copyright © 2025 Ping Identity Corporation

Status	State Model	Action	Description
`ID_VERIFICATION_IN_PROGRESS`		• `poll`	Indicates that ID verification is in progress.
`ID_VERIFICATION_COMPLETED`		• `continueAuthentication`	Indicates that ID verification completed successfully.
`ID_VERIFICATION_FAILED`	• `errorDetails`	• `cancelAuthentication`	Indicates that ID verification failed and a retry is not allowed.
`ID_VERIFICATION_TIMED_OUT`		• `retryVerification` • `cancelAuthentication`	Indicates that ID verification timed out.

Action models

Action	Action Model	Error	Description
`poll`	(void)		Gets the current ID transaction status.
`retryVerification`	(void)		Retries verification after a time out. Retry must be enabled in the adapter instance configuration.

Attribute models

Attribute name	Type	Description
`verifyAppName`	String	The name of the verification app. This name is provided from the Verify App Name field in the adapter instance configuration. You can use this name to help customers find the right verification app in the Google Play Store or Apple App Store.

Copyright © 2025 Ping Identity Corporation

Attribute name	Type	Description
`qrUrl`	String	The URL of the QR code image associated with the verification session. Use this URL to show the QR code to the user. The user scans the QR code with the verification app to begin the verification process. To display the QR code, your app needs to perform a GET request to the URL provided in `qrUrl`. Include an <img> element, and use the result of the GET request as the source. QR URL provided by PingFederate: `"qrUrl":"https://api.pingone.com/v1/idValidations/shortcode/008030285946/qr"` GET request to get the QR code image: `GET "https://api.pingone.com/v1/idValidations/shortcode/008030285946/qr"` GET request response: `...2wBDAQkJCQwLDBgNDRgyIRwhMjIyMjIyMj` HTML element that displays the image in your app: `<img src="data:image/jpegbase64,9j4AAQSkZJRgABAgAAAQABAADIsIxwcKDcpLDAxNDQ0Hyc5PTgyPC4zNDL...2wBDAQkJCQwLDBgNDRgyIRwhMjIyMjIyMj"/>`
`verificationCode`	String	A code that the user can enter instead of scanning the QR code image.
`errorDetails`	List< `CustomAuthnErrorDetail` >	A list of errors.

Troubleshooting

Enabling debug logging

To help with troubleshooting or monitoring, you can turn on activity logging for the adapter.

About this task

Copyright © 2025 Ping Identity Corporation

These steps are optional. You can find more general information about logging in Enabling debug messages and console logging in the PingFederate documentation.

Steps

1. Open the `<pf_install>/pingfederate/server/default/conf/log4j2.xml` file for editing.
2. If you want to log activity for PingFederate and all adapters, do the following.
    1. Find the following section:

```
<AsyncRoot level="INFO" includeLocation="false">
        <!-- <AppenderRef ref="CONSOLE" /> -->
        <AppenderRef ref="FILE" />
</AsyncRoot>
```

    2. Change `INFO` to `DEBUG`.

```
<AsyncRoot level="DEBUG" includeLocation="false">
        <!-- <AppenderRef ref="CONSOLE" /> -->
        <AppenderRef ref="FILE" />
</AsyncRoot>
```

    3. If you want to see the adapter activity in the console, remove the comment tags.

```
<AsyncRoot level="INFO" includeLocation="false">
        <AppenderRef ref="CONSOLE" />
        <AppenderRef ref="FILE" />
</AsyncRoot>
```

3. If you want to log activity just for the PingOne Verify IdP Adapter, add the following line.

```
<Logger name="com.pingidentity.adapters.pingone.verify" level="DEBUG"/>
```

4. Save the file.

Release notes

Changelog

The following is the change history for the PingOne Verify Integration Kit.

Copyright © 2025 Ping Identity Corporation

PingOne Verify Integration Kit 1.1 – July 2021

Added support for changes to PingOne Verify, including:

- The adapter now gets the user's `verifyEnabledState` attribute when getting the verification result
- Updated the list of core contract attributes to support changes to the structure of the `verifiedUserData` attribute
- Added support for change to the flow behavior when the `verifyDocument` flag is off
- Updated the core contract to support changes in PingOne Verify:

Change description	Version 1.0	Version 1.1
Attribute name change	`verificationStatus`	`transactionStatus`
Attribute name change	`dateOfBirth`	`birthDate`
Attribute name change/merge	`addressLine1`	`addressStreet`
Attribute name change/merge	`addressLine2`	
Attribute name change	`city`	`addressCity`
Attribute name change	`state`	`addressState`
Attribute name change	`postalCode`	`addressZip`
Attribute name change	`documentId`	`idNumber`
New attribute		`idType`

> **Tip**
>
> For upgrade instructions, see Deploying the integration files.

PingOne Verify Integration Kit 1.0 – January 2021

- Initial release.
- Added the ability to direct users to an identity verification app as part of a sign-on or registration flow.
- Added the ability to control which message pages (templates) are shown to the user.
- Added the ability to customize template messages using a language pack file.
- Added the ability to test the connection to PingOne Verify.
- Added a setting to automatically provision new users to PingOne.
- Added settings to control how the adapter handles sign-on and registration attempts when errors occur.
- Added support for the platform connection to PingOne introduced in PingFederate 10.2.

Copyright © 2025 Ping Identity Corporation

- Added support for the PingFederate authentication API
- Added support for the JavaScript Widget for the PingFederate Authentication API.
- Added settings for API connection and request timeouts.
- Added settings to override the PingFederate system-default proxy settings.

Known issues and limitations

The following are known issues or limitations for the PingOne Verify Integration Kit.

Known issues

There are no known issues.

Known limitations

There are no known limitations.

Download manifest

The following files are included in the PingOne Verify Integration Kit `.zip` archive:

- `ReadMeFirst.pdf` – Contains links to this online documentation
- `Legal.pdf` – Copyright and license information
- `dist/pingfederate/server/default` – Contains the integration files
  - `deploy` – Contains the Java libraries
    - `pf-pingone-verify-adapter-<version>.jar` – JAR file that contains the PingOne Verify IdP Adapter.
  - `conf` – contains the HTML template that presents the PingOne sign-on form.
    - `language-packs` – Contains files with customizable user-facing messages
      - `pingone-verify-messages.properties` – a variable file that customizes the messages that appear on the template file.
    - `template` – contains user-facing HTML template files
      - `pingone-verify-completed.html` – a sample template that indicates the verification step completed successfully.
      - `pingone-verify-failed.html` – a sample template that indicates that the user failed the verification challenge.
      - `pingone-verify-required.html` – a sample template that asks the user to complete the verification challenge.

Copyright © 2025 Ping Identity Corporation

- `pingone-verify-timed-out.html` – a sample template that indicates that the verification challenge took too long.
- `assets` – contains functional scripts and files used by the template
  - `css` – contains CSS files for the templates.
    - `pingone-verify.css` – a CSS file that customizes the appearance of the template files.
    - `end-user/<version>/end-user.css` – a CSS file that customizes the appearance of the template files.
  - `fonts/end-user` – contains template fonts and icons
    - `icons` – contains template icons
    - `proxima-nova` – contains template fonts
  - `images` – contains template image files
    - `ping-logo.svg` – an image file with company branding
    - `spinner.svg` – an image file used in a spinner animation
  - `scripts` – contains script files used to collect and send information
    - `jquery-<version>.min.js` – a JavaScript file that contains the jQuery library.
- `lib/pf-authn-api-sdk-<version>.jar` – a JAR file that contains the PingFederate Authentication API SDK

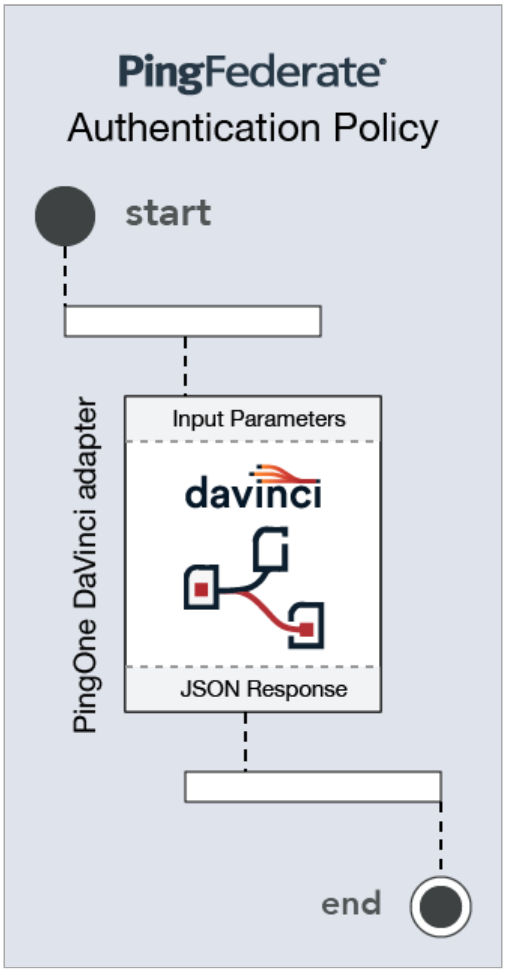
PingOne DaVinci Integration Kit

The PingOne DaVinci Integration Kit allows PingFederate to use PingOne as an identity provider (IdP).

Features

Embeds PingOne flows in your PingFederate authentication policy.

Copyright © 2025 Ping Identity Corporation



Components

PingOne DaVinci IdP Adapter

- Allows PingFederate to use DaVinci to process sign-on requests and get user information.
  - Learn more about the DaVinci widget in the widget method section in the PingOne for Developers documentation.
  - Learn more about using the DaVinci API in the API method section.

Templates

- Allows the adapter to show the DaVinci widget.

Language packs

- Allows you to customize or localize the messages shown on the templates during authentication. Learn more in Localizing messages for end users in the PingFederate documentation.

Intended audience

This document is intended for PingFederate administrators.

Copyright © 2025 Ping Identity Corporation

If you need help during the setup process, learn more in the following sections of the PingFederate documentation:

- Identity provider SSO configuration
- Managing IdP adapters
- Authentication policies
- SP connection management

System requirements

- PingFederate 10.3 or later.

Overview of the SSO flow

With the PingOne DaVinci Integration Kit, PingFederate integrates DaVinci into the sign-on flow.

Description

1. The adapter makes a back-channel POST call to PingOne with the application programming interface (API) key and target ID. API-based flows use the API key directly to authenticate the request. Widget-based flows include parameters to request an access token.
2. For widget-based flows, the adapter renders the widget using the access token.
3. DaVinci handles the flow in PingOne. After the flow completes, a JSON response is sent back to the adapter.

> **Note**
>
> If using the adapter's full-page redirect ability, the flow might look different depending on your configuration.

4. PingFederate redirects the user to the web application with the user attributes from the response.

Setup

Adding an application in DaVinci

Create an application to associate your PingFederate integration with your flow.

Steps

1. Sign on to the DaVinci admin portal.
2. Create a blank flow:
    1. Go to Flows and click + Add Flow.
    2. Select Blank Flow.
    3. On the Add Flow dialog, enter a name for your flow. Click Create.

Copyright © 2025 Ping Identity Corporation

4. Complete the flow as described in Building a flow in DaVinci.

3. Add an application:

    1. Go to Applications.
    2. Click + Add Application
    3. On the Add Application dialog, enter a name for your application. Click Create.
    4. Click your application.

4. Add a flow policy to your application:

    1. On the Flow Policy tab, click + Add Flow Policy.
    2. Select the flow that you made in step 2. Select Latest Version or another version. Click Create My Flow Policy.
    3. On the Edit Your Weight Distribution dialog, in the Analytics - Select Success Nodes list, select your Send success JSON response node. Click Save My Flow Policy.
    4. On the Flow Policy tab, note your Policy ID. You'll use this to configure the DaVinciadapter.

5. On the General tab, note your Company ID and API Key. You'll use these to configure the DaVinci adapter.

Deploying the integration files

To get started with the integration, deploy the PingOne DaVinci Integration Kit files to your PingFederate directory.

Before you begin

> **Important**
>
> If upgrading from PingOne DaVinci Integration Kit Beta, you must delete any Beta adapter configuration and existing files from `/conf` related to the PingOne DaVinci Integration Kit before updating PingFederate with content from the new `.zip`.

> **Important**
>
> If running the kit after an upgrade results in a JSON schema validation error, you must add the `nonce` parameter to the input schema in DaVinci. For more information, see Building a flow in DaVinci.

About this task

> **Note**
>
> If you operate PingFederate in a cluster, the following steps refer to the console node.

Steps

1. Download the PingOne DaVinci Integration Kit `.zip` archive from the Ping Identity Integration Directory.
2. Stop PingFederate.

Copyright © 2025 Ping Identity Corporation

3. If you're upgrading an existing deployment, back up your customizations and delete earlier versions of the integration files:
    1. Back up any PingOne DaVinci Integration Kit files that you customized in the `<pf_install>/pingfederate/server/default/conf/` directory.
    2. Delete the `pf-pingone-davinci-adapter-<version>.jar` file from your `<pf_install>/pingfederate/server/default/deploy` directory.
4. Extract the `.zip` archive and merge the contents of the `dist` directory with your `<pf_install>/pingfederate` directory.
5. If you backed up any customized files, modify the new files with your customizations.
6. Start PingFederate.
7. If you operate PingFederate in a cluster, repeat steps 2-6 for each engine node.

Configuring an adapter instance

Configure the PingOne DaVinci IdP Adapter to determine how PingFederate communicates with PingOne DaVinci.

Steps

1. In the PingFederate administrative console, create a new IdP adapter instance:

    Choose from:

    - For PingFederate 10.1 or later: go to Authentication > Integration > IdP Adapters. Click Create New Instance.
    - For PingFederate 10.0 or earlier: go to Identity Provider > Adapters. Click Create New Instance.
2. On the Type tab, set the basic adapter instance attributes.
    1. In the Instance Name field, enter a name for the adapter instance.
    2. In the Instance ID field, enter a unique identifier for the adapter instance.
    3. From the Type list, select PingOne DaVinci IdP Adapter. Click Next.
3. Optional: On the IdP Adapter screen, in the Simple Parameter Mappings section, select pre-defined parameters to send from PingFederate to DaVinci.

    > **Note**
    >
    > The adapter sends your **Simple Parameter Mappings** and **Advanced Parameter Mappings** to DaVinci. In DaVinci, you can access these parameters by configuring your flow's input schema to expect the **DaVinci Parameter Name** values you define here.

    > **Note**
    >
    > For more information about other input parameter options, see the Parameter reference in the PingFederate SDK documentation and Extended properties in the PingFederate documentation.

    1. Click Add a new row to 'Simple Parameter Mappings (optional)'.

Copyright © 2025 Ping Identity Corporation

2. In the Source list, select the pre-defined parameter you want to send.
3. In the DaVinci Attribute field, enter a key name to use when sending the parameter to PingOne.
4. In the Action column, click Update.
5. To add more attributes, repeat steps a - d.

4. (Optional) On the IdP Adapter screen, in the Advanced Parameter Mappings section, define custom parameters to send from PingFederate to DaVinci.
    1. Click Add a new row to 'Advanced Parameter Mappings (optional)'.
    2. In the Source Type list, select the type of parameter you want to send.

        Chained Attributes

        The attributes that are made available by other adapters and selectors in your PingFederate authentication policy.

        Extended Properties

        These parameters store additional information about connections, OAuth clients, or both. You can find details in Extended properties in the PingFederate documentation.

        Request Claims

        The claims PingFederate received within an OAuth/OpenID Connect Request Object or the parameters of a pushed authorization request.

        Tracked HTTP Request Parameters

        The tracked HTTP request parameters that were included in the initial HTTP request of the current transaction.

    3. In the Source Parameter field, enter the exact name of the parameter you want to send to DaVinci. The parameter must be available to the adapter from the Source Type that you selected.
    4. In the PingOne Attribute field, enter a key name to use when sending the parameter to DaVinci.
    5. In the Action column, click Update.
    6. To add more attributes, repeat steps a-e.
5. (Optional) On the IdP Adapter screen, in the DaVinci Response Mappings section, map attributes from the PingOne response to the attribute contract.

    These attributes become available in your PingFederate authentication policy.

    > **Note**
    >
    > Before using the table, check the **Extended Contract** tab to see which attributes are already included in the core contract.

    1. Click Add a new row to 'DaVinci Response Mappings (optional)'.

Copyright © 2025 Ping Identity Corporation

2. In the Local Attribute field, enter a name for the new attribute.
3. In the DaVinci Attribute Mapping field, enter the JSON Pointer syntax for the value of the matching PingOne attribute.

   For help, see JSON Pointer syntax reference.
4. In the Action column, click Update.
5. To add more attributes, repeat steps a-d.

6. On the IdP Adapter screen, enter the values for Company ID, Policy ID, API Key, and Subject Attribute Path.

> **Note**
>
> You noted the values for **Company ID**, **Policy ID**, and **API Key** when you created the application in Adding an application in DaVinci.
> In Building a flow in DaVinci, you gave the **Name of the field containing the additional properties** field a value of `additionalProperties`. We added the `username` property to `additionalProperties`, so the **Subject Attribute Path** value becomes `/additionalProperties/username`.

7. On the Extended Contract tab, add any Local Attributes that you added in the DaVinci Response Mappings table. Click Next.
8. On the Adapter Attributes tab, set pseudonym and masking options as shown in Set pseudonym and masking options in the PingFederate documentation. Click Next.
9. On the Adapter Contract Mapping tab, configure the contract fulfillment details for the adapter as shown in Define the IdP adapter contract in the PingFederate documentation. Click Next.
10. On the Summary tab, check and save your configuration:

    Choose from:

    - For PingFederate 10.1 or later: click Save.
    - For PingFederate 10.0 or earlier: click Done. On the Manage IdP Adapter Instances tab, click Save.

PingOne DaVinci IdP Adapter settings reference

Field descriptions for the PingOne DaVinci IdP Adapter configuration screen.

> **Important**
>
> If you're using an EU, Canada, or Asia DaVinci tenant, you must update the **DaVinci API URL**, **DaVinci Auth URL**, and **DaVinci Widget URL**. Replace `.com` with `.eu` for EU tenants, `.ca` for Canada tenants, or `.asia` for Asia tenants. Failure to do so results in an error in the PingFederate logs.

Copyright © 2025 Ping Identity Corporation

Standard fields

Field	Description
Company ID	**The company ID that you noted in Adding an application in DaVinci.**
Policy ID	**The policy ID that you noted in Adding an application in DaVinci.**
Flow Type	**The type of DaVinci flow that you're using: Widget-based flow or API-based flow.** ***Widget-based flows*** **Designed for user interaction. They can, for example, redirect the browser, present HTML forms or screens, prompt for MFA, or display a message. If your integration requires user interaction, select Widget-based flow.** ***API-based flows*** **Run silently and don't present any opportunity for user interaction. They execute more quickly than widget-based flows (when API-based flow is selected).** **API-based flows can, for example, run a risk analysis, check entitlements, or request an authorization policy decision. If your integration doesn't require user interaction and must be highly performant, select API-based flow.**
API Key	**The API key that you noted in Adding an application in DaVinci.**
Subject Attribute Path	**The attribute from DaVinci to use as the subject in the contract.** **Map using JSON Pointer (RFC 6901) syntax.**

Advanced fields

Field	Description
Additional Properties Attribute	**The name of the attribute that contains the additional properties from DaVinci.** **The default value is `additionalProperties`.**

Copyright © 2025 Ping Identity Corporation

Field	Description
Nonce Attribute Path	**The attribute sent back from PingOne DaVinci for nonce validation.** **Map using JSON Pointer (RFC 6901) syntax.** **The default value is `/parameters/nonce`.**
HTML Template	**Identifies the HTML template the adapter uses. If you customized the template file name in `/server/default/conf/template`, enter the new name here.** **The default value is `pingone-davinci.html`.** **Used for widget-based flows.**
Messages File	**Identifies the customizable language-pack file that the adapter uses. If you customized the language-pack file name in `/server/default/conf/language-packs`, enter the new name here.** **The default value is `pingone-davinci-messages`.** **Used for widget-based flows.**
DaVinci API URL	**The DaVinci API URL.** **The default value is `https://orchestrate-api.pingone.com/`.**
DaVinci Auth URL	**The DaVinci Auth base URL.** **The default value is `https://auth.pingone.com/`.**
DaVinci Widget URL	**The DaVinci widget JS URL. Used for widget-based flows.** **The default value is `https://assets.pingone.com/davinci/latest/davinci.js`.** **Note** To determine whether you need to add the DaVinci widget URL to the allow list for your content security policy (CSP) header, check the `response-header-runtime-config.xml` file in the `<pf_install>/pingfederate/server/default/data/config-store` directory. Learn more in this support article.
Authorization Callback Endpoint	**The PingFederate endpoint that DaVinci uses to respond to authorization requests. Used for widget-based flows.** **Note** The **Application Return To URL** setting in the DaVinci connector configuration is `<PingFederate Base URL>/ext/<Authorization Callback Endpoint>`. **The default value is `/davinci-authn`.**

Copyright © 2025 Ping Identity Corporation

Field	Description
Service Unavailable Failure Mode	Determines whether the adapter should block the user's sign-on attempt or bypass when the adapter is unable to access PingOne. **Important** Setting to bypass could result in a security risk if the flow is not configured properly.
API Request Timeout	The amount of time in milliseconds that PingFederate waits for DaVinci to respond to requests. A value of `0` disables the timeout. The default value is `5000`.
Proxy Settings	Defines proxy settings for outbound HTTP requests. The default value is System Defaults. **Note** To confirm whether you're using an outbound proxy, check the `run.properties` file of your PingFederate instance. Outbound proxies are defined by host name and port using the `http.proxyHost` and `http.proxyPort` parameters (and their HTTPS equivalents). If an outbound proxy is configured, select **Custom** and specify the host name and port in **Custom Proxy Host** and **Custom Proxy Port**, respectively.
Custom Proxy Host	The proxy server host name to use when Proxy Settings is set to Custom. This field is blank by default.
Custom Proxy Port	The proxy server port to use when Proxy Settings is set to Custom. This field is blank by default.

JSON Pointer syntax reference

JavaScript Object Notation (JSON) Pointer defines a syntax for identifying a specific value within a JSON payload. Using the sample payload and JSON Pointer examples below, identify the attributes that you want to use to populate your attribute contract.

For a complete technical description of JSON Pointer syntax, see JavaScript Object Notation (JSON) Pointer on ietf.org.

Copyright © 2025 Ping Identity Corporation

Example DaVinci JSON payload

```
"response": {
    "companyId": "A70vYu1GnH3OlRByrxWWA70vYu0sktst",
    "flowId": "66d90d8a8eb0aa5e5d8a8eb9579a767d",
    "interactionId": "0342a92b5954586eb7e7a0d8a8eb90fa7637dc8d8a8eb948448dd2a6d763a0d",
    "success": true,
    "parameters": {
        "myparameter": "user.exampleparameter"
    },
    "additionalProperties": {
        "myproperty": "user.exampleproperty"
    }
}
```

JSON Pointer syntax

Description	JSON Pointer	Example value
Success status	`/success`	`true`
Parameters	`/parameters/myparameter`	`user.exampleparameter`
Additional properties	`/additionalProperties/myproperty`	`user.exampleproperty`

> **Note**
>
> To populate a parameter with the entire JSON response, leave the **PingOne Attribute Mapping** field blank.

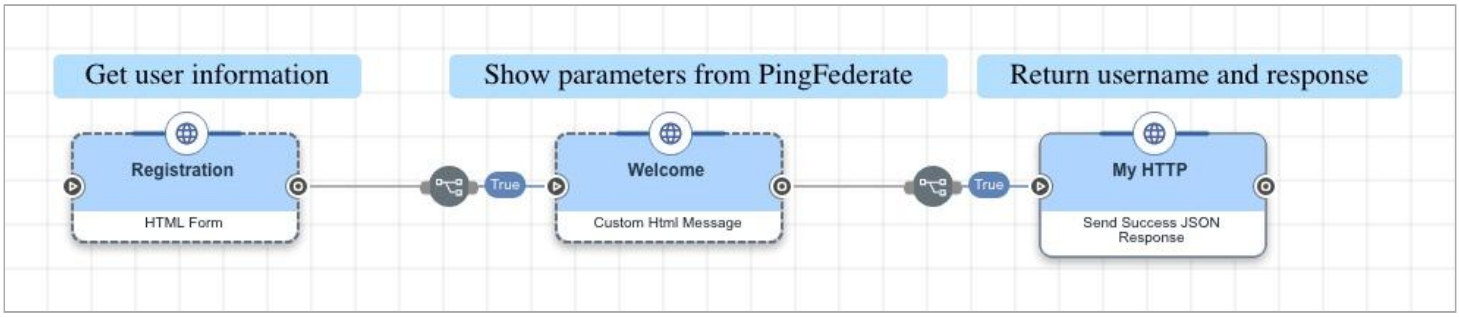
Building a flow in DaVinci

To integrate with PingFederate, configure your DaVinci flow to expect the parameters that you defined in the PingOne DaVinci IdP Adapter and end the flow with a JavaScript Object Notation (JSON) response.

About this task

> **Note**
>
> The following example builds and configures a widget-based DaVinci flow for integration with PingFederate. To show that the integration works, it displays the passed PingFederate values in a user-facing page.
> Flows can have different user-facing elements or none at all, depending on whether you build a widget-based flow or an API-based flow.

Copyright © 2025 Ping Identity Corporation

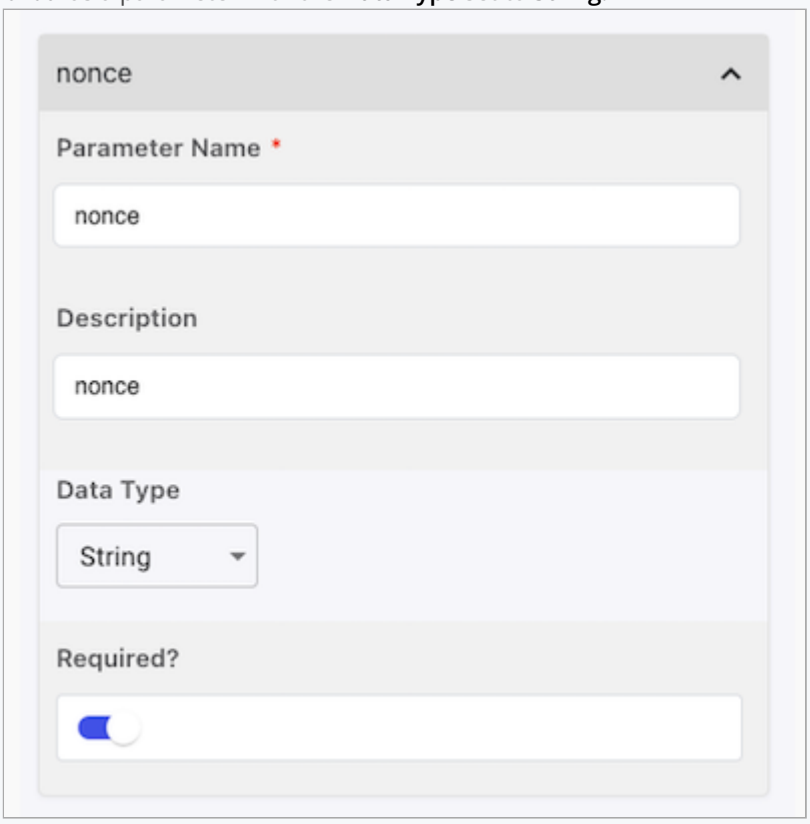


Steps

1. In the DaVinci admin portal, go to Flows, and open the blank flow that you created in Adding an application in DaVinci.
2. Add PingFederate to your flow's input schema.
    1. Go to Flow Options (⋮) > Input Schema.
    2. Click Add.
    3. In the Parameter Name field, enter the exact name that you defined in the DaVinci Parameter Name column in your DaVinci adapter configuration.

    Important

    You must add `nonce` as a parameter with the **Data Type** set to **String**.

    4. Repeat steps b-c for each parameter that you defined in the Simple Parameter Mappings and Advanced Parameter Mappings tables.
    5. Click Save.

Copyright © 2025 Ping Identity Corporation

3. Optional: If you're building a widget-based flow, create a registration form.

> **Note**
>
> This example node represents the functional part of the flow. When you build your own flow, this step is where you customize the flow.

1. On the flow canvas, add the HTTP connector and select the HTML Form capability. Select the node that appears in your flow.
2. In the Title field, enter a title, such as `Registration`.
3. In the Fields List section, add fields for the following:
   - Username
   - Password

   > **Tip**
   >
   > For help, see the HTTP connector.
4. In the Next Button Text field, enter `Sign Up`.
5. Click Apply.

4. Optional: For widget-based flows, show the parameters DaVinci received from PingFederate.

> **Note**
>
> This node is for demonstration purposes only. You don't need it when you build your own flow.

1. Following your HTML Form node in your flow, add the HTTP connector and select the Custom HTML Message capability. Select the node that appears in your flow.
2. In the Title field, enter a title, such as `Welcome`.
3. In the Message field, enter the following:

```
Confirm your information:

Chained Attributes

Specified chained val:

Extended Properties

Specified extended array:

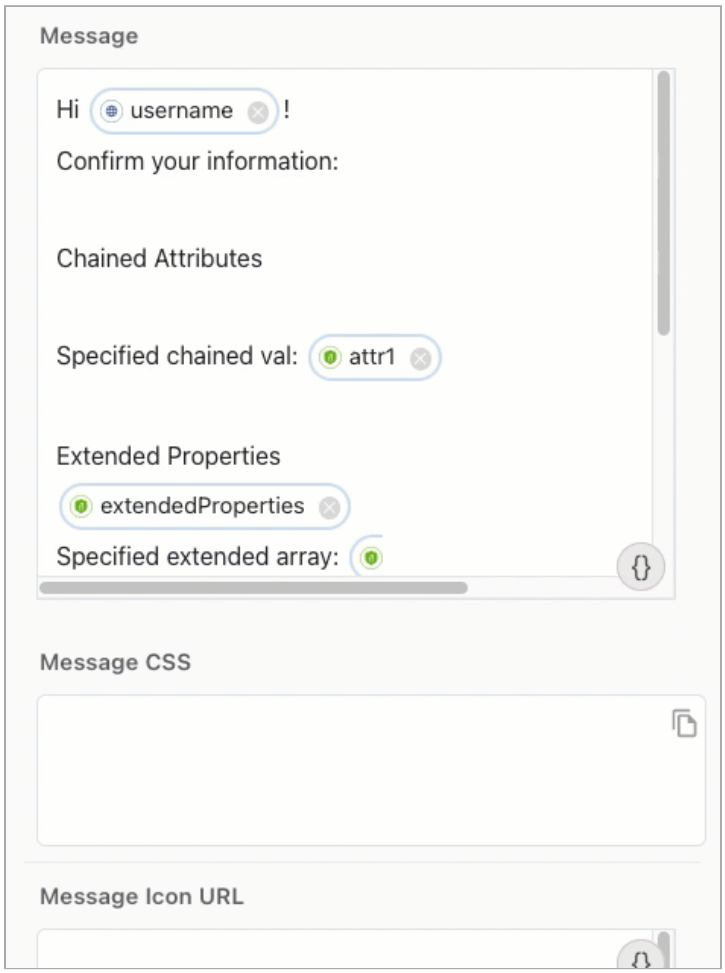
PAR Object

Specified PAR:

Server Base URL:
```

Copyright © 2025 Ping Identity Corporation

4. Populate the message by clicking {} and inserting variables from your flow's input schema.



5. In the Next Button Text field, enter `Sign Up`.

6. Click Apply.

5. Send a JSON response back to PingFederate.

> **Note**
>
> When you build your own flow, make sure that you end the flow with this node.

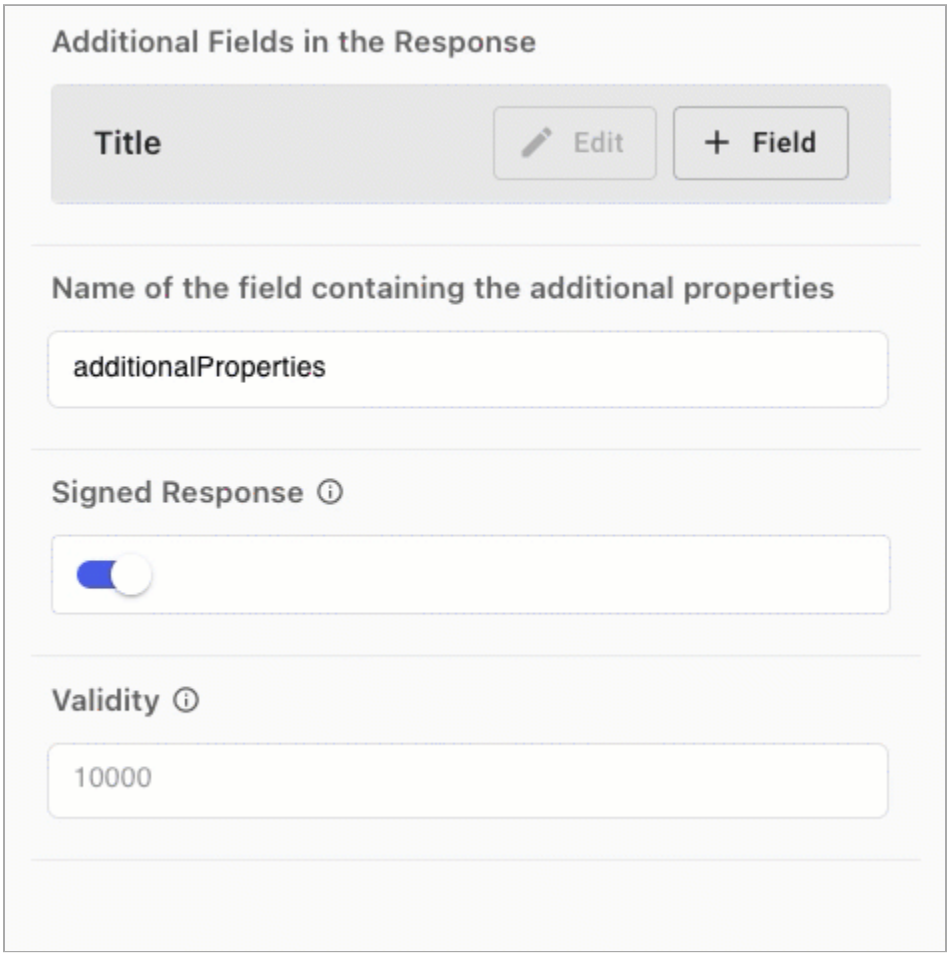
1. Following the Custom HTML Message node in your flow, add the HTTP connector and select the Send Success JSON Response capability. Select the node that appears in your flow.

2. Make sure that Return Request Parameters is enabled.

3. In the Additional Fields in the Response section, click + Field.

4. In the Value field, click \{} and select the username variable from your HTML Form node.

Copyright © 2025 Ping Identity Corporation



5. In the Name of the field containing the additional properties field, enter `additionalProperties`.

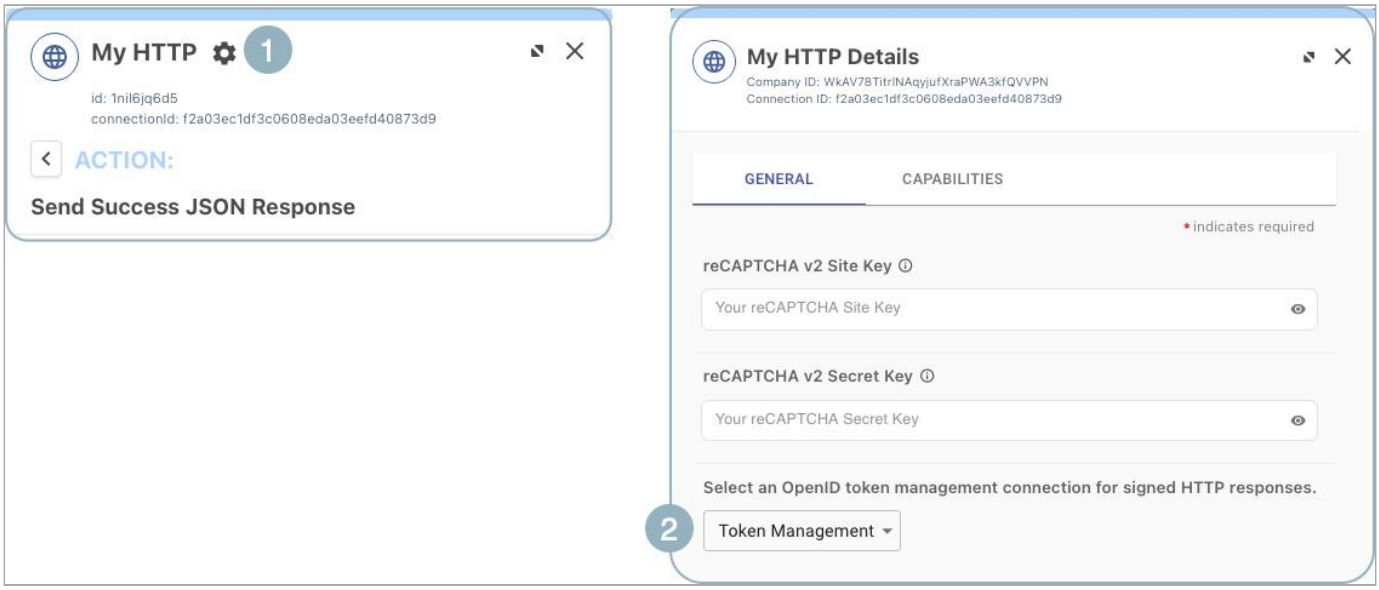
   If you customized the Additional Properties Attribute field in your PingOne DaVinci IdP Adapter configuration, enter the same value here.

6. Make sure Signed Response is enabled.

7. Click Apply.

6. If you're building a widget-based flow, on the HTTP node, click the Gear icon to open the connector configuration. In the Select an OpenID token management connection for signed HTTP responses list, select Token Management.

Copyright © 2025 Ping Identity Corporation



> **Note**
>
> If your list doesn't have a token management connector to select, save your flow, go to **Connections** > **New Connection**, and then add a **Token Management** connection.

7. Save and deploy your flow.

Troubleshooting

Enabling debug logging

To help with troubleshooting or monitoring, you can turn on activity logging for the adapter.

About this task

These steps are optional. You can find more general information about logging in Enabling debug messages and console logging in the PingFederate documentation.

Steps

1. Open the `<pf_install>/pingfederate/server/default/conf/log4j2.xml` file for editing.
2. If you want to log activity for PingFederate and all adapters, do the following.
    1. Find the following section:

```
<AsyncRoot level="INFO" includeLocation="false">
        <!-- <AppenderRef ref="CONSOLE" /> -->
        <AppenderRef ref="FILE" />
</AsyncRoot>
```

    2. Change `INFO` to `DEBUG`.

Copyright © 2025 Ping Identity Corporation

```
<AsyncRoot level="DEBUG" includeLocation="false">
        <!-- <AppenderRef ref="CONSOLE" /> -->
        <AppenderRef ref="FILE" />
</AsyncRoot>
```

3. If you want to see the adapter activity in the console, remove the comment tags.

```
<AsyncRoot level="INFO" includeLocation="false">
        <AppenderRef ref="CONSOLE" />
        <AppenderRef ref="FILE" />
</AsyncRoot>
```

3. If you want to log activity just for the PingOne DaVinci IdP Adapter, add the following line.

```
<Logger name="com.pingidentity.adapters.pingone.davinci" level="DEBUG"/>
```

4. Save the file.

Release notes

Changelog

The following is the change history for the PingOne DaVinci Integration Kit.

PingOne DaVinci Integration Kit 1.3 - April 2025

- Added the ability to use the default PingFederate template parameters, including extended properties, in the `pingone-davinci.html` template. Learn more in the updated template file and Customizable user-facing pages in the PingFederate documentation.

PingOne DaVinci Integration Kit 1.2.1 - July 2024

- Fixed an issue that caused an unexpected system error when using a DaVinci connector that redirected to PingFederate.

PingOne DaVinci Integration Kit 1.2 – July 2023

- Added the ability to select API-based DaVinci flows for a highly performant integration to support use cases that don't require user interaction. Use the new Flow Type adapter setting to specify either widget-based or API-based DaVinci flows.

PingOne DaVinci Integration Kit 1.1.2 – June 2023

- Fixed an issue with error handling in certain scenarios with the HTTP connector.

Copyright © 2025 Ping Identity Corporation

PingOne DaVinci Integration Kit 1.1.1 – April 2023

- Fixed an issue that caused an inconsistent response error.
- Added support to custom DaVinci domains.

PingOne DaVinci Integration Kit 1.1 – March 2023

- Added support for the Nonce Attribute Path field.
- Resolved a potential security vulnerability.

PingOne DaVinci Integration Kit 1.0.1 – November 2022

- Fixed an issue that caused the adapter to return an error during an SLO event.

PingOne DaVinci Integration Kit 1.0 – August 2022

- Initial release.
- Added support for DaVinci connectors with redirect.
- Added the ability for the adapter to bypass or fail when PingOne is unavailable.
- Added the ability to send specific attributes to send to DaVinci using simple and advanced parameter definitions.
- Added the raw response from DaVinci to the log in DEBUG mode.

Known issues and limitations

The following are known issues or limitations for the PingOne DaVinci Integration Kit.

Known issues

There are no known issues.

Known limitations

There are no known limitations.

Download manifest

The following files are included in the PingOne DaVinci Integration Kit `.zip` archive:

- `Legal.pdf` – copyright and license information.
- `dist/pingfederate/server/default` – Contains the integration files
    - `deploy` – contains the Java libraries
        - `pf-pingone-davinci-adapter-<version>.jar` – JAR file that contains the PingOne DaVinci IdP Adapter.

Copyright © 2025 Ping Identity Corporation

- `conf` – contains the HTML template that presents the PingOne sign-on form.
  - `language-packs` – Contains files with customizable user-facing messages
    - `pingone-davinci.properties` – a variable file that customizes the messages that appear on the template files.
  - `template` – contains user-facing HTML template files
    - `pingone-davinci.html` – a template that opens a pop-up window to prompt the user to sign on

PingAM Integration Kit

The PingAM Integration Kit allows PingFederate to communicate with PingAM and provide a rich sign-on orchestration experience.

> **Note**
>
> The ForgeRock Intelligent Access Integration Kit and the PingAM Integration Kit use different adapters, so it isn't possible to upgrade from one to the other. Going forward, enhancements will only be added to the PingAM Integration Kit.

PingAM can deploy at scale and across any platform. It's a software as a service (SaaS) application with self-managed options for:

- Public cloud deployment
- Private cloud deployment
- On-premises virtual machine deployment
- Bare metal server deployment

PingFederate outsources user sign-on experience to PingAM.

Components

PingAM IdP Adapter

- When a user signs on through PingFederate, the adapter validates whether there's a valid session cookie and whether it needs to create a new session with PingAM.

Intended audience

This document is intended for PingFederate administrators.

If you need help during the setup process, learn more in the following resources:

- Step 2. Deploy AM

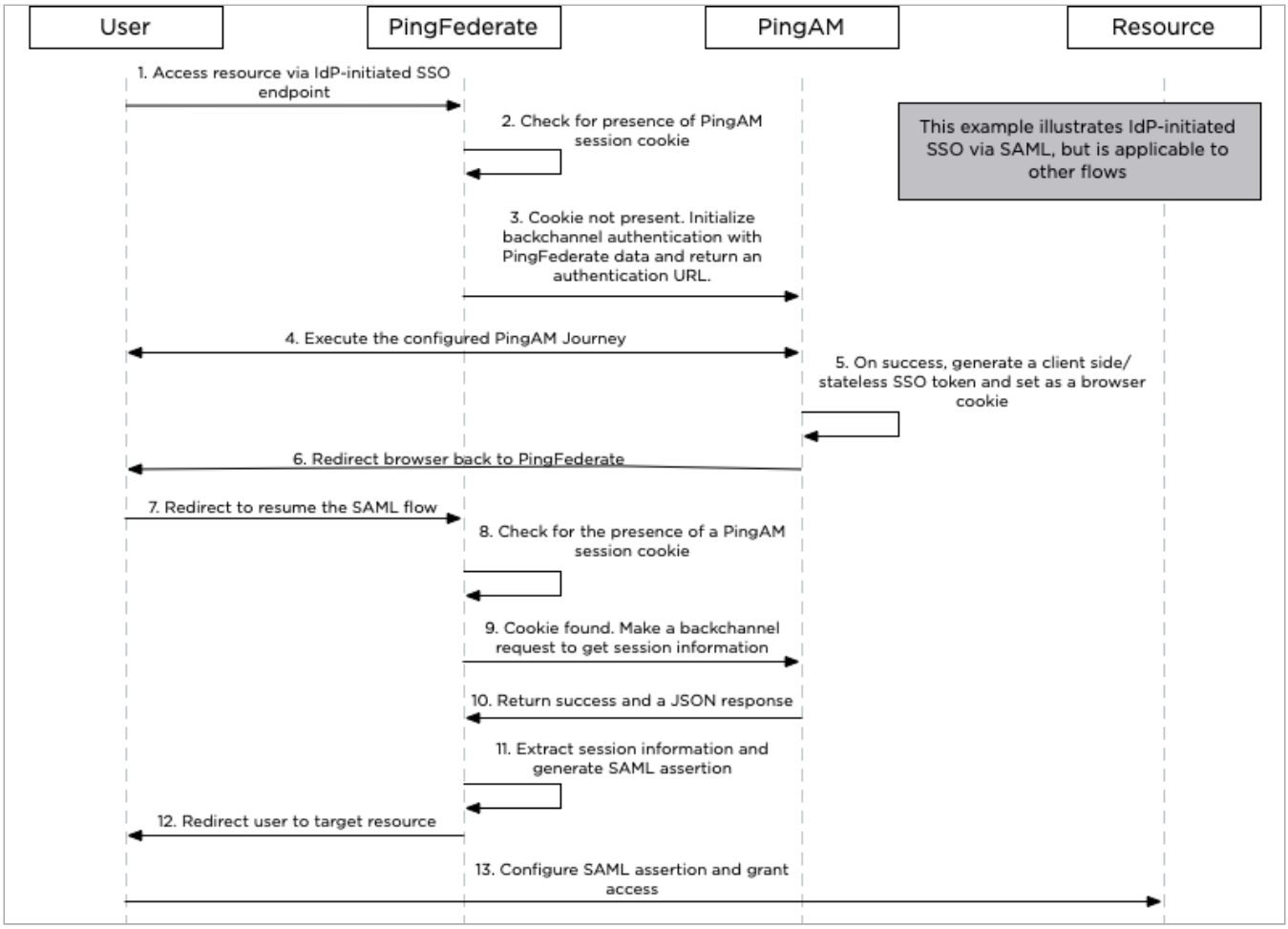
Copyright © 2025 Ping Identity Corporation

System requirements

- PingFederate 11.3 or later.
- PingAM Integration Kit 1.2 (requires PingAM 8.0 or later).
- You must deploy PingFederate and PingAM on the same cookie domain because the current implementation relies on the presence and validation of a PingAM single sign-on (SSO) cookie.

Overview of the SSO flow

The following figure illustrates an example single sign-on (SSO) process flow.



In summary:

1. A user initiates the sign-on process by requesting access to a protected resource.
2. If PingFederate detects that the PingAM cookie is not present, it gets an access token using the OAuth credentials, then initializes a backchannel authentication using the access token, and uses the redirect URI returned to send the user to orchestrate authentication. It also appends a PingFederate URL as a request parameter to resume the flow post-login.

   If a session cookie is present, PingFederate makes a backchannel request to get session information from PingAM.

Copyright © 2025 Ping Identity Corporation

3. On a success, PingFederate extracts session information from the JSON response provided by PingAM and generates a SAML assertion.
4. PingFederate redirects the user to the protected resource and configures the SAML assertion. The user is granted access.

Setup

Deploying the integration files

To get started with the integration, deploy the PingAM Integration Kit files to your PingFederate directory.

Before you begin

You must sign up for a PingAM account.

Steps

1. Download the PingAM Integration Kit `.zip` archive from the Add-ons tab of the PingFederate Downloads page.
2. Stop PingFederate.
3. Extract the `.zip` archive and merge the contents of the `dist` directory with your `<pf_install>/pingfederate/server/default/deploy` directory.
4. Start PingFederate.
5. If you operate PingFederate in a cluster, repeat steps 2-4 for each engine node.

Next steps

Configure an adapter instance.

Configuring an adapter instance

Configure the PingAM IdP Adapter to determine how PingFederate communicates with PingAM.

About this task

To begin the integration, deploy the PingAM Integration Kit files to your PingFederate directory.

Steps

1. In the PingFederate administrative console, go to Authentication > Integration > IdP Adapters and click Create New Instance.
2. On the Type tab, set the basic adapter instance attributes:
    1. In the Instance Name field, enter a name for the adapter instance.
    2. In the Instance ID field, enter a unique identifier for the adapter instance.
    3. In the Type list, select PingAM IdP Adapter. Click Next.

Copyright © 2025 Ping Identity Corporation

3. (Optional) On the IdP Adapter tab, in the Simple Parameter Mappings section, select pre-defined parameters to send from PingFederate to PingAM.

> **Note**
>
> - If you have configured an OAuth client in PingAM and provided the **Client ID** and **Client Secret**, the adapter sends your **Simple Parameter Mappings** and **Advanced Parameter Mappings** to PingAM. In PingAM, you can access these parameters by configuring your flow's input schema to expect the **Parameter Name** values you define in steps 3b and 4b.
> - You can find more information about other input parameter options in the Parameter reference in the PingFederate SDK documentation and Extended properties in the PingFederate documentation.

   1. Click Add a new row to 'Simple Parameter Mappings (optional)'.
   2. In the Parameter Name field, enter a key name to use when sending the parameter to PingAM.

      For example, `appName`.
   3. In the Source list, select the pre-defined parameter you want to send.

      For example, Application Name.
   4. In the Action column, click Update.
   5. To add more attributes, repeat steps a - d.

4. (Optional) On the IdP Adapter tab, in the Advanced Parameter Mappings section, define custom parameters to send from PingFederate to PingAM.

   1. Click Add a new row to 'Advanced Parameter Mappings (optional)'.
   2. In the Parameter Name field, enter a key name to use when sending the parameter to PingAM.

      For example, `trackedParameters`.
   3. In the Source Type list, select the type of parameter you want to send.

      For example, Tracked HTTP Request Parameters.

      The following options are available:

      Chained attributes

      The attributes that are made available by the other adapters and selectors in your PingFederate authentication policy.

      Extended properties

      These parameters store additional information about connections, OAuth clients, or both. You can find more detail in Extended properties in the PingFederate documentation.

      Request claims

      The claims PingFederate recieved within an OAuth/OpenID Connect Request Object or the parameters of a pushed authorization request.

Copyright © 2025 Ping Identity Corporation

Tracked HTTP request parameters

The tracked HTTP request parameters that were included in the initial HTTP request of the current transaction.

4. In the Source Parameter field, enter the exact name of the parameter that you want to send to PingAM. The parameter must be available to the adapter from the Source Type that you selected.

   Leave this field empty to send all parameters of the Source Type.

5. In the Action column, click Update.

6. To add more attributes, repeat steps a - e.

5. (Optional) On the IdP Adapter tab, in the Journey Response Mappings section, map attributes from the PingAM response to the attribute contract:

   1. Click Add a new row to 'Journey Response Mappings (optional)'.

   2. In the Local Attribute field, enter a name of your choosing for an attribute.

   3. In the Journey Attribute Mapping field, map the local attribute to a remote attribute using JSON pointer syntax.

      Learn more about JSON pointer syntax in RFC 6901.

   4. In the Action column, click Update.

   5. To add more attributes, repeat steps a - d.

      These attributes become available in your PingFederate authentication policy.

6. On the IdP Adapter tab, configure the adapter instance by referring to PingAM IdP Adapter settings reference. Click Next.

7. On the Extended Contract tab, add any attributes that you included in the PingAM Response Mappings section of the IdP Adapter tab. Click Next.

8. On the Adapter Attributes tab, set pseudonym and masking options as shown in Set pseudonym and masking options in the PingFederate documentation. Click Next.

9. On the Adapter Contract Mapping tab, configure the contract fulfillment details for the adapter as shown in Define the IdP adapter contract in the PingFederate documentation. Click Next.

10. On the Summary tab, review your configuration. Click Save.

Next steps

Review Preparing to use a PingAM journey.

PingAM IdP Adapter settings reference

Field descriptions for the PingAM IdP Adapter configuration screen.

Copyright © 2025 Ping Identity Corporation

Field	Description
Base URL	The PingAM base URL, including context for the sign-on redirect. For example: `https://am.example.com/am`
Realm	The PingAM Realm to call. For example: `alpha`
Journey	The PingAM Journey or Tree to call. For example: `Login`
Cookie Name	The PingAM session cookie name. For example: `iPlanetDirectoryPro`
Client ID	The `Client ID` defined in PingAM for backchannel authentication. If this field is left blank, the adapter uses the PingAM Integration Kit 1.1 authentication mechanism and logs a warning in the PingFederate server log. **Note** This field is required, if you want to use backchannel communication to send PingFederate data to PingAM.
Client Secret	The `Client Secret` defined in PingAM for backchannel authentication. If this field is left blank, the adapter uses the PingAM Integration Kit 1.1 authentication mechanism and logs a warning in the PingFederate server log. **Note** This field is required if you want to use backchannel communication to send PingFederate data to PingAM.
Logout Mode	Defines handling for application sign off. Select one of the following options: *PingAM* PingFederate sends a sign-off request to PingAM after a user signs off from the protected application. This deletes the PingAM session. *None* PingFederate doesn't send a sign-off request to PingAM after a user signs off from the protected application. The default value is `PingAM`.

Copyright © 2025 Ping Identity Corporation

Field	Description
Username Path	The JSON Pointer path that indicates the location of the username in the payload that PingAM returns. If you set a Username Path, the adapter validates this value against the incoming user ID that it receives from the authentication policy. The default value is `/username`.
API Request Timeout	The amount of time in milliseconds that PingFederate waits for PingAM to respond to requests and connection establishment. A value of `0` disables the timeout. The default value is `5000`.
Proxy Settings	Defines proxy settings for outbound HTTP requests. The default value is System Defaults.
Custom Proxy Host	The proxy server host name to use when Proxy Settings is set to Custom. This field is blank by default.
Custom Proxy Port	The proxy server port to use when Proxy Settings is set to Custom. This field is blank by default.

Preparing to use a PingAM journey

To use a PingAM journey, you must install PingAM and perform the additional configuration steps described in this document.

Before you begin

Complete the steps in the PingAM installation guide.

About this task

In the PingAM administrative console:

Steps

1. If you do not have an alpha realm in your environment yet:
    1. Go to Realms and click New Realm.
    2. In the Name field, enter `alpha`.
    3. Click Use Client-Side Sessions.
    4. Click Create to save your configuration.

Copyright © 2025 Ping Identity Corporation

> **Important**
>
> You should have an alpha realm because it's best practice to reserve the root realm for administrative operations. Learn more about configuring a realm in create a new realm.

2. Configure client-side sessions by completing the steps in the configure client-side sessions procedure.
3. Enable goto and redirects for the validation service.

   By default, PingAM denies all goto and redirects after the sign on flow is complete.

   1. In the alpha realm, go to Services.
   2. If the Validation Service is not in the list of services, click Add a Service and in the Choose a service type list, select Validation Service.
   3. In the Valid goto URL Resources field, enter one or more valid URL patterns to allow.

      Example:

      - `https://<my_ping_federate_url>:*/*`
      - `https://<my_ping_federate_url>:*/*?*`
4. Configure push authentication journeys by completing the steps in the push authentication journeys guide.
5. (Optional) To adjust authentication session lifetimes:
   1. In the alpha realm, go to Services, click Session, then click Create.
   2. On the Dynamic Attributes tab, enter the desired values in the Maximum Session Time and Maximum Idle Time fields.
   3. Click Save Changes.

      Learn more in the session service configuration reference.
6. If you're using PingAM Integration Kit 1.2, create an OAuth client as described in Create a client profile.
   1. In the Core section, make sure to set the `write` and `back_channel_authentication` scopes.
   2. In the Advanced section, make sure to configure the `Client Credentials` grant type, and that the Token Endpoint Authentication Method is using `client_secret_basic`.

Collecting session data from PingAM

Get session data from a PingAM journey (tree) to use in PingFederate.

Steps

1. Create a script to collect the desired session data from the PingAM journey that you specified in the adapter settings:
   1. In the PingAM administrative console, go to the realm that the journey is in, then go to Scripts and click + New Script.
   2. In the Name field, give the script a meaningful name.

Copyright © 2025 Ping Identity Corporation

3. In the Description field, give the script a meaningful description.

4. In the Script Type list, select Decision node script for authentication trees.

5. In the Language section, select JavaScript.

6. In the Evaluator Version section, confirm that the value is Legacy.

7. In the Script field, enter the following sample code.

> **Note**
>
> The following script sets the username, email and telephone number attributes that PingAM stores as part of authentication. Use this script as a starting point for the session data that you want to make available to PingFederate.

```
/*
- Data made available by nodes that have already executed are available in the sharedState variable.
- The script should set outcome to either "true" or "false".
- Note: This script is not fault-tolerant. It is simply meant to give an idea of how script nodes may
be used.
*/

var fr = JavaImporter(org.forgerock.openam.auth.node.api.Action);

var userId = nodeState.get("username").asString();

// lookup attributes by LDAP attribute name
var mail = idRepository.getAttribute(userId, "mail").iterator().next();
var telephoneNumber = idRepository.getAttribute(userId, "telephoneNumber").iterator().next();

// for each attribute, add the 'putSessionProperty' method
action = fr.Action.goTo("true").putSessionProperty("am.protected.sessionUsername", userId)
.putSessionProperty("am.protected.mail", mail)
.putSessionProperty("am.protected.telephoneNumber", telephoneNumber)
.build();

outcome = "true";
```

8. Click Validate.

2. Use the script in the authentication journey:

> **Note**
>
> This procedure assumes that the mail, username, and telephone number attributes are available from a previous node in the journey.

1. Go to Authentication > Trees and open the journey that's being used in the adapter.

2. Drag the Scripted Decision node onto the journey.

3. Select the Scripted Decision node and give the node a meaningful name.

4. In the Script list, select the script that you created in the previous step.

Copyright © 2025 Ping Identity Corporation

5. In the Outcomes field, enter `true`. Press Enter.
6. Connect the `True` outcome of the DataStore Decision node to the Scripted Decision node.
7. Connect the `True` outcome of the Scripted Decision node to the Success node.
8. Click Save.

3. Add the properties that the script sets in the session to the allow list:
    1. In the PingAM administrative console, go to Services.
    2. Add or edit the Session Property Whitelist Service. To add this service:
        1. Click + Add a Service.
        2. In the Choose a service type list, search for `Session Property Whitelist Service` and select it in the list.
    3. On the Session Property Whitelist Service page, in the Allowlisted Session Property Names field, add the following properties:
        - `am.protected.sessionUsername`
        - `am.protected.mail`
        - `am.protected.telephoneNumber`
    4. Click Save Changes.
4. Map the user attributes that you set in the script to their session attribute equivalents:
    1. Go to Authentication > Settings and click the Post Authentication Processing Settings tab.
    2. In the User Attribute Mapping to Session Attribute field, add the following attribute mappings:
        - `mail|mail`
        - `username|sessionUsername`
        - `telephoneNumber|telephoneNumber`
    3. Click Save Changes.
5. Test the user journey.

Extracting data from PingFederate

Extract data collected from PingFederate through the simple and advanced parameter mappings to use in a PingAM journey (tree).

Copyright © 2025 Ping Identity Corporation

Steps

1. Create a script to extract data coming from PingFederate into the PingAM journey that you specified in the adapter settings:
    1. In the PingAM administrative console, go to the realm that the journey is in, then go to Scripts and click + New Script.
    2. In the Name field, give the script a meaningful name.
    3. In the Description field, give the script a meaningful description.
    4. In the Script Type list, select Decision node script for authentication trees.
    5. In the Language section, select JavaScript.
    6. In the Evaluator Version section, confirm that the value is Legacy.
    7. In the Script field, enter the following sample code.

    > **Note**
    >
    > Use this script as a starting point. In this example, the data sent from PingFederate is configured with the **Parameter Name** `appName`.

    ```
    /*
    - Data made available by nodes that have already executed are available in the sharedState variable.
    - The script should set outcome to either "true" or "false".
    */

    var fr = JavaImporter(org.forgerock.openam.auth.node.api.Action);

    var pfApplicationName = nodeState.get("appName").asString();

    logger.error("The PingFederate application name passed in is {}", pfApplicationName);

    //for each attribute, add the 'putSessionProperty' method
    action = fr.Action.goTo("true").putSessionProperty("am.pf.appName", pfApplicationName)
    .build();

    outcome = "true";
    ```

    8. Click Validate.
2. Use the script in the authentication journey:
    1. Go to Authentication > Trees and open the journey that's being used in the adapter.
    2. Drag the Scripted Decision node onto the journey.
    3. Select the Scripted Decision node and give the node a meaningful name.
    4. In the Script list, select the script that you created in the previous step.
    5. In the Outcomes field, enter `true`. Press Enter.
    6. Connect the `True` outcome of the DataStore Decision node to the Scripted Decision node.

Copyright © 2025 Ping Identity Corporation

7. Connect the `True` outcome of the Scripted Decision node to the Success node.
8. Click Save.

3. Add the properties that the script sets in the session to the allow list:
    1. In the PingAM administrative console, go to Services.
    2. Add or edit the Session Property Whitelist Service. To add this service:
        1. Click + Add a Service.
        2. In the Choose a service type list, search for `Session Property Whitelist Service` and select it in the list.
    3. On the Session Property Whitelist Service page, in the Allowlisted Session Property Names field, add the properties that you set in the script:

        For example:

        - `am.pf.appName`
    4. Click Save Changes.
4. Test the user journey.

Troubleshooting

Enabling debug logging

To help with troubleshooting or monitoring, you can turn on activity logging for PingFederate, the PingAM IdP Adapter, or both.

About this task

You can use logging for troubleshooting and analytics.

Learn more about logging in Enabling debug messages and console logging in the PingFederate documentation.

Steps

1. Open the `<pf_install>/pingfederate/server/default/conf/log4j2.xml` file for editing.
2. To log activity for PingFederate and all adapters:
    1. Find the following section:

```
<AsyncRoot level="INFO" includeLocation="false">
        <!-- <AppenderRef ref="CONSOLE" /> -->
        <AppenderRef ref="FILE" />
</AsyncRoot>
```

    2. Change `INFO` to `DEBUG`:

Copyright © 2025 Ping Identity Corporation

```
<AsyncRoot level="DEBUG" includeLocation="false">
        <!-- <AppenderRef ref="CONSOLE" /> -->
        <AppenderRef ref="FILE" />
</AsyncRoot>
```

3. Optional: To see the adapter activity in the console, remove the comment tags that surround the `CONSOLE` line:

```
<AsyncRoot level="INFO" includeLocation="false">
        <AppenderRef ref="CONSOLE" />
        <AppenderRef ref="FILE" />
</AsyncRoot>
```

3. To log activity relating to the PingAM IdP Adapter:

Choose from:

- To log activity for the PingAM IdP Adapter as well as its HTTPS and component activity, add the following line:

```
<Logger name="com.pingidentity.adapters.pingam" level="DEBUG"/>
```

- To log activity for the adapter's HTTPS activity and other components but not for the adapter itself, add the following line:

```
<Logger name="com.pingidentity.adapters.pingam.shade" level="DEBUG"/>
```

- To log activity for the PingAM IdP Adapter but not its HTTPS or component activity, add the following lines:

```
<Logger name="com.pingidentity.adapters.pingam" level="DEBUG"/>
<Logger name="com.pingidentity.adapters.pingam.shade" level="INFO"/>
```

4. Save the file.

Tracking transactions between PingFederate and PingAM

To help with application log debugging, you can track PingFederate and PingAM transactions using a unique request ID that's set by the adapter.

About this task

PingFederate and PingAM both have methods to keep track of transactions across multiple applications:

- In PingFederate, the `httprequestid` can be used to track transactions that come from outside PingFederate.
- In PingAM, the `X-ForgeRock-TransactionId` header tracks related requests throughout the ForgeRock platform.

Copyright © 2025 Ping Identity Corporation

To track transactions between PingFederate and PingAM, the PingAM IdP Adapter injects the `X-ForgeRock-TransactionId` header with the `httprequestid` value in all API calls to PingAM.

Complete the following procedure to finish the setup in PingAM and see the unique request ID in the audit logs:

Steps

1. To see the value set for `httprequestid` in the `X-ForgeRock-TransactionId` header, configure PingFederate to log the `httprequestid`:
    1. In the `<pf_install>/pingfederate/server/default/conf/log4j2.xml` file, go to the `SecurityAudit2File` RollingFile and add the httprequestid field in the Pattern section.

        Learn more in PingFederate security audit logging and the PingFederate log files section of the PingFederate documentation.

    2. Save the file.
2. Configure PingAM to accept the `X-ForgeRock-TransactionId` header:
    1. In the PingAM admin UI, go to Configure > Server Defaults > Advanced and go to the bottom of the list.
    2. In the Property Name column, enter `org.forgerock.http.TrustTransactionHeader`.
    3. In the corresponding Property Value column, enter `true`.
    4. To add the property and save your work, click the + icon.

Result

You can now track the unique request ID between PingFederate and PingAM in the PingFederate security audit log and the PingAM access audit log.

Release notes

Changelog

The following is the change history for the PingAM Integration Kit.

PingAM Integration Kit 1.2 - February 2025

- Added the ability to use backchannel communication with PingAM to pass data from PingFederate to PingAM securely.

  To send PingFederate parameters to PingAM, you must create an OAuth client in PingAM and update the PingAM IdP Adapter configuration with the associated Client ID and Client Secret.

  Learn more in step 6 of Preparing to use a PingAM journey and refer to the Client ID and Client Secret table entries in PingAM IdP Adapter settings reference.

- Added the ability to configure whether the PingAM session should be deleted after a user signs off from the protected application.

Copyright © 2025 Ping Identity Corporation

Learn more in the Logout Mode table entry in the PingAM IdP Adapter settings reference.

PingAM Integration Kit 1.1 – November 2024

- Added the ability to track PingFederate and PingAM transactions using a unique request ID that's set by the adapter. This enhances the application log debugging experience. You can find configuration instructions in Tracking transactions between PingFederate and PingAM.
- Added the ability to perform username validation. To do so, configure the Username Path between the username that the adapter receives from the authentication policy and a value in the PingAM session info payload.
- Fixed an issue that caused an infinite loop between PingFederate and PingAM if the PingAM and adapter cookie name values didn't match. The adapter now fails if it doesn't receive the expected cookie name from PingAM.

PingAM Integration Kit 1.0 – August 2024

- Initial release

Known issues and limitations

The following are known issues or limitations for the PingAM Integration Kit.

Known limitations

- The adapter doesn't support the PingFederate authentication API because it isn't feasible to generate verified access challenge response values externally.
- PingAM Integration Kit 1.2 requires PingAM 8.0.
- As of PingAM Integration Kit 1.2, the adapter uses an OAuth client to do backchannel authentication. Learn more in Changelog.

Download manifest

The following files are included in the PingAM Integration Kit `.zip` archive.

- `Legal.pdf` – Copyright and license information
- `dist/pingfederate/server/default` – Contains the integration files
  - `deploy` – Contains the Java libraries
    - `pf-pingam-integration-kit-<version>.jar` – The PingAM IdP Adapter

Magic Link Integration Kit

The Magic Link Integration Kit enables PingFederate to send users a one-time link through their verified email address to sign on without a password.

Copyright © 2025 Ping Identity Corporation

Features

- Supports the PingFederate Authentication application programming interface (API), allowing you to integrate it into your existing web-based sign-on application.
- Supports existing PingFederate notification publishers like SMTP and Amazon SNS.

Components

Magic Link IdP Adapter

- Allows PingFederate to trigger a notification publisher to send a one-time link to a user during sign on.
- Accesses user contact information from data store attributes or from earlier in the authentication flow.
- Validates that the user has clicked the one-time link.

Language packs

- Allow you to customize the messages shown to the user during authentication.

Templates

- Allow you to modify the appearance of pages shown to the user during authentication.

Intended audience

This document is intended for PingFederate administrators.

Before starting, we recommend that you familiarize yourself with the following sections of the PingFederate documentation:

- Authentication API
- Managing notification publisher instances
- Managing IdP adapters
- Authentication policies

System requirements

- PingFederate 11.2 or later
- To improve security, it's best to use the Magic Link IdP Adapter as part of two-factor authentication (2FA) instead of single-factor authentication.
- You should only use the Magic Link IdP Adapter with verified email addresses.

> **Note**
>
> Create user accounts and verify email addresses through a different means before using the Magic Link IdP Adapter.

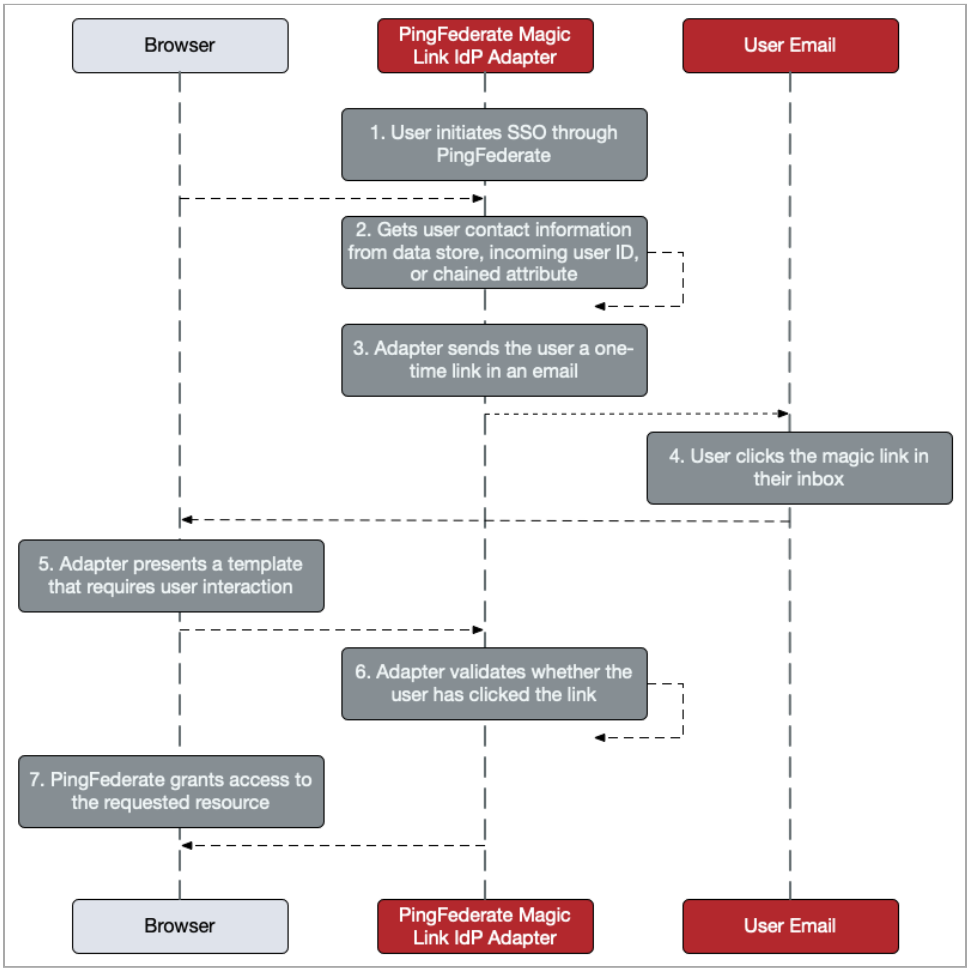
Copyright © 2025 Ping Identity Corporation

Overview of the SSO flow

With the Magic Link Integration Kit, PingFederate sends a one-time link to the user as part of the sign-on flow.

Same device SSO flow description

The following figure shows how the Magic Link IdP Adapter is integrated into the sign-on process when the user authenticates on the same device:



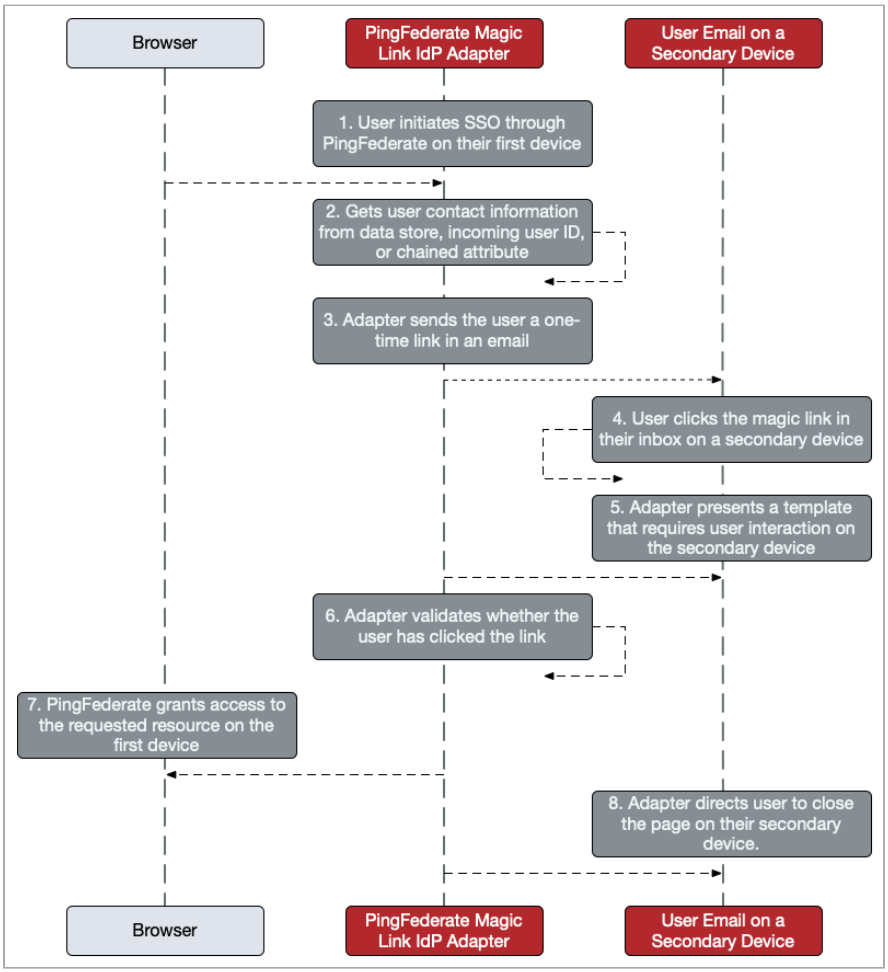
1. The user initiates single sign-on with PingFederate.
2. The adapter gets the user's contact information from a data store, an incoming user ID, or an attribute from earlier in the authentication flow.
3. The adapter sends the user a one-time link in an email and presents the user with a template that instructs them to check their email.
4. The user opens their email clicks the magic link in their inbox on the same device that they used to initiate the sign-on process.
5. If Show Confirmation is enabled, the adapter presents the user with a template that requires user interaction to confirm authentication.
6. The adapter validates whether the user has clicked the magic link.

Copyright © 2025 Ping Identity Corporation

7. PingFederate grants access to the requested resource in the window where the user initiated the sign-on process.

Different device SSO flow description

The following figure shows how the Magic Link IdP Adapter is integrated into the sign-on process when the user authenticates on a different device:



1. The user initiates single sign-on with PingFederate on one device, such as their laptop.
2. The adapter gets the user's contact information from a data store, an incoming user ID, or an attribute from earlier in the authentication flow.
3. The adapter sends the user a one-time link in an email and presents the user with a template that instructs them to check their email.
4. The user clicks the magic link in their inbox on a different device, such as their phone.
5. If Show Confirmation is enabled, the adapter presents the user with a template that requires user interaction to confirm authentication on their secondary device.
6. The adapter validates whether the user has clicked the magic link.
7. PingFederate grants access to the requested resource on the user's original device, in the window where the user initiated the sign-on process.

Copyright © 2025 Ping Identity Corporation

8. The adapter directs the user to close the page on their secondary device and continue on their original device.

Authentication API Support

You can use the PingFederate authentication application programming interface (API) to integrate the Magic Link IdP Adapter into your application.

The PingFederate Authentication API provides access to the current state of the authentication flow as a user steps through the PingFederate authentication policy. Learn more in PingFederate authentication API in the PingFederate documentation.

To integrate the Magic Link IdP Adapter into your authentication flow, configure your application based on the information in this section.

> **Note**
>
> For this adapter, only the different device SSO flow supports the PingFederate Authentication API. For more information on the different device SSO flow, see Overview of the SSO flow.
> The application using the authentication API polls for status and continues with the same flow ID and cookie once the link is approved. This is because the link verification endpoint requires a redirect back to the resume URL, and the application should control this behavior.

> **Tip**
>
> You can also explore the process using the PingFederate Authentication API Explorer. Learn more in Exploring the Authentication API in the PingFederate documentation.

Models, attributes, and error codes

When using the Magic Link IdP Adapter through the PingFederate authentication API, the adapter uses the following state models, action models, objects, and error codes.

Copyright © 2025 Ping Identity Corporation

State models

Status	Response Model	Action	Description
`LINK_INITIATED`	• `transactionID (string)` : The transaction ID that can be used to check the status of the magic link request. • `email (string)` : The masked email address that the magic link was sent to. This value can be `null` if no email was found, or if the Show Email advanced setting isn't configured in the adapter instance.	• `poll`	The link has been generated and sent to the user. PingFederate is awaiting action from the user.
`LINK_COMPLETED`	No model	• `continue`	The user has clicked the link and completed the process. This allows the application to present a screen to the user if desired.
`LINK_EXPIRED`	No model	• `continue`	The transaction ID that's associated with the link has expired. A new one must be generated.

Action Models

Action	Request Model	Errors	Description
`poll`	No model	None	Checks the status of a magic link transaction to see if the user has clicked the link yet.
`continue`	No model	None	Continues the flow.

Setup

Copyright © 2025 Ping Identity Corporation

Deploying the integration files

To get started with the integration, deploy the Magic Link Integration Kit files to your PingFederate directory.

Steps

1. Download the Magic Link Integration Kit `.zip` archive from the Ping Identity Integration Directory.
2. Stop PingFederate.
3. If you're upgrading from a previous deployment, do the following:
    1. Back up any Magic Link Integration Kit files that you've customized in the `<pf_install>/pingfederate/server/default/conf/` directory.
    2. Delete the `pf-magic-link-adapter-<version>.jar` file from the `<pf_install>/pingfederate/server/default/deploy` directory.
    3. Delete the `pf-authn-api-sdk-<version>.jar` file from the `<pf_install>/pingfederate/server/default/lib` directory if it's older than the version in the integration files.
4. Extract the `.zip` archive and merge the contents of the `dist` directory with your `<pf_install>/pingfederate/server/default/deploy` directory.
5. To use the SMTP Notification Publisher, merge the contents of the `pingfederate-smtp-magiclink-messages.properties` file into the `<pf_install>/pingfederate/server/default/conf/language-packs/pingfederate-email-messages.properties` file.
6. If you backed up any templates or messages from a previous deployment, modify the new files with your customizations.
7. Start PingFederate.
8. If you operate PingFederate in a cluster, repeat steps 2-7 for each engine node.

Configuring an adapter instance

Configure the Magic Link IdP Adapter to enable PingFederate to use notification publishers to send one-time links to users during sign on.

Before you begin

Deploy the Magic Link Integration Kit files to your PingFederate directory.

Steps

1. In the PingFederate administrative console, go to Authentication > Integration > IdP Adapters and click Create New Instance.
2. On the Type tab, set the basic adapter instance attributes:
    1. In the Instance Name field, enter a name for the adapter instance.
    2. In the Instance ID field, enter a unique identifier for the adapter instance.

Copyright © 2025 Ping Identity Corporation

3. In the Type list, select Magic Link IdP Adapter. Click Next.

3. On the IdP Adapter tab, configure the adapter instance by referring to Magic Link IdP Adapter settings reference. Click Next.
4. On the Actions tab, test your connection to the database. Resolve any issues that are reported, and then click Next.
5. On the Extended Contract tab, add any attributes that you expect to retrieve in addition to the core contract attributes. Click Next.
6. Complete the adapter configuration.
7. On the Summary tab, review your configuration. Click Save.

Next steps

Modify your PingFederate authentication policy to include your Magic Link IdP Adapter instance. Configure the Magic Link IdP Adapter as one of several multi-factor authentication (MFA) steps.

Learn more about configuring authentication policies in Authentication Policies.

Magic Link IdP Adapter settings reference

Field descriptions for the Magic Link IdP Adapter configuration screen.

Field	Description
Notification Publisher	Select the notification publisher that sends the one-time link to the user.
Link Expiration	Define the duration, in minutes, that the magic link remains valid.
Attribute Source	The source of the preferred delivery method and contact attributes. Select a data store, or: • Select Chained Attributes if the adapter receives the attributes from earlier in the authentication flow. • Select Incoming User ID if the adapter receives the username from the user ID mapping.
Email Attribute Name	The attribute that the notification publisher uses to email the user. For example, `email` or `mail`. This attribute can be from a data store or from earlier in the authentication flow. **Note** If the **Attribute Source** field is set to **Incoming User ID**, the adapter ignores the **Email Attribute Name** field.

Copyright © 2025 Ping Identity Corporation

Field	Description
Search String	The string that the adapter uses to search the data store to find the user. • For JDBC, enter a `select` statement. For example, `select email, phone from <db.table> where username=${userid}`. • For LDAP, enter an LDAP filter. For example, `sAMAccountName=${userid}`. • For a PingOne data store, enter the attribute. For example, `username=${userid}` or `id=${userid}`. • For REST API data stores, enter the resource path that's appended to the base URL of the REST API data store. For example, `/users?uid=${userid}`. The `${userid}` variable contains the user ID. Your adapter instance receives this from earlier in your PingFederate authentication flow. **Note** If the **Attribute Source** field is set to **Chained Attributes** or **Incoming User ID**, the adapter ignores the **Search String** field.
Base DN	The base DN that the adapter uses when connecting to an LDAP data store. **Note** If the **Attribute Source** field is set to **Chained Attributes** or **Incoming User ID**, the adapter ignores the **Base DN** field.
Notification Publisher Failure Mode	Determines whether the user authentication flow continues or fails when the notification publisher fails. **Note** If you select **Continue**, the **Magiclink.status** contract attribute records either `BYPASS` or `SUCCESS`. This can indicate when a notification publisher failure has occurred. Configure a rule using this attribute to create a backup authentication method.
Show Confirmation	Determines whether PingFederate shows a template or state that requires user interaction after clicking a magic link. Select this checkbox if email scanners are in use, or if links are pre-clicked in the user's inbox through some other means.

Copyright © 2025 Ping Identity Corporation

Field	Description
Magic Link Endpoint	**The endpoint that the magic link goes to.** **The default value is `/onetime-link`.**
Magic Link API Endpoint	**The API endpoint that the front end can use to poll magic link status.** **The default value is `/onetime-link-api`.**
Test User ID	**The user ID that's used to test the configuration on the Actions tab.** **This field is blank by default.**
Show Email	**Determines whether the adapter shows the masked email that magic links are sent to in templates and API responses.** **This checkbox is selected by default.**
HTML Template Prefix	**Identifies the HTML templates that the adapter uses.** **Note** If you added a template file name in the `/server/default/conf/template` directory, enter the new prefix here. **The default value is `magic-link`.**
Messages File	**Identifies the customizable language-pack file that the adapter uses for HTML templates.** **Note** If you added a language-pack file name in the `/server/default/conf/language-packs` directory, enter the new name here. **The default value is `magic-link-messages`.**
Email Template	**Identifies the email template that the adapter uses to send the magic link.** **Note** If you added a language-pack file name in the `/server/default/conf/template/mail-notifications` directory, enter the new name here. **The default value is `pingfederate.magiclink.adapter.template.html`.**

Copyright © 2025 Ping Identity Corporation

Field	Description
LDAP Search Scope	When the Attribute Source is an LDAP data store, this setting determines the scope of the user search. *Single Level* Searches the immediate children of the base object, but excludes the base object itself. *Include Subtree (default)* Searches all child objects as well as the base object.

Release Notes

Changelog

The following is the change history for the Magic Link Integration Kit. Updated April 26, 2024.

Magic Link Integration Kit 1.0 — April 2024

- Initial release.

Known Issues and Limitations

The following are known issues or limitations for the Magic Link Integration Kit.

Known issues

There are no known issues.

Known limitations

- When PingFederate is operating in standalone mode, changes to an existing notification publisher configuration do not take effect until the corresponding adapter instance is saved again.

Download Manifest

The following files are included in the Magic Link Integration Kit:

- `Legal.pdf` – copyright and license information.

Copyright © 2025 Ping Identity Corporation

- `dist/pingfederate/server/default` – a directory that contains integration files.
  - `conf` – a directory that contains configuration files.
    - `language-packs` – a directory that contains language template files.
      - `magic-link-messages.properties` – a variable file that customizes the messages that appear in template files.
      - `pingfederate-smtp-magiclink-messages.properties` – a variable file that customizes the messages that appear on the SMTP template file.
    - `template` – a directory that contains user-facing HTML template files.
      - `assets` – a directory that contains functional scripts and files used by the template.
        - `css` – a directory that contains CSS files for the templates.
          - `magiclink.css` – a CSS file that customizes the appearance of the template file.
          - `end-user/<version>/end-user.css` – a CSS file that customizes the appearance of the template files.
        - `scripts/jquery-<version>.min.js` – a JavaScript file that contains the jQuery library.
      - `magic-link-confirmation.html` – presents the user with a form when the adapter requires user interaction after a magic link has been clicked.
      - `magic-link-invalid.html` – presents the user with a message when the one-time link expires.
      - `magic-link-sent.html` – presents the user with a message when the one-time link is sent.
      - `magic-link-success.html` – presents the user with a message when authentication through the one-time link is completed successfully.
      - `mail-notifications` – a directory that contains email template files.
        - `pingfederate.magiclink.adapter.template.html` – a sample template file that the SMTP Notification Publisher can use to send a one-time link in an email.
  - `deploy` – a directory that contains the Java libraries necessary to run the adapter.
    - `pf-magic-link-adapter-<version>.jar` – JAR file that contains the Magic Link IdP Adapter.
  - `lib` – a directory that contains PingFederate core library files.
    - `pf-authn-api-sdk-<version>.jar` – JAR file that contains the PingFederate Authentication API SDK.

PingOne Protect Integration Kit

The PingOne Protect Integration Kit allows PingFederate to communicate with PingOne Protect for risk-based authentication.

By sending transaction information and an optional device profile to PingOne when a user signs on, PingFederate can get a security risk evaluation for the sign-on event. Including the risk evaluation in your PingFederate authentication policy allows you to dynamically adjust the user's authentication requirements each time they sign on.

Copyright © 2025 Ping Identity Corporation

Download

To download the PingOne Protect Integration Kit, see PingFederate Downloads on the Ping Identity site.

Components

PingOne Protect IdP Adapter

- When a user signs on through PingFederate, the adapter sends the transaction information to PingOne Protect, and retrieves a Result and other information about the user's current and previous transactions.

PingOne Protect Provider and SDK

- The provider works with the HTML Form adapter in your policy. The provider includes the SDK and can evaluate risk and detect bots before the password credential validator (PCV) is triggered. It can work together or separately with the PingOne Protect IdP Adapter. Like the adapter, the provider supports the authentication API and widget.

  The provider can be configured in the following ways:

  - The provider uses the SDK and performs risk evaluation without using the PingOne Protect IdP Adapter.
  - The provider uses the SDK and sends the payload to the PingOne Protect IdP Adapter, which checks the risk.
  - The provider is not used, such as in the case of an active session. The PingOne Protect IdP Adapter loads the device profiling page and checks the risk.

Template and script files

- When a user signs on through PingFederate and device profiling is enabled, these files create a device profile for the adapter to send to PingOne Protect.

Intended audience

This document is intended for PingFederate administrators.

If you need help during the setup process, learn more in the following resources:

- PingOne Protect on the Ping Identity site
- PingOne Protect in the PingOne documentation
- The following sections of the PingFederate documentation:
  - Managing IdP adapters
  - Authentication policies

System requirements

- PingFederate 11.3 or later

Copyright © 2025 Ping Identity Corporation

> **Note**
>
> If you have an older version of PingFederate, use the PingOne Risk Integration Kit.

- To allow PingFederate to make outbound HTTPS connections, you might need to allow the following host names in your firewall:
  - https://api.pingone.com, https://api.pingone.asia, or https://api.pingone.eu
  - https://auth.pingone.com, https://auth.pingone.asia, or https://auth.pingone.eu
- A PingOne Protect or PingOne Risk license

  You can find information on creating a trial account in Creating an organization and environment in PingOne.

Overview of PingOne Protect

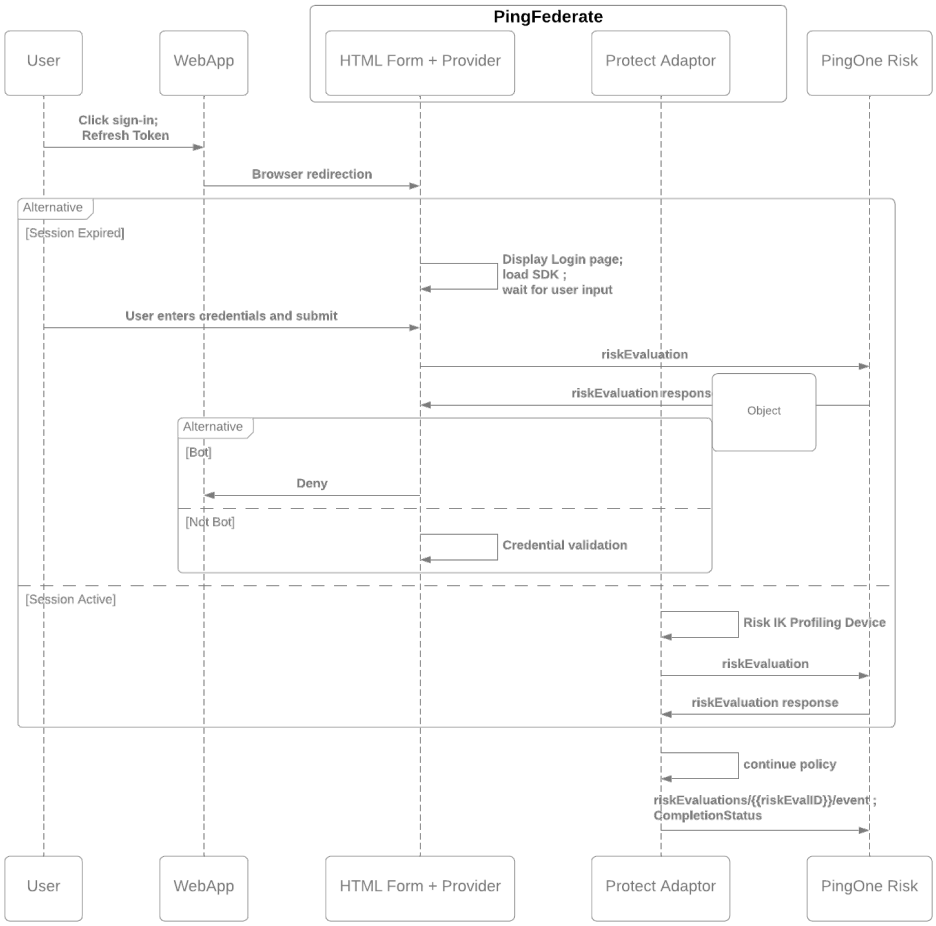
PingOne Protect evaluates the level of security risk for a user sign-on event based on a device profile and transaction information. Learn more in PingOne Protect in the PingOne documentation.

Overview of the SSO Flow

With the PingOne Protect Integration Kit, PingFederate includes PingOne Protect in the sign-on flow.

The following figure shows how PingOne Protect is integrated into the sign-on process

Copyright © 2025 Ping Identity Corporation



Description

1. A user initiates the sign-on process by requesting access to a protected resource.
2. When device profiling is enabled, one of the following occurs, depending on the device profiling method:
    - An adapter that is earlier in the authentication flow runs a script that creates a device profile. The script passes the device profile to the PingOne Protect IdP Adapter in a series of HTTP cookies.
    - The PingOne Protect IdP Adapter creates a device profile.
3. The PingOne Protect IdP Adapter collects transaction information, such as the user's IP address.
4. The adapter sends the transaction information and optional device profile to PingOne Protect.
5. PingOne Protect returns a JSON payload with the risk result and other information, such as the IP reputation, to the adapter.
6. The PingOne Protect IdP Adapter makes the risk result and other information available in the PingFederate authentication policy.
7. PingFederate executes the authentication policy, which branches based on the risk result provided by the adapter.
8. PingFederate returns the resource that the user requested.

Copyright © 2025 Ping Identity Corporation

9. The adapter notifies PingOne Protect whether authentication ultimately succeeded. This helps PingOne Protect evaluate subsequent sign-on attempts.

Device profiling methods

In addition to the basic transaction information, PingOne Protect can include an optional device profile in its risk evaluation.

There are different methods for collecting the device profile depending on your preferences and whether users are authenticating directly through PingFederate or through the PingFederate authentication API.

Complete steps for each device profiling method are available when you reach the Integrating device profiling part of the setup process.

Captured by the provider and HTML Form adapter

The PingOne Protect provider is the recommended device profiling method and is critical for the best risk detection. It is required for the newest risk predictors such as New Device. The PingOne Protect provider uses the built-in SDKs with the HTML Form adapter to evaluate the risk.

Captured by the PingOne Protect IdP Adapter

The PingOne Protect IdP Adapter inserts a page into the sign-on flow that shows a spinner animation while it collects the device profile. The page appears in the sign-on flow in the order set by your PingFederate authentication policy, typically after the first-factor sign-on page.

If Dynamic Device Profiling is enabled, the PingOne Protect IdP Adapter will only load the device profiling page if it hasn't already received the payload from the provider.

This method does not require modifications to any other pages, but it adds a wait time to the sign-on process while the device profile is collected. The length of the wait depends on your environment.

Captured by a previous adapter

You add a device profiling script, such as Identifier First, to a page that appears earlier in the sign-on flow than the PingOne Protect IdP Adapter. The script creates the device profile and stores it in a series of HTTP cookies. When the PingOne Protect IdP Adapter is triggered by the PingFederate authentication policy, it picks up the device profile from the cookies. The adapter sends the device profile to PingOne Protect along with the transaction information.

By adding the device profiling script to your first-factor authentication page, for example, the device profile is created while the user types in their credentials. This can reduce the perceived wait time during sign-on.

You can integrate the device profiling script into any page that meets the following criteria:

- The page appears before the PingOne Protect IdP Adapter in the sign-on flow.
- The page is hosted in the same domain as your PingFederate server. This allows HTTP cookies to pass the transaction information to the PingOne Protect IdP Adapter. You might be able to work around this requirement by consolidating your domains with a reverse proxy server.
- You have access to the page to add the script.

Copyright © 2025 Ping Identity Corporation

Captured by your web application first-factor sign-on page

You add a device profiling script to your web application sign-on page. The script creates the device profile and stores it in a series of HTTP cookies. When the PingOne Protect IdP Adapter is triggered by the PingFederate authentication policy, it picks up the device profile from the cookies. The adapter sends the device profile to PingOne Protect along with the transaction information.

PingFederate setup

To configure the PingOne Protect Integration Kit with PingFederate, follow these steps:

1. Deploying the integration files
2. Integrating device profiling
3. Configuring an adapter instance
4. Configuring a provider instance
5. Using custom risk predictors
6. Adding risk level results to your authentication policy
7. Configuring PingFederate to forward IP addresses

Deploying the integration files

To get started with the integration, deploy the PingOne Protect Integration Kit files to your PingFederate directory.

Steps

1. Download the PingOne Protect Integration Kit `.zip` archive from the Ping Identity Integration Directory.
2. Stop PingFederate.
3. Go to your `pingfederate/server/default/deploy` directory and add the `pf-pingone-protect-adapter-<version>.jar` file.
4. Go to your `pingfederate/server/default/conf` directory and add the following files:
    1. Add the `pingone-protect-messages.properties` file to the `language-packs` directory.
    2. Add the `pingone-protect-template.html` file to the `template` directory.
    3. Add the `pingone-protect.css` file to the `template/assets/css` directory.
    4. Add the `pingone-protect-device-profiling.js` and `signals-sdk-<version>.js` files to the `template/assets/scripts` directory.
    5. Add the `signals.js` file to the `template/assets/scripts/captcha` directory.

> **Note**
>
> When using the default templates with PingFederate 11.3, see Troubleshooting information.

Copyright © 2025 Ping Identity Corporation

Creating an organization and environment in PingOne

The PingOne platform uses an organization-based model to define tenant accounts and their related entities. Create an organization to contain your PingOne Protect environments, application connections, user populations, and settings.

Steps

1. If you don't have a PingOne Protect license, sign up for a free trial.
    1. Go to Try Ping for Free on the Ping Identity site.
    2. Complete the sign up form. Click Submit.
    3. Verify your email using the link emailed to you.
    4. Go to the unique login link provided when you verified your email address.
    5. Enter your email address and password. Click Sign On.
2. On the PingOne console, add an environment to contain your PingOne Protect connections and settings.
    1. Click Add Environment.
    2. On the Create Environment page, in the What kind of environment do you want to create? section, select Build your own solution.
    3. Select the PingOne Protect and PingFederate tiles. Click Next.
    4. Select your deployment preferences. Click Next.
    5. In the Environment Name field, enter a name of your choosing.
    6. Optional: In the Description field, enter a description.
    7. From the Region list, select the region closest to you for the best performance.
    8. From the License list, select your PingOne Protect trial license.
    9. Click Finish.
3. On the Your Environments page, open the PingOne Protect console by clicking the PingOne Protect icon for your new environment.

Integrating device profiling

You can optionally send a device profile to PingOne Protect.

About this task

You can find detailed descriptions of each method in Device profiling methods.

Steps

- To send a device profile to PingOne Protect, do one of the following:

Copyright © 2025 Ping Identity Corporation

Method	Steps
Direct authentication mode - Captured by the PingOne Protect provider/HTML Form adapter	No extra steps are necessary. Continue to Configuring a provider instance.
Direct authentication mode - Captured by the PingOne Protect IdP Adapter	No extra steps are necessary. Continue to Configuring an adapter instance.
Direct authentication mode - Captured by a previous adapter, such as the Identifier First adapter	Complete the steps in Adding device profiling to an authentication API-based application.
Authentication API mode - Captured by your web application first-factor sign-on page	Complete the steps in Adding device profiling to an authentication API-based application.

Tip

You can find help with the authentication API in PingFederate authentication API and Exploring the Authentication API in the PingFederate documentation.

Adding device profiling to a browser-based authentication page

About this task

You can adapt these instructions to add device profiling to any web page included in an adapter, such as the Identifier First Adapter or a custom adapter. The page must meet the criteria listed in Device profiling methods.

Note

To use the HTML Form Adapter, follow the instructions in Configuring an adapter instance.

Steps

1. If you are modifying an external web application, copy the following files from the integration `.zip` archive to a location that your page can access.
    - `signals-sdk-<version>.js`
    - `signals.js` in the `captcha` directory
    - `pingone-protect-device-profiling-implementation.js`. (If you are using version 1.0.1 or earlier of the integration kit, the file is called `pingone-protect-device-profiling.js`)
2. Add the following external script references to the sign-on page:

Important

The scripts must be added in the following order.

Copyright © 2025 Ping Identity Corporation

```
<script type="text/javascript" src="signals-sdk-<version>.js"></script>
<script type="text/javascript" src="pingone-protect-device-profiling-implementation.js"></script>
<script type="text/javascript" src="signals.js"></script>
```

> **Note**
>
> If you are using version 1.0.1 or earlier of the integration kit, replace the reference to `pingone-protect-device-profiling-implementation.js` with a reference to `pingone-protect-device-profiling.js`

3. When you complete the steps in Configuring an adapter instance, you can enable risk evaluation for active sessions. Follow the steps in Configuring authentication sessions to set up sessions for the PingOne Protect Adapter, and complete steps 2a, 2b, and 3.

Adding device profiling to an authentication API-based application

About this task

Use the following instructions to add device profiling to a web application using the PingOne Protect SDK. For more information about the authentication API, see Authentication API in the PingFederate documentation or PingOne Protect Native SDKs in the PingOne Native SDK documentation.

Steps

1. Implement the Signals SDK by following the steps in PingOne Protect Native SDKs.
2. Configure functions to format the device profile and submit it to the authentication API. Use the following code sample as a guide.

> **Note**
>
> The `pingone-protect-device-profiling.js` assumes you are using the web SDK. If using a mobile device and you want to send the `deviceProfile` check the SDK documentation and use accordingly.

Copyright © 2025 Ping Identity Corporation

```
function onCompletion(flowId, deviceProfile) {
        submitDeviceProfile(flowId, deviceProfile);
}

function submitDeviceProfile(flowId, deviceProfile) {
        var myHeaders = new Headers();
        myHeaders.append("X-XSRF-Header", "test");
        myHeaders.append("Content-Type", "application/vnd.pingidentity.submitDeviceProfile+json");

        var raw = JSON.stringify({
        "signalsSdkDeviceProfile": deviceProfile
        });

        var requestOptions = {
        method: 'POST',
        headers: myHeaders,
        body: raw,
        redirect: 'follow'
        };

        fetch("https://pf_host:pf_port/pf-ws/authn/flows/" + flowId, requestOptions)
        .then(response => response.text())
        .then(result => console.log(result))
        .catch(error => console.log('error', error));
}
```

When the authentication API is in the `DEVICE_PROFILE_REQUIRED` state, your application should `submitDeviceProfile` with the value from the `profileDevice(callback)` method.

3. When you complete the steps in Configuring an adapter instance, set the device profiling method. The provider or the adapter can receive the payload sent by the authentication API.

   Choose from:

   - If the provider is configured, you must add the payload to the `checkUsernamePassword` `requestOptions` API call in the captchaResponse.

     ```
     {
       "username": "string",
       "password": "string",
       "rememberMyUsername": "boolean",
       "thisIsMyDevice": "boolean",
       "captchaResponse": "string"
     }
     ```

   - When using the PingOne Protect IdP Adapter, submit the payload following the `submitDeviceProfile` `requestOptions` action of the authentication API.

     > **Note**
     >
     > For more information on these options, see Exploring the authentication API in the PingFederate documentation.

Copyright © 2025 Ping Identity Corporation

Configuring an adapter instance

To get started with the integration, deploy the PingOne Protect Integration Kit files to your PingFederate directory.

Steps

1. In the PingFederate administrative console, go to Authentication > Integration > IdP Adapters. Click Create New Instance.
2. On the Type tab, set the basic adapter instance attributes:
    1. In the Instance Name field, enter a name for the adapter instance.
    2. In the Instance ID field, enter a unique identifier for the adapter instance.
    3. In the Type list, select PingOne Protect IdP Adapter. Click Next.
3. Optional: On the IdP Adapter tab, in the Additional User Attributes (optional) section, you can configure additional attributes to send to PingOne Protect.
4. Optional: In the Additional Risk Predictors (optional) section, you can configure additional risk predictors to send to PingOne Protect.
    1. Click Add a new row to 'Additional Risk Predictors (optional)'.
    2. In the Incoming Attribute Name field, enter the name of an attribute from any authentication source that appears earlier in your PingFederate authentication policy than the PingOne Protect IdP Adapter.
    3. In the PingOne Protect Attribute list, select the PingOne attribute that you want to populate.
    4. In the Action column, click Update.
    5. To add more attributes, repeat steps a - d.
5. Optional: On the IdP Adapter tab, in the PingOne Protect API Response Mappings section, map the attributes from PingOne Protect Evaluation API response to the attribute contract.

    These attributes will become available in your PingFederate authentication policy.

    1. Click Add a new row to 'PingOne Protect API Response Mappings'.
    2. In the Local Attribute field, enter a name of your choosing for an attribute.
    3. In the PingOne Protect API Attribute Mapping field, enter the JSON Pointer syntax for the source PingOne attribute as shown in JSON Pointer syntax reference.

        Example:

        For example, the JSON pointer `/details/ipAddressReputation/level` will return the IP address repuation level, such as `LOW`.
    4. In the Action column, click Update.
    5. To add more attributes, repeat steps a - d.

Copyright © 2025 Ping Identity Corporation

> **Note**
>
> If you skip performing a fraud evaluation in the adapter, the response mappings might not be returned.

6. On the IdP Adapter tab, configure the adapter instance by referring to IdP Adapter settings reference. Click Next.
7. On the Actions tab, test your connection to PingOne Protect. Resolve any issues that are reported, and then click Next.
8. On the Extended Contract tab, add any attributes that you included in the PingOne Protect API Response Mappings section of the IdP Adapter tab. Click Next.
9. On the Adapter Attributes tab, set pseudonym and masking options as shown in Set pseudonym and masking options in the PingFederate documentation. Click Next.
10. On the Adapter Contract Mapping tab, configure the contract fulfillment details for the adapter as shown in Define the IdP adapter contract in the PingFederate documentation. Click Next.
11. On the Summary tab, check and save your configuration. Click Save.

JSON Pointer syntax reference

You can find a complete technical description of JSON Pointer syntax in JavaScript Object Notation (JSON) Pointer.

Copyright © 2025 Ping Identity Corporation

Example PingOne Protect JSON payload

Copyright © 2025 Ping Identity Corporation

```
{
    "_links": {
        "self": {
            "href": "https://api.pingone.eu/v1/environments/abfba8f6-49eb-49f5-a5d9-80ad5c98f9f6/riskEvaluations/
5fa661b4-6be7-468c-afa4-e1213cc15d5a"
        },
        "environment": {
            "href": "https://api.pingone.eu/v1/environments/abfba8f6-49eb-49f5-a5d9-80ad5c98f9f6"
        },
        "event": {
            "href": "https://api.pingone.eu/v1/environments/abfba8f6-49eb-49f5-a5d9-80ad5c98f9f6/riskEvaluations/
5fa661b4-6be7-468c-afa4-e1213cc15d5a/event"
        }
    },
    "id": "5fa661b4-6be7-468c-afa4-e1213cc15d5a",
    "environment": {
        "id": "abfba8f6-49eb-49f5-a5d9-80ad5c98f9f6"
    },
    "createdAt": "2023-06-28T09:40:36.682Z",
    "updatedAt": "2023-06-28T09:40:36.682Z",
    "event": {
        "completionStatus": "IN_PROGRESS",
        "targetResource": {
            "id": "{{targetResourceID}}",
            "name": "Jira"
        },
        "ip": "156.35.85.124",
        "flow": {
            "type": "AUTHENTICATION"
        },
        "session": {
            "id": "{{sessionID}}"
        },
        "user": {
            "id": "john",
            "name": "John DeMock",
            "type": "EXTERNAL",
            "groups": [
                {
                    "name": "dev"
                },
                {
                    "name": "sre"
                }
            ]
        },
        "sharingType": "SHARED",
        "browser": {
            "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_6) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/80.0.3987.122 Safari/537.36"
        },
        "sdk": {
            "signals": {}
        }
    },
    "riskPolicySet": {
        "id": "a2c47c78-39fd-4f2d-9633-1d61960893c9",
        "name": "Default policy"
    },
    "result": {
```


Copyright © 2025 Ping Identity Corporation

```
            "value": "",
            "level": "HIGH",
            "score": 80,
            "source": "SCORE_AGGREGATION",
            "recommendedAction": "BOT_MITIGATION",
            "type": "VALUE"
        },
        "details": {
            "ipAddressReputation": {
                "score": 0,
                "domain": {
                    "asn": 766,
                    "sld": "uniovi",
                    "tld": "es",
                    "organization": "universidad de oviedo",
                    "isp": "entidad publica empresarial red.es"
                },
                "level": "LOW"
            },
            "anonymousNetworkDetected": false,
            "country": "spain",
            "device": {
                "id": "Id-4c0013c4-5739-440f-91f4-147f460dd2ec",
                "estimatedDistance": 0,
                "os": {
                    "name": "Mac OS X"
                },
                "browser": {
                    "name": "Chrome"
                }
            },
            "state": "asturias",
            "city": "oviedo",
            "impossibleTravel": false,
            "suspiciousDevice": {
                "level": "LOW",
                "type": "DEVICE"
            },
            "ipVelocityByUser": {
                "level": "LOW",
                "threshold": {
                    "source": "MIN_NOT_REACHED"
                },
                "velocity": {
                    "distinctCount": 1,
                    "during": 3600
                },
                "type": "VELOCITY"
            },
            "geoVelocity": {
                "level": "LOW",
                "type": "GEO_VELOCITY"
            },
            "botDetectionBasic": {
                "level": "HIGH",
                "reason": "Browser loading anomaly",
                "detected": {
                    "rule": {
                        "id": 628
                    }
                },
                "type": "BOT"
```


Copyright © 2025 Ping Identity Corporation

```
        },
        "userRiskBehavior": {
            "reason": "Not enough information to assess risk score",
            "status": "NOT_AVAILABLE",
            "type": "USER_RISK_BEHAVIOR"
        },
        "ipRisk": {
            "level": "LOW",
            "type": "IP_REPUTATION"
        },
        "userBasedRiskBehavior": {
            "reason": "Not enough information to assess risk score",
            "status": "IN_TRAINING_PERIOD",
            "type": "USER_RISK_BEHAVIOR"
        }
    }
}
```


JSON Pointer syntax

Description	JSON Pointer	Example value
Name of the user	`/event/user/name`	`John DeMock`
IP Address Reputation level	`/details/ipAddressReputation/level`	`LOW`

> **Note**
>
> To populate an attribute with the entire JSON response, leave the **PingOne Protect API Attribute Mapping** field blank.

IdP Adapter settings reference

Field descriptions for the PingOne Protect IdP Adapter configuration screen.

Standard fields

Field	Value	Description
PingOne Environment	*<PingOne Connection>*	**Your PingOne Environment. Create connections in System > External Systems > PingOne Connections. This field is blank by default.**
PingOne Risk Policy	*<PingOne Risk Policy Name>*	**The risk policy used by PingOne for the risk evaluation. Overrides the environment and global default policy selections. This list is populated when you select a PingOne Environment.**

Copyright © 2025 Ping Identity Corporation

Advanced fields

Field	Value	Description
Include Dynamic Device Profile	• Enabled • Disabled	When enabled, PingFederate will include a device profile in the risk evaluation.
Device Profiling Timeout	`5000` (default)	The amount of time in milliseconds that PingFederate waits for the device profiling script to collect device details. Applies only if Device Profiling Method is set to Captured by this adapter.
Device Profile Cookie Name	`pingone.protect.device.profile` (default)	The name of the cookie that indicates whether the device profile has been captured.
Failure Mode	• Continue with fallback policy decision • Fail (default)	When PingOne Protect is unavailable or an error occurs, this setting determines whether the user's sign-on attempt should fail or continue with a pre-determined policy decision.
Fallback Policy Decision Value	• `LOW` • `MEDIUM` (default) • `HIGH` • `unknown`	The fallback fraud evaluation level to use in the authentication policy when the PingOne Protect service is unavailable or an error occurs, and Failure Mode is set to Continue with fallback policy decision.
API Request Timeout	`2000` (default)	The amount of time in milliseconds that PingFederate allows when establishing a connection with PingOne Protect or waiting for a response to a request. A value of `0` disables the timeout.
Proxy Settings	• No Proxy • System Defaults (default) • Custom	Defines proxy settings for outbound HTTP requests.
Custom Proxy Host	*<Proxy server host>*	The proxy server host name to use when Proxy Settings is set to Custom. This field is blank by default.

Copyright © 2025 Ping Identity Corporation

Field	Value	Description
Custom Proxy Port	*<Proxy server port>*	The proxy server port to use when Proxy Settings is set to Custom. This field is blank by default.
Custom Connection Pool	`50` (default)	The number of connections to PingOne Protect. Can be between `25` and `200`. Recommended that the number be left at the default value.

Configuring a provider instance

The PingOne Protect provider works similarly to CAPTCHA providers and can detect bot activity before the password credential validator (PCV) is triggered.

About this task

> **Note**
>
> A provider instance isn't required, but having one is necessary for bot detection.

Steps

1. In the PingFederate administrative console, go to System > External Systems > CAPTCHA and Risk Providers. Click Create New Instance.
2. On the Type tab, set the basic provider instance attributes:
    1. In the Instance Name field, enter a name for the provider instance.
    2. In the Instance ID field, enter a unique identifier for the adapter instance.
    3. In the Type list, select PingOne Protect Provider. Click Next.
3. On the Instance Configuration tab:
    1. In the PingOne Environment field, select your PingOne Connection and Environment Name.
    2. Click Show Advanced Fields.
    3. Optional: To allow the adapter to evaluate the risk rather than the provider, clear the Enable Risk Evaluation check box.
    4. In the Field Value list for PingOne Risk Policy section, select your risk policy.
    5. The Custom Connection Pool setting controls the number of connections to PingOne Protect.

       This value can be between `25` and `200`, but best practice is to use the default value.
    6. Click Next.
4. On the Summary tab, click Save.

Copyright © 2025 Ping Identity Corporation

5. Enable the HTML Form adapter to use the provider:
    1. Go to Authentication > Integration > IdP Adapters. Select your HTML Form IdP Adapter from the list.
    2. On the IdP Adapter tab, click Show Advanced Fields.
    3. In the Risk Provider list, select your PingOne Protect provider instance.
    4. Select one or more of the following check boxes.

Check box	Description
Risk for authentication	Enable for the login form to prevent automated attacks
Risk for password change	Enable for the password change form to prevent automated attacks
Risk for password reset	Enable for the password reset and account unlock features to prevent automated attacks
Risk for username recovery	Enable for the username recovery features to prevent automated attacks

6. Optional: Set the device profile settings for the PingOne Protect adapter:
    1. Go to Authentication > Integration > IdP Adapters.
    2. Select your PingOne Protect IdP adapter from the list.
    3. On the IdP Adapter tab, click Show Advanced Fields.
    4. Optional: Enable Include Device Dynamic Profile.

    When enabled, the adapter will load the device profiling page if it hasn't already received the device profile payload from the provider.

Setting risk evaluation for active sessions

To allow risk evaluation for users with active sessions, you can enable a session to the PingOne Protect IdP Adapter.

To configure a session to the adapter, see Configuring authentication sessions.

Using custom risk predictors

The PingOne Protect IdP Adapter lets you use attributes from your PingFederate authentication flow as risk predictors in PingOne Protect.

About this task

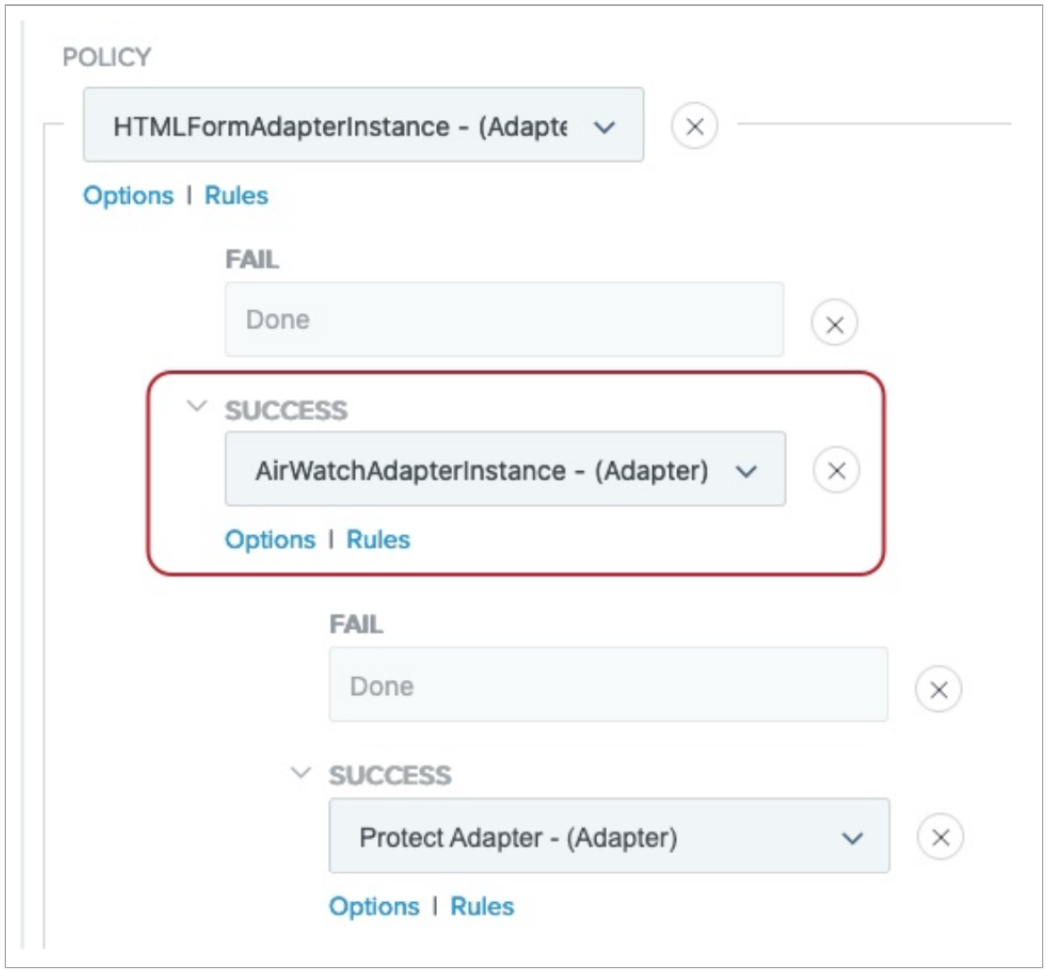
You can find an overview of risk predictors in Predictors in the PingOne Protect documentation.

Copyright © 2025 Ping Identity Corporation

The following steps provide an example that shows how to include the device security state from a mobile device management (MDM) service in the PingOne Protect risk evaluation.

Steps

1. **Make the predictor available as an attribute in your PingFederate authentication policy:**



   1. **Add the source of the predictor data to your authentication policy.**

      MDM Example

      1. **Add a mobile device management adapter.**
      2. **On the Extended Contract tab of the configuration for that adapter instance, the attribute that holds the security state of the user's device is called `ComplianceStatus`.**

   2. **Later in the flow, add the PingOne Protect IdP Adapter that you configured in Configuring an adapter instance.**

2. **In PingOne Protect, add the risk predictor and include it in your risk policy.**

   MDM Example

   1. **Add a predictor with the JSON pointer `$\{event.ComplianceStatus}`.**

      Learn more in Predictors in the PingOne Protect documentation.

Copyright © 2025 Ping Identity Corporation

3. In the Risk Predictors table of your PingOne Protect IdP Adapter configuration, map the predictor attribute from your PingFederate authentication policy to the JSON pointer you defined in PingOne Protect.

   Learn more in Configuring an adapter instance.

   MDM Example

   1. Map the PingFederate `ComplianceStatus` attribute to the PingOne Protect predictor in your PingOne Protect IdP Adapter configuration.



      During the authentication flow, the PingOne Protect IdP Adapter gets the predictor attribute from the PingFederate authentication policy and passes it to PingOne Protect. Next, PingOne Protect compares the value to the risk levels you defined and includes it in the risk evaluation.

Adding risk level results to your authentication policy

By modifying your PingFederate authentication policy to include the risk evaluation (`LOW`, `MEDIUM`, and `HIGH`) from PingOne Protect, you can dynamically change authentication requirements based on security risk level.

About this task

These steps are designed to help you add to an existing authentication policy. Learn more about configuring authentication policies in Authentication policies in the PingFederate documentation.

For new deployments, you should allow for a training period. To do this, configure your policy to pass traffic through the PingOne Protect IdP Adapter and continue regardless of the risk evaluation result. When you are ready to end the training period, adjust your authentication policy as described here.

> **Important**
>
> When the authentication flow finishes, PingFederate informs PingOne Protect whether the user ultimately succeeded or failed. This is an important consideration when designing your authentication flow.
> For example, a user receives a risk evaluation of `HIGH`, but ultimately completes the PingFederate authentication policy successfully. Based on that success, PingOne Protect now considers the user authentic and lowers the risk evaluation to `MEDIUM` or `LOW` on the next attempt.

Steps

1. In the PingFederate administrative console, go to the Policies tab.

   Choose from:

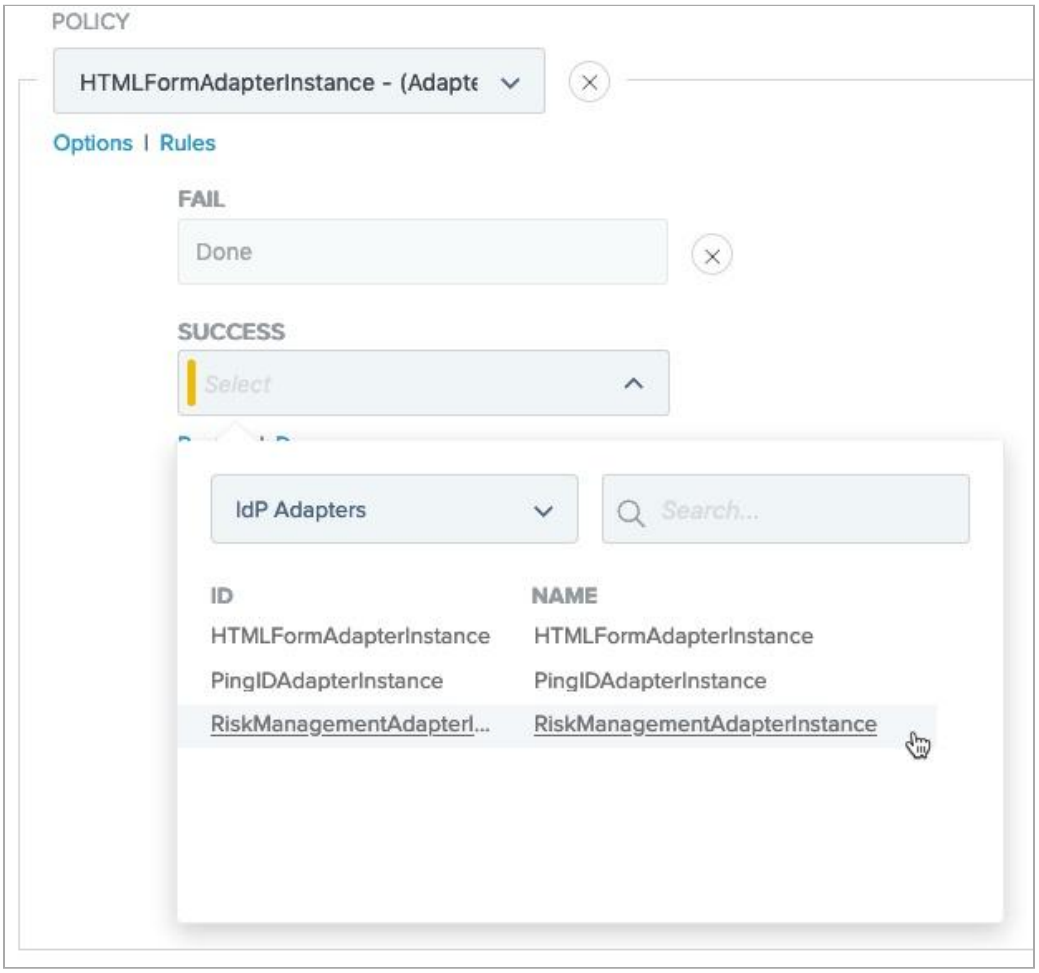
   - For PingFederate 10.1 or later: go to Authentication > Policies > Policies.

Copyright © 2025 Ping Identity Corporation

- For PingFederate 10.0 or earlier: go to Identity Provider > Authentication Policies > Policies.

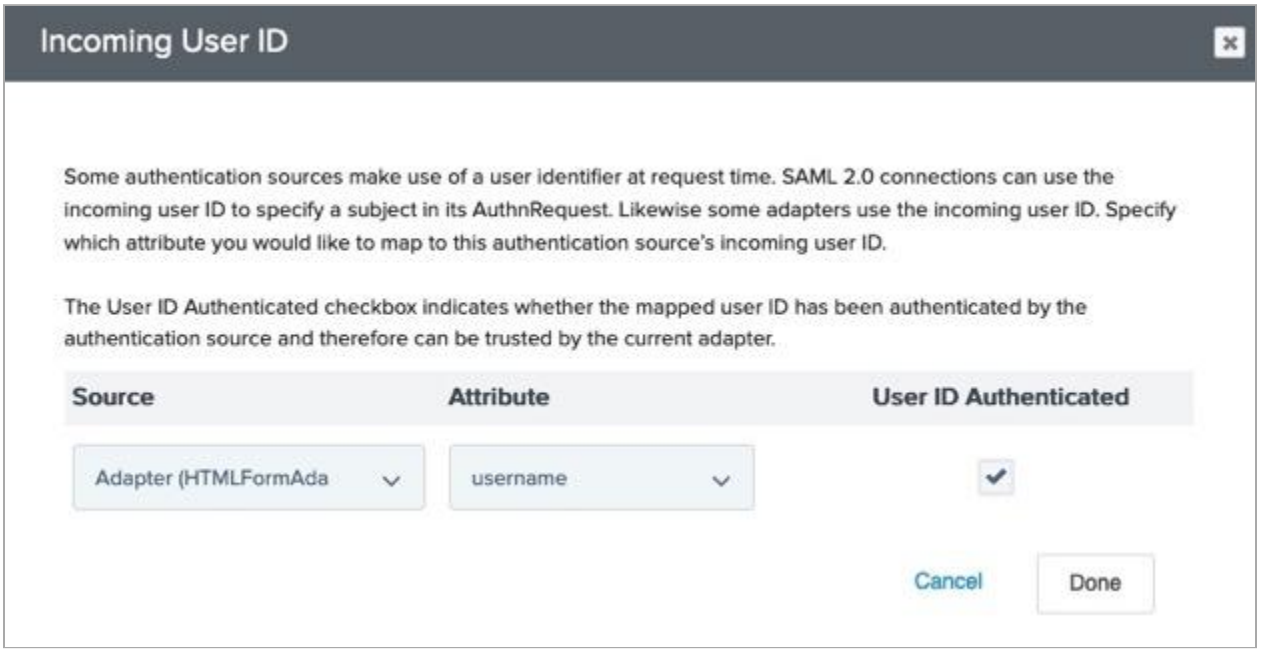
2. Select the IdP Authentication Policies checkbox.
3. Open an existing authentication policy, or click Add Policy.

   Learn more in Defining authentication policies.
4. In the Policy area, in the Select list, select a PingOne Protect IdP Adapter instance.



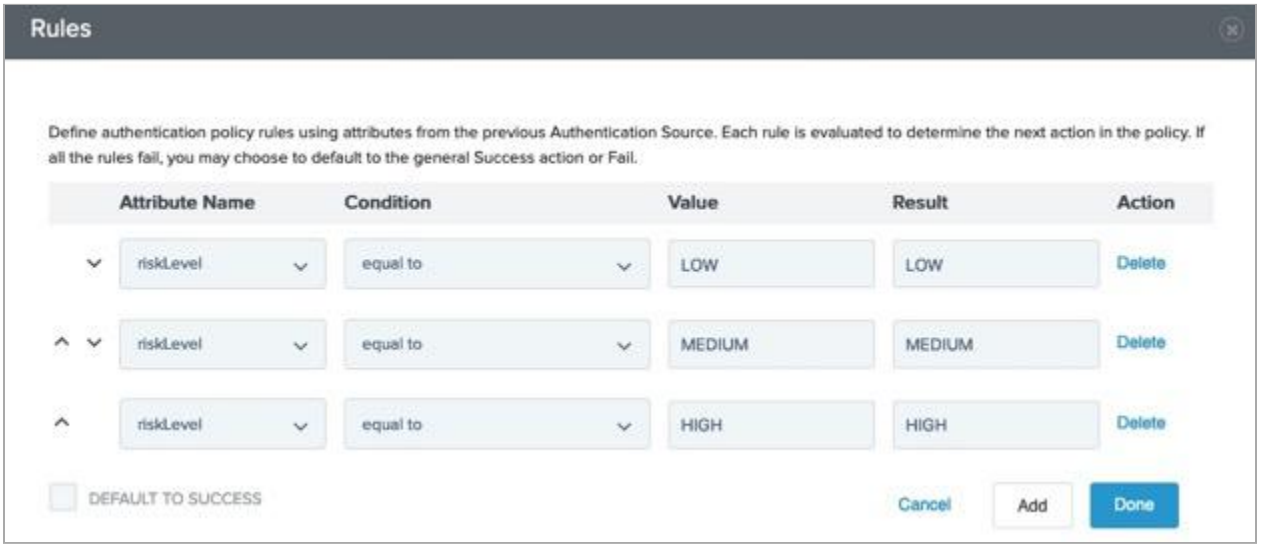
5. Map the user ID into the PingOne Protect IdP Adapter instance:

Copyright © 2025 Ping Identity Corporation



1. Under the PingOne Protect IdP Adapter instance, click Options.
2. In the Options dialog, in the Source list, select a previous authentication source that collects the user ID.
3. In the Attribute list, select the user ID. Click Done.

6. Define policy paths based on risk results:



1. Under the PingOne Protect IdP Adapter instance, click Rules.
2. In the Rules dialog, from the Attribute Name list, select riskLevel or riskValue.
3. In the Condition list, select equal to.
4. In the Value field, if you selected riskLevel, enter `LOW`, `MEDIUM`, or `HIGH`.

   If you selected riskValue, enter one of the risk values that you configured in PingOne.

Copyright © 2025 Ping Identity Corporation

5. In the Result field, enter a name.

   This appears as a new policy path that branches from the authentication source.

6. Optional: To add more policy paths, click Add and repeat steps 6b-6e.

7. Optional: Clear the Default to success check box.

8. Click Done.

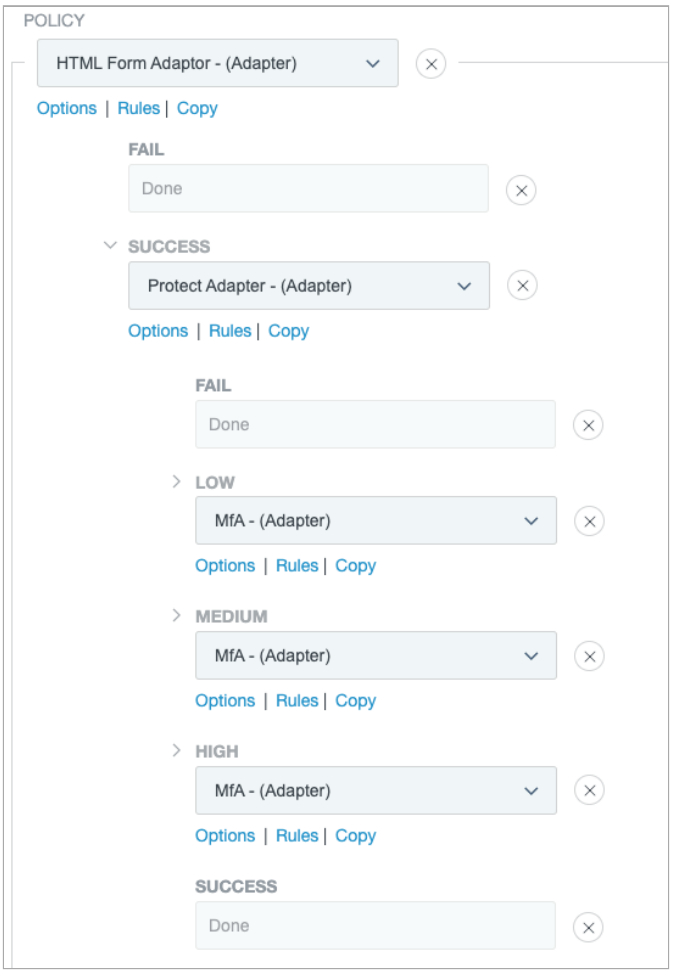
7. Complete the authentication policy:

   1. Configure each of the policy paths.

   2. Optional: To allow users continue to sign on by satisfying stricter authentication requirements when PingOne Protect is unreachable or returns an error, do one of the following.

      Choose from:

      - In your PingOne Protect IdP Adapter instance, set the Failure mode as shown in IdP Adapter settings reference.
      - In your authentication policy, set the Fail outcome of the PingOne Protect IdP Adapter instance to point to a second authentication factor, as shown in the example below.



8. Click Done.

Copyright © 2025 Ping Identity Corporation

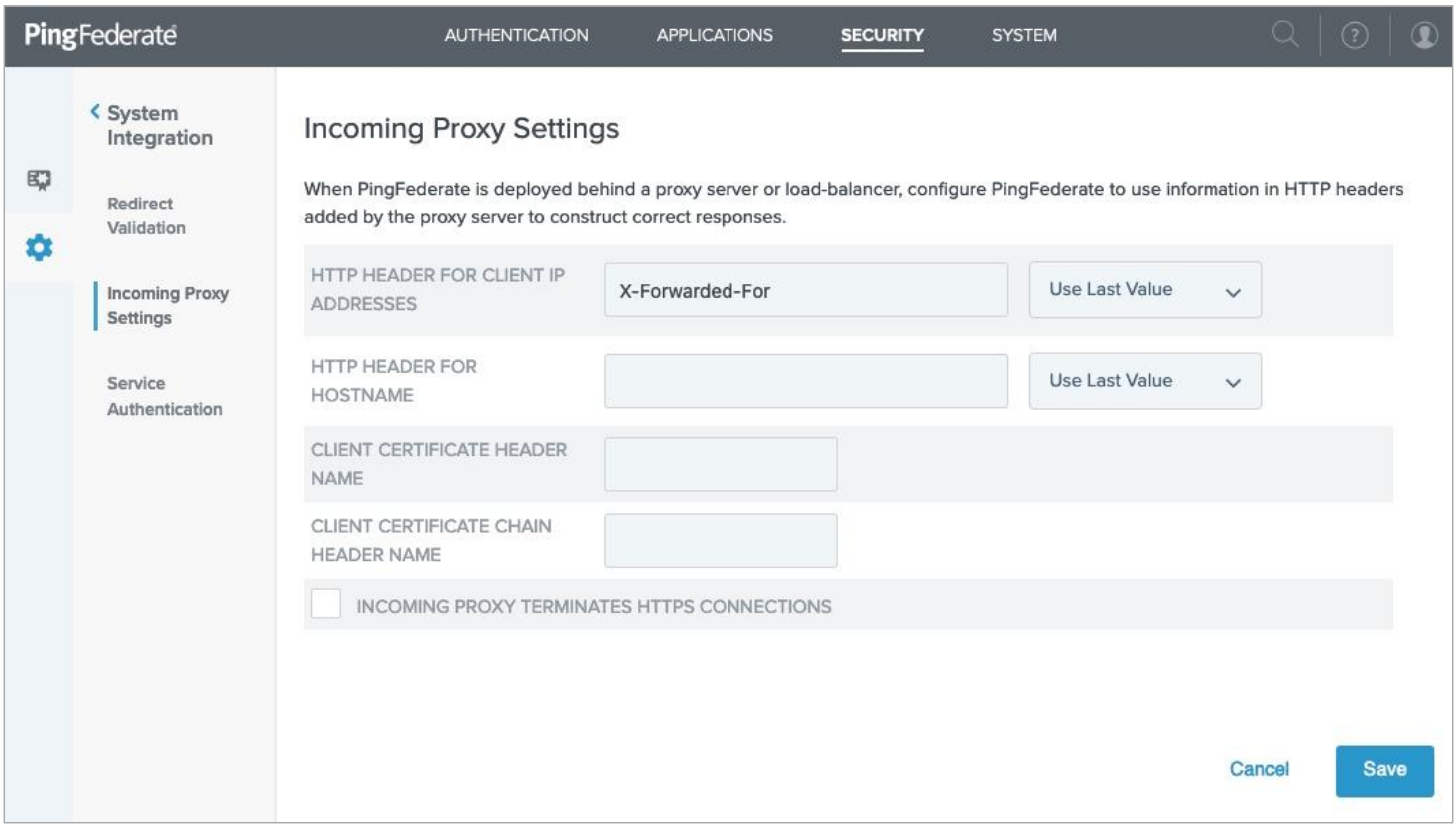
9. In the Policies window, click Save.

Configuring PingFederate to forward IP addresses

The PingOne Protect service needs the client's public-facing (rather than local) IP address for the risk assessment.

Steps

1. In the PingFederate administrative console, go to Security > System Integration > Incoming Proxy Settings.
2. In the HTTP Header for Client IP Addresses field, enter `X-Forwarded-For`.
3. Leave Use Last Value selected.
4. Click Save.



PingOne Protect Native SDKs

You can use the PingOne Protect SDK to collect information for additional risk-related variables and then pass this information on to the risk evaluation.

Learn more in the following SDKs:

- PingOne Protect SDK for Web
- PingOne Protect SDK for iOS
- PingOne Protect SDK for Android

Copyright © 2025 Ping Identity Corporation

Troubleshooting

Enabling debug logging

To help with troubleshooting or monitoring, you can turn on activity logging for PingFederate, the PingOne Protect IdP Adapter, or both.

About this task

This task is optional. You can use logging for troubleshooting or analytics.

Learn more about logging in Enabling debug messages and console logging in the PingFederate documentation.

Steps

1. Open the `<pf_install>/pingfederate/server/default/conf/log4j2.xml` file for editing.
2. To log activity for PingFederate and all adapters:
    1. Find the following section in the file:

    ```
    <AsyncRoot level="INFO" includeLocation="false">
            <!-- <AppenderRef ref="CONSOLE" /> -->
            <AppenderRef ref="FILE" />
    </AsyncRoot>
    ```

    2. Change `INFO` to `DEBUG`.

    The following code snippet shows `DEBUG` in bold for visibility:

    ```
    <AsyncRoot level="DEBUG" includeLocation="false">
            <!-- <AppenderRef ref="CONSOLE" /> -->
            <AppenderRef ref="FILE" />
    </AsyncRoot>
    ```

    3. Optional: To see the adapter activity in the console and the log file, remove the comment tags (`<!--` and `-->`) that surround the `CONSOLE` line:

    ```
    <AsyncRoot level="INFO" includeLocation="false">
            <AppenderRef ref="CONSOLE" />
            <AppenderRef ref="FILE" />
    </AsyncRoot>
    ```

3. To log activity relating to the PingOne Protect IdP Adapter, do one of the following.

> **Tip**
>
> You can use this information with a third-party log analysis tool to monitor for important events, such as when a sign-on event has a high-risk Result.

Copyright © 2025 Ping Identity Corporation

Choose from:

- To log activity for the PingOne Protect IdP Adapter and its HTTPS and component activity, add the following line:

```
<Logger name="com.pingidentity.adapters.pingone.risk" level="DEBUG"/>
```

- To log activity for the adapter's HTTPS activity and other components but not the adapter itself, add the following line:

```
<Logger name="com.pingidentity.adapters.pingone.risk.shade" level="DEBUG"/>
```

- To log activity for the PingOne Protect IdP Adapter but not its HTTPS or component activity, add the following lines:

```
<Logger name="com.pingidentity.adapters.pingone.risk" level="DEBUG"/>
<Logger name="com.pingidentity.adapters.pingone.risk.shade" level="INFO"/>
```

4. Save the file.

Troubleshooting information

The following information addresses technical situations that you might encounter after setting up the PingOne Protect Integration Kit.

Copyright © 2025 Ping Identity Corporation

Situation	Information
Using the Risk Provider with PingFederate 11.3 results in the following error: `Refused to load https://apps.<PingOne regional domain>/signals/sdk/pong.css?body=H0kaJgZiVxYYAC0DFghGXkdwQwNFRkdLMgFVRgIEGy9PDhApCgoLA0BXCEkU&e=2 because it does not appear in the style-src directive of the Content Security Policy.`	Update the HTML form template in `$PF_HOME/server/default/conf/templates`. HTML pages implementing ContentSecurityPolicy restrictions might require updating the `style-src` CSP settings when adding the `signals-sdk` JavaScript file to the page. Update `style-src` to include the https://apps.*<PingOne regional domain>* host name. For example, PingFederate 11.3 has updated default templates with strict CSP settings. To use the `signals-sdk` JavaScript file with PingFederate 11.3 default templates, update the following line in the template's CSP settings: `<meta http-equiv="Content-Security-Policy" content="script-src 'self' 'nonce-$CSPNonce'; style-src 'self'; img-src 'self'; font-src 'self';" />` Updated line: `<meta http-equiv="Content-Security-Policy" content="script-src 'self' 'nonce-$CSPNonce'; style-src 'self' https://apps.<PingOne regional domain>; img-src 'self'; font-src 'self';" />` **Note** For more information on your PingOne regional domain, see IP address and domain reference in the PingOne documentation.

Copyright © 2025 Ping Identity Corporation

Situation	Information
Include Device Profile is selected in the adapter configuration, but the device profile does not affect the risk result or appear in the response from PingOne Protect.	When this setting is enabled, it is possible for an error to prevent the device profile from reaching PingOne Protect. Because PingOne Protect considers the device profile to be optional, it still successfully returns a Result to the adapter. The adapter logs a warning in the PingFederate error log about the missing device profile and returns a `Success` result to the authentication policy. As a result, the process succeeds but no device profile information is available. To address the problem generating or sending the device profile, review the steps in Integrating device profiling. Make sure you have completed the correct set of steps (authentication page versus web application) and completed the steps exactly as described.
Authentication fails, and the log files show a NullPointerException.	Prior to release 1.0.3 of the PingOne Protect Integration Kit, there were cases where a NullPointerException occurred if the flow involved an X.509 certificate. Verify that you are using version 1.0.3 or higher.

Release notes

Updated January 2024.

Changelog

The following is the change history for the PingOne Protect Integration Kit.

PingOne Protect Integration Kit 1.0.5 — March 2025

- Modifications have been made to allow use of the PingID Device Trust risk predictor in risk evaluations, based on the data provided by the PingID Device Trust agent installed on user computers.
- The included PingOne Protect (Signals) SDK has been updated to version 5.5.0.

Copyright © 2025 Ping Identity Corporation

PingOne Protect Integration Kit 1.0.4 — November 2024

- The included PingOne Protect (Signals) SDK has been updated to version 5.3.7.
- PingFederate now takes into account the additional "recommended actions" that were added for risk evaluations.
- PingFederate now takes into account the flow subtype if one was specified for a risk evaluation.

PingOne Protect Integration Kit 1.0.3 — January 2024

- There were cases where user authentication would fail if the flow involved an X.509 certificate. This issue has been fixed (STAGING-21876).

PingOne Protect Integration Kit 1.0.2 — November 2023

- A new setting called Custom Connection Pool has been added to allow adjustment of the number of connections to PingOne Protect. Default value is 50.
- General performance improvements

PingOne Protect Integration Kit 1.0.1 — July 2023

- Critical bug fix

PingOne Protect Integration Kit 1.0 — July 2023

- Initial release

Known Issues and Limitations

The following are known issues or limitations for the PingOne Protect Integration Kit.

Known issues

There are no known issues.

Known limitations

There are no known limitations.

Download manifest

The following files are included in the PingOne Protect Integration Kit `.zip` archive:

- `ReadMeFirst.pdf` – Contains links to this online documentation
- `Legal.pdf` – Copyright and license information

Copyright © 2025 Ping Identity Corporation

PingOne Credentials Integration Kit

The PingOne Credentials Integration Kit allows PingFederate to verify and return requested credentials through a credential verification presentation session.

Features

- Initiate a credential verification process as part of a user sign-on flow.
- Supports the PingFederate authentication API.
- Supports the JavaScript Widget for the PingFederate Authentication API.

Components

PingOne Credentials IdP Adapter

- Allows PingFederate to use PingOne Credentials to trigger a credential verification challenge as part of the PingFederate authentication policy.

Templates

- Allow the adapter to show the status of the credential verification process.
- Allow you to modify the appearance of pages shown to the user during authentication.

Language packs

- Allow you to customize or localize the messages returned by the PingFederate authentication application programming interface (API) and shown on the templates during authentication. Learn more in Localizing messages for end users in the PingFederate documentation.

Intended audience

This document is intended for PingFederate administrators.

If you need help during the setup process, see the following resources:

- PingOne Credentials in the PingOne documentation
- The following sections of the PingFederate documentation:
  - PingFederate authentication API
  - Managing IdP adapters
  - Authentication policies

System requirements

- PingFederate 11.2 or later

Copyright © 2025 Ping Identity Corporation

- To allow PingFederate to make outbound HTTPS connections, you might need to allow the following host names in your firewall:
    - https://api.pingone.com, https://api.pingone.asia, https://api.pingone.com.au/, or https://api.pingone.eu
    - https://auth.pingone.com, https://auth.pingone.asia, https://auth.pingone.com.au/, or https://auth.pingone.eu

Overview of the SSO flow

With the PingOne Credentials Integration Kit, PingFederate includes PingOne Credentials in the sign-on flow.

Different device SSO flow description

1. The user initiates SSO with PingFederate on their first device. For example, a laptop. They complete the first authentication step, which might be provided by an HTML Form Adapter instance, for example.
2. The PingOne Credentials IdP Adapter contacts PingOne Credentials to initiate the credential verification process.
3. PingOne Credentials provides a QR code image URL and a Open Wallet button URL to the adapter.
4. The PingOne Credentials IdP Adapter presents an HTML page that shows the QR code and Open Wallet button to the user on their first device.
5. The user scans the QR code through their wallet app on a second device.
6. The wallet app on the user's second device presents a consent prompt to the user to confirm whether the requested credentials can be presented for verification.
7. The user confirms or cancels the request.
8. If the user confirms that the requested credentials may be shared, the wallet app presents the requested credentials to PingOne Credentials for verification, following the configured Presentation Protocol.
9. The PingOne Credentials IdP Adapter polls PingOne Credentials while it waits for the result of the credential verification process.
10. PingOne Credentials provides the adapter with the result of the verification process.
11. (Optional) The PingOne Credentials IdP Adapter presents an HTML page on the user's first device that shows the success or failure message, depending on the verification result.
12. If the user completed the verification process successfully, PingFederate provides access to the requested resource on the user's first device.

Same device SSO flow description

1. The user initiates SSO with PingFederate on the same device that their wallet app is on. For example, a cellphone. They complete the first authentication step, which might be provided by an HTML Form Adapter instance, for example.
2. The PingOne Credentials IdP Adapter contacts PingOne Credentials to initiate the credential verification process.
3. PingOne Credentials provides a QR code image URL and a Open Wallet button URL to the adapter.

Copyright © 2025 Ping Identity Corporation

4. The PingOne Credentials IdP Adapter presents an HTML page that shows the QR code and Open Wallet button to the user on the same device.
5. The user clicks Open Wallet and is redirected to the wallet app on the same device.
6. The wallet app presents a consent prompt to the user to confirm whether the requested credentials can be presented for verification.
7. If the user confirms that the requested credentials may be shared, the wallet app shares the requested credentials with PingOne Credentials, following the selected Presentation Protocol.
8. The PingOne Credentials IdP Adapter polls PingOne Credentials while it waits for the result of the verification process.
9. PingOne Credentials provides the adapter with the result of the credential verification process.
10. (Optional) The PingOne Credentials IdP Adapter presents an HTML page on the user's device that shows the success or failure message, depending on the verification result.
11. If the user completed the verification process successfully, PingFederate provides access to the requested resource on the same device.

Setup

Deploying the integration files

To get started with the integration, deploy the PingOne Credentials Integration Kit files to your PingFederate directory.

Steps

1. Download the PingOne Credentials Integration Kit `.zip` archive from the Ping Identity Integration Directory or the PingFederate product downloads page.
2. Stop PingFederate.
3. If you're upgrading an existing deployment, back up your customizations and delete earlier versions of the integration files:
    1. Back up any PingOne Credentials Integration Kit files that you customized in the `<pf_install>/pingfederate/server/default/conf/` directory.
    2. Delete the `pf-pingone-credentials-adapter-<version>.jar` file from your `<pf_install>/pingfederate/server/default/deploy` directory.
4. Extract the `.zip` archive and merge the contents of the `dist` directory with your `<pf_install>/pingfederate` directory.
5. If there is more than one version of the `pf-authn-api-sdk-<version>.jar` file in your `<pf_install>/pingfederate/server/default/lib` directory, delete all but the latest version of the file.
6. If you backed up any customized files, modify the new files with your customizations.
7. Start PingFederate.

Copyright © 2025 Ping Identity Corporation

8. If you operate PingFederate in a cluster, repeat steps 2-7 for each engine node.

Connecting PingFederate to PingOne

To allow PingFederate to communicate with PingOne, create a connection.

PingFederate 10.2 or later

Connecting PingFederate 10.2 or later
About this task

Create a connection between PingFederate and PingOne. In PingFederate 10.2 and later, you can use this connection for all PingFederate components that communicate with PingOne.

> **Note**
>
> If you already have a connection from PingOne to PingFederate, ensure that the **Identity admin** and **Environment admin** roles are enabled. Skip the steps below.

Steps

1. Create credentials in PingOne:
    1. On the PingOne administrative console, go to Integrations > PingFederate and click the + icon.
    2. On the Create Connection Profile window, enter a connection name, then click Save.
    3. After the New Credential Created window opens, click Copy To Clipboard, then Done.
2. Use the credentials to create a connection in PingFederate:
    1. On the PingFederate administrative console, go to System > External Systems > PingOne Connections and click + Add Connection.
    2. On the Add Connection window, paste the credential information in the Paste The Credential Here field.
    3. In the Connection Name field, enter a unique name and click Save.

Copyright © 2025 Ping Identity Corporation

PingFederate 10.1 or earlier

Connecting PingFederate 10.1 or earlier
About this task

In PingOne, create a set of credentials for PingFederate. For PingFederate 10.1 and earlier, you must enter these credentials in each PingFederate component that communicates with PingOne.

Steps

1. On the PingOne administrative console, go to Applications > Applications and click the + icon.
2. On the Add Application window, enter a unique Application Name.
3. In the Application Type list, select Worker, then Save.
4. On the Roles tab, click Grant Roles, then select the Identity Data Administrator checkbox for the environment.
5. On the Overview or Configuration tab, note the Client ID, Client Secret, and Environment ID.

   You'll use these credentials in Configuring an adapter instance.
6. Click the toggle at the top of the application to activate it.

Configuring an adapter instance

Configure the PingOne Credentials IdP Adapter to determine how PingFederate communicates with PingOne Credentials.

Steps

1. In the PingFederate administrative console, go to Authentication > Integration > IdP Adapters. Click Create New Instance.
2. On the Type tab, set the basic adapter instance attributes.
   1. In the Instance Name field, enter a name for the adapter instance.
   2. In the Instance ID field, enter a unique identifier for the adapter instance.
   3. From the Type list, select PingOne Credentials IdP Adapter. Click Next.
3. On the IdP Adapter tab, in the Credential Types and Attributes section, define the credential types and attributes that you want users to present for verification.
   1. Click Add a new row to 'Credential Types and Attributes'.
   2. In the Credential Type list, select a credential type that you want users to present for verification.

      Example:

      `verifiedEmployee`
   3. In the corresponding Credential Attribute field, enter any attributes that you want users to present for verification.

Copyright © 2025 Ping Identity Corporation

Separate the attribute values with commas.

Example:

`firstName, lastName`

4. In the Action column, click Update.
5. To add more credential types and attributes, repeat steps a-d.

Because the credential issuer and credential verifier can be in different environments, the Credential Type list doesn't auto-populate credential types for the selected environment. Use the following issuer filters, Requested Issuer(s) by Environment ID and Requested Issuer(s) by Decentralized Identifiers (DID) to filter through environments and find the credential types that you want to include in the credential presentation request.

4. Optional: In the Requested Issuer(s) by Environment ID section, enter the PingOne environment ID of a trusted issuer.
    1. Click Add a new row to 'Requested Issuer(s) by Environment ID'.
    2. In the Issuer by Environment ID field, enter the environment ID of the issuer.
    3. In the Action column, click Update.
    4. To add more issuers, repeat steps a-d.
5. Optional: In the Requested Issuer(s) by Decentralized Identifiers (DID) section, enter the decentralized identifier of a trusted issuer.
    1. Click Add a new row to 'Requested Issuer(s) by Decentralized Identifiers (DID)'.
    2. In the Issuer by Decentralized Identifier field, enter the decentralized identifier of the issuer.
    3. In the Action column, click Update.
    4. To add more issuers, repeat steps a-d.
6. On the IdP Adapter tab, configure the adapter instance by referring to PingOne Credentials IdP Adapter settings reference. Click Next.
7. On the Extended Contract tab, add all of the attributes that you defined in the Credential Types and Attributes table. Then click Next.

    Using the following format, define each attribute separately:

    ```
    <credentialType>.<credentialAttribute>
    ```

    Example:

    For a `verifiedEmployee` credential type that has two attributes, entered in the Credential Attribute field as `firstName, lastName`, add the `firstName` attribute first:

    ```
    verifiedEmployee.firstName
    ```

    Then add the `lastName` attribute:

Copyright © 2025 Ping Identity Corporation

```
verifiedEmployee.lastName
```

8. On the Adapter Attributes tab, for subject, select the Pseudonym check box. Click Next.
9. On the Adapter Contract Mapping tab, click Configure Adapter Contract and fill out the configuration as necessary. Click Done, then Next.

> **Note**
>
> Make sure to click **Configure Adapter Contract** and complete a configuration, even if it's just **Configure Adapter Contract > Next > Next > Next > Done > Next**. Otherwise, the adapter contract fulfillment isn't mapped to the adapter.

10. On the Summary tab, review your configuration. Click Save.

Next steps

Complete the Extending the contract procedure.

PingOne Credentials IdP Adapter settings reference

Field descriptions for the PingOne Credentials IdP Adapter configuration screen.

Standard fields

Field	Description
PingOne Environment	For PingFederate 10.2 and later. Select the PingOne connection that you created in Connecting PingFederate to PingOne. This field is blank by default.
Environment ID	For PingFederate 10.1 and earlier. The environment ID that you noted in Connecting PingFederate to PingOne. This field is blank by default.
Region	For PingFederate 10.1 and earlier. Determines the PingOne API that the adapter communicates with. Select the region that appears on Settings > Environment > Properties in PingOne.
PingFederate Connection Client ID	For PingFederate 10.1 and earlier. The client ID that you noted in Connecting PingFederate to PingOne. This is required for automatic device pairing. This field is blank by default.

Copyright © 2025 Ping Identity Corporation

Field	Description
PingFederate Connection Client Secret	For PingFederate 10.1 and earlier. The client secret that you noted in Connecting PingFederate to PingOne. This is required for automatic device pairing. This field is blank by default.
Digital Wallet Application	The name of the application that captures the mobile application attributes for the wallet. This list is populated when you select a PingOne Environment.
App Consent Prompt	The message prompt that's shown to the user in the app before they consent to the presentation request. The default value is `Please accept`.
Presentation Protocol	The presentation protocol that the app should use to complete the presentation request. Available options are: • Ping Native • OpenID4VP

Advanced fields

Field	Description
Refresh Interval	The interval, in milliseconds, at which the adapter checks if the credential has been shared. The default value is `5000`.
Cookie Signing Key	The key that the adapter uses to sign application instance ID cookies. This value can't exceed 1000 characters. The default value is `some-secret-key`.
HTML Template Prefix	Identifies the set of HTML templates that the adapter uses. If you customize the template file names in the `/server/default/conf/template` directory, enter the new prefix here. The default value is `pingone-credentials`. For a description of the template files, see Download manifest.
Messages Files	Identifies the customizable language-pack file that the adapter uses to show messages on the templates. If you customize the `pingone-credentials-messages.properties` file name in the `/server/default/conf/language-packs` directory, enter the new name here. The default value is `pingone-credentials-messages`.

Copyright © 2025 Ping Identity Corporation

Field	Description
Error Message Key Prefix	Identifies the error messages in the language-pack file that the adapter uses to show messages on the templates. If you customize the error message names in `/server/default/conf/language-packs/pingone-credentials-messages.properties`, enter the new prefix here. The default value is `pingone.credentials.error.`.
Allow Credential Verification Retries	When credential verification fails, this setting determines whether to start a new session. This check box is cleared by default.
Service Unavailable Failure Mode	When the adapter is unable to access PingOne, this setting determines whether the adapter should block the user's sign-on attempt or bypass authentication. The default selection is `Bypass authentication`.
Show Success Screens	When the presentation session is successful, this setting determines whether the adapter shows a success page. This check box is selected by default.
Show Failed Screens	When the presentation session fails, this setting determines whether the adapter shows a failure page. This check box is selected by default.
Show Timeout Screens	When a presentation session times out, this setting determines whether the adapter shows a timeout page. This check box is selected by default.
State Timeout	The amount of time in seconds that the adapter will stay in the same presentation session state before the adapter times out. The default value is `300`. The minimum value is `60`.
API Request Timeout	The amount of time in milliseconds that PingFederate allows when establishing a connection with PingOne Credentials or waiting for a response to a request. A value of 0 disables the timeout. The default value is `5000`.
Proxy Settings	Defines proxy settings for outbound HTTP requests. The default value is System Defaults.
Custom Proxy Host	The proxy server host name to use when Proxy Settings is set to Custom. This field is blank by default.

Copyright © 2025 Ping Identity Corporation

Field	Description
Custom Proxy Port	The proxy server port to use when Proxy Settings is set to Custom. This field is blank by default.

Extending the contract

Apply the same extended contract changes to the policy contract and OpenToken SP adapter. Map the policy contract to the authentication policy's IdP adapter success step and the browser SSO of the SP and IdP connections.

Before you begin

This procedure assumes that you've created:

1. An OpenToken SP adapter. Learn more in Configuring an OpenToken SP adapter instance.
2. The authentication policy that you want to use the PingOne Credentials IdP Adapter in. Learn more in Authentication policies.
3. An SP connection for the OpenToken SP adapter and an IdP connection for the PingOne Credentials IdP Adapter. Learn more in SP connection management and Managing IdP adapters.

About this task

Complete this procedure to finish configuring the PingOne Credentials IdP Adapter and successfully use the same device or different device SSO flow to collect the credential attribute values presented from the user's digital wallet. Chained adapters can then use the collected attribute values later in the configured PingFederate authentication policy flow.

Credential attributes can't be added to the core attribute contract because administrators might require different credential types and attributes for verification. Unlike with the PingOne Verify integration kit, administrators must configure the PingOne Credentials IdP Adapter credential fields manually, then extend the attribute contract to match.

Steps

1. Create the same extended contract from the Configuring an adapter instance procedure in a PingFederate policy contract:

   Learn more in Policy contracts.

   > **Note**
   >
   > You will use this policy contract in steps 2, 4, and 5.

   1. Go to Authentication > Policies > Policy Contracts and click Create New Contract.
   2. On the Contract Info tab, in the Contract Name field, enter a unique value, then click Next.

      For example, `PingOne Credentials contract`.
   3. On the Contract Attributes tab, add the same attributes that you added to the extended contract in step 7 of the Configuring an adapter instance procedure. Click Save, then click Next.

Copyright © 2025 Ping Identity Corporation

For example, `verifiedEmployee.firstName` and `verifiedEmployee.lastName`.

4. On the Summary tab, click Save.

2. Map the policy contract that you created in step 2 to the PingOne Credentials IdP Adapter contract on the adapter's success step in the authentication policy:

   Learn more in Applying policy contracts or identity profiles to authentication policies.

   1. Go to Authentication > Policies > Policies and edit the authentication policy that you've included the PingOne Credentials IdP Adapter in.
   2. In the Policy section, expand the Success step containing the PingOne Credentials IdP Adapter, and click Contract Mapping under the adapter's Success step.
   3. On the Contract Fulfillment tab, add the same attributes that you added to the extended contract in step 7 of the Configuring an adapter instance procedure.

      For each attribute, select the adapter as the Source and add the attribute as the Value.

   4. On the Contract Fulfillment tab, click Done, then click Next.
   5. On the Summary tab, click Done.

3. Create the same extended contract from the Configuring an adapter instance procedure in your OpenToken SP adapter instance:

   Learn more in Extending an SP adapter contract.

   1. Go to Applications > SP Adapters and open your OpenToken SP adapter configuration.
   2. On the Extended Contract tab, add the same attributes that you added to the extended contract in step 7 of the Configuring an adapter instance procedure. Click Save, then click Next.
   3. On the Summary tab, click Save.

4. Map the policy contract that you created in step 2 to the Browser SSO section of the SP connection:

   Learn more in Configuring SP browser SSO.

   1. Go to Applications > SP Connections and open the connection that you used for your SP adapter.
   2. On the Browser SSO tab, click Configure Browser SSO, go to the Assertion Creation tab, click Configure Assertion Creation, and then go to the Attribute Contract tab.
   3. In the Extend the Contract section, add the same attributes that you added to the extended contract in step 7 of the Configuring an adapter instance procedure. Click Save, then Next.

      For each attribute, use `urn:oasis:names:tc:SAML:2.0:attrname-format:basic` as the Attribute Name Format.

   4. On the Authentication Source Mapping tab, confirm that your PingOne Credentials policy contract is mapped in the Authentication Policy Contract Name section.
   5. Open the authentication policy mapping, go to the Attribute Contract Fulfillment tab, and add the same attributes that you added to the extended contract in step 7 of the Configuring an adapter instance procedure.

      For each attribute, use Authentication Policy Contract as the Source and the attribute as the Value.

Copyright © 2025 Ping Identity Corporation

6. Click Save, then Done.

5. Map the policy contract that you created in step 2 to the Browser SSO section of the IdP connection:

   Learn more in Configuring IdP Browser SSO.

   1. Go to Authentication > IdP Connections and open the connection associated with the PingOne Credentials IdP Adapter.
   2. On the Browser SSO tab, click Configure Browser SSO, go to the User-Session Creation tab, click Configure User-Session Creation, and then go to the Attribute Contract tab.
   3. In the Extend the Contract section, add the same attributes that you added to the extended contract in step 7 of the Configuring an adapter instance procedure. Click Save, then Next.
   4. On the Target Session Mapping tab, confirm that your PingOne Credentials IdP Adapter is mapped in the Adapter Instance Name section.
   5. Open the adapter instance mapping, go to the Adapter Contract Fulfillment tab, and add the same attributes that you added to the extended contract in step 7 of the Configuring an adapter instance procedure.

      For each attribute, use Assertion as the Source and the attribute as the Value.
   6. Click Save, then Done.

Authentication API Support

You can use the PingFederate authentication API to integrate the PingOne Credentials IdP Adapter into your application.

The PingFederate Authentication API provides access to the current state of the authentication flow as a user steps through the PingFederate authentication policy. Learn more in PingFederate authentication API in the PingFederate documentation.

To integrate the PingOne Credentials IdP Adapter into your authentication flow, configure your application based on the information in this section.

> **Tip**
>
> You can also explore the process using the PingFederate Authentication API Explorer. Learn more in Exploring the Authentication API in the PingFederate documentation.

Models, attributes, and error codes

When using the PingOne Credentials IdP Adapter through the PingFederate authentication API, the adapter uses the following state models, action models, objects, and error codes.

Copyright © 2025 Ping Identity Corporation

State models

Status	Response Model	Action	Description
`CREDENTIAL_VERIFICATION_REQUIRED`	• `sessionID (string)` : The ID that can be used to check the status of the credential verification request. • `sessionStatus (SessionStatus)` : The status of the credential verification request. • `qUrl (string)` : The URL that can be used to get the QR code. Used in the different device SSO flow to open the wallet app. • `appOpenUrl (string)` : The URL that can be used to open the wallet app. • `redirect (boolean)` : Used in the same device SSO flow to open the wallet app after the user clicks Open Wallet. Uses the `appOpenUrl` . • `errorDetails (List<CustomAuthnErrorDetail>)` : A list of the errors that caused the credential verification request to fail.	• `poll` • `cancel` • `sameDevice`	Indicates that credential verification is required.
`CREDENTIAL_VERIFICATION_COMPLETED`	No model	• `continue`	Indicates that credential verification completed successfully.

Copyright © 2025 Ping Identity Corporation

Status	Response Model	Action	Description
`CREDENTIAL_VERIFICATION_FAILED`	• `sessionID (string)` : The ID that can be used to check the status of the credential verification request. • `sessionStatus (SessionStatus)` : The status of the credential verification request. • `errorDetails (List<CustomAuthnErrorDetail>)` : A list of the errors that caused the credential verification request to fail.	• `cancel`	Indicates that credential verification failed.
`CREDENTIAL_VERIFICATION_TIMED_OUT`	No model	• `retryVerification` • `cancel`	Indicates that credential verification timed out.

Action models

Action	Request Model	Errors	Description
`cancel`	No model	None	Cancels the verification flow.
`continue`	No model	None	Continues the verification flow.
`poll`	No model	None	Gets the current presentation session status.
`retryVerification`	No model	None	Retries verification after time out or failure.

Copyright © 2025 Ping Identity Corporation

Action	Request Model	Errors	Description
`sameDevice`	No model	None	Indicates that the user prefers to complete verification on the same device. Learn more in Overview of the SSO flow.

Troubleshooting

Enabling debug logging

To help with troubleshooting or monitoring, you can turn on activity logging for the adapter.

About this task

These steps are optional. For general information about logging, see Enabling debug messages and console logging in the PingFederate documentation.

Steps

1. Open the `<pf_install>/pingfederate/server/default/conf/log4j2.xml` file for editing.
2. If you want to log activity for PingFederate and all adapters, do the following.
    1. Find the following section.

```
<AsyncRoot level="INFO" includeLocation="false">
        <!-- <AppenderRef ref="CONSOLE" /> -->
        <AppenderRef ref="FILE" />
</AsyncRoot>
```

    2. Change `INFO` to `DEBUG`.

```
<AsyncRoot level="DEBUG" includeLocation="false">
        <!-- <AppenderRef ref="CONSOLE" /> -->
        <AppenderRef ref="FILE" />
</AsyncRoot>
```

    3. If you want to see the adapter activity in the console, remove the comment tags.

```
<AsyncRoot level="INFO" includeLocation="false">
        <AppenderRef ref="CONSOLE" />
         <AppenderRef ref="FILE" />
</AsyncRoot>
```

Copyright © 2025 Ping Identity Corporation

3. If you want to log activity just for the PingOne Credentials IdP Adapter, add the following line.

```
<Logger name="com.pingidentity.adapters.pingone.credentials" level="DEBUG"/>
```

4. Save the file.

Release notes

Changelog

The following is the change history for the PingOne Credentials Integration Kit.

PingOne Credentials Integration Kit 1.0 - August 2024

- Initial release.
- Added the ability to verify user credentials as part of a sign-on flow.
- Added the ability to control which message pages (templates) are shown to the user.
- Added the ability to customize template messages using a language pack file.
- Added settings to control how the adapter handles sign-on attempts when errors occur.
- Added support for the platform connection to PingOne introduced in PingFederate 10.2.
- Added support for the PingFederate authentication API.
- Added support for the JavaScript Widget for the PingFederate Authentication API.
- Added settings for API connection and request timeouts.

Known issues and limitations

The following are known issues or limitations for the PingOne Credentials Integration Kit.

Known issues

There are no known issues.

Known limitations

There are no known limitations.

Copyright © 2025 Ping Identity Corporation

Download manifest

The following files are included in the PingOne Credentials Integration Kit `.zip` archive:

- `ReadMeFirst.pdf` – contains links to this online documentation
- `Legal.pdf` – copyright and license information
- `dist/pingfederate/server/default` – contains the integration files
  - `deploy` – contains the Java libraries
    - `pf-pingone-credentials-adapter-<version>.jar` – JAR file that contains the PingOne Credentials IdP Adapter.
  - `conf` – contains the HTML template that presents the PingOne sign-on form.
    - `language-packs` – contains files with customizable user-facing messages
      - `pingone-credentials-messages.properties` – a variable file that customizes the messages that appear on the template file.
    - `template` – contains user-facing HTML template files
      - `pingone-credentials-completed.html` – a sample template that indicates the presentation session completed successfully.
      - `pingone-credentials-failed.html` – a sample template that indicates that the user failed the presentation session.
      - `pingone-credentials-required.html` – a sample template that asks the user to complete the presentation session.
      - `pingone-credentials-timed-out.html` – a sample template that indicates that the presentation session took too long.
    - `assets` – contains functional scripts and files used by the template
      - `css` – contains CSS files for the templates.
        - `pingone-credentials.css` – a CSS file that customizes the appearance of the template files.
        - `end-user/<version>/end-user.css` – a CSS file that customizes the appearance of the template files.
      - `fonts/end-user` – contains template fonts and icons
        - `icons` – contains template icons
        - `proxima-nova` – contains template fonts
      - `images` – contains template image files
        - `ping-logo.svg` – an image file with company branding

Copyright © 2025 Ping Identity Corporation

        - `spinner.svg` – an image file used in a spinner animation
    - `scripts` – contains script files used to collect and send information
        - `jquery-<version>.min.js` – a JavaScript file that contains the jQuery library.
- `lib/pf-authn-api-sdk-<version>.jar` – a JAR file that contains the PingFederate Authentication API SDK

Copyright © 2025 Ping Identity Corporation

RSA SecurID integrations

You can integrate PingFederate with RSA SecurID for multi-factor authentication (MFA) both in the cloud and on-premises. Both integrations support users with SecurID passcodes, hardware tokens, and software tokens.

RSA SecurID Cloud Authentication integration guide (cloud)

This guide describes how to configure PingFederate and the RSA SecurID Cloud Authentication Service for MFA.

Because this integration uses a standard SAML configuration, you don't need to download additional software.

RSA SecurID Integration Kit (on-premises)

The RSA SecurID Integration Kit allows PingFederate to use the RSA Authentication Manager service for MFA. This allows PingFederate to act as an RSA Authentication Agent for on-premises deployments of RSA SecurID.

RSA SecurID Cloud Authentication Integration Guide

This guide describes how to configure PingFederate and the RSA SecurID Cloud Authentication Service for multi-factor authentication (MFA). Because this integration uses a standard SAML configuration, you don't need to download additional software.

Features

- Supports MFA for users with RSA SecurID passcodes, hardware tokens, and software tokens

Intended audience

This document is intended for PingFederate administrators.

If you need help during the setup process, see the following resources:

- The following sections of the PingFederate documentation:
  - Identity provider SSO configuration
  - Managing IdP adapters
  - Authentication policies
- The following sections of the RSA SecurID documentation
  - RSA SecurID Access Overview
  - Ping Identity Corporation - Technology Integrations
  - Ping Identity PingFederate 10 - RSA SecurID Access Implementation Guide
  - Video: PingFederate SAML Integration user experience

Copyright © 2025 Ping Identity Corporation

System requirements

- PingFederate 11.3 or later
- RSA SecurID with the Cloud Authentication Service

Integrating PingFederate with RSA SecurID

RSA provides complete setup steps for both PingFederate and the RSA Cloud Authentication Service. After you finish the RSA instructions, you can customize the SAML assertion in PingFederate.

Steps

1. Follow the Ping Identity PingFederate 9.3 - RSA Ready SecurID Access Implementation Guide in the RSA documentation. Your integration type and use case determine which steps to follow.
2. (Optional) Modify the outgoing SAML assertion to suit your needs.
    1. Enabling and disabling expressions in the PingFederate documentation.
    2. Open the connection to RSA that you created while following the RSA implementation guide.
    3. Go to Browser SSO > Protocol Settings > Configure Protocol Settings. On the Assertion Consumer Service URL tab, click Show Advanced Customizations.
    4. In the Message Type list, select AuthnRequestDocument.
    5. In the Expression field, enter your custom OGNL expression.

       Learn more in Example OGNL expressions.

Example OGNL expressions

The following examples show common ways you might want to customize your outgoing SAML assertion.

Example 1: AuthnContextClassRef based on SAML_AUTHN_CTX

To send `AuthnContextClassRef` based on the `SAML_AUTHN_CTX` attribute from your PingFederate authentication policy, use the following expression:

```
#req = #AuthnRequestDocument.getAuthnRequest(),
#newctx = #ChainedAttributes.get("SAML_AUTHN_CTX"),
#newctx && (
#req.isSetRequestedAuthnContext() && #req.unsetRequestedAuthnContext(),
#ctx = #req.addNewRequestedAuthnContext(),
#ctx.addAuthnContextClassRef(#newctx.toString())
)
```

In this example, the value of the `SAML_AUTHN_CTX` attribute is "Password" and the expression sends the following:

Copyright © 2025 Ping Identity Corporation

```
<AuthnRequest Version="2.0" ID="gjmkj6OVk9tVhd1kvno63j92pqb" IssueInstant="2021-05-11T21:36:42.953Z"
xmlns="urn:oasis:names:tc:SAML:2.0:protocol" xmlns:urn="urn:oasis:names:tc:SAML:2.0:assertion">
 <urn:Issuer>localhost:default:entityId</urn:Issuer>
 <urn:Subject>
  <urn:NameID>rsa@demo.com</urn:NameID>
 </urn:Subject>
 <NameIDPolicy AllowCreate="true"/>
 <RequestedAuthnContext>
  <urn:AuthnContextClassRef>Password</urn:AuthnContextClassRef>
 </RequestedAuthnContext>
</AuthnRequest>
```

Example 2: AuthnContextClassRef based on SP entity ID

To send `AuthnContextClassRef` based on the service provider (SP) entity ID, use the following expression:

```
#Salesforce = "salesforceSPConnection", // this would be the target
#req = #AuthnRequestDocument.getAuthnRequest(),

#newctx = #Salesforce == #FedHubSpConnPartnerId ? "urn:rsa:names:tc:SAML:
2.0:ac:classes:spec:stepup:PingFed_Lev_Low": null,
#newctx && (
#req.isSetRequestedAuthnContext() && #req.unsetRequestedAuthnContext(),
#ctx = #req.addNewRequestedAuthnContext(),
#ctx.addAuthnContextClassRef(#newctx.toString())
)
```

In this example, the SP connection used is Salesforce. The value `urn:rsa:names:tc:SAML:2.0:ac:classes:spec:stepup:PingFed_Lev_Low` is sent as the AuthnContextClassRef. `#FedHubSpConnPartnerId` is the SP entity ID.

> **Tip**
>
> For other variables, see Message types and available variables in the PingFederate documentation.

The expression sends the following:

```
<AuthnRequest Version="2.0" ID="gjmkj6OVk9tVhd1kvno63j92pqb" IssueInstant="2021-05-11T21:36:42.953Z"
xmlns="urn:oasis:names:tc:SAML:2.0:protocol" xmlns:urn="urn:oasis:names:tc:SAML:2.0:assertion">
 <urn:Issuer>localhost:default:entityId</urn:Issuer>
 <urn:Subject>
  <urn:NameID>rsa@demo.com</urn:NameID>
 </urn:Subject>
 <NameIDPolicy AllowCreate="true"/>
 <RequestedAuthnContext>
  <urn:AuthnContextClassRef>urn:rsa:names:tc:SAML:2.0:ac:classes:spec:stepup:PingFed
  _Lev_Low</urn:AuthnContextClassRef>
 </RequestedAuthnContext>
</AuthnRequest>
```


Copyright © 2025 Ping Identity Corporation

RSA SecurID Integration Kit

The RSA SecurID Integration Kit allows PingFederate to use the RSA Authentication Manager service for multi-factor authentication (MFA). This allows PingFederate to act as an RSA Authentication Agent for on-premises deployments of RSA SecurID.

With the RSA SecurID Integration Kit, PingFederate can authenticate users that use SecurID passcodes, hardware tokens, and software tokens for multi-factor authentication (MFA).

> **Note**
>
> This integration is for the on-premises RSA Authentication Manager. To integrate with RSA Cloud Authentication, see the RSA SecurID Cloud Authentication Integration Guide.

Components

- RSA SecurID IdP Adapter
  - Allows PingFederate to communicate with the RSA Authentication API to process sign-on requests and other user activities supported by the templates.
- Templates
  - Allow the adapter to present information and forms to the user during the sign-on process, such as token challenges and PIN changes. You can customize the HTML portions of the templates to suit your organization's needs.
  - Allow you to modify the appearance of pages shown to the user during authentication.

Intended audience

This document is intended for PingFederate administrators.

If you need help during the setup process, see the following resources:

- The following sections of the PingFederate documentation:
  - Identity provider SSO configuration
  - Managing IdP adapters
  - Authentication policies
- The following sections of the RSA SecurID documentation
  - RSA SecurID Access Overview
  - RSA Authentication Manager in the RSA Authentication Manager documentation

System requirements

- PingFederate 11.3 or later

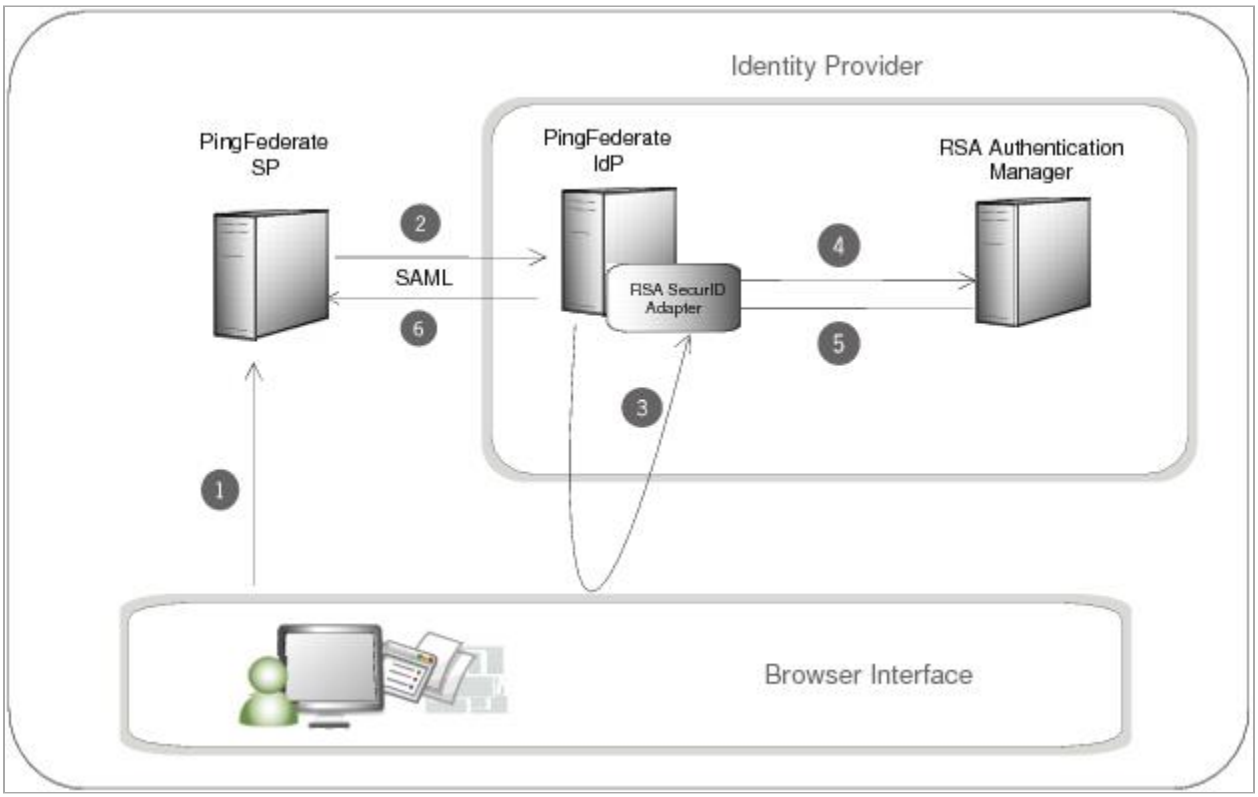
Copyright © 2025 Ping Identity Corporation

- RSA Authentication Manager 8.7 SP2 or later
- RSA SecurID authenticator hardware or software tokens

Overview of the SSO flow

With the RSA SecurID Integration Kit, PingFederate includes an RSA Authentication Manager server in the sign-on flow.

The following figure illustrates a single sign-on (SSO) scenario in which PingFederate authenticates users to an SP application using the RSA SecurID IdP Adapter.



Description

1. The user initiates SSO from an SP application through the PingFederate SP server.

   > **Note**
   >
   > This SP-initiated scenario represents the optimal use case, where both the identity provider (IdP) and SP use PingFederate. If your SP partner does not support this scenario, however, PingFederate accepts any valid SAML authentication request.
   > You can also enable IdP-initiated SSO. In this case, the SSO flow would not include this step or the next one.

2. The PingFederate SP server generates a SAML `AuthnRequest` and sends it to the PingFederate IdP server.
3. The PingFederate IdP server requests user authentication using the RSA SecurID Adapter. The adapter challenges the user for a RSA SecurID passcode.
4. The adapter sends authentication credentials to RSA Authentication Manager.

Copyright © 2025 Ping Identity Corporation

5. The RSA Authentication Manager validates the credentials sent by the adapter and sends a response to PingFederate.
6. If the validation fails, user access is denied. If validation succeeds, the PingFederate IdP server generates a SAML assertion with the username as the Subject and passes it to the PingFederate SP server.

Setup

Deploying the integration files

To get started with the integration, deploy the RSA SecurID Integration Kit files to your PingFederate directory.

Steps

1. Download the RSA SecurID Integration Kit `.zip` archive from the Ping Identity Integration Directory.
2. Stop PingFederate.
3. If you are upgrading an existing deployment, back up your customizations and delete older versions of the integration files.
    1. Back up any RSA SecurID Integration Kit files that you customized in `<pf_install>/pingfederate/server/default/conf/templates`:
        - `RSASecurIDIdPAdapter.form.template.html`
        - `RSASecurIDIdPAdapter.nexttoken.template.html`
        - `RSASecurIDIdPAdapter.reauthenticate.template.html`
        - `RSASecurIDIdPAdapter.systempinreset.template.html`
        - `RSASecurIDIdPAdapter.userpinreset.template.html`
        - `RSASecurIDIdPAdapter.approve.template.html`
        - `RSASecurIDIdPAdapter.cas.approve.template.html`
        - `RSASecurIDIdPAdapter.token.template.html`
    2. Delete the following from `<pf_install>/pingfederate/server/default/deploy`:
        - `pf-rsa-securid-idp-adapter-<version>.jar`
4. Extract the `.zip` archive and merge the contents of the `dist` directory with your `<pf_install>/pingfederate` directory.
5. If you have existing customizations, manually modify the new `rsa-securid-messages.properties` file based on your previous customizations.

   You can find more information on customizing messages in Localizing messages for end users in the PingFederate documentation.
6. If there is more than one version of the `pf-authn-api-sdk-<version>.jar` file in your `<pf_install>/pingfederate/server/default/lib` directory, delete all but the latest version of the file.

Copyright © 2025 Ping Identity Corporation

7. Start PingFederate.
8. If you operate PingFederate in a cluster, repeat steps 2 - 7 for each engine node.

Registering PingFederate as an agent in the RSA Security Console

To process sign-on requests, PingFederate requires access to the RSA Authentication API. You can grant access in the RSA Security Console.

About this task

You can find more information about registering authentication agents in Add an Authentication Agent in the RSA Authentication Manager documentation.

Steps

1. Sign on to the RSA Security Console.
2. Go to Access > Authentication Agents > Add new.
3. In the Authentication Agent Basics section, in the Hostname field, enter a unique name of your choosing, such as `MyPingFederate`. Click Save.
4. If your RSA Authentication Manager uses a self-signed certificate, add it as a trusted certificate authority (CA) in PingFederate.
    1. Export the certificate by following the steps in How to export RSA SecurID Access Authentication Manager Root Certificate in the RSA Authentication Manager documentation.
    2. In the PingFederate administrative console, on the Security > Trusted CAs screen, import the certificate. Click Save.

Configuring an adapter instance

Configure the RSA SecurID IdP Adapter to determine how PingFederate communicates with the RSA Authentication Manager server.

Steps

1. If you are upgrading from version 2.x of the integration kit, note the configuration details of your existing adapter instance, and then delete the adapter instance. Restore your configuration as you complete the steps below.
2. In the PingFederate administrative console, go to Authentication > Integration > IdP Adapters. Click Create new Instance.
3. On the Type tab, set the basic adapter instance attributes.
    1. In the Instance Name field, enter a name for the adapter instance.
    2. In the Instance ID field, enter a unique identifier for the adapter instance.
    3. From the Type list, select RSA SecurID IdP Adapter. Click Next.

Copyright © 2025 Ping Identity Corporation

4. (Optional) On the IdP Adapter screen, configure additional RSA Authentication Manager servers to try, in order, in case the primary server fails to respond.
    1. In the Failover Servers section, click Add a new row to 'Failover Servers'.
    2. In the RSA Base API URL field, enter the complete URL and endpoint of the RSA Authentication Manager server.
    3. If you want to skip hostname verification to prevent errors, clear the Verify HTTPS Hostname checkbox.
    4. In the Action column, click Update.
    5. To add another property, repeat steps a - d.
5. On the IdP Adapter tab, configure the adapter instance by referring to RSA SecurID IdP Adapter settings reference. Click Next.
6. On the Actions screen, test your connection to the RSA Authentication API. Resolve any issues that are reported and then click Next.
7. On the Extended Contract tab, add any attributes that you want to include in the contract. Click Next.
8. On the Adapter Attributes tab, set pseudonym and masking options as shown in Set pseudonym and masking options in the PingFederate documentation. Click Next.
9. On the Adapter Contract Mapping tab, configure the contract fulfillment details for the adapter as shown in Define the IdP adapter contract in the PingFederate documentation. Click Next.
10. On the Summary tab, check and save your configuration. Click Save.
11. Create or modify a connection to the service provider using the RSA SecurID IdP Adapter instance. Learn more in SP connection management in the PingFederate documentation.

RSA SecurID IdP Adapter settings reference

Field descriptions for the RSA SecurID IdP Adapter configuration screen.

Standard Fields

Field	Description
RSA Authentication Agent	The unique name that you entered in the Hostname field in Registering PingFederate as an agent in the RSA Security Console , such as `MyPingFederate`. PingFederate uses this to identify itself to the RSA Authentication Manager API. **Note** If you've integrated the Authentication Manager with the RSA Cloud Authentication Service (CAS), you can leave this field blank if you fill out the **Assurance Policy ID** field instead.

Copyright © 2025 Ping Identity Corporation

Field	Description
RSA Base API URL	The base URL of the primary RSA Authentication Manager including the hostname, port number and REST URL root path. For example: `https://RSA_Authentication_Manager_Hostname:REST_API_Port/mfa/v1_1`. The default REST API port is `5555`.
RSA Access ID	A unique string that the RSA Authentication Manager uses to identify individual REST API client. This is required if the security key type is HMAC.
RSA Access Key	A unique string that the RSA Authentication Manager generates and uses as a shared secret with REST API clients.

Advanced Fields

Field	Description
Use Custom Cipher Suites	Cipher suites are used to send information securely when the adapter makes TLS requests to RSA Authentication Manager. Cleared (default) – The adapter uses all cipher suites available to the adapter. You can find a complete list in Enum CipherSuites in the OkHttp documentation. Selected – Restricts the adapter to the cipher suites entered in the Custom Cipher Suites field. This allows your organization to use only cipher suites that meet your unique security standards. Select this if your environment has special requirements. This check box is cleared by default.
Custom Cipher Suites	The cipher suites that the adapter uses when Use Custom Cipher Suites is selected. Separate multiple ciphers with a comma. For example, `TLS_RSA_WITH_AES_128_GCM_SHA256, TLS_RSA_WITH_AES_256_GCM_SHA384, TLS_RSA_WITH_AES_128_CBC_SHA256`. You can find a complete list in Enum CipherSuites in the OkHttp documentation. This field is blank by default.
Challenge Retries	The number of failed user authentications after which the account locking service blocks future attempts. **Note** To enable this feature, follow the steps in Account lockout protection in the PingFederate documentation. The default value is `5`.

Copyright © 2025 Ping Identity Corporation

Field	Description
Security Key Type	The method of security key authentication to use against the RSA Authentication REST API. If Access Key is enabled, the plain key will be used. If HMAC is enabled an HMAC calculated from the Access Key, a hash of the request body, the Access ID, and other request-specific information will be used.
Assurance Policy ID	The access policy name that's configured in the Cloud Administration Console. You can get the access policy name from your Cloud Authentication Service Super Admin. **Note** If you're using the RSA Authentication Manager integrated with the Cloud Authentication Service, you must enter a valid value for either the **Assurance Policy ID** field or the **RSA Authentication Agent** field.
Logout Path	Path on the PingFederate server to end a user's IdP session. Must include the initial slash. For example, `/mylogoutpath`. The value is added to the following to create the logout URL: https://*pf_host*:*port*/ext This field is blank by default.
Logout Redirect	The URL that the adapter redirects the user to after they log out. Applies only when Logout Path is set above. When provided, this URL takes precedence over any Logout Template specified below. This field is blank by default.
Logout Template	HTML template to render after the user logs out. Applies when Logout Path is set above and Logout Redirect is blank. The template file must be located in `<pf_home>/server/default/conf/template`. The default value is: `idp.logout.success.page.template.html`
Authentication Context Value	Additional information provided to the SP to assess the level of confidence in the assertion. This value will override the default authentication context used by the adapter. This field is blank by default.
Verify HTTPS Hostname	When a connection is established with RSA Authentication Manager, PingFederate matches the target host name against the names stored inside the server's X.509 certificate. This security measure ensures that PingFederate is connecting to the correct server. This check box is selected by default.

Copyright © 2025 Ping Identity Corporation

Field	Description
Override Internal User ID	Allows you to specify a custom user identifier attribute for authentication with RSA SecurID. By default, the adapter takes the `username` attribute from the PingFederate authentication policy and uses it for both frontend display and backend authentication with RSA Authentication Manager. To use a different attribute for backend authentication, select this check box and enter the custom attribute in the Internal User ID Attribute field. ***Not selected (default)*** Frontend display: `username` Backend authentication: `username` ***Selected*** Frontend display: `username` Backend authentication: *<custom attribute>* **Note** In either case, if no `username` attribute is available, the user-facing template shows an error. This check box is cleared by default.
Internal User ID Attribute	When Override Internal User ID is selected, this field determines the user identifer attribute used to authenticate the user with RSA Authentication Manager. The attribute must be available in the PingFederate authentication policy. The attribute name is case sensitive. This field is blank by default.
Test Username	The username that's used to test the configuration on the Actions tab.
HTML Template Prefix	A file prefix that identifies the customizable HTML templates that the adapter instance uses. The template files must be located in `<pf_home>/server/default/conf/template`. **Important** If you customize the template file names in `/server/default/conf/template`, make sure to use the same prefix consistently and enter that new prefix in this field. The default value is: `RSASecurIDIdPAdapter.`
Messages Files	Identifies the customizable language-pack file that the adapter uses. If you customize the `rsa-securid-messages.properties` file name in the `/server/default/conf/language-packs` directory, enter the new name here. The default value is `rsa-securid-messages`.
Error Message Key Prefix	Prefix for error messages in the language pack. The default value is `rsa.securid.error`.

Copyright © 2025 Ping Identity Corporation

Field	Description
API Request Timeout	The amount of time in milliseconds that PingFederate allows when establishing a connection with RSA Authentication Manager or waiting for a response to a request. A value of 0 disables the timeout. The default value is `5000`.
Proxy Settings	Defines proxy settings for outbound HTTP requests. The default value is System Defaults.
Custom Proxy Host	The proxy server host name to use when Proxy Settings is set to Custom. This field is blank by default.
Custom Proxy Port	The proxy server port to use when Proxy Settings is set to Custom. This field is blank by default.

Customizing templates

The RSA SecurID Integration Kit includes templates that can display RSA SecurID authentication challenges and other messages. You can customize the HTML portions of these templates to suit your organization's branding and presentation needs.

The following table lists the templates and their use. For more information about templates, see Customizable user-facing pages in the PingFederate documentation.

Template files included in the RSA SecurID Integration Kit

Template file	Description
`RSASecurIDIdPAdapter.nexttoken.template.html`	Form presented when the user is required to enter the next token.
`RSASecurIDIdPAdapter.reauthenticate.template.html`	Login form asking the user to authenticate again after resetting their PIN.
`RSASecurIDIdPAdapter.systempinreset.template.html`	Presents the user with a new, system-generated PIN.
`RSASecurIDIdPAdapter.userpinreset.template.html`	Presents the user with a form to input a new PIN.
`RSASecurIDIdPAdapter.form.template.html`	Main login form, presented under normal conditions.
`RSASecurIDIdPAdapter.cas.approve.template.html`	Presents the user with the Cloud Authentication Service options that are available to them for authentication.
`RSASecurIDIdPAdapter.approve.template.html`	Presents the user with a wait page while waiting for a push notification to complete on the user device.

Copyright © 2025 Ping Identity Corporation

Template file	Description
`RSASecurIDIdPAdapter.token.template.html`	Presents the user with the form for the `TOKEN` authentication method for the Cloud Authentication Service.

Authentication API Support

You can use the PingFederate authentication API to integrate the RSA SecurID IdP Adapter into your application.

The PingFederate Authentication API provides access to the current state of the authentication flow as a user steps through the PingFederate authentication policy. Learn more in PingFederate authentication API in the PingFederate documentation.

To integrate the RSA SecurID IdP Adapter into your authentication flow, configure your application based on the information in this section.

> **Tip**
>
> You can also explore the process using the PingFederate Authentication API Explorer. Learn more in Exploring the Authentication API in the PingFederate documentation.

Troubleshooting

Enabling debug logging

To help with troubleshooting or monitoring, you can turn on activity logging for the adapter.

About this task

These steps are optional. For general information about logging, see Enabling debug messages and console logging in the PingFederate documentation.

Steps

1. Open the `<pf_install>/pingfederate/server/default/conf/log4j2.xml` file for editing.
2. If you want to log activity for PingFederate and all adapters, do the following.
    1. Find the following section.

```
<AsyncRoot level="INFO" includeLocation="false">
        <!-- <AppenderRef ref="CONSOLE" /> -->
        <AppenderRef ref="FILE" />
</AsyncRoot>
```

    2. Change `INFO` to `DEBUG`.

Copyright © 2025 Ping Identity Corporation

```
<AsyncRoot level="DEBUG" includeLocation="false">
        <!-- <AppenderRef ref="CONSOLE" /> -->
        <AppenderRef ref="FILE" />
</AsyncRoot>
```

3. If you want to see the adapter activity in the console, remove the comment tags.

```
<AsyncRoot level="INFO" includeLocation="false">
        <AppenderRef ref="CONSOLE" />
        <AppenderRef ref="FILE" />
</AsyncRoot>
```

3. If you want to log activity just for the RSA SecurID IdP Adapter, add the following line.

```
<Logger name="com.pingidentity.adapters.securid.idp" level="DEBUG"/>
```

4. Save the file.

Release notes

Changelog

The following is the change history for the RSA SecurID Integration Kit.

RSA SecurID Integration Kit 4.1.1 – December 2024

- Fixed an issue that caused the `Approve` method's push notification template to render twice.
- Fixed an issue that caused the RSA SecurID IdP Adapter to fail when using PingFederate in BC FIPS mode.

RSA SecurID Integration Kit 4.1 – April 2024

- Added support for the `Approve` method's `Selection` mode. Support for the `Approve` method was introduced in RSA SecurID Integration Kit 4.0.

> **Note**
>
> To use `Selection` mode, you must have RSA Authentication Manager 8.7 SP2 or later, integrated with the RSA Cloud Authentication Service (CAS). You must enable the `Selection` feature in RSA CAS, and use version 4.3 or later of the RSA mobile app.

- Added the ability to define a template file prefix and customize the pages that the adapter displays per adapter instance.

Copyright © 2025 Ping Identity Corporation

> **Note**
>
> If you modify a template file, rename the template file to give it a new prefix. Make sure to enter the new prefix in the HTML Template Prefix field.

RSA SecurID Integration Kit 4.0.1 – February 2024

- Fixed an issue that caused an unavailable authentication method to display incorrectly.

RSA SecurID Integration Kit 4.0 – October 2023

- Added support for the RSA Authentication Manager integrated with the RSA Cloud Authentication Service (CAS). The adapter supports the `SecurID`, `Token`, and `Approve` methods.

RSA SecurID Integration Kit 3.2.2 – December 2022

- Updated the security dependencies used by the adapter to the latest available.

RSA SecurID Integration Kit 3.2.1 – September 2021

- Fixed two issues that caused certain authentication attempts to be logged incorrectly in `audit.log` and `server.log`.

RSA SecurID Integration Kit 3.2 – June 2021

- Added support for the PingFederate authentication API.
- Added support for the JavaScript Widget for the PingFederate Authentication API.
- Improved the template files to use the PingFederate localization framework. If you have existing customizations, you need to manually modify the new `rsa-securid-messages.properties` file. For help customizing messages, see Localizing messages for end users in the PingFederate documentation.
- Fixed an issue that, after upgrading the adapter, caused an error when using the administrative API to bulk import an earlier version of the adapter.
- Fixed an issue where the previous adapter wouldn't failover to its replicas when the virtual machine was running but RSA services was not.
- Fixed a user impersonation vulnerability. See security bulletin SECADV026.

RSA SecurID Integration Kit 3.1.1 – March 2021

- Fixed an issue that caused unexpected behavior when `LockoutPeriod` was set to `0` in the account lockout settings file.
- Improved the description for the Challenge Retries setting in the adapter configuration.

RSA SecurID Integration Kit 3.1 – November 2020

- Added the ability to customize the ciphers used in outbound HTTP requests to RSA SecurID.
- Improved error handling for cases where the user leaves the username or passcode field blank.

Copyright © 2025 Ping Identity Corporation

- Fixed an issue that could cause the adapter to cancel some sign-on attempts.

RSA SecurID Integration Kit 3.0.1 – October 2019

- Fixed a serialization issue that occurred when PingFederate was used in a cluster.

RSA SecurID Integration Kit 3.0 – August 2019

- Added support for RSA Authentication Manager 8.4
- Added support for Java 11 by updating the adapter to use the RSA SecurID Authentication API.
- Added the ability to override the user ID attribute that is sent to the authentication API without affecting the ID shown to the user.
- Added the ability to configure failover servers for the primary RSA Authentication API endpoint.
- Added proxy connection override settings.
- Added connection read and timeout settings.
- Improved templates to support multi-factor authentication and CSRF protection.
- Improved template usability and error messages.

RSA SecurID Integration Kit 2.1 – June 2018

- Added support for RSA Authentication Manager 8.3
- Improved variable names in the template pages
- Added new variables available to the template pages
- Added configuration field to set the log level of the RSA Authentication Agent SDK
- Added validation for user-generated PINs in the adapter and template

RSA SecurID Integration Kit 2.0 – May 2017

- Added support for RSA Authentication Manager 8.2
- Removed support for versions 6.x and 7.x of RSA Authentication Manager
- Updated the SecurID Adapter to use RSA Authentication Agent SDK 8.6 for Java

RSA SecurID Integration Kit 1.2.2 – January 2017

- Resolved issue when configuring RSA in a clustered Windows environment

RSA SecurID Integration Kit 1.2.1 – December 2015

- Security fixes
- Fixed serialization issue when token is in new PIN mode

Copyright © 2025 Ping Identity Corporation

RSA SecurID Integration Kit 1.2 – September 2015

- Qualified for RSA Authentication Manager 7.1 SP4
- Updated SecurID adapter to be utilized as second factor authentication
- Added session state management

RSA SecurID Integration Kit 1.1 – November 2013

- RSA Authentication Agent API 8.1 SP2

RSA SecurID Integration Kit 1.0 – March 2010

- Initial Release

Known issues and limitations

The following are known issues or limitations for the RSA SecurID Integration Kit.

Known issues

There are no known issues.

Known limitations

- The RSA SecurID Integration Kit is designed to work with the on-premises RSA Authentication Manager (AM).

  > **Note**
  >
  > This integration kit works with the RSA AM when the AM is integrated with the RSA Cloud Authentication Service (CAS), but does not work directly with the RSA CAS.

- When a push notification request that's triggered by the RSA CAS's `Approve` method times out on a registered device, the adapter automatically fails the authentication method.
- Only RSA AM 8.7 SP2 and later integrated with CAS supports the `Approve` method's `Selection` mode.
- The adapter requires you to set a specific JVM option if you're using PingFederate deployed in JDK 17 in BC FIPS mode. Update the `JAVA_OPTS` to set `--add-opens=java.base/sun.security.internal.spec=ALL-UNNAMED` in the `run.sh` and `run.bat` files.

  For example, in `<pf_install>/pingfederate/bin/run.sh`, add:

Copyright © 2025 Ping Identity Corporation

```
# java 17 support
if [[ $JAVA_MAJOR_VERSION -eq "17" ]]; then
JAVA_OPTS="$JAVA_OPTS --add-opens=java.base/java.lang=ALL-UNNAMED"
JAVA_OPTS="$JAVA_OPTS --add-opens=java.base/java.util=ALL-UNNAMED"
JAVA_OPTS="$JAVA_OPTS --add-opens=java.base/sun.security.internal.spec=ALL-UNNAMED"
JAVA_OPTS="$JAVA_OPTS --add-exports=java.base/sun.security.x509=ALL-UNNAMED"
JAVA_OPTS="$JAVA_OPTS --add-exports=java.base/sun.security.util=ALL-UNNAMED"
JAVA_OPTS="$JAVA_OPTS --add-exports=java.naming/com.sun.jndi.ldap=ALL-UNNAMED"
JAVA_OPTS="$JAVA_OPTS --add-exports=java.base/sun.net.util=ALL-UNNAMED"
JAVA_OPTS="$JAVA_OPTS --add-exports=java.base/sun.security.pkcs=ALL-UNNAMED"
JAVA_OPTS="$JAVA_OPTS --add-exports=java.base/sun.security.pkcs10=ALL-UNNAMED"
fi
```

In `<pf_install>/pingfederate/bin/run.bat`, add:

```
if %JAVA_PRODUCT_VERSION% == 17 (
set "PF_ENDORSED_DIRS_FLAG="
set JAVA_OPTS=%JAVA_OPTS% ^
--add-opens=java.base/java.lang=ALL-UNNAMED ^
--add-opens=java.base/java.util=ALL-UNNAMED ^
--add-opens=java.base/sun.security.internal.spec=ALL-UNNAMED ^
--add-exports=java.naming/com.sun.jndi.ldap=ALL-UNNAMED ^
--add-exports=java.base/sun.net.util=ALL-UNNAMED ^
--add-exports=java.base/sun.security.pkcs=ALL-UNNAMED ^
--add-exports=java.base/sun.security.pkcs10=ALL-UNNAMED ^
--add-exports=java.base/sun.security.x509=ALL-UNNAMED ^
--add-exports=java.base/sun.security.util=ALL-UNNAMED
)
```

Download manifest

The following files are included in the RSA SecurID Integration Kit `.zip` archive:

- `Legal.pdf`: Copyright and license information
- `dist/pingfederate/server/default`: Contains the integration files
  - `deploy`: Contains the Java libraries
    - `pf-rsa-securid-idp-adapter-<version>.jar` – JAR file that contains the RSA SecurID IdP Adapter.
  - `conf` – contains the HTML template that presents the RSA SecurID sign-on form.
    - `template`: Contains user-facing HTML template files
      - `RSASecurIDIdPAdapter.form.template.html` – main login form, presented under normal conditions.
      - `RSASecurIDIdPAdapter.nexttoken.template.html` – form presented when the user is required to enter the next token.

Copyright © 2025 Ping Identity Corporation

- `RSASecurIDIdPAdapter.reauthenticate.template.html` – login form asking the user to authenticate again after resetting their PIN.
- `RSASecurIDIdPAdapter.systempinreset.template.html` – presents the user with a new, system-generated PIN.
- `RSASecurIDIdPAdapter.userpinreset.template.html` – presents the user with a form to input a new PIN.
- `RSASecurIDIdPAdapter.cas.approve.template.html` - presents the user with the Cloud Authentication Service options that are available to them for authentication.
- `RSASecurIDIdPAdapter.approve.template.html` - presents the user with a wait page while waiting for a push notification to complete on the user device.
- `RSASecurIDIdPAdapter.token.template.html` - presents the user with the form for the `TOKEN` authentication method for the Cloud Authentication Service.

- `assets`: Contains functional scripts and files used by the template
  - `css` – contains CSS files for the templates.
    - `rsa-securid.css` – a CSS file that customizes the appearance of the template files.
    - `end-user/<version>/end-user.css` – a CSS file that customizes the appearance of the template files.
  - `fonts/end-user/icons`: Contains template icons

- `lib/pf-authn-api-sdk-<version>.jar`: A `.jar` file that contains the PingFederate Authentication API SDK

Copyright © 2025 Ping Identity Corporation

Overview of Salesforce integrations

Salesforce Cloud Identity Connector

The Salesforce Cloud Identity Connector allows PingFederate to use Salesforce as an identity provider (IdP). This allows users to access service provider (SP) applications by signing on with Salesforce.

Salesforce Connector

The PingFederate Salesforce Connector extends PingFederate capabilities to allow enterprises to provide secure single sign-on (SSO) with user provisioning to Salesforce Customer Relations Management (CRM) and Chatter platforms.

Salesforce Contacts Connector

The Salesforce Contacts Connector allows PingFederate to provision "lead" and "contact" records to Salesforce Contacts based on changes in a datastore attached to PingFederate.

Salesforce Login Integration Kit

The Salesforce Login Integration Kit allows PingFederate to use Salesforce as an identity provider (IdP). This allows users to access service provider (SP) applications by signing on with Salesforce.

This allows users who have logged on to any enterprise Salesforce service to perform single sign-on (SSO) to SaaS applications.

Intended audience

This document is intended for PingFederate administrators.

You can get help during the setup process by referencing the following resources:

- PingFederate documentation:
  - Identity provider SSO configuration
  - Managing IdP adapters
  - Authentication policies
  - SP connection management

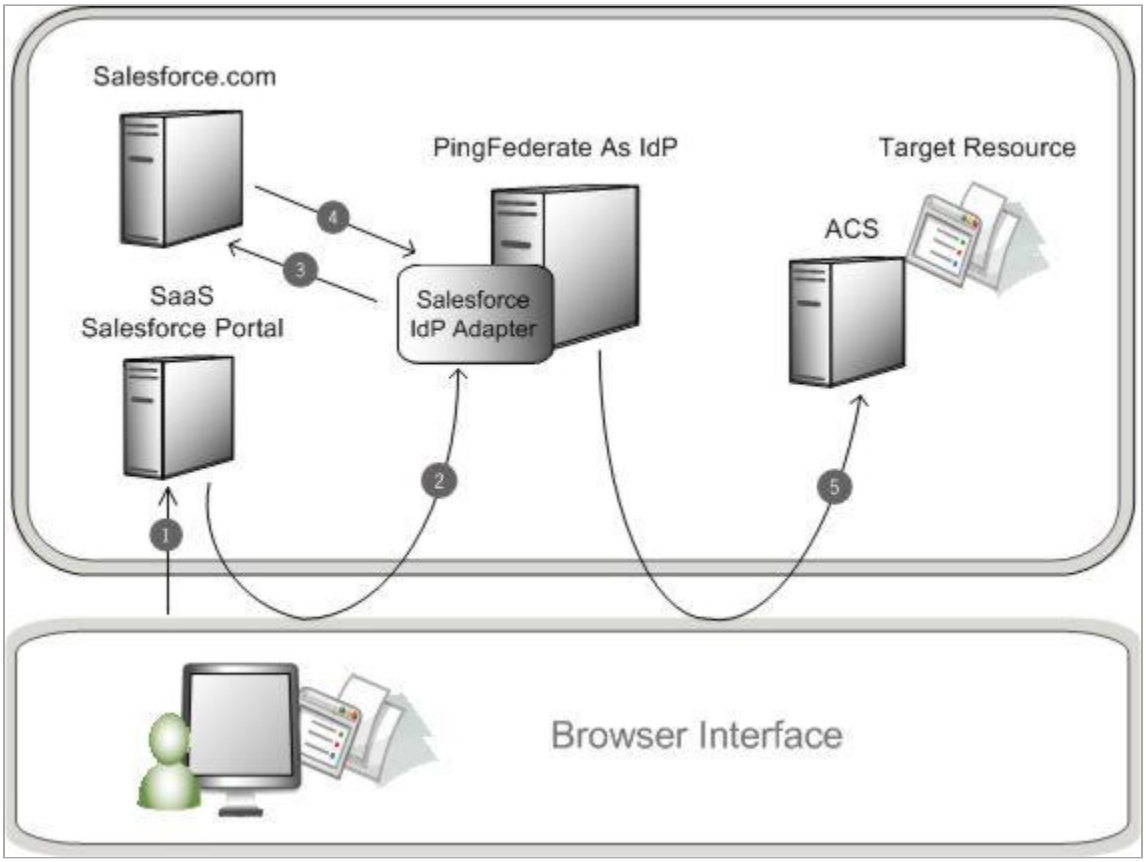
System requirements

- PingFederate 6.2 or later

Copyright © 2025 Ping Identity Corporation

Overview of the SSO flow

The following diagram illustrates the single sign-on (SSO) processing flow using the Salesforce Cloud Identity Connector in a SaaS environment as an example implementation:



Processing Steps

1. On the enterprise Salesforce site, a user clicks a custom link for access to a protected resource.

   > **Important**
   >
   > The user must be signed on to Salesforce.

2. The link goes to PingFederate and includes the user's Salesforce session ID and service URL as query parameters. Learn more in Define the SSO URL in Salesforce.
3. The Salesforce IdP Adapter makes a SOAP (Simple Object Access Protocol) request to Salesforce to obtain attributes for the user.
4. Salesforce validates the session and returns requested user attributes in the SOAP response.
5. PingFederate issues a SAML (Security Assertion Markup Language) assertion to the SP-connection Assertion Consumer Service (ACS).

Copyright © 2025 Ping Identity Corporation

> **Note**
>
> Alternatively, for onsite target resources within the same security context as PingFederate, SSO can be accomplished through adapter-to-adapter mapping without using a SAML connection:
>
> - For an external SP partner, configure an SP connection using the instructions under SSO to an SP partner.
> - For SSO to an application at your site in the same security domain, a standard SAML connection is not necessary. You can use direct IdP-to-SP adapter mapping using the instructions under SSO to an onsite application.

6. (Not shown) The user is logged on to the target resource.

Setup

Install the Salesforce connector

Steps

1. In the distribution `dist` directory, copy the following files:
   - `pf-salesforce-idp-adapter-1.0.jar`
   - `salesforce-partner-api-20.jar`

   into the directory:

   ```
   <pf_install>/pingfederate/server/default/deploy
   ```

2. If it exists, delete the file:

   `salesforce-partner.jar`

   from the directory:

   ```
   <pf_install>/pingfederate/server/default/lib
   ```

   This library is an older version of the Salesforce API and is not compatible with the PingFederate Salesforce Connector.

3. Start or restart PingFederate.

Configure the IdP adapter

About this task

Copyright © 2025 Ping Identity Corporation

> **Tip**
>
> For this configuration, you need to know your Salesforce.com Organization ID. You can use more than one ID as needed. Organization IDs are listed under **Company Information** in your Salesforce Administration Setup.

Steps

1. Sign on to the PingFederate administrative console and click Adapters under My IdP Configuration on the Main Menu.
2. On the Manage IdP Adapter Instances page, click Create New Instance.
3. On the Type page, enter an Instance Name and Instance ID.

   The Instance Name is any name you choose for identifying this adapter instance. The Instance ID is used internally and must be alphanumeric without any spaces.

   > **Tip**
   >
   > Make a note of the adapter ID for later use.

4. Select Salesforce.com Adapter from the Type list and click Next.
5. On the IdP Adapter page, under Allowed Organization(s):
   1. Click Add a new row to 'Allowed Organization(s)'.
   2. Enter your Salesforce Org ID.
   3. Click Update.
   4. Repeat these steps for any other Salesforce IDs at your site as needed.
6. (Optional) If SSO to a target application requires the user's organizational role and profile, select the associated checkbox.

   Note that this selection marginally increases processing time for the SSO transaction.
7. Click Next.
8. On the Adapter Attributes page, select the checkbox next to subject under Pseudonym.

   Pseudonyms are opaque subject identifiers used for SAML account linking and might not be applicable in the context of cloud-identity deployments. To ensure correct PingFederate performance under all circumstances, however, a selection is required. Learn more about account linking in the PingFederate Administrator's Manual or click Help on this page.
9. On the Summary page, verify that the information is correct and click Done.
10. On the Manage IdP Adapter Instances page, click Save.

SSO to an SP partner

Steps

- Use the Salesforce IdP Adapter Instance (configured earlier) in an SP Connection.

Copyright © 2025 Ping Identity Corporation

Result

You select the Adapter Instance during the IdP Adapter Mapping setup under Assertion Creation.

Learn more in the PingFederate Administrator's Manual.

> **Note**
>
> Ensure you activate the connection on the **Activation & Summary** page when you are ready to test the connector deployment.

SSO to an onsite application

Steps

1. **On the Main Menu, click Server Settings.**
2. **On the Roles and Protocols page in the Server Settings configuration, ensure that both the IdP and SP roles are enabled.**

   > **Note**
   >
   > The choice of protocol is not relevant for either role to implement the Salesforce Connector for onsite single sign-on (SSO), but a selection is required to enable a role.

   > **Note**
   >
   > If updates are needed on the page, make sure you click **Save**.

3. **Configure an SP Adapter Instance, if one is not already configured or if you want to use a new one.**

   Click Adapters under SP Configuration on the Main Menu.

   Use any adapter type, including the OpenToken Adapter bundled with PingFederate. Learn more in the PingFederate Administrator's Manual.

4. **On the Main Menu under System Settings, click IdP-to-SP Adapter Mapping and follow the page flow to complete this configuration.**

   Select the Salesforce IdP Adapter Instance configured earlier as the source instance and any SP Adapter Instance as the target.

   Learn more in the PingFederate Administrator's Manual or click Help on any screen.

5. **(Optional) On the Main Menu under SP Configuration, click Default URLs.**

   If the default SSO URL in the top text box is unspecified and the SP configuration will be used only to set up this Salesforce Connector, you can enter the target-application URL as the default.

   > **Note**
   >
   > The default URL for single logout (SLO) in the second text box does not apply for the Salesforce Connector. SLO is not supported.

Copyright © 2025 Ping Identity Corporation

Alternatively, you can enter a fallback URL or leave an existing entry unchanged, and provide the target-application URL as a query parameter in the Salesforce link. This recommended process is described in the next section.

Learn more about how default URLs are used in the PingFederate Administrator's Manual or click Help.

Define the SSO URL in Salesforce

In your Salesforce administrative web interface, add a custom link to initiate single sign-on (SSO) where needed for your Salesforce deployment. The content of the link depends on the PingFederate configuration.

> **Tip**
>
> Salesforce navigational details and identification of pages and selections in these steps are subject to change. Only the configuration options directly relevant to the PingFederate Connector are described. Some configuration steps are summarized and different configurations may be possible. Ensure you consult the Salesforce documentation for complete and up-to-date information as needed.

Add an SSO link in Salesforce (example)

Create a dynamic single sign-on (SSO) link and add it to your Salesforce page.

Steps

1. In the Salesforce administrative console, go to Setup > Customize > Home > Custom Links.
2. On the Home page, in the Custom Links section, click New.
3. Complete the Label, Name, and Description fields.
4. In the Content Source list, select URL.
5. In the text area, enter one of the following URLs:

   > **Tip**
   >
   > You can insert `{!API.Session_ID}` and `{!API.Partner_Server_URL_200}` or equivalent by selecting **$Api** from the **Select Field Type** list and then selecting the variable you want from the **Insert Field** list. Salesforce dynamically populates these variables when the user clicks the link. Learn more about Global Variables in the Salesforce documentation.

   Choose from:

   - For SSO through an SP-partner connection in PingFederate:

     ```
     https://pf_host:pf_port/idp/startSSO.ping?
     sid={!API.Session_ID}&apiendpoint={!API.Partner_Server_URL_200}&
     TargetResource=destination_URL&
     PartnerSpId=connection_ID
     ```

Copyright © 2025 Ping Identity Corporation

Replace the variables as follows:

- *pf_host*: The host name or IP address of the PingFederate server.
- *pf_port*: The port number of the PingFederate server.
- *destination_URL*: The fully-qualified URL of the target application or other protected resource.
- *connection_ID*: The Partner's Entity ID (Connection ID) of the service provider (SP) connection that you configured.

  Example URL:

```
https://pingfederate.example.com:9031/idp/startSSO.ping?sid={!API.Session_ID}&apiendpoint={!
API.Partner_Server_URL_200}&TargetResource=https://portal.example.com/
welcome&PartnerSpId=SFConnection1
```

- For SSO through IdP-to-SP adapter mapping in PingFederate:

```
https://pf_host:pf_port/pf/adapter2adapter.ping?
sid={!API.Session_ID}&apiendpoint={!API.Partner_Server_URL_200}&
TargetResource=destination_URL&
IdpAdapterId=adapter_ID
```

Replace the variables as follows:

- *pf_host*: The host name or IP address of the PingFederate server.
- *pf_port*: The port number of the PingFederate server.
- *destination_URL*: The fully-qualified URL of the target application or other protected resource.
- *adapter_ID*: The Instance ID of the Salesforce IdP Adapter that you configured.

  Example URL:

```
https://pingfederate.example.com:9031/pf/adapter2adapter.ping?sid={!API.Session_ID}
&apiendpoint={!API.Partner_Server_URL_200}&TargetResource=https://portal.example.com/
welcome&IdpAdapterId=SFAdapter1
```

6. Click Save.
7. Go to Setup > Customize > Home Page Components.
8. Add a new link component using the custom link that you created.
9. Go to Setup > Customize > Home Page Layout.
10. Edit your existing layout or create a new layout to include the new link component.
11. Check that the user profile is configured to use the page layout that you modified.

Copyright © 2025 Ping Identity Corporation

Release notes

Changelog

Salesforce Cloud Identity Connector 1.0 – January 2011

Initial Release

Known issues and limitations

Known Issues, Problems, and Limitations

- A customer portal user cannot look up a role name. All other attributes for a customer portal user can be looked up.
- Profile name cannot be looked up for customer and partner portal users.
- Adapter-to-adapter mapping with this adapter is supported only with PingFederate versions 6.2 and later.

Exclusions

- Due to a limitation with PingFederate 8.1 and earlier versions, when configuring two SP connections with the same provisioner, the second connection built may be pre-populated with the channel from the first connection. To avoid conflicts, delete this pre-populated channel and create a unique channel for each connection.
- The adapter was not tested on 64-bit systems.
- For clustered testing, a software load balancer was used for testing, instead of a hardware load balancer.

Download manifest

The distribution `.zip` archive for the connector contains the following:

- `/docs` : Contains this documentation:
  - `Salesforce_Cloud_Identity_Connector_Qualification_Statement.pdf` : Testing and platform information
  - `Salesforce_Cloud_Identity_Connector_User_Guide.pdf` : This document
- `/dist` : Contains libraries needed for the adapter:
  - `pf-salesforce-idp-adapter-1.0.jar` : Salesforce Adapter `.jar` file
  - `salesforce-partner-api-20.jar` : Salesforce application programming interface (API)

Salesforce Provisioner

The Salesforce Provisioner allows PingFederate to integrate with Salesforce customer relations management (CRM) products, such as Sales, Communities, and Chatter. The integration enables single sign-on (SSO) and allows PingFederate to provision users and groups to Salesforce.

Copyright © 2025 Ping Identity Corporation

Provisioning connector

Allows PingFederate to manage users and groups in Salesforce based on changes in an external user datastore. Includes a quick-connection template that helps you create a connection to Salesforce by pre-populating some configuration settings.

The provisioner allows you to enable the create, update, and disable capabilities independently and customize other provisioning behaviors.

Features

- Manages users
  - Creates, updates, and disables users.
  - Allows you to enable the create, update, and disable capabilities independently.
  - Allows you to provision users with a "disabled" status.
  - Allows you to deprovision users with a "freeze" or "disable" action.
  - Allows you to merge or overwrite user permission sets.
- Manages groups
  - Creates and deletes groups.
  - Updates group memberships.
- Enables browser-based SSO initiated by the service provider (SP) or identity provider (IdP).
- Pre-populates some connection settings with the included quick connection template.

Intended audience

This document is intended for PingFederate administrators.

You can get help during the setup process by referencing the following resources:

- The following sections of the PingFederate documentation:
  - Datastores
  - SP connection management
  - Configuring outbound provisioning
  - Configuring outbound provisioning settings

System requirements

- PingFederate `8.0` or later.
- Administrator access to a Salesforce organization account. The account must have "View All Data" permissions. Learn more about account permissions in Enable OAuth Settings for API Integration in the Salesforce documentation.

Copyright © 2025 Ping Identity Corporation

- To provision to Salesforce Communities, the communities feature must be enabled in Salesforce.
- To allow PingFederate to make outbound connections to Salesforce, you might need to allow the following endpoint in your firewall: https://*your_salesforce_domain*.com.

User and group management

The Salesforce Connector links users and groups from the datastore to Salesforce. The behavior of each provisioning capability is described below.

Synchronizing existing users

The provisioning connector synchronizes users from the datastore to Salesforce based on the `Username` attribute.

To set up synchronization, use the SP Connection > Configure Channels > Channel > Attribute Mapping page to populate the `Username` attribute with a matching attribute from the datastore.

For example:

- In Salesforce, Janet's `Username` is `jsmith@example.com`.
- In your datastore, Janet's `mail` is `jsmith@example.com`.
- On the Attribute Mapping page, you map the `Username` attribute to `mail`.
- When the provisioning connector runs, the datastore user is provisioned with a `Username` of `jsmith@example.com`. That matches Janet's existing `Username` in Salesforce, so her information in the datastore is synchronized to her Salesforce account.

User provisioning

Triggered by any of the following:

- A user is added to the datastore group or filter that is targeted by the provisioning connector.
- A user with "disabled" status is added to the datastore group or filter that is targeted by the provisioning connector, and the Provision disabled users provisioning option is enabled.

The target is determined by the Source Location page in the provisioning connector configuration.

User updates

Triggered when a change occurs to a user attribute that is mapped in the provisioning connector configuration.

User deprovisioning

Triggered by any of the following:

- A user is deleted from the user store.
- A user is disabled in the user store.
- A user is removed from the datastore group or filter that is targeted by the provisioning connector.

Copyright © 2025 Ping Identity Corporation

The provisioning connector disables or freezes the user depending on whether Freeze users instead of disable is selected. Learn more in Provisioning options reference.

Synchronize existing groups

The provisioning connector synchronizes groups from the datastore to Salesforce based on the group name.

For example:

- In Salesforce, there is a group named `Accounting`.
- In your datastore, there is a group with a `CN` of `Accounting`.
- When the provisioning connector runs, the two groups are synchronized.

Group provisioning

Triggered when a group is added to the datastore filter that is targeted by the provisioning connector.

The target is determined by the Source Location page in the provisioning connector configuration.

Group name updates

Renaming the group in the datastore triggers PingFederate to rename the group in Salesforce.

Group membership updates

Changing group memberships through the group's properties or a user's properties triggers PingFederate to update the group membership in Salesforce.

Group memberships in the datastore overwrites the group memberships in Salesforce.

Group deletion

Triggered by any of the following:

- Deleting the group in the datastore triggers PingFederate to delete the group in Salesforce. Group deletions are permanent and can't be undone.
- The group is removed from the datastore group or filter that is targeted by the provisioning connector.

Salesforce Community users

The Salesforce Connector can create, update, and deprovision users and groups in Salesforce Communities.

User creation

When a user is created in the datastore and Enable Communities is selected in the connection configuration, PingFederate provisions the user:

1. Checks if the user exists as a "contact" in Salesforce and then creates, or updates the contact.
2. Associates the contact with a Salesforce business account. The account ID comes from the attribute mappings configured in Creating a connection.

Copyright © 2025 Ping Identity Corporation

3. Creates a user in Salesforce and links it to the contact. The user is assigned a profile, role, and permission sets based on the attribute mappings configured in Creating a connection.

Use cloned community profiles

For security reasons, Salesforce prevents using standard external profiles for self-registration and user creation.

To allow the PingFederate provisioner to create users with community profiles, clone the standard community profiles and note the IDs of the cloned profiles. Use these IDs in the datastore attribute that you will map to the `Profile ID` attribute in Salesforce in Creating a connection.

Alternately, you can override this security setting on the Communities > Community Settings page in Salesforce by selecting the Allow using standard external profiles for self-registration and user creation checkbox. This is not recommended.

You can find details and a list of affected profiles in Prevent Using Standard External Profiles for Self-Registration and User Creation in the Salesforce Winter 2020 Release Notes.

Key user attributes

Salesforce Communities users must have the following attributes:

- Account ID: The ID of the business account that you want to assign to a user when provisioning to Salesforce.
- Profile ID: The ID associated with a user profile type in Salesforce. The profile determines the type of user and some permissions. For communities, this ID needs to be a "customer" or "partner" profile.

Setup

Upgrade an existing deployment

If you're upgrading from a previous version of the Salesforce Provisioner, note your existing service provider (SP) connection configuration and create a new connection.

Steps

1. Back up your current PingFederate configuration as shown in Configuration archive in the PingFederate documentation.
2. In the PingFederate administrative console, open your existing SP connection.

   Choose from:

   - For PingFederate 10.1 or later: Go to Applications > Integration > SP Connections and select your connection.
   - For PingFederate 10.0 or earlier: Go to Identity Provider > SP Connections and select your connection.
3. Note the attribute mappings for your existing SP connection.

   You can get help by referencing Mapping attributes in the PingFederate documentation.
4. Delete your existing SP connection.

Copyright © 2025 Ping Identity Corporation

Choose from:

- For PingFederate 10.1 or later:
    1. Go to Applications > Integration > SP Connections.
    2. For your existing connection, click Select action and then click Delete. Click Confirm.
- For PingFederate 10.0 or earlier:
    1. Go to Identity Provider > SP Connections > Manage All.
    2. For your existing connection, click Select action and then click Delete. Click Save.

5. Complete the steps in Deploying the integration files.
6. Complete the steps in Creating a connection.
    - In Outbound Provisioning > Manage Channels > Channel, on the Attribute Mapping tab, configure the attribute mappings based on your notes.

Deploying the integration files

To get started with the integration, deploy the Salesforce Provisioner files to your PingFederate directory.

Steps

1. Download the Salesforce Provisioner `.zip` archive from the Ping Identity Integration Directory.
2. Stop PingFederate.
3. If you're upgrading an existing deployment, delete the `pf-salesforce-quickconnection-<version>.jar` file from your `<pf_install>/pingfederate/server/default/deploy` directory.
4. Extract the `.zip` archive and merge the contents of the `dist` directory with your `<pf_install>/pingfederate/server/default/deploy` directory.
5. Enable the PingFederate provisioning engine:
    1. Open your `<pf_install>/pingfederate/bin/run.properties` file.
    2. Change `pf.provisioner.mode` to `STANDALONE`.
    3. Save the file.

    > **Note**
    >
    > Learn more about how you can configure `FAILOVER` mode instead in Deploying provisioning failover in the PingFederate Server Clustering Guide.

6. Start PingFederate.
7. If you operate PingFederate in a cluster, repeat steps 2 - 4 and step 6 for each engine node.

Copyright © 2025 Ping Identity Corporation

Enabling provisioning and single sign-on (SSO) in PingFederate

To use PingFederate for provisioning and SSO, configure an external datastore and set a SAML entity ID.

About this task

Your external datastore acts as the source of data for provisioning. PingFederate also uses an internal datastore to store the state of synchronization between the source datastore and the target datastore.

Learn more about Datastores and Configuring outbound provisioning settings in the PingFederate documentation.

Enabling provisioning and SSO in PingFederate 10.1 or later

Steps

1. In the PingFederate administrative console, configure the datastore for PingFederate to use as the source of user data.

   Learn more about Datastores in the PingFederate documentation.

2. Go to System > Server > Protocol Settings > Federation Info.
3. In the SAML 2.0 Entity ID field, enter a name for PingFederate to use when identifying itself to SAML partners.
4. On the Outbound Provisioning tab, in the Provisioner Data Store list, select the internal database that will store the synchronization state. Click Save.

   Learn more about Configuring outbound provisioning settings in the PingFederate documentation.

Enabling provisioning and SSO in PingFederate 10.0 or earlier

Steps

1. In the PingFederate administrative console, configure the datastore for PingFederate to use as the source of user data.

   Learn more about Datastores in the PingFederate documentation.

2. Go to System > Protocol Settings > Roles & Protocols.
3. Select the Enable Identity Provider IdP Role and Support the Following checkbox.
4. Select the SAML 2.0 and Outbound Provisioning checkboxes. Click Next.
5. Click the Federation Info tab.
6. In the SAML 2.0 Entity ID field, enter a name for PingFederate to use to identify itself to SAML partners.
7. On the Outbound Provisioning tab, in the Provisioner Data Store list, select the internal database that will store the synchronization state. Click Save.

   Learn more about Configuring outbound provisioning settings in the PingFederate documentation.

Copyright © 2025 Ping Identity Corporation

Registering PingFederate as a connected app in Salesforce

About this task

Before PingFederate can access the Salesforce API, it needs an access token. To get the access token, you need to register PingFederate as an application in the Salesforce Setup console. You can find detailed configuration instructions in Create a Connected App in the Salesforce documentation.

> **Note**
>
> The following steps follow the Salesforce Lightning interface. You can find more information about Salesforce Classic in the Salesforce documentation.

Steps

1. Sign on to Salesforce as an administrator.
2. On the Setup console, create a new connected app:
    1. In the search bar, enter `App Manager`.
    2. Click Apps > App Manager.
    3. On the Salesforce Lightning App Manager window, click New Connected App.
3. Under Basic Information, enter your PingFederate server details:
    1. In the Connected App Name field, enter a name that you choose, such as `PingFederate Provisioning`.
    2. In the API Name field, enter a name that you choose without spaces or special characters, such as `PingFederateProvisioning`.
    3. In the Contact Email field, enter your email address.
    4. (Optional) In the Description field, enter a description to display to end users.
4. Under API (Enable OAuth Settings), configure OAuth:
    1. Select Enable OAuth Settings.
    2. In the Callback URL field, enter `https://oauth.pingone.com/ocs/ppm/rest/v1/oauth/oastempcredresponse/`.
    3. In the Available OAuth Scopes list, add the following:
        - Access and manage your data (`api`)
        - Access custom permissions (`custom_permissions`)
        - Perform requests on your behalf at any time (`refresh_token` , `offline_access`)
5. Click Save. If you receive a message that says allow 2 - 10 minutes for the changes to take effect, click Continue.
6. On the application detail page, note the Consumer Key and Consumer Secret.

   You will use these credentials in Getting an API access token from Salesforce
7. Click Manage and then click Edit Policies.

Copyright © 2025 Ping Identity Corporation

8. On the Connected App Edit window, under OAuth policies for Refresh Token Policy, verify that Refresh token is valid until revoked is selected. Click Save.

Registering PingFederate as an SSO provider in Salesforce

About this task

The Salesforce Connector quick-connection template helps you create a connection to Salesforce by pre-populating some configuration settings. Some of the configuration information comes from a SAML metadata XML file that you can export from the Salesforce console after registering PingFederate as a single sign-on (SSO) provider. You can find detailed configuration instructions in Configure SAML Settings for SSO in the Salesforce documentation.

> **Note**
>
> SAML XML transmissions containing assertions must be digitally signed. If you haven't done so, export the public certificate for the signing key to be used for the Salesforce connections. Note the file location. Certificate export is available through the **Digital Signing** section in the security section of the PingFederate Main Menu.

Steps

1. Sign on to Salesforce as an administrator.
2. On the Setup console, on the search bar, enter `Single Sign-On Settings` and then click Single Sign-On Settings.
3. On the Single Sign-On Settings page, under SAML Single Sign-On Settings, click New.
4. On the SAML Single Sign-On Settings page, enter the SAML information for your PingFederate server.
5. In the Name field, enter a name that you choose, such as `My PingFederate Instance`.
6. In the API Name field, enter a name that you choose without spaces or special characters, such as `MyPingFederateInstance`.
7. In the Issuer field, do one of the following:

   Choose from:

   - If you want PingFederate to use a custom entity ID with Salesforce, enter a name that you choose. When you complete Creating a connection, add the name in the Virtual Server IDs field.
   - Otherwise, enter the SAML 2.0 Entity ID that you chose in Enabling provisioning and single sign-on (SSO) in PingFederate.
8. For Identity Provider Certificate, upload your PingFederate signing certificate.
9. In the Entity ID field, do one of the following:

   Choose from:

   - If you want to integrate with Salesforce Communities, enter the URL for your Salesforce community.

     On the Salesforce Setup console, on the Platform Tools > Feature Settings > Communities > All Communities page, copy your community URL.

Copyright © 2025 Ping Identity Corporation



- Otherwise, enter your Salesforce domain. For example, "mycompany.my.salesforce.com" in the URL https://mycompany.my.salesforce.com.

10. (Optional) In the Identity Provider Login URL field, enter the PingFederate endpoint URL for SSO.

> **Tip**
>
> After you complete the steps in Creating a connection, you can copy the SSO URL from the connection summary page.

11. (Optional) In the Custom Logout URL field, enter the logout URL for your IdP application. Salesforce redirects a user to this URL after ending the Salesforce session.
12. Click Save.
13. On the SAML Single Sign-On Settings page, click Download Metadata. Save the `SAMLSP-xxxxxxxxxxxxxxx.xml` file.

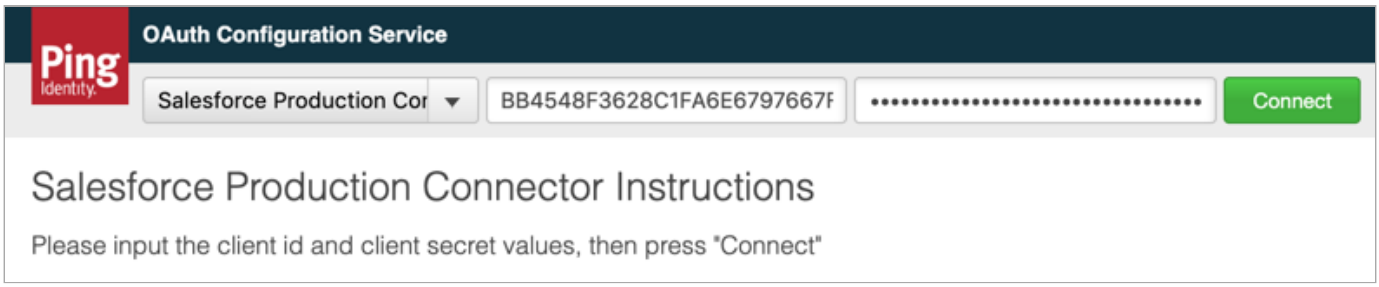
Getting an API access token from Salesforce

About this task

PingFederate and PingOne need an access token to manage users and groups using the Salesforce API. You can use your consumer key and secret with the Ping Identity OAuth Configuration Service (OCS) tool to get your Salesforce access token.

Steps

1. In a browser, open Ping Identity's OAuth Configuration Service (OCS).
2. In the service list, select Salesforce Connector.
3. In the Client ID field, enter the Consumer Key that you noted in Registering PingFederate as a connected app in Salesforce.
4. In the Client Secret field, enter your Consumer Secret.
5. Click Connect.



Copyright © 2025 Ping Identity Corporation

6. Authorize the request through Salesforce:
    1. If you receive a message asking you to sign on to Salesforce, enter your Salesforce administrator credentials and then click Log In.
    2. On the Salesforce Allow Access dialog, click Allow.
7. On the Service Response dialog, note the Access Token and Refresh Token.

    You will use these tokens in Creating a connection



Creating a connection

To allow PingFederate to act as an identity provider and manage users in Salesforce, create a service provider (SP) connection:

Steps

1. In the PingFederate administrator console, create a new SP connection:

    Choose from:

    - For PingFederate 10.1 or later: Go to Applications > Integration > SP Connections. Click Create Connection.
    - For PingFederate 10.0 or earlier: Go to Identity Provider > SP Connections. Click Create Connection.
2. Configure the basic connection details with the Salesforce quick connection template:
    1. On the Connection Template tab, select Use a template for this connection.
    2. In the Connection Template list, select Salesforce Provisioner.
    3. On the Metadata File row, upload the SAMLSP-xxxxxxxxxxxxxxx.xml file that you saved in Registering PingFederate as an SSO provider in Salesforce. Click Next.

Copyright © 2025 Ping Identity Corporation

4. On the Connection Type tab, select Browser SSO Profiles and Outbound Provisioning. Click Next.
5. On the Connection Options tab, click Next.
6. On the General Info tab, if you configured a custom entity ID in the Issuer field in Registering PingFederate as an SSO provider in Salesforce, enter the name in the Virtual Server IDs field and then click Add.
7. In the Connection Name field, enter a name that you choose. Click Next.

3. On the Browser SSO page, configure browser SSO with the following details.

   You can find more information in Configuring IdP Browser SSO and Configuring SSO token creation in the PingFederate documentation.

   If you want to integrate with Salesforce Communities, set your Salesforce Communities URL as the default for SSO:

   1. On the Browser SSO > Protocol Settings > Assertion Consumer Service URL tab, find your Salesforce Communities URL.
   2. In the Actions column, click Edit.
   3. In the Default column, select the checkbox. Click Update.

4. On the Credentials page, configure the digital signature settings with the following details:

   1. On the Digital Signature Settings page, in the Signing Certificate list, select your certificate.
   2. Select Include the certificate in the signature <keyinfo> element. Click Done.

      Learn more about Configuring credentials in the PingFederate documentation.

5. On the Outbound Provisioning page, configure provisioning with the following details:

   Learn more about Configuring outbound provisioning in the PingFederate documentation.

   1. On the Target tab, in the Client ID field, enter the Consumer Key that you noted in Registering PingFederate as a connected app in Salesforce.
   2. In the Client Secret field, enter the Consumer Secret that you noted in Registering PingFederate as a connected app in Salesforce.
   3. In the OAuth Access Token field, enter the Access Token that you noted in Getting an API access token from Salesforce.
   4. In the OAuth Refresh Token field, enter the Refresh Token that you noted in Getting an API access token from Salesforce.
   5. If you want to provision to Salesforce Communities, select Enable Communities.
   6. Under Provisioning Options, customize the provisioning connector behavior. Click Next.

      Learn more about Provisioning options reference.

   7. On the Manage Channels > Attribute Mapping tab, at the bottom of the attribute list, click Refresh Fields to get fields and specifications from your Salesforce site. Complete the attribute mappings by referring to Supported attributes reference.

      You can learn more in Managing channels in the PingFederate documentation.

Copyright © 2025 Ping Identity Corporation

> **Caution**
> If you're provisioning to Salesforce Communities, you must map attributes for any Salesforce fields that are required, including custom fields in users and contacts.

6. On the Activation and Summary page, above the Summary section, note the SSO Application Endpoint.

   Use this value for the Identity Provider Login URL of the provider that you configured in Registering PingFederate as an SSO provider in Salesforce.

7. Turn on the connection and then click Save.

Provisioning options reference

The Salesforce Connector allows you to enable the create, update, and disable capabilities independently and customize other provisioning options.

Field Name	Description
User Create	Selected (default): PingFederate creates users in Salesforce. Cleared: PingFederate does not create users in Salesforce.
User Update	Selected (default): PingFederate updates existing users in Salesforce. Cleared: PingFederate does not update existing users in Salesforce.
User Disable	Selected (default): PingFederate disables users in Salesforce. Cleared: PingFederate does not disable users in Salesforce.
Freeze Users Instead of Disable	This setting only applies when User Disable is True. Selected: When deprovisioning, PingFederate freezes a user. The frozen user can't log in, but the user's data, such as profile and activity, is still visible to other users in Salesforce. Learn more about how to Freeze or Unfreeze User Accounts in the Salesforce documentation. Cleared (default): When deprovisioning, PingFederate disables a user. The user can't log in and their data is not visible to other users in Salesforce. **Note** PingFederate can only unfreeze users when this is selected. PingFederate can only reenable users when this is cleared.
Provision Disabled Users	This setting only applies when User Disable is True and the provisioning connector targets a user in the datastore with a "disabled" status. Selected (default): PingFederate creates a user in Salesforce with a "disabled" status. Cleared: PingFederate does not create a user in Salesforce.

Copyright © 2025 Ping Identity Corporation

Field Name	Description
Note If any of the preceding options are cleared, PingFederate logs a warning in the user workflow section of `provisioner.log` when the related action fails.	
Permission Set Management	Merge with permission sets in Salesforce (default): When a permission set is added in the datastore, PingFederate adds it to the user's existing permission sets in Salesforce. PingFederate doesn't remove any permission sets added in Salesforce by other sources. Overwrite permission sets in Salesforce: When a permission set is added or removed in the datastore, PingFederate overwrites the user's permission sets in Salesforce with those from the datastore.

Supported attributes reference

The following table lists common attributes that can be mapped for user provisioning. The Salesforce Connector dynamically retrieves custom attributes that you configure in the Salesforce console.

You can find a more complete list of Salesforce attributes in the Salesforce Users documentation.

Attribute	Description
Username	The user's username and Salesforce login. This value must be in the format of an email.
Community Nickname	The user's nickname used in Salesforce Communities. Must be `40` characters or fewer. This must be unique in your Salesforce organization. Consider matching the username attribute.
Alias	The user's short name used on list pages, reports, and other pages where the entire name doesn't fit. Must be eight characters or fewer. If the attribute value exceeds eight characters, you can use an OGNL expression to truncate the value. The following code example shows where `sAMAccountName` is truncated to eight characters: `#result=#this.get("sAMAccountName").toString(),` `#result.length() > 8 ? #result.substring(0,8):#result` Learn more about Attribute mapping expressions in the PingFederate documentation.
Email	The user's email address.

Copyright © 2025 Ping Identity Corporation

Attribute	Description
Email Encoding Key	The email encoding. A default set of email encoding options is provided based on your Salesforce environment.
Language Locale Key	The user's language.
Last name	The user's last name.
Locale Sid Key	The locale of the user. A default set of options is provided based on your Salesforce environment.
Profile ID	The identifier associated with a user profile type in Salesforce. The profile determines the type of user and some permissions. Learn more about Profiles in the Salesforce documentation. **Note** You can find out more about community profiles in **Use cloned community profiles** in Salesforce Community users. This attribute is required.
Time Zone SID Key	The user's timezone. A default set of options is provided based on your Salesforce environment.
Account ID	The identifier associated with a company in Salesforce. This field is required for Salesforce Communities users when provisioning "customer" and "partner" users. Leave this field blank when provisioning "internal" users.
City	The user's city.
Company Name	The user's company name.
Country	The user's country.
Department	The user's department.
Division	The user's division.
Extension	The user's phone extension.
Fax	The user's fax number.
Federation Identifier	The unique identifier for a user for identity federation and single sign-on (SSO). This identifier must be unique in your Salesforce organization. Consider matching the username attribute.
First name	The user's first name.
Mobile Phone	The user's mobile phone number.

Copyright © 2025 Ping Identity Corporation

Attribute	Description
Permission Sets	The permissions associated with the user's account. Learn more about Permission Sets in the Salesforce documentation. A list of permission sets is provided during configuration.
Phone	The user's home phone number.
Postal Code	The user's postal or zip code.
State	The user's state or province.
Street	The user's street address.
Title	The user's professional title.
User Role ID	One or more roles the user has.

Release notes

Changelog

The following is the change history for the Salesforce Provisioner.

Salesforce Connector 7.0.1 – April 2020

- Fixed an issue that caused the connection check to return a success when the wrong subdomain was entered.

Salesforce Connector 7.0 – September 2019

- Added support for provisioning to Salesforce Community Cloud, which includes customer, partner, and custom communities.
- Added the ability to create connections to custom Salesforce domains.
- Added support for version `46.0` of the Salesforce REST API.
- Added configuring options to manage permission sets by merging or overwriting.
- Added support for additional Salesforce attributes.
- Added the ability to create multiple Salesforce connections in one PingFederate instance.
- Added default attribute mappings for channels that use PingDirectory and Oracle Directory Server datastores.
- Improved error handling and reporting for cases where users in the target application do not have an ID.
- Improved error handling and reporting for cases where groups are updated or deleted but don't exist in the target application.

Copyright © 2025 Ping Identity Corporation

- Fixed an issue that prevented users with certain special characters from being provisioned to Salesforce.
- Fixed an issue that caused a user to be updated in Salesforce when the user was being disabled and the Update Users option was turned off.

Salesforce Connector 6.1 – July 2017

- Added support for disabling user without updating any user attributes.
- Added configuration control of provisioning disabled users.
- Added custom attribute support.
- Removed non-mappable attributes from attribute mapping page.

Salesforce Connector 6.0 – October 2016

- Added support for additional user attributes.
- Added support for group provisioning.
- Added support for adding users to groups.
- Added configuration options for CRUD capabilities.
- Improved exception handling and reporting.
- Added support for Salesforce REST `v37.0` API.
- Added support for the OAuth Authentication with the OAuth Configuration Service (OCS).
- Removed support for Delegated Authentication in favor of SAML (deprecated).
- Removed support for outbound provisioning to partner and customer portals (deprecated).

Salesforce Connector 5.1.1 – November 2015

- Added support for Salesforce API endpoint change (January 1, 2016).

Salesforce Connector 5.1 – August 2015

- Added support for SSO to Veeva iRep mobile clients through delegated authentication with Salesforce.
- Fixed quick-connection UI performance issues.
- Added support for the ability to freeze user accounts instead of deactivating them through the provisioner.
- Added support for overriding the provisioning target environment endpoint.
- Enhanced support for SP-Initiated SSO through deep linking and SP-initiated SLO by adding a Salesforce Login URL and a Single Logout Landing Page field.

Salesforce Connector 5.0 – October 2014

- Added ability to have one LDAP user mapped to two different Salesforce administrator accounts.

Copyright © 2025 Ping Identity Corporation

- Salesforce API was updated from version `20` to version `31`.
- Added `Permission Set` attribute to SP Connection, which enables mapping of permission sets in Salesforce to an LDAP field.
- Added support for proxy authentication.

Salesforce Connector 4.1 - June 2011

- Added support for Salesforce Chatter and Chatter Free user provisioning.

Salesforce Connector 4.0.1 - November 2010

- Corrected compatibility issue with PingFederate `6.4`.

Salesforce Connector 4.0 - November 2010

- Added quick-connection SSO template and enabled SaaS provisioning to the Salesforce Customer Portal.
- Added quick-connection SSO template and enabled SaaS provisioning to the Salesforce Partner Portal.
- Added quick-connection SSO template to support Salesforce CRM SAML capabilities including metadata export from Salesforce and import into PingFederate, plus optional SaaS provisioning.

Salesforce Connector 3.3 - May 2009

- Added support for Salesforce "Connect for Microsoft Outlook" version `3.2.513`.
- Added an option to remove the email domain from the Salesforce Username for OpenToken authentication by the Authentication Web Service.

Salesforce Connector 3.2 - January 2009

- Added a quick-connection plug-in component for automated SSO configuration and SaaS provisioning support for use with PingFederate `5.3`.
- Added support for Mutual Transport Layer Security (TLS) between Salesforce and the PingFederate Authentication Service.
- Added an option to remove the email domain from the Salesforce User Name for authentication to a LDAP directory.
- Added support for delegated authentication to Salesforce through Salesforce Mobile Clients.
- Combined the former Salesforce User Guide into the Salesforce Quick Connection Guide.

Salesforce Connector 3.1 - September 2008

- Support for URL deep linking.
- Obfuscated LDAP password in the configuration file.
- Enabled Single Logout from Salesforce.
- Included Salesforce Quick Connection Guide for PingFederate `5.2`.

Copyright © 2025 Ping Identity Corporation

- Changed the product name to Salesforce Connector.

Salesforce.com Integration Kit 3.0 - May 2008

- Support for Salesforce "Connect for Microsoft Outlook" plug-in through a new Desktop Proxy Service that allows the use of corporate credentials for access to Salesforce.
- Uses OpenToken for the encrypted token format rather than PFTOKEN.

Salesforce.com Integration Kit 2.0 - July 2007

- Provisioning (new user creation on the Salesforce side).
- PingFederate `v4` administrative console Salesforce configuration for optional provisioning and direct Salesforce login.
- Salesforce Web Service user-validation changes.
- Certification on Salesforce AppExchange.

Salesforce.com Integration Kit 1.1 - November 2006

Use of the PingFederate Standard Adapter, eliminating the Salesforce.com Adapter that was specific to this Integration Kit.

Salesforce.com Integration Kit 1.0 - June 2006

Initial release.

Known issues and limitations

The following are known issues or limitations with the Salesforce Connector.

Known issues

- In PingFederate 9.3 and earlier, PingFederate provisions users as groups, creating unwanted groups. PingFederate also tries to provision groups as users, resulting in error log messages. This only impacts configurations where both group provisioning is enabled and the datastore is Oracle Directory or PingDirectory.

Known limitations

- Attributes
  - The provisioning connector can’t clear user attributes once they’ve been set.
- Certificates
  - Adding a new certificate to PingFederate’s trusted CA store for use in a secure LDAP or LDAPS connection requires a server restart when a secure LDAP connection has already been attempted or established.
- Deprovisioning
  - When deprovisioning a Salesforce customer or partner user, the provisioning connector doesn’t unlink the user from the associated contact.

Copyright © 2025 Ping Identity Corporation

- If a customer or partner user is unlinked in Salesforce from the associated contact, any changes to the user in the datastore causes the provisioning connector to create a new user in Salesforce and links it to the existing contact.
- Guest users in Salesforce can't be frozen. If Freeze users instead of disable is selected in your provisioning options, the guest user won't be disabled or frozen.
- After deleting an LDAP user account, the provisioner doesn't remove the user in the next provisioning cycle when Group DN is specified until a new user is added to the targeted group. This limitation is compounded when the User Create provisioning option is disabled. You can find more details in SaaS provisioner does not remove the user when Group DN is specified in the Ping Identity Knowledge Base.

• Groups

- Group synchronization is based on group name. If multiple groups have the same name, the provisioner syncs to the group that is returned first.

• Performance

- The Salesforce Provisioner dynamically retrieves data from your Salesforce instance. Depending on your Salesforce environment, this could cause a delay when you create an SP connection to Salesforce.
- If multiple PingFederate administrators are creating connections to Salesforce at the same time, the attribute mapping page might not show attributes from Salesforce correctly.

• Refresh tokens

- Refresh token policy must be set to Refresh token is valid until revoked for OAuth as expiring refresh tokens are not supported.

• Salesforce Communities

- The provisioner can link users to "customer" and "partner" business accounts, but not to "person" accounts. Learn more about Accounts in the Salesforce documentation.

Download manifest

The following files are included in the Salesforce Provisioner `.zip` archive:

- `legal`: A directory that contains the legal document
  - `Legal.pdf`: Copyright and license information
- `dist`: Contains the integration files
  - `pf-salesforce-quickconnection-<version>.jar`: The Salesforce Provisioner quick connection template

Salesforce Contacts Provisioner

The Salesforce Contacts Provisioner allows PingFederate to provision Lead and Contact records to Salesforce.

Copyright © 2025 Ping Identity Corporation

Features

- Creates, updates, and deletes Lead or Contact records.
- Allows you to enable the create, update, and delete capabilities independently.

Components

Salesforce Contacts Provisioner connection type:

- Helps you create a service provider connection to Salesforce with the necessary settings and attributes.

Intended audience

This document is intended for PingFederate administrators.

You can get help during the setup process by referencing the following resources:

- The following sections of the PingFederate documentation:
  - Datastores
  - SP connection management
  - Configuring outbound provisioning
  - Configuring outbound provisioning settings

System requirements

- PingFederate `9.0` or later.
- Administrator access to a Salesforce organization account. The account must have "View All Data" permissions. Learn more about account permissions in Enable OAuth Settings for API Integration in the Salesforce documentation.
- To allow PingFederate to make outbound connections to Salesforce, you might need to allow the following endpoint in your firewall: https://*your_salesforce_domain*.com.

Lead and Contact management

The Salesforce Contacts Provisioner links users from a data store to Lead or Contact records in Salesforce. The behavior of each provisioning capability is described below.

Synchronizing existing Lead and Contact records

The provisioning connector synchronizes users from the data store to Salesforce based on the `Email` attribute.

When you configure the attribute mappings in your service provider connection, you can set up synchronization by populating the `Email` attribute with a matching attribute from the data store.

Copyright © 2025 Ping Identity Corporation

For example:

- In Salesforce, Janet's `Email` is `jsmith@example.com`.
- In your data store, Janet's `mail` is `jsmith@example.com`.
- On the Attribute Mapping page, you map the `Email` attribute to `mail`.
- When the provisioning connector runs, the datastore user is provisioned with an `Email` of `jsmith@example.com`. That matches Janet's existing `Email` in Salesforce, so her information in the datastore is synchronized to her Salesforce account.

Lead and Contact record provisioning

Triggered by any of the following:

- A user is added to the datastore group or filter that is targeted by the provisioning connector.

The target is determined by the Source Location page in the provisioning connector configuration.

Lead and Contact record updates

Triggered when a change occurs to a user attribute that is mapped in the provisioning connector configuration.

Lead and Contact record deprovisioning

Deletion is triggered by any of the following:

- A user is deleted from the user store.
- A user is disabled in the user store.
- A user is removed from the datastore group or filter that is targeted by the provisioning connector.

Setup

Upgrading an existing deployment

If you're upgrading from a previous version of the Salesforce Contacts Provisioner, note your existing service provider (SP) connection configuration and create a new connection.

Steps

1. Back up your current PingFederate configuration as shown in Configuration archive in the PingFederate documentation.
2. In the PingFederate administrative console, open your existing SP connection.

   Choose from:

   - For PingFederate 10.1 or later: Go to Applications > Integration > SP Connections and select your connection.
   - For PingFederate 10.0 or earlier: Go to Identity Provider > SP Connections and select your connection.

Copyright © 2025 Ping Identity Corporation

3. Note the attribute mappings for your existing SP connection. You can get help by referencing Mapping attributes in the PingFederate documentation.
4. Delete your existing SP connection.

   Choose from:

   - For PingFederate 10.1 or later:
     1. Go to Applications > Integration > SP Connections.
     2. For your existing connection, click Select action and then click Delete. Click Confirm.
   - For PingFederate 10.0 or earlier:
     1. Go to Identity Provider > SP Connections > Manage All.
     2. For your existing connection, click Select action and then click Delete. Click Save.
5. Complete the steps in Deploying the integration files.
6. Complete the steps in Creating a provisioning connection.
   - In Outbound Provisioning > Manage Channels > Channel, on the Attribute Mapping tab, configure the attribute mappings based on your notes.

Deploying the integration files

To get started with the integration, deploy the Salesforce Contacts Provisioner files to your PingFederate directory.

Steps

1. Download the Salesforce Contacts Provisioner `.zip` archive from the Ping Identity Integration Directory.
2. Stop PingFederate.
3. If you're upgrading an existing deployment, delete `pf-salesforce-contacts-quickconnection-<version>.jar` from `<pf_install>/pingfederate/server/default/deploy`.
4. In the `.zip` archive, copy the contents of `dist` to `<pf_install>/pingfederate/server/default`.
5. Enable the PingFederate provisioning engine.
   1. In `<pf_install>/pingfederate/bin`, open `run.properties` for editing.
   2. Change `pf.provisioner.mode` to `STANDALONE`. Save the file.

   > **Note**
   >
   > Learn more about how to configure `FAILOVER` mode instead in Deploying provisioning failover in the PingFederate Server Clustering Guide.
6. Start PingFederate.
7. If you operate PingFederate in a cluster, repeat steps 2 - 6 for each engine node.

Copyright © 2025 Ping Identity Corporation

Enabling provisioning in PingFederate

To use PingFederate for provisioning, configure an external datastore and enable outbound provisioning.

About this task

Your external datastore acts as the source of data for provisioning. PingFederate also uses an internal datastore to store the state of synchronization between the source datastore and the target datastore.

Learn more in Datastores and Configuring outbound provisioning settings in the PingFederate documentation.

Steps

1. Configure the external datastore that PingFederate will use as the source of user data. You can find instructions in Adding a new datastore in the PingFederate documentation.

   > **Note**
   >
   > When targeting users and groups for provisioning, exclude the user account that you will use to administer users in your connection to Salesforce. This prevents the PingFederate provisioning engine from interfering with the account that provisions users and groups.

2. Do one of the following:

   Choose from:

   - For PingFederate 10.1 or later: Go to System > Server > Protocol Settings.
   - For PingFederate 10.0 or earlier:
     1. Go to System > Protocol Settings > Roles & Protocols.
     2. Select Enable Identity Provider IdP Role and Support the Following.
     3. Select Outbound Provisioning. Click Next.

3. On the Outbound Provisioning tab, select the PingFederate internal datastore. Click Save.

   You can find more information in Configuring outbound provisioning settings in the PingFederate documentation.

Registering PingFederate as a connected app in Salesforce

About this task

Before PingFederate can access the Salesforce API, it needs an access token. To get the access token, you need to register PingFederate as an application in the Salesforce Setup console. You can find detailed configuration instructions in Create a Connected App in the Salesforce documentation.

> **Note**
>
> The following steps follow the Salesforce Lightning interface. Reference the Salesforce documentation for more information about Salesforce Classic.

Copyright © 2025 Ping Identity Corporation

Steps

1. Sign on to Salesforce as an administrator.
2. On the Setup console, create a new connected app.
    1. In the search bar, enter `App Manager`.
    2. Click Apps > App Manager.
    3. On the Salesforce Lightning App Manager window, click New Connected App.
3. In Basic Information, enter your PingFederate server details.
    1. In the Connected App Name field, enter a name that you choose, such as `PingFederate Provisioning`.
    2. In the API Name field, enter a name that you choose without spaces or special characters, such as `PingFederateProvisioning`.
    3. In the Contact Email field, enter your email address.
    4. (Optional) In the Description field, enter a description to display to end users.
4. In API (Enable OAuth Settings), configure OAuth.
    1. Select Enable OAuth Settings.
    2. In the Callback URL field, enter `https://oauth.pingone.com/ocs/ppm/rest/v1/oauth/oastempcredresponse/`.
    3. In the Available OAuth Scopes list, add the following:
        - Access and manage your data (`api`).
        - Access custom permissions (`custom_permissions`).
        - Perform requests on your behalf at any time (`refresh_token`, `offline_access`).
5. Click Save. If you receive a message that says allow 2 - 10 minutes for the changes to take effect, click Continue.
6. On the application detail page, note the Consumer Key and Consumer Secret.

   You will use these credentials in Getting an API access token from Salesforce.
7. Click Manage and then click Edit Policies.
8. On the Connected App Edit window, under OAuth policies for Refresh Token Policy, verify that Refresh token is valid until revoked is selected. Click Save.

Getting an API access token from Salesforce

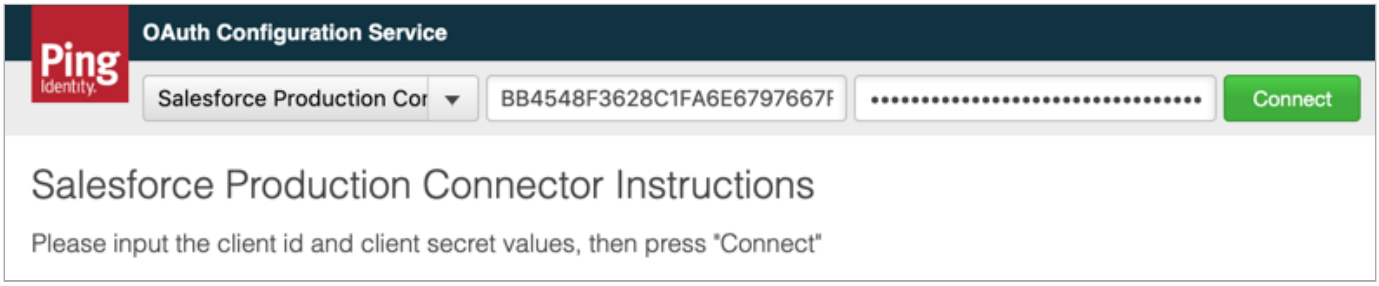
About this task

PingFederate needs an access token to manage users and groups using the Salesforce API. You can use your consumer key and secret with the Ping Identity OAuth Configuration Service (OCS) tool to get your Salesforce access token.

Copyright © 2025 Ping Identity Corporation

Steps

1. In a browser, open Ping Identity's OAuth Configuration Service (OCS).
2. In the service list, select Salesforce Connector.
3. In the Client ID field, enter the Consumer Key that you noted in Registering PingFederate as a connected app in Salesforce.
4. In the Client Secret field, enter your Consumer Secret.
5. Click Connect.



6. Authorize the request through Salesforce.
    1. If you receive a message asking you to sign on to Salesforce, enter your Salesforce administrator credentials and then click Log In.
    2. On the Salesforce Allow Access dialog, click Allow.
7. On the Service Response dialog, note the Access Token and Refresh Token.

   You will use these tokens in Creating a provisioning connection.



Copyright © 2025 Ping Identity Corporation

Creating a provisioning connection

To allow PingFederate to manage users in Salesforce, configure a service provider (SP) connection.

About this task

Configure PingFederate to provision users and groups to Salesforce and enable single sign-on (SSO). You can find more general information in Configuring outbound provisioning in the PingFederate documentation.

> **Note**
>
> You can complete the following steps to create a new SP connection or you can modify an existing connection.

Steps

1. In the PingFederate administrator console, create a new SP connection:

   Choose from:

   - For PingFederate 10.1 or later: Go to Applications > Integration > SP Connections. Click Create Connection.
   - For PingFederate 10.0 or earlier: Go to Identity Provider > SP Connections. Click Create Connection.

2. On the Connection Template tab, select Do not use a template for this connection. Click Next.
3. On the Connection Type tab, select only Outbound Provisioning. In the Type list, select Salesforce Contacts Provisioner. Click Next.
4. On the General Info tab, configure the basic connection information.
   1. In the Entity ID field, enter any value. This field is not used for this integration.
   2. In the Connection Name field, enter a name that you choose. Click Next.
5. On the Outbound Provisioning tab, configure provisioning with the following details:

   You can find more information in Configuring outbound provisioning in the PingFederate documentation:

   1. On the Target tab, in the Client ID field, enter the Consumer Key that you noted in Registering PingFederate as a connected app in Salesforce.
   2. In the Client Secret field, enter the Consumer Secret that you noted in Registering PingFederate as a connected app in Salesforce.
   3. In the OAuth Access Token field, enter the Access Token that you noted in Getting an API access token from Salesforce.
   4. In the OAuth Refresh Token field, enter the Refresh Token that you noted in Getting an API access token from Salesforce.
   5. In the Salesforce Domain field, enter the domain of your Salesforce site. For example, `mycompany.my.salesforce.com` in the URL https://mycompany.my.salesforce.com.
   6. In the Salesforce Record Type list, select the type of record you want to create in Salesforce.

Copyright © 2025 Ping Identity Corporation

> **Note**
>
> You can find more information about Contacts and Leads in the Salesforce documentation.

> **Note**
>
> Although you can change this setting after creating the connection, it requires that you refresh and remap all attributes. Instead, you can create a new connection with the other record type.

7. Under Provisioning Options, customize the provisioning connector behavior by referring to Provisioning options reference.
8. On the Manage Channels > Attribute Mapping tab, at the bottom of the attribute list, click Refresh Fields to get fields and specifications from your Salesforce site. Complete the attribute mappings by referring to Supported attributes reference.

   You can find more information in Managing channels in the PingFederate documentation.

6. On the Activation and Summary tab, above the Summary section, click the toggle to turn on the connection. Click Save.

Provisioning options reference

The Salesforce Contacts Provisioner allows you to enable actions independently and customize other provisioning options.

Field Name	Description
Record Create	*Selected (default)* PingFederate creates Contacts and Leads in Salesforce. *Cleared* PingFederate does not create Contacts and Leads in Salesforce.
Record Update	*Selected (default)* PingFederate updates existing Contacts and Leads in Salesforce. *Cleared* PingFederate does not update existing Contacts and Leads in Salesforce.
Record Delete	*Selected* PingFederate deletes Contacts and Leads in Salesforce. *Cleared (default)* PingFederate does not delete Contacts and Leads in Salesforce.

Copyright © 2025 Ping Identity Corporation

Field Name	Description

> **Note**
> If any of the preceding options are cleared, PingFederate logs a warning in the user workflow section of `provisioner.log` when the related action fails.

Supported attributes reference

The supported attributes are different for Lead and Contact records in Salesforce. You can reference one of the following tables for the record type that you want to provision:

Contact record attributes reference

The following table lists the core attributes that can be mapped for Contact records in Salesforce. Learn more in Contact in the Salesforce documentation.

Attribute	Description
Email	The Contact's email address. This attribute is required.
Last name	The Contact's last name. This attribute is required.
Assistant's Name	The Contact's assistant's name.
Birthdate	The Contact's birthdate.
Clean Status	Indicates the record's clean status compared with Data.com. Values include: `Matched`, `Different`, `Acknowledged`, `NotFound`, `Inactive`, `Pending`, `SelectMatch`, or `Skipped`.
Contact Description	A description of the Contact.
Data.com Key	The Contact's Contact ID in Data.com.
Department	The Contact's department.
EmailBouncedDate	The date and time that an email bounced. Applies when bounce management is activated and when an email sent to the Contact bounced.
EmailBouncedReason	The reason that an email bounced. Applies when bounce management is activated and when an email sent to the Contact bounced.

Copyright © 2025 Ping Identity Corporation

Attribute	Description
Fax	The Contact's fax number.
First name	The Contact's first name.
Home Phone	The Contact's home phone number.
Individual ID	ID of the data privacy record associated with this Contact. This field is available when Data Protection and Privacy are enabled in Salesforce.
Industry	The Contact's industry.
Lead Source	The Contact's source, such as web, phone inquiry, partner referral, purchased list, other.
Mailing City	The Contact's mailing city.
Mailing Country Code	The Contact's mailing country code.
Mailing Country	The Contact's mailing country.
Mailing Geocode Accuracy	Accuracy level of the geocode for the mailing address.
Mailing Latitude	Used with Longitude to specify the precise geolocation of an address. Acceptable values are numbers between `-90` and `90` up to `15` decimal places.
Mailing Longitude	Used with Latitude to specify the precise geolocation of an address. Acceptable values are numbers between `-180` and `180` up to `15` decimal places.
Mailing Phone	The Contact's mailing phone number.
Mailing State/Province Code	The Contact's mailing state or province code.
Mailing State/Province	The Contact's mailing state or province.
Mailing Street	The Contact's mailing street.
Mailing Zip/Postal Code	The Contact's mailing ZIP or postal code.
Mobile Phone	The Contact's mobile phone number.
Other City	The Contact's "other" city.
Other Country Code	The Contact's "other" country code.
Other Country	The Contact's "other" country.
Other Geocode Accuracy	Accuracy level of the geocode for the "other" address.

Copyright © 2025 Ping Identity Corporation

Attribute	Description
Other Latitude	Used with Longitude to specify the precise geolocation of an address. Acceptable values are numbers between `-90` and `90` up to `15` decimal places.
Other Longitude	Used with Latitude to specify the precise geolocation of an address. Acceptable values are numbers between `-180` and `180` up to `15` decimal places.
Other Phone	The Contact's "other" phone number.
Other State/Province Code	The Contact's "other" state or province code.
Other State/Province	The Contact's "other" state or province.
Other Street	The Contact's "other" street.
Other Zip/Postal Code	The Contact's "other" ZIP or postal code.
Owner ID	The ID of the Contact's owner.
Phone	The Contact's phone number.
Reports to ID	The ID of the person that the Contact reports to.
Salutation	The Contact's salutation.
Title	The Contact's job title.

Lead record attributes reference

The following table lists the core attributes that can be mapped for Lead records in Salesforce. You can find more details in Lead in the Salesforce documentation.

Attribute	Description
Email	The Lead's email address. This attribute is required.
Company	The Lead's company name. This attribute is required for Lead records.
Last name	The Lead's last name. This attribute is required.
Annual Revenue	The annual revenue for the Lead's company.

Copyright © 2025 Ping Identity Corporation

Attribute	Description
City	The Lead's city.
Clean Status	Indicates the record's clean status compared with Data.com. Values include: `Matched`, `Different`, `Acknowledged`, `NotFound`, `Inactive`, `Pending`, `SelectMatch`, or `Skipped`.
Company D-U-N-S Number	The Lead's company DUNS number.
Country	The Lead's country.
Country Code	The Lead's country code.
D&B Company	The Lead's D&B company number.
Data.com Key	The Lead's contact ID in Data.com.
Description	A description of the Lead.
EmailBouncedDate	The date and time that an email bounced. Applies when bounce management is activated and when an email sent to the Lead bounces.
EmailBouncedReason	The reason that an email bounced. Applies when bounce management is activated and when an email sent to the Lead bounces.
Employees	The number of employees in the Lead's company.
Fax	The Lead's fax number.
First name	The Lead's first name.
Geocode Accuracy	Accuracy level of the geocode for the address.
Individual ID	ID of the data privacy record associated with this Lead. This field is available when Data Protection and Privacy are enabled in Salesforce.
Industry	The Lead's industry.
Latitude	Used with Longitude to specify the precise geolocation of an address. Acceptable values are numbers between `-90` and `90` up to `15` decimal places.
Lead Source	The Lead's source, such as web, phone inquiry, partner referral, purchased list, other.
Longitude	Used with Latitude to specify the precise geolocation of an address. Acceptable values are numbers between `-180` and `180` up to `15` decimal places.
Mobile Phone	The Lead's mobile phone number.
Owner ID	The ID of the Lead's owner.

Copyright © 2025 Ping Identity Corporation

Attribute	Description
Phone	The Lead's phone number.
Rating	The Lead's rating, such as cold, warm, hot.
Salutation	The Lead's salutation.
State/Province	The Lead's state or province.
State/Province Code	The Lead's state or province code.
Status	The Lead's status, such as open, working, closed.
Street	The Lead's street.
Title	The Lead's job title.
Unread By Owner	If true, the Lead has been assigned, but hasn't been viewed.
Website	The Lead's website.
Zip/Postal Code	The Lead's ZIP or postal code.

Release notes

Changelog

The following is the change history for the Salesforce Contacts Provisioner.

Salesforce Connector 1.0 – September 2020

- Initial release.
- Added the ability to manage Leads and Contacts in Salesforce.
- Added the ability to enable the create, update, and disable capabilities independently.

Known issues and limitations

The following are known issues or limitations for the Salesforce Contacts Provisioner.

Known issues

There are no known issues.

Copyright © 2025 Ping Identity Corporation

Known limitations

- Converted contacts and leads
  - When a Contact record is converted to a User in Salesforce:
    - The Salesforce Contacts Provisioner can continue to update the Contact record, but changes are not reflected in the new User record.
    - The Salesforce Contacts Provisioner can't delete the Contact record. Instead, it shows the following error:

```
[{"message":"Your attempt
to delete jsmith could not be completed because it is associated with
the following portal users.: jsmith@example.com\n","errorCode":"DELETE_FAILED","fields":[]}]
```

  - When a Lead record is converted to another record type in Salesforce:
    - The Salesforce Contacts Provisioner can still delete the Lead record, but can't update it. Instead, it shows the following error:

```
"[{"message":"cannot reference converted lead",
"errorCode":"CANNOT_UPDATE_CONVERTED_LEAD", "fields":[]}]"
```

    - If the Lead record is deleted from your data store, but not deleted from Salesforce and a new Lead is created in the directory with the same email address, the synchronization fails with the preceding message.
- Attributes
  - The provisioning connector can't clear user attributes after they've been set.
- Certificates
  - Adding a new certificate to PingFederate's trusted CA store for use in a secure LDAP (or LDAPS) connection requires a server restart when a secure LDAP connection has already been attempted or established.
- Deprovisioning
  - After deleting an LDAP user account, the provisioner does not remove the user in the next provisioning cycle when Group DN is specified until a new user is added to the targeted group. This limitation is compounded when the User Create provisioning option is disabled. Learn more in SaaS provisioner does not remove the user when Group DN is specified article in the Ping Identity Knowledge Base.
- Performance
  - The Salesforce Contacts Provisioner dynamically retrieves data from your Salesforce instance. Depending on your Salesforce environment, this could cause a delay when you create an SP connection to Salesforce.
  - If multiple PingFederate administrators are creating connections to Salesforce at the same time, the attribute mapping page might not show attributes from Salesforce correctly.

Copyright © 2025 Ping Identity Corporation

- Refresh tokens
    - Refresh token policy must be set to "Refresh token is valid until revoked" for OAuth because expiring refresh tokens are not supported.

Download manifest

The following files are included in the Salesforce `.zip` archive:

- `ReadMeFirst.pdf`: Contains links to this online documentation
- `legal`: A directory that contains the legal document
    - `Legal.pdf`: Copyright and license information
- `dist`: Contains the integration files
    - `deploy`: Contains the Java libraries
        - `pf-salesforce-contacts-quickconnection-<version>.jar`: The Salesforce Contacts Provisioner quick connection template

Copyright © 2025 Ping Identity Corporation

SAP NetWeaver Integration Kit

The PingFederate Integration Kit for SAP NetWeaver provides Identity Provider (IdP) and Service Provider (SP) functionality to PingFederate. This kit allows an enterprise to extend its existing NetWeaver investment by expanding the reach of the NetWeaver domain to federated partner applications.

Additionally, this kit enables an SP enterprise to accept SAML assertions and provide single sign-on (SSO) to NetWeaver applications. The assertions may be sent using either the SAML protocol (version 2.0 or 1.x) or the WS-Federation passive-requestor protocol.

Intended audience

This document is intended for PingFederate administrators.

Before you start, you should be familiar with the following parts of the PingFederate documentation:

- Identity provider SSO configuration
- Managing IdP adapters

System requirements

- PingFederate 6 or later
- NetWeaver Application Server 7.0, NetWeaver Application Server 7.3, or NetWeaver Application server 7.4
- For an IdP:
  - SAP Single Sign-on (SSO) Extension Library (`sapssoext`), which existing customers can obtain from SAP. See Single Sign-On to Non-SAP Systems and Applications in the SAP documentation.
- For a SP:
  - The PingFederateOpenToken Adapter 2.5.1 or later installed on PingFederate
  - `PFLoginModule` , installed on the SAP NetWeaver Application Server

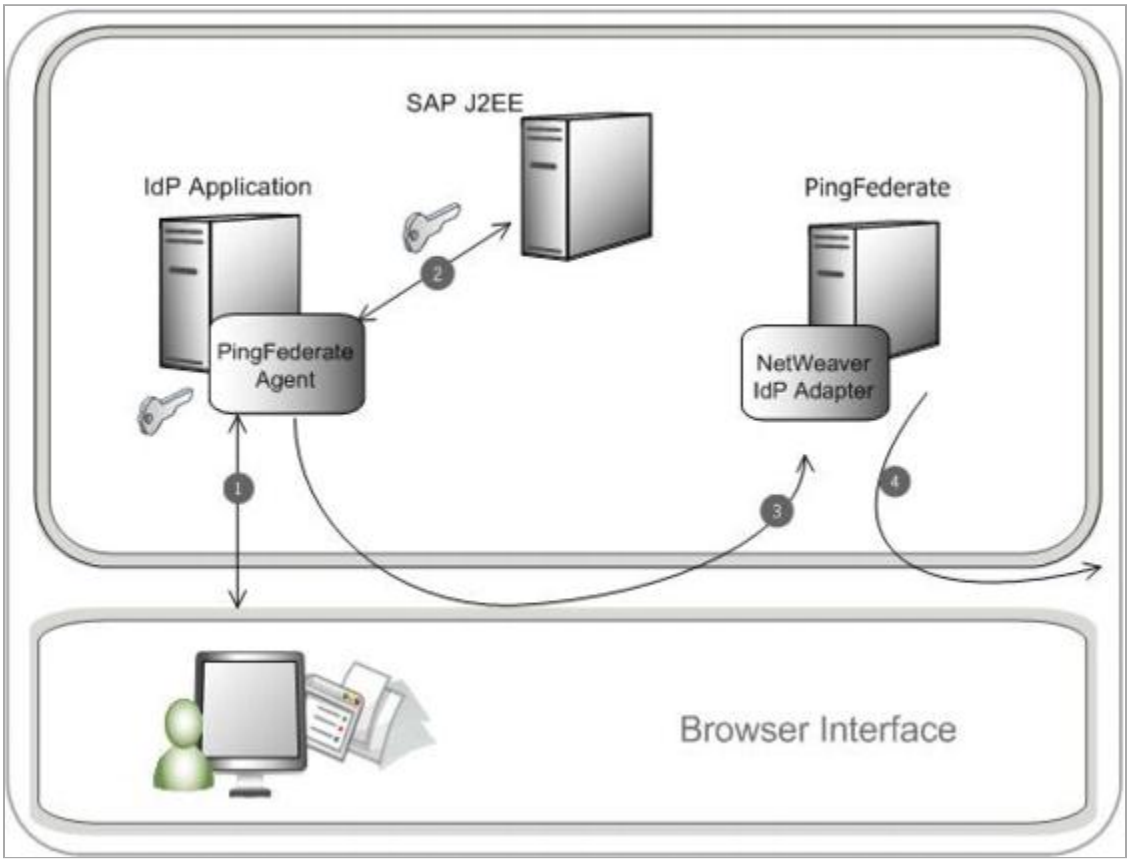
Setup

Implementing IdP functionality

The NetWeaver IdP Adapter uses the SAP SSO extension library to decrypt the session ticket and pass the attributes to the PingFederate server, which maps the values into an assertion and sends the assertion to the SP's federation gateway. For more information about configuration setup and attribute mapping, see Configuring an IdP adapter instance and User attributes in the PingFederate documentation.

The following figure illustrates the request flow and how the NetWeaver IdP Adapter is used in generating a SAML assertion:

Copyright © 2025 Ping Identity Corporation



Processing Steps

1. **The user's browser accesses the IdP application.**
2. **The SAP J2EE Server authenticates the user and creates a session ticket.**
3. **The user clicks a link that initiates a Single Sign-on (SSO) transaction to the partner application. The request is redirected to the PingFederate IdP Server.**
4. **The NetWeaver IdP Adapter retrieves the session ticket from the session cookie, decrypts the session ticket, and then transfers the attributes to the PingFederate IdP Server.**
5. **The PingFederate IdP server generates a SAML assertion and redirects the request, with the assertion, back through the user's browser to the SP site.**

IdP installation and setup

About this task

This section describes how to install and configure the PingFederate Integration Kit for SAP NetWeaver for an IdP.

Steps

1. **Shut down the PingFederate server.**
2. **Unzip the distribution `.zip` archive.**

Copyright © 2025 Ping Identity Corporation

3. Copy the `pf-netweaver-adapter-1.0.jar` file from the `dist` directory to this location:

   `[PingFederate installation directory]/server/default/deploy`

4. Obtain the SSO extension library from SAP and install the libraries on the system running PingFederate (see System requirements).

5. Restart the PingFederate server.

6. Log on to the PingFederate administrative console and click Adapters from the My IdP Configuration side of the Main Menu.

   For more information, see Configuring an IdP adapter instance in the PingFederate documentation.

7. Click Create New Adapter Instance.

8. Enter the Instance Name and Instance ID. Select PF4 NetWeaver IdP Adapter v1.0 as the Type and click Next.

9. Enter the values for adapter configuration described below and click Next.

Property	Description
Domain Name	Your domain name, preceded by a period (e.g., ".pingidentity.com").
SAP TicketName	The name of SAP session ticket for SSO.
PSE File	The path to `verify.pse`. This file contains the public key for verifying the digital signature. Contact SAP support to obtain this file.
PAB Password	The private address-book password for accessing the public key.
Login URL	An optional URL where the user is redirected if SAP session ticket is not found.

10. Optionally, on the Extended Contract tab, you can configure additional attributes for the adapter. For more information, see Key concepts in the PingFederate documentation.

    For instance, you can use the extended adapter contract for Policy Server response-object attributes. Click Next.

11. On the Adapter Attributes tab, select userId under Pseudonym. You can also select any extended attributes specified in the previous tab. Click Next.

    For more information about this tab, see Set pseudonym and masking options in the PingFederate documentation.

12. On the Summary tab, verify that the information is correct and click Done.

13. Click Save to complete the adapter configuration.

SAP J2EE setup

Steps

- Configure NetWeaver J2EE to create SSO tickets.

Copyright © 2025 Ping Identity Corporation

For instructions, see the SAP Help.

Testing the IdP adapter

About this task

You can test this adapter using the samples applications that are included in the Java Integration Kit.

Steps

1. Download the Java Integration kit from the PingFederate server add-ons page.
2. Complete the steps in Sample application setup in the Java Integration Kit documentation to set up an IdP application.
3. Configure an instance of the NetWeaver IdP Adapter as shown in IdP installation and setup.
4. Reconfigure the SP connection to use the NetWeaver Adapter instance.

   Delete the existing adapter instance and map the NetWeaver Adapter instance in its place. For more information, see Managing mappings in the PingFederate documentation.
5. On a web page protected by the NetWeaver Application Server, create an SSO link to the PingFederate `startSSO` endpoint, including the sample SP's connection ID, in the following format:

   ```
   [.codeph]``http[s]://<PF_host>:<port>/IdP/startSSO.ping ``
   [.codeph]``?PartnerSpId=<connection_id>``
   ```

   where:

   - `<PF_host>` is the machine running the PingFederate server,
   - `<port>` is the PingFederate port (refer to the PingFederate *Administrator's Manual*),
   - `<connection_id>` is the Connection ID of the SP connection.
6. Access the protected web page by authenticating through NetWeaver, and click the SSO link.

Result

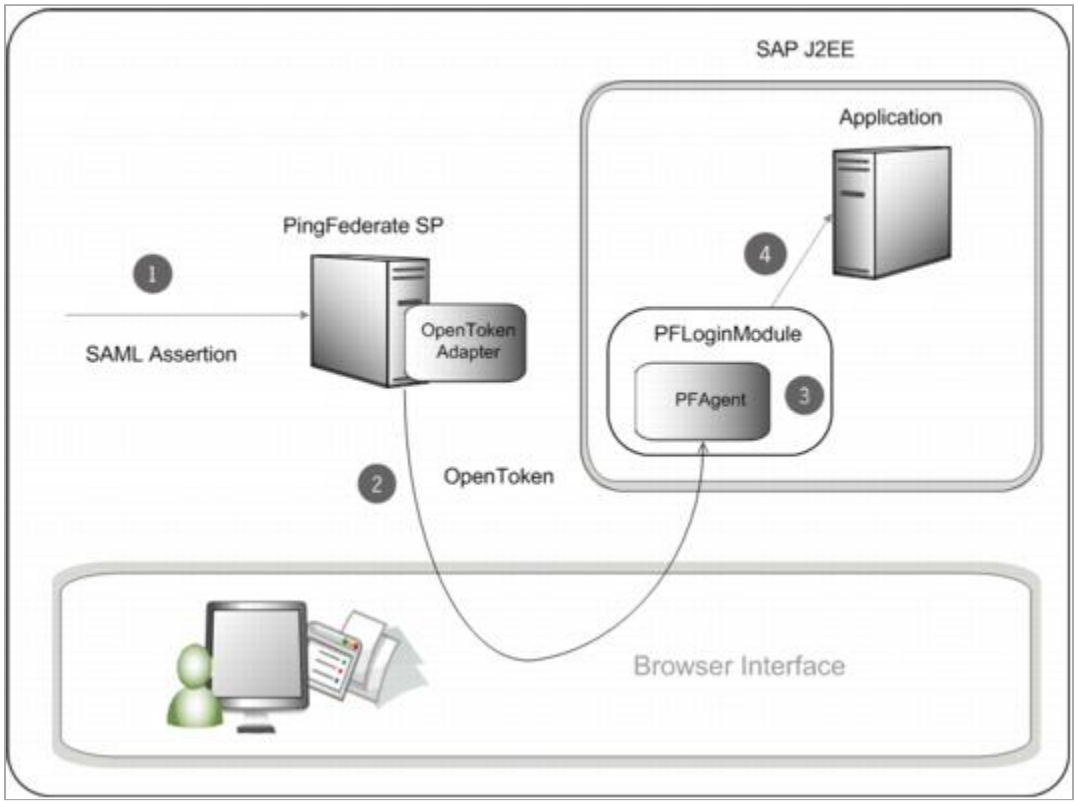
You are logged on to the Java sample application.

Implementing SP functionality

The PingFederate SP server receives an assertion (see Service provider SSO configuration in the PingFederate documentation), wraps the received attributes into `OpenToken`, and redirects to an application protected by NetWeaver. The `PFLoginModule` configured in NetWeaver extracts the `UserID` from OpenToken and authenticates the user. Note that `UserID` is the value of the "subject" attribute in the OpenToken.

The following figure illustrates the request flow and how the PingFederate OpenToken SP Adapter wraps attributes from the assertion into `OpenToken` and passes them to SAP NetWeaver (J2EE Engine):

Copyright © 2025 Ping Identity Corporation



Processing Steps

1. The PingFederate SP server receives a SAML assertion from the IdP.
2. The PingFederate SP server wraps the attributes from the SAML assertion into an `OpenToken` and redirects the token through the user's browser to the application(s) deployed on the SAP J2EE Server.
3. `PFLoginModule`, installed in SAP J2EE Server, parses the `OpenToken` and retrieves the `UserID`.
4. The SAP J2EE server authenticates the user using this UserID and grants access to the SAP Application.

OpenToken adapter setup

About this task

> **Note**
>
> If you have already deployed version 2.5.1 (or later) of the OpenToken Adapter skip steps 1 - 4 in the following procedure.

Steps

1. Stop the PingFederate server if it is running.
2. Remove any existing OpenToken Adapter files (`opentoken*.jar`) from the `<PF_install>/pingfederate/server/default/deploy` directory.

   The adapter JAR file is `opentoken-adapter-<version>.jar`.

Copyright © 2025 Ping Identity Corporation

> **Note**
>
> In the same directory (`server/default/deploy`), delete the `opentoken-java-1.x.jar` file, if it exists.

> **Important**
>
> If it exists, delete the `opentoken-adapter.jar` file from the `<PF_install>/pingfederate/server/default/lib` directory.

3. **Unzip the integration-kit distribution file and copy `opentoken-adapter-2.5.1.jar` from the `/dist` directory to the PingFederate directory:**

   `<PF_install>/pingfederate/server/default/deploy`

4. **Start or restart the PingFederate server.**
5. **Configure an instance of the OpenToken Adapter. You can find instructions in OpenToken Adapter in the PingFederate documentation.**
6. **On the Adapter Actions page in the adapter setup steps, click the Invoke Download link and then click Export to download the `agent-config.txt` properties file to a directory that's readable by the SAP J2EE Server.**

SAP J2EE setup for NetWeaver 7.0

About this task

- **To allow for deep linking for SP-initiated SSO, the login module appends the target-resource URL to the `ssoUrl` property. This feature is supported only for NetWeaver portals; for other applications the target resource is not appended and the user will go to the Default URL configured in PingFederate. For more information, see Configuring default URLs in the PingFederate documentation.**
- **The login module JAR file (`PFLoginModuleJAR.jar`), along with supporting JARS included with this distribution, can be used to create a custom SDA for the NetWeaver platform. For more information see the SAP Help.**

Steps

1. **Deploy the login module included with this distribution (`PFLoginModuleLibrary.sda`) to NetWeaver using the Software Deployment Manager (SDM).**

   > **Note**
   >
   > For information on how to deploy a login module, please refer to SAP Help.

2. **Add a reference to the `Classloader` through the Config Tool, using this value for the library `PingIdentity-PFLoginModuleLibrary` .**

   > **Note**
   >
   > For information on how to add a reference, see Adding a Reference to the Classloader of the Security Provider in the SAP Help Portal.

3. **Configure the login module through the Visual Administrator, using the class name `com.pingidentity.adapters.netweaver.sp.PFLoginModuleClass` and the following options:**

Copyright © 2025 Ping Identity Corporation

Option	Description
`agentPropertiesFileName`	Filename with full path to the location of OpenToken properties file. For example, `C:\agent-config.txt`.
`pfBaseUrl`	Base URL to the PingFederate SP instance.
`enableSPSSO`	If `true`, `PFLoginModule` redirects to the `ssoUrl` (below) if `OpenToken` is not found in the request. This enables SP-initiated SSO functionality for NetWeaver. The default value is `false`.
`ssoUrl`	URL for redirect if SP-initiated SSO, required only if is enabled (above). The value required is PingFederate's application endpoint to start the SSO: http[s]://<PF_host>:<port>/SP/startSSO.ping ?PartnerIdpId=<connection_id> For more information, see Developer Notes below.
`excludeUrI`	List of excluded resource URIs using regular expressions. For example: `/webdynpro.`
`enableSSOCookie`	If `true` and `enableSPSSO` is set to `true`, `PFLoginModule` redirects only if a cookie (an SSO Cookie, defined below) is found in the request. The SP sets an SSO Cookie in the user's browser during an initial IdP-initiated SSO event. When the user arrives at the NetWeaver SP in the future, with the SSO Cookie, the user is redirected to the `ssoUrl`. If `false` and `enableSPSSO` is set to `true`, the `PFLoginModule` redirects any user to the `ssoUrl`, regardless of any SSO Cookie. The default value is `false`
`ssoCookieName`	The name of the SSO cookie to set in the user's browser, required only if `enableSSOCookie` is set to `true`.

> **Note**
>
> For information on how to configure a login module, please refer to the SAP Help.

4. Configure an application to use the login module. A sample configuration, which allows for both SSO and direct authentication, is shown below:

Login Module	Flag
`EvaluateTicketLoginModule`	`SUFFICIENT`
`PFLoginModule`	`REQUISITE`
`BasicPasswordLoginModule`	`REQUISITE`
`CreateTicketLoginModule`	`OPTIONAL`

Copyright © 2025 Ping Identity Corporation

> **Note**
>
> For information on how to configure an application, see Configuring an Application to Use the Login Module in the SAP Help Portal.

SAP J2EE setup for NetWeaver 7.3

About this task

- To allow for deep linking for SP-initiated SSO, the login module appends the target-resource URL to the `ssoUrl` property. This feature is supported only for NetWeaver portals; for other applications the target resource is not appended and the user will go to the Default URL configured in PingFederate. For more information, see Configuring default URLs in the PingFederate documentation.
- The login module JAR file (`PFLoginModuleJAR.jar`), along with supporting JARS included with this distribution, can be used to create a custom EAR for the NetWeaver platform. For more information, see Configuring the Login Module on the AS Java in the SAP Help Portal.

Steps

1. Deploy the login module included with this distribution (`PFLoginModuleLibrary.ear`) to NetWeaver using the appropriate version of SAP NetWeaver Developer Studio.

   > **Note**
   >
   > For information on how to deploy a login module, please refer to SAP Help.

2. Configure the login module through the NetWeaver Administrator, using the following options:

Option	Description
`agentPropertiesFileName`	Filename with full path to the location of OpenToken properties file (for example, `C:\agent-config.txt)` .
`pfBaseUrl`	Base URL to the PingFederate SP instance.
`enableSPSSO`	If `true` , `PFLoginModule` redirects to the `ssoUrl` (below) if `OpenToken` is not found in the request. This enables SP-initiated SSO functionality for NetWeaver. The default value is `false` .
`ssoUrl`	URL for redirect if SP-initiated SSO, required only if is enabled (above). The value required is PingFederate's application endpoint to start the SSO: http[s]://<PF_host>:<port>/SP/startSSO.ping `?PartnerIdpId=<connection_id>` For more information, see Developer Notes below.
`excludeUrI`	List of excluded resource URIs using regular expressions. For example: `./webdynpro.`

Copyright © 2025 Ping Identity Corporation

Option	Description
`enableSSOCookie`	If `true` and `enableSPSSO` is set to `true`, `PFLoginModule` redirects only if a cookie (an SSO Cookie, defined below) is found in the request. The SP sets an SSO Cookie in the user's browser during an initial IdP-initiated SSO event. When the user arrives at the NetWeaver SP in the future, with the SSO Cookie, the user is redirected to the `ssoUrl`. If `false` and `enableSPSSO` is set to `true`, the `PFLoginModule` redirects any user to the `ssoUrl`, regardless of any SSO Cookie. The default value is `false`.
`ssoCookieName`	The name of the SSO cookie to set in the user's browser, required only if `enableSSOCookie` is set to `true`.

> **Note**
>
> For information on how to configure a login module, please refer to the SAP Help.

3. Configure an application to use the login module. A sample configuration which allows for both SSO and direct authentication is shown below:

Login Module	Flag
`EvaluateTicketLoginModule`	`SUFFICIENT`
`PFLoginModule`	`REQUISITE`
`BasicPasswordLoginModule`	`REQUISITE`
`CreateTicketLoginModule`	`OPTIONAL`

> **Note**
>
> For information on how to configure an application, see Configuring an Application to Use the Login Module in the SAP Help Portal.

SAP J2EE setup for NetWeaver 7.3 or 7.4

About this task

- To allow for deep linking for SP-initiated SSO, the login module appends the target-resource URL to the ssoUrlproperty. This feature is supported only for NetWeaver portals; for other applications the target resource is not appended and the user will go to the Default URL configured in PingFederate. For more information, see Configuring default URLs in the PingFederate documentation.
- The login module JAR file (`PFLoginModuleJAR.jar`), along with supporting JARS included with this distribution, can be used to create a custom EAR for the NetWeaver platform. For more information, see Configuring the Login Module on the AS Java in the SAP Help Portal.

Copyright © 2025 Ping Identity Corporation

Steps

1. **Deploy the login module included with this distribution (`PFLoginModuleLibrary.ear`) to NetWeaver using the appropriate version of SAP NetWeaver Developer Studio.**

   > **Note**
   >
   > For information on how to deploy a login module, including how to deploy through shell scripts, please refer to SAP Help.

2. **Configure the login module through the NetWeaver Administrator, using the following options:**

Option	Description
Option	**Description**
`agentPropertiesFileName`	**Filename with full path to the location of OpenToken properties file for example, `C:\agent-config.txt` .**
`pfBaseUrl`	**Base URL to the PingFederate SP instance.**
`enableSPSSO`	**If true, `PFLoginModule` redirects to the `ssoUrl` (below) if `OpenToken` is not found in the request. This enables SP-initiated SSO functionality for NetWeaver. The default value is `false` .**
`ssoUrl`	**URL for redirect if SP-initiated SSO, required only if is enabled (above). The value required is PingFederate's application endpoint to start the SSO: `http[s]://<PF_host>:<port>/SP/startSSO.ping?PartnerIdpId=<connection_id>` For more information, see Developer Notes below.**
`excludeUrI`	**List of excluded resource URIs using regular expressions. For example: `./webdynpro.`**
`enableSSOCookie`	**If true and `enableSPSSO` is set to `true` , `PFLoginModule` redirects *only* if a cookie (an SSO Cookie, defined below) is found in the request. The SP sets an SSO Cookie in the user's browser during an initial IdP-initiated SSO event. When the user arrives at the NetWeaver SP in the future, with the SSO Cookie, the user is redirected to the ssoUrl. If false and `enableSPSSO` is set to `true` , the PFLoginModule redirects any user to the ssoUrl, regardless of any SSO Cookie. (Default: `false`)**
`ssoCookieName`	**The name of the SSO cookie to set in the user's browser, required only if `enableSSOCookie` is set to true.**

Copyright © 2025 Ping Identity Corporation

> **Note**
>
> For information on how to configure a login module, please refer to the SAP Help.

3. Configure an application to use the login module. A sample configuration which allows for both SSO and direct authentication is shown below:

Option	Description
`Login Module`	Flag
`EvaluateTicketLoginModule`	SUFFICIENT
`PFLoginModule`	REQUISITE
`BasicPasswordLoginModule`	REQUISITE
`CreateTicketLoginModule`	OPTIONAL

> **Note**
>
> For information on how to configure an application, see Configuring an Application to Use the Login Module in the SAP Help Portal.

Testing the SP adapter

About this task

You can test this adapter using the samples applications that are included in the Java Integration Kit.

Steps

1. Download the Java Integration kit from the PingFederate server add-ons page.
2. Complete the steps in Sample application setup in the Java Integration Kit documentation to set up an SP application.
3. Configure an instance of the OpenToken SP Adapter and PFLoginModule. For more information, see Implementing SP functionality.
4. Reconfigure the IdP connection to use the OpenToken Adapter instance configured for NetWeaver.

   Delete the existing adapter instance for the connection and map the OpenToken Adapter instance in its place. See Managing mappings in the PingFederate documentation.
5. Protect a web page using NetWeaver.
6. On the same NetWeaver server, create an unprotected web page with a hyperlink to PingFederate's SP-initiated SSO endpoint in the following format:

Copyright © 2025 Ping Identity Corporation

```
http[s]://<PF_host>:<port>/sp/startSSO.ping (1)(2)
?TargetResource=<protected_resource> (3)
?PartnerIdpId=<connection_id> (4)
```

1 `<PF_host>` is the machine running the PingFederate server.
2 `<port>` is the port (refer to the PingFederate documentation).
3 `<protected_resource>` is the web page protected in the previous step.
4 `<connection_id>` is the Connection ID of the IdP connection.

7. Click the SSO link on the unprotected web page.

   The SP application login page displays.

8. Add at least one of the users in the username drop-down list to NetWeaver.

9. On the IdP Application's login page, log in with a username managed by NetWeaver.

   You are redirected to the NetWeaver-protected web page.

Release notes

Changelog

> **Note**
>
> For internal configuration management, the current release of the SAP NetWeaver Integration Kit supports the OpenToken 2.5.1 Adapter and OpenToken Agent.

NetWeaver Integration Kit Version 2.3 – December 2015 (Current Release)

- Added support for NetWeaver 7.4.

NetWeaver Integration Kit Version 2.2.1 – December 2012

- Updated to address security issue found since the previous release.
- Added support for the 2.5.1 version of the OpenToken Adapter and OpenToken Agent

NetWeaver Integration Kit Version 2.2 – August 2012

- Added support for SAP NetWeaver 7.3
- Added support for determining applications that the login module (`PFLoginModule)` apply to using `excludeUrI`
- Added support for the 2.5.0 version of the OpenToken Agent and Adapter

NetWeaver Integration Kit Version 2.1 – March 2012

- Deep linking for SP-Initiated SSO added to the NetWeaver Login Module

Copyright © 2025 Ping Identity Corporation

- Extended the NetWeaver Login Module to support anonymous access for portal applications

NetWeaver Integration Kit Version 2.0 – July 2008

- SP Initiated SSO functionality added to the NetWeaver Login Module
- SP functionality updated to use OpenToken
- SAP Certification for PingFederate 5 and NetWeaver 7

NetWeaver Integration Kit Version 1.1 – April 2007

- SP Functionality for SAP NetWeaver
- SAP Certification

NetWeaver Integration Kit Version 1.0 – October 2006

PingFederate Integration Kit Version 1.0 for NetWeaver was the initial release of the product. It provided the IdP functionality for SAP NetWeaver, whereby users who have been authenticated by NetWeaver can gain access to partner applications using SAML or WS-Federation.

Known issues and limitations

Known issues and limitations

- Logout does not integrate with the JAAS module to perform Single Log-Out (SLO) because of the sequence SLO must take. Applications wishing to use SLO functionality must be modified to include a reference to PingFederate's SLO service within the application itself.
- Account linking is not supported.
- In cases where the SP-initiated SSO property (`enableSPSSO`) is set to true, and the `ssoURL` property is set incorrectly, authentication through PingFederate and access to the administrative console is not possible. IdP-initiated SSO requests can be made along with the NetWeaver administrative console URL as a Target Resource. This temporarily allows access to the NetWeaver administrative console to set the correct SSO URL.

  For example: https://<hostname>:<port>/idp/startSSO.ping?PartnerSpId=<hostname>:default:entityId&TargetResource=http://<hostname>:<port>/nwa

Download manifest

The distribution `.zip` archive for the PingFederate Integration Kit for SAP NetWeaver contains the following:

- `ReadMeFirst.pdf` – contains links to this online documentation
- `/legal` – contains this document:
  - `Legal.pdf` – copyright and license information
- `/dist` – contains the following libraries that are needed to run the adapter:
  - `pf-netweaver-adapter-1.0.jar` – the NetWeaver IdP Adapter JAR file (IdP only)

Copyright © 2025 Ping Identity Corporation

- `PFLoginModuleJAR.jar` – **PingFederate Login Module JAR file that can be used to create a custom Software Development Archive (SDA) or Enterprise Archive (EAR)**
- `PFLoginModuleLibrary.sda` – **SDA for NetWeaver 7.0 that contains `PFLoginModule`**
- `PFLoginModuleLibrary.ear` – **EAR for NetWeaver 7.3 and 7.4 that contains `PFLoginModule`**
- `opentoken-adapter-2.5.1.jar` – **OpenToken Adapter JAR file**
- `opentoken-agent-2.5.1.jar` – **OpenToken Agent JAR file**
- `commons-collections-3.2.jar` – **Apache Commons Collections library**
- `commons-beanutils.jar` – **Apache Commons Bean Utility library**
- `commons-logging.jar` – **Apache Commons Logging library**

Copyright © 2025 Ping Identity Corporation

SCIM Provisioner

The SCIM Provisioner allows PingFederate to integrate with a wide range of services that support the System for Cross-domain Identity Management (SCIM) for user provisioning and single sign-on (SSO).

Features

- Manages users in the target service based on changes in an external datastore that is attached to PingFederate:
  - Creates, updates, disables, and deletes users
  - Allows you to enable the create, update, disable, and delete capabilities independently
  - Allows you to choose whether to disable or delete users when deprovisioning
  - Allows you to provision disabled users
- Manages groups in the target service based on changes in an external datastore that is attached to PingFederate:
  - Creates, updates, and deletes groups
  - Updates group memberships
- Enables browser-based SSO initiated by the service provider (SP) or identity provider (IdP).

Specifications

The SCIM Connector implements the official specifications provided from simplecloud.info.

Feature	Outbound provisioning
SCIM specification	1.1, 2.0
Data format	JavaScript Object Notation (JSON)
User and group CRUD operations	Yes
Custom schema support	*Users* Yes *Groups* No
Filtering support	*Users* Yes *Groups* The connector allows group filtering by retrieving all groups and finding a match.

Copyright © 2025 Ping Identity Corporation

Feature	Outbound provisioning
PATCH	*Users* No *Groups* Yes. Learn more in the SCIM provisioner settings reference.
Authentication method	HTTP Basic Authentication, OAuth bearer token and OAuth client credentials
Source data stores	Active Directory and other LDAPv3-compliant directory servers

Components

The SCIM provisioning and SSO connector:

- Allows PingFederate to manage users in the service based on changes in an external user data store
- (Optional configuration) Allows PingFederate to create an SSO connection to the service
- Includes a quick-connection template that pre-populates some configuration settings

Intended audience

This document is intended for PingFederate administrators. If you need help during the setup process, see the following resources:

- PingFederate documentation:
  - SP connection management
  - Identity provider SSO configuration
  - Managing digital signing certificates and decryption keys
  - Datastores
  - Configuring outbound provisioning
  - Configuring outbound provisioning settings
- The SCIM 1.1 Developer Guide on the Ping Identity Developer site
- The SCIM specification on simplecloud.info

System requirements

- PingFederate 9.0 or later.
- To allow PingFederate to make outbound connections, you might need to allow SCIM endpoints in your firewall.

Copyright © 2025 Ping Identity Corporation

User and group management

The SCIM Provisioner synchronizes users and groups from your datastore to the target service. The following describes the behavior of each provisioning capability.

> **Tip**
>
> You can configure the following capabilities and specify which users to provision when you get to the Creating a provisioning connection part of the setup process.

Synchronizing existing users

PingFederate synchronizes users based on the `userName` attribute in the target service. If a user already exists in your datastore and the target service, mapping this attribute correctly links the two records together. For example:

- In the target service, Janet's `userName` is `jsmith`.
- In your datastore, Janet's `sAMAccountName` is `jsmith`.
- On the Attribute Mapping tab of your provisioning connection configuration, map the `userName` attribute to `sAMAccountName`.
- When the provisioning connector runs, the datastore user is provisioned with a `userName` of `jsmith`. That matches Janet's existing `userName` in the target service, so her information in the datastore is synchronized to her the target service account.

User provisioning

PingFederate provisions users when any of the following happens:

- A user is added to the datastore group or filter that is targeted by the provisioning connector.
- A user with `disabled` status is added to the datastore group or filter that is targeted by the provisioning connector, and theProvision disabled users provisioning option is enabled. This feature is not available in all provisioning connector versions.

You can define which users PingFederate targets for provisioning on the Source Location tab of your provisioning connection configuration.

User updates

PingFederate updates users when a user attribute changes in your datastore.

You can define which attributes PingFederate monitors for changes on the Attribute Mapping tab of your provisioning connection configuration.

Copyright © 2025 Ping Identity Corporation

User deprovisioning

PingFederate deprovisions users when any of the following happens:

- A user is deleted from the user store.
- A user is disabled in the user store.
- A user is removed from the datastore group or filter that is targeted by the provisioning connector.

The Remove User Action setting in the connection configuration determines whether the deprovisioning action disables or deletes the user.

Synchronizing existing groups

PingFederate synchronizes groups from the datastore to the target service based on the group name. For example:

- In the target service, there is a group is named `Accounting`.
- In your datastore, there is a group with a `CN` of `Accounting`.
- When the provisioning connector runs, the two groups are synchronized.

Group provisioning

PingFederate provisions groups when a group is added to the datastore filter that is targeted by the provisioning connector.

You can define which groups PingFederate targets for provisioning and monitors for changes on the Source Location tab in your provisioning connection configuration.

> **Note**
>
> When provisioning groups, only required attributes, such as `displayName` and `members`, are supported. Common attributes, such as `id` and `externalID`, are not supported because they're optional attributes.

Group name updates

PingFederate renames groups when they are renamed in the datastore.

Group membership updates

PingFederate updates group memberships when memberships change in the datastore, whether the change is in the group's properties or a user's properties.

Group memberships in the datastore overwrite the group memberships in the target service.

Group deletion

PingFederate deletes groups when any of the following happens:

- The group is deleted in the datastore.

Copyright © 2025 Ping Identity Corporation

- The group is removed from the datastore group or filter that is targeted by the provisioning connector.

> **Important**
>
> Group deletions are permanent and cannot be undone.

Setup

Upgrading an existing deployment

If you're upgrading from a previous version of the SCIM Provisioner, note your existing service provider (SP) connection configuration and create a new connection.

Upgrading an existing deployment in PingFederate 10.1 or later

Steps

1. Back up your current PingFederate configuration as shown in Configuration archive in the PingFederate documentation.
2. In the PingFederate administrative console, go to Applications > Integration > SP Connections and select your connection.
3. Note the attribute mappings for your existing SP connection.

   Learn more in Mapping attributes in the PingFederate documentation.
4. Delete your existing SP connection.
    1. Go to Applications > Integration > SP Connections.
    2. For your existing connection, click Select action, and then click Delete. Click Confirm.
5. Complete the steps in Deploying the integration files.
6. Complete the steps in Creating a provisioning connection.
    - Go to Outbound Provisioning > Manage Channels > Channel. On the Attribute Mapping tab, configure the attribute mappings based on your notes.
7. (Optional) Complete the steps in Adding SSO to a connection.

Upgrading an existing deployment in PingFederate 10.0 or earlier

Steps

1. Back up your current PingFederate configuration as shown in Configuration archive in the PingFederate documentation.
2. In the PingFederate administrative console, go to Identity Provider > SP Connections and select your connection.
3. Note the attribute mappings for your existing SP connection.

Copyright © 2025 Ping Identity Corporation

Learn more in Mapping attributes in the PingFederate documentation.

4. Delete your existing SP connection.
    1. Go to Identity Provider > SP Connections > Manage All.
    2. For your existing connection, click Select action, and then click Delete. Click Save.
5. Complete the steps in Deploying the integration files.
6. Complete the steps in Creating a provisioning connection.
    - Go to Outbound Provisioning > Manage Channels > Channel. On the Attribute Mapping tab, configure the attribute mappings based on your notes.
7. (Optional) Complete the steps in Adding SSO to a connection.

Deploying the integration files

To get started with the integration, deploy the SCIM Provisioner files to your PingFederate directory.

Steps

1. Download the SCIM Provisioner `.zip` archive from the Ping Identity Integration Directory.
2. Stop PingFederate.
3. If you're upgrading an existing deployment, delete the `pf-scim-quickconnection-<version>.jar` file from your `<pf_install>/pingfederate/server/default/deploy` directory.
4. Extract the `.zip` archive and merge the contents of the `dist` directory with your `<pf_install>/pingfederate/server/default/deploy` directory.
5. Enable the PingFederate provisioning engine:
    1. Open your `<pf_install>/pingfederate/bin/run.properties` file.
    2. Change `pf.provisioner.mode` to `STANDALONE`. Save the file.

    > **Note**
    >
    > To configure the `FAILOVER` mode instead, see Deploying provisioning failover in the PingFederate Server Clustering Guide.

6. Start PingFederate.
7. If you operate PingFederate in a cluster, repeat steps 2-4 and step 6 for each engine node.

Enabling provisioning and single sign-on in PingFederate

To use PingFederate for provisioning and single sign-on, configure an external datastore and set a SAML entity ID.

About this task

Copyright © 2025 Ping Identity Corporation

Your external datastore acts as the source of data for provisioning. PingFederate also uses an internal datastore to store the state of synchronization between the source datastore and the target datastore.

Learn more in Datastores and Configuring outbound provisioning settings in the PingFederate documentation.

Enabling provisioning and single sign-on in PingFederate 10.1 or later

Steps

1. In the PingFederate administrative console, configure the datastore for PingFederate to use as the source of user data.

   Learn more in Datastores in the PingFederate documentation.

2. Go to System > Server > Protocol Settings > Federation Info.
3. In the SAML 2.0 Entity ID field, enter a name for PingFederate to use when identifying itself to SAML partners.
4. On the Outbound Provisioning tab, in the Provisioner Data Store list, select the internal database that will store the synchronization state. Click Save.

   Learn more in Configuring outbound provisioning settings in the PingFederate documentation.

Enabling provisioning and single sign-on in PingFederate 10.0 or earlier

Steps

1. In the PingFederate administrative console, configure the datastore for PingFederate to use as the source of user data.

   Learn more in Datastores in the PingFederate documentation.

2. Go to System > Protocol Settings > Roles & Protocols.
3. Select the Enable Identity Provider IdP Role and Support the Following checkbox.
4. Select the SAML 2.0 and Outbound Provisioning checkboxes. Click Next.
5. Click the Federation Info tab.
6. In the SAML 2.0 Entity ID field, enter a name for PingFederate to use to identify itself to SAML partners.
7. On the Outbound Provisioning tab, in the Provisioner Data Store list, select the internal database that will store the synchronization state. Click Save.

   Learn more in Configuring outbound provisioning settings in the PingFederate documentation.

Gathering information from the service provider

Before you configure the SCIM Connector, collect the connection details from the target service.

Steps

1. Note your SCIM URL from the SCIM-enabled SP.

Copyright © 2025 Ping Identity Corporation

2. Note your client credentials from the SCIM-enabled SP. This will depend on the authentication method used by the SP, which might include:
    - Basic Authentication: Username and password.
    - OAuth 2.0 Bearer Token: Access token.
    - OAuth 2.0 Client Credentials: Client ID, client secret, token endpoint.

Creating a provisioning connection

To allow PingFederate to manage users in the target service, create a service provider (SP) connection.

Steps

1. In the PingFederate administrative console, create a new SP connection:

   Choose from:

    - For PingFederate 10.1 or later: go to Applications > Integration > SP Connections. Click Create Connection.
    - For PingFederate 10.0 or earlier: go to Identity Provider > SP Connections. Click Create Connection.

2. Configure the basic connection details with the the target service quick connection template:
    1. On the Connection Template tab, select Use a template for this connection.
    2. In the Connection Template list, select SCIM Provisioner. Click Next.
    3. On the Connection Type tab select Outbound Provisioning. Click Next.
    4. On the General Info tab, in the Connection Name field, enter a name of your choosing. Click Next.
3. On the Outbound Provisioning tab, configure provisioning with the following details.

   Learn more in Configuring outbound provisioning in the PingFederate documentation.

    1. On the Target screen, enter the URL and authentication details that you noted in Gathering information from the service provider.

       > **Note**
       >
       > PingFederate verifies the authentication details when you activate the channel and SP connection.

    2. Under Provisioning Options, customize the provisioning connector behavior by referring to SCIM provisioner settings reference. Click Next.
    3. On the Manage Channels > Attribute Mapping tab, complete the attribute mappings by referring to Supported attributes reference.

       Learn more in Managing channels in the PingFederate documentation.
4. On the Activation and Summary tab, above the Summary section, click the toggle to turn on the connection. Click Save.

Copyright © 2025 Ping Identity Corporation

SCIM provisioner settings reference

Configuration settings and provisioning options for the SCIM Provisioner.

Field Name	Description
SCIM URL	The SCIM base URL for the target service. For example: http://scim-example.com/v2/
SCIM Version	The SCIM version supported by the target service. The options are: • 2.0 (default) • 1.1.
Authentication Methods	The authentication method expected by the target service. The options are: • None (default) • Basic Authentication • OAuth 2 Bearer Token • OAuth 2 Client Credentials. **Note** When an authentication method is selected, only the data required for that method will be processed. Entries in fields for other authentication methods will be ignored.
Basic Authentication	
Basic Authentication Username	The username of the administrator account on the target service.
Basic Authentication Password	The password of the administrator account on the target service.
OAuth 2 Bearer Token	
Access Token	The OAuth access token for the target service.
OAuth 2 Client Credentials	
Token Request Endpoint	The endpoint that the connector uses to get an access token. For example: https://scim-example.com/as/token.oauth2
Client ID	The client ID for the target service.

Copyright © 2025 Ping Identity Corporation

Field Name	Description
Client Secret	The client secret the target service.
Scope	An optional field that allows an admin to specify a comma-delimited list of OAuth scopes that access tokens requested from the SCIM provider should contain.
SCIM Overrides	
Unique User Identifier	The attribute that uniquely identifies a user when PingFederate does not have access to the unique user ID that the target application assigns to a user. *userName (default)* The value of the user name attribute. This sets the following filter expression: `userName eq  <userName_value>` *workEmail* The value of the work email attribute. Use when synchronization based on the username is not desirable. This sets the following filter expression: `emails eq  <workEmail_value>` To override a default filter, use the Filter Expression field. **Important** To change the unique user identifier: 1. Delete the existing SP connection. 2. Restart PingFederate. 3. Create an SP Connection with the new Unique User Identifier setting.

Copyright © 2025 Ping Identity Corporation

Field Name	Description
Filter Expression	A rule that determines how the connector uses the unique user identifier to match existing users in the target application to users in the data store. This expression overrides the default filter expression that is set by the Unique User Identifier field. The filter expression contains three parts: `<attribute_name> <operation> <attribute_value>` The *<attribute_value>* is represented by `"%s"` in the expression. It is populated by the value of the Unique User Identifier. Example filter expressions: • `username eq "%s"` • `email co "%s"` **Note** Check the target service documentation and the SCIM Filtering specification to see which filter expressions are supported.
Authorization Header Type	The type of HTTP authorization header used. For example, `oauth2`.
Users API Path	The users API path is used when the users endpoint deviates from the SCIM specification (`/Users` is used by default when left blank).
Groups Path API	The groups API path is used when the groups endpoint deviates from the SCIM specification (`/Groups` is used by default when left blank).
Results Per Page	Determines the number of groups that PingFederate requests per `GET` request when searching all groups for a match. If the target service has a limit, change this value to match. A value of `-1` retrieves the largest page size allowed by the target service. The default value is `1000`.
Provisioning Options	

Copyright © 2025 Ping Identity Corporation

Field Name	Description
User Create	***Selected (default)*** PingFederate creates users in the target service. ***Cleared*** PingFederate does not create users in the target service.
User Update	***Selected (default)*** PingFederate updates existing users in the target service. ***Cleared*** PingFederate does not update existing users in the target service.
User Disable / Delete	***Selected (default)*** PingFederate removes users from the target service according to the Remove User Action setting. **Note** You might need to enable **User Update** for this to work with some services. ***Cleared*** PingFederate does not remove users from the target service.
Provision Disabled Users	***Selected*** PingFederate creates users in the target service with a "disabled" status. ***Cleared (default)*** If a user has a "disabled" status, PingFederate does not create the user in the target service.
	Note If any of the previous provisioning options are cleared, PingFederate logs a warning in the user workflow section of `provisioner.log` when the related action fails.

Copyright © 2025 Ping Identity Corporation

Field Name	Description
Remove User Action	*Delete* PingFederate deletes the user from the target service. *Disable (default)* PingFederate disables the user in the target service. **Note** Some target applications do not support hard deleting users through external interfaces. For those services, users are disabled. This option applies when User Disable / Delete is selected, and either: • A previously provisioned user no longer meets the condition set on the Source Location tab. • A user has been disabled or deleted from the data store.
Group Name Source	*Common Name (CN) (default)* PingFederate provisions groups to the target service with a name equal to the common name (CN) of the group in the datastore. *Distinguished Name (DN)* PingFederate provisions groups with a name equal to the distinguished name (DN) of the group in the datastore.
Use PATCH for Group Updates	*Selected* PingFederate uses the `PATCH` method to update groups in the target service. Select this option if the target service supports `PATCH` updates to use lighter API calls. *Cleared (default)* PingFederate uses the `PUT` method to update groups in the target service.

Copyright © 2025 Ping Identity Corporation

Field Name	Description
Custom Attribute Schema URNs	An optional field that allows an admin to explicitly specify a comma-delimited list of schema URNS for which to look for custom attributes. This is only required if the SCIM provider doesn't follow the standard naming convention for schema extensions that define custom attributes. For example, URNs such as: urn:ietf:params:scim:schemas:extension:<*Organization Name*>:2.0:User.

Supported attributes reference

The following standard SCIM attributes can be mapped for user provisioning. Different attributes will be supported depending on the target service.

Learn more about SCIM attributes in SCIM specification and the target service documentation.

Attribute	Description
`userName`	A service provider's unique identifier for the user. This attribute is required.
`givenName`	The given name of the user, or first name in most Western languages. For example, `Barbara`, given the full name Ms. Barbara Jane Jensen, III.
`familyName`	The family name of the user, or last name in most Western languages. For example, `Jensen`, given the full name Ms. Barbara Jane Jensen, III.
`middleName`	The middle name or names of the user. For example, `Jane`, given the full name Ms. Barbara Jane Jensen, III.
`honorificPrefix`	The honorific prefix or prefixes of the user, or title in most Western languages. For example, `Ms.`, given the full name Ms. Barbara Jane Jensen, III.
`honorificSuffix`	The honorific suffix or suffixes of the user, or suffix in most Western languages. For example, `III`, given the full name Ms. Barbara Jane Jensen, III.

Copyright © 2025 Ping Identity Corporation

Attribute	Description
`formattedName`	The full name, including all middle names, titles, and suffixes as appropriate, formatted for display. For example, `Ms. Barbara Jane Jensen, III`.
`workEmail`	Work email for the user. For example, `bjensen@example.com`.
`homeEmail`	Home email for the user. For example, `bjensen@example.com`.
`otherEmail`	Other email for the user. For example, `bjensen@example.com`.
`displayName`	The name of the user, suitable for display to end-users.
`title`	The user's title, such as `Vice President`.
`externalId`	A string that is an identifier for the resource as defined by the provisioning client.
`password`	This attribute is intended to be used as a means to set, replace, or compare (for example, filter for equality) a password.
`preferredLanguage`	Indicates the user's preferred written or spoken languages and is generally used for selecting a localized user interface.
`userType`	Used to identify the relationship between the organization and the user. Typical values used might be `Contractor`, `Employee`, `Intern`, `Temp`, `External`, and `Unknown`, but any value can be used.
`locale`	Used to indicate the user's default location for purposes of localizing such items as currency, date time format, or numerical representations.
`nickName`	The casual way to address the user in real life. For example, `Bob` or `Bobby` instead of `Robert`.
`profileUrl`	A URI that is a uniform resource locator that points to a location representing the user's online profile. For example, a web page.

Copyright © 2025 Ping Identity Corporation

Attribute	Description
`profilePhotoUrl`	A URI that is a uniform resource locator that points to the user's profile photo. The resource must be a file (for example, a GIF, JPEG, or PNG image file) rather than a web page containing an image.
`profileThumbnailUrl`	A URI that is a uniform resource locator that points to the user's profile thumbnail. The resource must be a file (for example, a GIF, JPEG, or PNG image file) rather than a web page containing an image.
`timezone`	The user's time zone, in IANA Time Zone database format. For example, `America/Los_Angeles`.
`workPhone`	The work phone number for the user. For example, `+1-201-555-0123`.
`mobilePhone`	The mobile phone number for the user. For example, `+1-201-555-0123`.
`pagerPhone`	The pager number for the user. For example, `+1-201-555-0123`.
`faxPhone`	The fax number for the user. For example, `+1-201-555-0123`.
`homePhone`	The home phone number for the user. For example, `+1-201-555-0123`.
`otherPhone`	Another phone number that can be used to reach the user. For example, `+1-201-555-0123`.
`workStreetAddress`	The work street address for the user, which could include house number, street name, P.O. box, and multiline extended street address information.
`workCity`	The work city or locality component for the user's mailing address.
`workState`	The work state or region component for the user's mailing address.
`workPostalCode`	The work ZIP or postal code component for the user's mailing address.

Copyright © 2025 Ping Identity Corporation

Attribute	Description
`workCountry`	The work country component for the user's mailing address. When specified, the value must be in ISO 3166-1 "alpha-2" code format [ISO3166]. For example, the United States and Sweden are `US` and `SE`, respectively.
`workFormattedAddress`	The user's full work address, formatted for display.
`homeStreetAddress`	The home street address for the user, which may include house number, street name, P.O. box, and multi-line extended street address information.
`homeCity`	The home city or locality component for the user's mailing address.
`homeState`	The home state or region component for the user's mailing address.
`homePostalCode`	The home ZIP or postal code component for the user's mailing address.
`homeCountry`	The home country component for the user's mailing address. When specified, the value MUST be in ISO 3166-1 "alpha-2" code format [ISO3166]. For example, the United States and Sweden are `US` and `SE`, respectively.
`homeFormattedAddress`	The user's full home address, formatted for display.
`otherStreetAddress`	An alternate street address for the user, which may include house number, street name, P.O. box, and multi-line extended street address information.
`otherCity`	The alternate city or locality component for the user's mailing address.
`otherState`	The alternate state or region component for the user's mailing address.
`otherPostalCode`	The alternate ZIP or postal code component for the user's mailing address.
`otherCountry`	The alternate country component for the user's mailing address. When specified, the value MUST be in ISO 3166-1 "alpha-2" code format [ISO3166]. For example, the United States and Sweden are `US` and `SE`, respectively.

Copyright © 2025 Ping Identity Corporation

Attribute	Description
`otherFormattedAddress`	The alternate address for the user, formatted for display.
`qqIm`	The QQ instant messaging address for the user.
`skypeIm`	The Skype instant messaging address for the user.
`gtalkIm`	The Google Talk instant messaging address for the user.
`aimIm`	The AOL Instant Messenger instant messaging address for the user.
`icqIm`	The ICQ instant messaging address for the user.
`yahooIm`	The Yahoo Messenger instant messaging address for the user.
`msnIm`	The MSN Messenger instant messaging address for the user.
`xmppIm`	The XMPP instant messaging address for the user.
`entitlements`	A list of entitlements for the user that represent a thing the user has. An entitlement could be an additional right to a thing, object, or service.
`roles`	A list of roles for the user that collectively represent who the user is. For example, `Student` and `Faculty`.
`certificates`	A list of certificates associated with the resource (for example, a user). Each value contains exactly one DER-encoded X.509 certificate (refer to Section 4 of RFC5280), which MUST be base64 encoded per Section 4 of RFC4648.
`employeeNumber`	A string identifier, typically numeric or alphanumeric, assigned to a person, often based on order of hire or association with an organization.
`costCenter`	The cost center for the user.
`organization`	The organization for the user.
`division`	The division for the user.
`department`	The department for the user.

Copyright © 2025 Ping Identity Corporation

Attribute	Description
`manager`	The `id` of the SCIM resource representing the user's manager.

Configuring single sign-on

To allow PingFederate to act as an identity provider for the target service, modify your connection to include single sign-on (SSO).

About this task

> **Note**
>
> SSO is not required for provisioning. If you only want to use the SCIM Provisioner for provisioning, skip these steps.

Steps

1. Complete the steps in Adding SSO to a connection.
2. On the target service, configure SSO by following the documentation provided by the service. When prompted for information about PingFederate, refer to PingFederate SSO details for the service provider.

Adding SSO to a connection

To enable single sign-on (SSO), modify the provisioning connection that you created in Creating a provisioning connection.

Steps

1. On the PingFederate administrative console, open your existing SP connection.

   Choose from:

   - For PingFederate 10.1 or later: go to Applications > Integration > SP Connections. Select your connection.
   - For PingFederate 10.0 or earlier: go to Identity Provider > SP Connections. Select your connection.

2. On the Connection Type tab select Browser SSO Profiles. Click Next.
3. On the Browser SSO tab, configure your SSO settings.

   You can find specific connection details in the documentation and administrative console provided by the target service.

   You can find configuration help in Configuring IdP Browser SSO in the PingFederate documentation.

4. On the Activation and Summary tab, above the Summary section, click the toggle to turn on the connection. Click Save.

Copyright © 2025 Ping Identity Corporation

PingFederate SSO details for the service provider

When enabling single sign-on (SSO) in the target service, you will require some or all of the following information from PingFederate.

Metadata file

Some target services allow you to import a Security Assertion Markup Language (SAML) metadata file that contains some of the following information. Learn more about exporting your metadata file in Metadata export in the PingFederate documentation.

SAML endpoint

The PingFederate SAML endpoint is:

```
https://<pf_hostname>:<pf_port>/idp/SSO.saml2
```

Identity provider issuer

This is SAML 2.0 Entity ID from PingFederate, which can be found on the Server Settings page. Learn more in Specifying federation information.

> **Tip**
>
> To override SAML 2.0 Entity ID on the **Server Settings** page for your SP Connection, go to the **General Info** page to add a Virtual Server ID. This value will be sent as the SAML Issuer URL.

Signing certificate

This is the public signing certificate that PingFederate uses to sign the SAML assertion. Learn more about exporting your certificate in Managing digital signing certificates and decryption keys.

Release notes

Changelog

The following is the change history for the SCIM Provisioner. Updated June 3, 2024.

SCIM Provisioner 1.6 – June 2024

- Added the ability to request specific OAuth scopes to the OAuth 2 client credentials authentication method. Learn more in SCIM provisioner settings reference.
- Fixed an issue that caused a JSON parsing error if non-standard fields were present in the SCIM 2.0 `Enterprise User Schema Extension`.

Copyright © 2025 Ping Identity Corporation

SCIM Provisioner 1.5.2 – September 2023

- Fixed an issue with `type` sub-attribute case sensitivity that caused a `NullPointerException` error. `Type` sub-attribute values for phone numbers and photos are now case insensitive.

SCIM Provisioner 1.5 – December 2022

- Added support for user custom attributes for SCIM 2.0.
- Added `homeEmail` and `otherEmail` attributes.

SCIM Provisioner 1.4.2 – August 2021

- Fixed an issue that caused the Check Connection function to fail when correct credentials were provided.
- Fixed an issue that caused a `NullPointerException` error when the `photo` attribute was empty.

SCIM Provisioner 1.4 – December 2020

- Added support for the PATCH method for group updates. Learn more in the SCIM provisioner settings reference.
- Improved backward compatibility for SCIM 1.1 headers.
- Fixed an issue that could cause the provisioner to parse groups incorrectly.

SCIM Provisioner 1.3 – March 2020

- Added support for the `application/scim+json` HTTP header type.
- Improved support for services that limit the number of groups that can be retrieved per request by adding a Results Per Page field in the connection configuration.
- Improved the SCIM URL field in the connection configuration to work regardless of whether a trailing slash (`/`) is included in the URL.
- Fixed an issue that caused connection tests to return a success when the wrong credentials were used.
- Fixed an issue that caused groups to be serialized incorrectly.

SCIM Provisioner 1.2 – March 2019

- Added configuration options for the Unique User Identifier.
- Improved error handling and reporting for cases where users in the target application do not have an ID.

SCIM Provisioner 1.1 – July 2018

- Added the option to provision groups with common name (CN) or distinguished name (DN).
- Fixed an issue where SCIM v2 requests included SCIM v1.1 schema URN's.
- Fixed a compatibility issue when multiple SCIM SaaS connectors are deployed.
- Fixed an issue where the NO_CONTENT HTTP response code was not being handled.
- Fixed an issue where the SERVER_ERROR HTTP response code was not being handled.

Copyright © 2025 Ping Identity Corporation

- Fixed an issue where the user's active status was not updated correctly on update requests.
- Fixed an issue where SCIM v2 errors descriptions were not logged correctly.
- Fixed an issue where an empty return body on a user PUT would cause a JSON parse exception.

SCIM Provisioner 1.0 – October 2017

- Initial release.
- Added support for user provisioning.
- Added support for group provisioning.
- Added support for adding users to groups.
- Added support for SCIM core and enterprise attributes.
- Added support for basic authentication, OAuth 2 bearer token and OAuth 2 client credentials authentication.
- Added support for SCIM overrides (Filter expression, authorization header type, users API path, groups API path).
- Added configuration options for CRUD capabilities.
- Added configuration options for provisioning disabled users.
- Added configuration options for deprovisioning actions.

Known issues and limitations

The following are known issues or limitations with the SCIM Provisioner.

Known issues

There are no known issues.

Known limitations

Service provider (SP) connections:

- The Unique User Identifier cannot be changed in an SP connection configuration. To change to a different Unique User Identifier, delete the existing connection, restart PingFederate, and then create a connection with the new Unique User Identitier.
- All SP connections with the same target must use the same Unique User Identifier. If multiple SP connections are created for the same target, every subsequent connection will use the Unique User Identifier configured in the first connection that was created.

Attributes:

- The connector has a limit of one value per type (home, work, other, and so on) for multi-value attributes (email, phone, address).

Copyright © 2025 Ping Identity Corporation

- If the SaaS does not specify type or primary information on multi-value attributes (email, phone, address), unexpected behavior can occur. During an update, existing attributes on the SaaS may not be removed, and the desired value may not be correctly set as primary.
- The connector cannot clear a user attribute once it has been set.
- If the target application supports two email attributes and one attribute is empty, the connector populates both attributes with the email address and sets both as "primary". This can produce unexpected effects in some target applications.

Group provisioning:

- When provisioning groups, only required attributes, such as `displayName` and `members`, are supported. Common attributes, such as `id` and `externalID`, are not supported because they're optional attributes.

Other:

- When an LDAP user is deleted in a targeted group distinguished name (DN), the provisioning connector doesn't propagate the deletion until a new user is added to the group. This limitation is compounded when the User Create provisioning option is disabled. You can find solutions in SaaS provisioner does not remove the user in the Knowledge Base.
- SCIM-compliant service providers can implement or interpret the SCIM standards differently, which could result in behavior that's not consistent with the SCIM Provisioner intended use.

Download manifest

The following files are included in the SCIM Provisioner `.zip` archive:

- `Legal.pdf`: Copyright and license information.
- `dist`: Contains the integration files.
  - `pf-scim-quickconnection-<version>.jar`: The SCIM Provisioner quick connection template.

Copyright © 2025 Ping Identity Corporation

ServiceNow Provisioner

The PingFederate ServiceNow Provisioner allows PingFederate to integrate with ServiceNow for user provisioning and single sign-on (SSO).

Features

- Manages users in ServiceNow based on changes in an external data store that is attached to PingFederate.
  - Creates, updates, and disables users.
  - Allows you to enable the create, update, and disable capabilities independently.
  - Allows you to provision disabled users.
- Browser-based single sign-on (SSO) and single logout (SLO) initiated by the service provider (SP) or identity provider (IdP).
- Pre-populates some connection settings with the included quick connection template and SAML metadata file.

Intended audience

This document is intended for PingFederate administrators.

Before you start, you should be familiar with the following:

- The following sections of the PingFederate documentation:
  - Identity provider SSO configuration
  - Datastores
  - Managing digital signing certificates and decryption keys
  - SP connection management
  - Configuring outbound provisioning
  - Configuring outbound provisioning settings
- The following sections of the ServiceNow documentation:
  - User administration
  - External single sign-on (SSO)
- Known issues and limitations for this integration.

System requirements

- PingFederate 9.0 or later.

Copyright © 2025 Ping Identity Corporation

- To allow PingFederate to make outbound connections to the ServiceNow API, you might need to whitelist the following domain in your firewall:
    - https://*yourinstance*.service-now.com
    - https://*yourinstance*.servicenowservices.com
- Administrator access to a ServiceNow instance running one of the following versions:
    - Vancouver
    - Utah
    - Tokyo
    - San Diego
    - Rome
    - Quebec
    - Paris
    - Orlando
    - New York
    - Madrid
    - London

User management

The ServiceNow Provisioner links users from the data store to ServiceNow. The behavior of each provisioning capability is described below.

Synchronizing existing users

By default, the ServiceNow Provisioner synchronizes users from the data store to the target service by matching an attribute in the data store to the `Username` attribute in the target service.

For example:

- In ServiceNow, Janet's `Username` is `jsmith`.
- In your data store, Janet's `sAMAccountName` is `jsmith`.
- On the Attribute Mapping tab, you map the `Username` attribute to `sAMAccountName`.
- When the provisioning connector runs, the data store user is provisioned with a `Username` of `jsmith`. That matches Janet's existing `Username` in ServiceNow, so her information in the data store is synchronized to her ServiceNow account.

You can map the matching data store attribute when you configure your channel in Creating a provisioning connection.

Copyright © 2025 Ping Identity Corporation

User provisioning

Triggered by any of the following:

- A user is added to the data store group or filter that is targeted by the provisioning connector.
- A user with "disabled" status is added to the data store group or filter that is targeted by the provisioning connector, and the Provision disabled users provisioning option is enabled.

The target is determined by the Source Location tab in the provisioning connector configuration.

User updates

Triggered when a change occurs to a user attribute that is mapped in the provisioning connector configuration.

User deprovisioning (disabling)

Triggered by any of the following:

- A user is deleted from the user store.
- A user is disabled in the user store.
- A user is removed from the data store group or filter that is targeted by the provisioning connector.

Setup

Upgrading an existing deployment

If you are upgrading from a previous version of the ServiceNow Provisioner, note your existing service provider (SP) connection configuration and create a new connection.

Steps

1. Back up your current PingFederate configuration as shown in Configuration archive in the PingFederate documentation.
2. In the PingFederate administrative console, open your existing SP connection.
    1. For PingFederate 10.1 or later: go to Applications > Integration > SP Connections and select your connection.
    2. For PingFederate 10.0 or earlier: go to Identity Provider > SP Connections and select your connection.
3. Note the attribute mappings for your existing SP connection. For help, see Mapping attributes in the PingFederate documentation.
4. Delete your existing SP connection.
    1. For PingFederate 10.1 or later:
        1. Go to Applications > Integration > SP Connections.

Copyright © 2025 Ping Identity Corporation

2. For your existing connection, click Select action, and then click Delete. Click Confirm.

2. For PingFederate 10.0 or earlier:

    1. Go to Identity Provider > SP Connections > Manage All.

    2. For your existing connection, click Select action, and then click Delete. Click Save.

5. Complete the steps in Deploying the integration files.

6. Complete the steps in Creating a provisioning connection.

    - From Outbound Provisioning > Manage Channels > Channel, on the Attribute Mapping tab, configure the attribute mappings based on your notes.

Deploying the integration files

To get started with the integration, deploy the ServiceNow Provisioner files to your PingFederate directory.

Steps

1. Download the ServiceNow Provisioner `.zip` archive from the Ping Identity Integration Directory.

2. Stop PingFederate.

3. If you're upgrading an existing deployment, delete the `pf-servicenow-quickconnection-<version>.jar` file from your `<pf_install>/pingfederate/server/default/deploy` directory.

4. Extract the `.zip` archive and merge the contents of the `dist` directory with your `<pf_install>/pingfederate/server/default/deploy` directory.

5. Enable the PingFederate provisioning engine: Open your `<pf_install>/pingfederate/bin/run.properties` file.Change `pf.provisioner.mode` to `STANDALONE`. Save the file.

    > **Note**
    >
    > To configure the `FAILOVER` mode instead, see Deploying provisioning failover in the PingFederate Server Clustering Guide.

6. Start PingFederate.

7. If you operate PingFederate in a cluster, repeat steps 2-4 and step 6 for each engine node.

Configuring provisioning

To allow PingFederate to manage users in ServiceNow, create a user to represent PingFederate in ServiceNow, and then create a service provider connection.

Steps

1. In your ServiceNow instance, check that every user, even those not targeted for provisioning, has a `Username` attribute set.

    The connector uses this attribute for synchronization with the data store.

Copyright © 2025 Ping Identity Corporation

2. If you want PingFederate to manage the departments or locations assigned to users in ServiceNow, merge any duplicate `department` or `location` objects with different capitalization, such as `Accounting` and `accounting`.

   For more information, see Known issues and limitations.

3. Complete the steps in Adding the Ping Identity provisioning role in ServiceNow.
4. Complete the steps in Creating a provisioning user in ServiceNow.
5. Complete the steps in Creating a provisioning connection.

Adding the Ping Identity provisioning role in ServiceNow

To allow PingFederate to manage users in ServiceNow, add the special Ping Identity provisioning role in your ServiceNow instance.

Steps

1. Sign on to https://*yourinstance*.servicenow.com as an administrator.
2. Install the Ping Identity Provisioning Solution app.

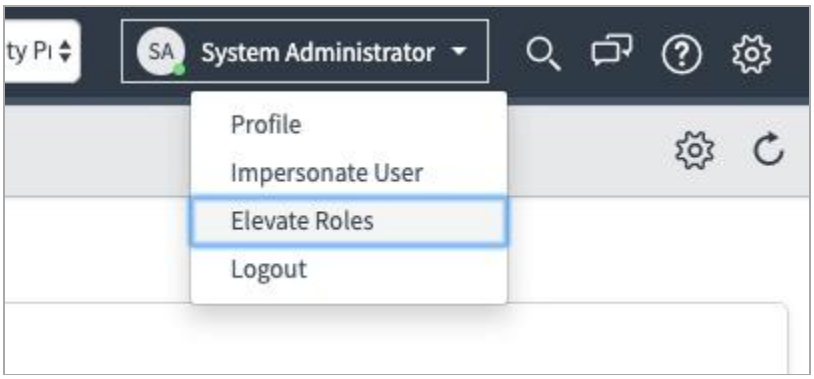
   The app contains the provisioning role.

   1. In the ServiceNow admin console, navigate to All Available Applications.
   2. Search for and select the Ping Identity Provisioning Solution app.
   3. Click Install.

3. Optional: If you want the ServiceNow connector to be able to remove roles from users, grant additional permissions to the provisioning role.

   > **Caution**
   >
   > Due to a limitation in the ServiceNow API, the ServiceNow Provisioner requires additional security permissions to be able to remove roles from users. We recommend that you only grant these permissions if you require the role removal functionality. Otherwise, skip these steps. For more details, see Known issues and limitations.

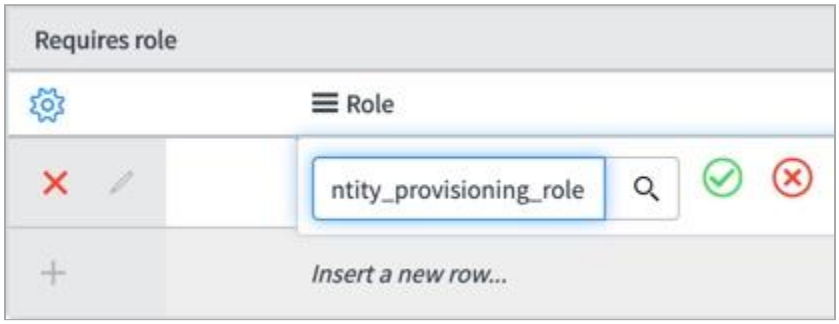
   1. In the upper-right corner, click your administrator account name. Click Elevate Roles.

   2. On the Elevate Roles dialog, select security_admin. Click OK.

Copyright © 2025 Ping Identity Corporation

3. Go to System Security > Access Control (ACL). Click New.
4. On the New record tab, from the Operation list, select Delete.
5. From the Name list, select User Role (sys_user_has_role).
6. In the Requires role section, double-click Insert a new row.
7. Enter `ping_identity_provisioning_role`, and then press enter.



8. Click Submit.

Creating a provisioning user in ServiceNow

To allow PingFederate to connect to ServiceNow, create a provisioning user in ServiceNow.

Steps

1. In your ServiceNow instance, create the user credentials that PingFederate will use to connect.
    1. Go to User Administration > Users. Click New.
    2. In the New Record window, in the User ID field, enter `ping_identity_provisioning_solution_user`.
    3. In the Password field, enter a password of your choosing.
    4. Select the Web service access only and Internal integration user check boxes.
    5. Click Submit.
2. Assign the provisioning role to the user.
    1. In your ServiceNow instance, go to User Administration > Users and click the newly created user.
    2. On the Roles tab, click Edit.
    3. In the Collection field, search for the `x_pir_ping_prov.ping_identity_provisioning_role` role.
    4. Click the right arrow to add it to the Roles list.
    5. Click Save.

Copyright © 2025 Ping Identity Corporation

Enabling provisioning in PingFederate

To use PingFederate for provisioning, configure an external datastore.

About this task

Your external datastore acts as the source of data for provisioning. PingFederate also uses an internal datastore to store the state of synchronization between the source datastore and the target datastore.

For more information, see Datastores in the PingFederate documentation.

Steps

1. Configure the datastore for PingFederate to use as the source of user data.

   For instructions, see Datastores in the PingFederate documentation.

   > **Tip**
   >
   > When targeting users and groups for provisioning, exclude the user account that you will use to administer users in your connection to ServiceNow. This prevents the PingFederate provisioning engine from interfering with the account that provisions users and groups.

2. Do one of the following:

   Choose from:

   - For PingFederate 10.1 or later: Go to System > Server > Protocol Settings.
   - For PingFederate 10.0 or earlier: Enable the identity provider (IdP) and outbound provisioning roles:
     1. Go to System > Protocol Settings > Roles & Protocols.
     2. Select Enable Identity Provider IdP Role and Support the Following.
     3. Select Outbound Provisioning. Click Next.

3. On the Outbound Provisioning tab, select the PingFederate internal datastore. Click Save.

   For help, see Configuring outbound provisioning settings in the PingFederate documentation.

Creating a provisioning connection

To allow PingFederate to manage users in ServiceNow, configure a service provider (SP) connection.

Steps

1. On the Identity Provider screen, in the SP Connections area, open an existing connection or create a new one as follows:
   1. Click Create new.
   2. On the Connection Template screen, select Use a template for this connection.
   3. In the Connection Template list, select ServiceNow Connector.

Copyright © 2025 Ping Identity Corporation

4. Click Choose File, select the `servicenow-saml-metadata.xml` file from the ServiceNow Provisioner `.zip` archive, and then click Open. Click Next.

2. On the Connection Type screen, select Outbound Provisioning and clear any unwanted types. Click Next.
3. On the General Info screen, in the Partner's Entity ID, Connection Name, and Base URL fields, change *yourinstance* to your ServiceNow instance name. Click Next.
4. On the Outbound Provisioning screen, configure the provisioning target and channel.

   See Configuring outbound provisioning in the PingFederate documentation.

   1. Click Configure Provisioning.
   2. On the Target screen, in the Administrator Username and Administrator Password fields, enter the credentials for the user that you created in Creating a provisioning user in ServiceNow.

      > **Note**
      >
      > PingFederate verifies the credentials when you activate the channel and SP connection.

   3. In the ServiceNow URL field, enter the URL of your ServiceNow instance. For example, `https://yourinstance.service-now.com`.
   4. Under Provisioning Options, customize the provisioning connector actions as shown in Provisioning options reference. Click Next.
   5. On the Manage Channels screen, create a channel. Click Done.

   See Managing channels in the PingFederate documentation.

   1. On the Outbound Provisioning screen, click Next.
5. On the Activation and Summary screen, above the Summary section, turn on the connection. Click Save.

Provisioning options reference

The following are setting descriptions for the ServiceNow provisioning capabilities.

Field Name	Description
User Create	***Selected (default)*** PingFederate creates users in ServiceNow. ***Cleared*** PingFederate does not create users in ServiceNow.

Copyright © 2025 Ping Identity Corporation

Field Name	Description
User Update	***Selected (default)*** PingFederate updates existing users in ServiceNow. ***Cleared*** PingFederate does not update existing users in ServiceNow.
User Disable	***Selected (default)*** PingFederate disables users in ServiceNow. ***Cleared*** PingFederate does not disable users in ServiceNow.
	Note If any of the above options are cleared, PingFederate logs a warning in the user workflow section of the `provisioner.log` file when the related action fails.
Provision Disabled Users	***Selected*** PingFederate creates users in ServiceNow with a "disabled" status. ***Cleared (default)*** If a user has a "disable" status, PingFederate does not create the user in ServiceNow.

Supported attributes reference

The following attributes can be mapped for user provisioning to ServiceNow.

For more information about these attributes, see Create a user in the ServiceNow documentation.

Attribute	Description
UserName	The user's unique identifier in ServiceNow. This attribute maps to the user's `user_id` in ServiceNow. The ServiceNow Provisioner uses this value to synchronize pre-existing users in ServiceNow with the data store. **Note** This user attribute is required for provisioning. It must be unique and is not cannot be updated. For more information, see Known issues and limitations.

Copyright © 2025 Ping Identity Corporation

Attribute	Description
First name	The user’s given name. For example, "Barbara" in "Ms. Barbara Jane Jensen, III".
Last name	The user’s family name. For example, "Jensen" given the full name "Ms. Barbara Jane Jensen, III".
Middle name	The user’s middle name or names. For example, "Jane" given the full name "Ms. Barbara Jane Jensen, III".
Prefix	The user’s honorific prefix or prefixes, such as "Ms.", "Mr.", or "Dr." This attribute maps to the user’s "introduction" in ServiceNow.
Gender	The user’s gender.
Email	The user’s email address.
Title	The user’s business title, such as "Software engineer".
Department	The user’s department or work group, such as "Sales".
Password	The user’s password. Passwords cannot be mapped from a source data store, so this field can be used to set a literal default value.
Password Needs Reset	When set to true, the user must change their password at next login.
Language	The user’s language. Uses the two-character language code as defined by the ISO 639-1 standard. This attribute has a default value in ServiceNow. Languages might need to be updated in ServiceNow to account for other allowable values for this field.
Location	The user’s geographical location.
Photo	The user’s photo.
Business Phone	The user’s business phone number.
Mobile Phone	The user’s mobile phone number.
Home Phone	The user’s home phone number.
Street	The user’s street address.
City	The user’s city or locale.
State/Province	The user’s state, province, or territory.
Zip/Postal Code	The user’s ZIP or postal code.

Copyright © 2025 Ping Identity Corporation

Attribute	Description
Country Code	The user's country. Uses the two-character country code as defined by the ISO-3166-1 alpha-2 standard. This attribute has a default value in ServiceNow. County Codes might need to be updated in ServiceNow to account for other allowable values for this field.
Roles	The user's role, such as "Student" or "Faculty". This attribute supports multiple values. The ServiceNow Provisioner can only assign roles that are already assigned to the user account that you use for provisioning. The ServiceNow Provisioner does not create new roles, so the roles must already exist in ServiceNow. By default, the ServiceNow Provisioner can set roles but not remove them. To remove this limitation, you can grant elevated security permissions, as shown in Adding the Ping Identity provisioning role in ServiceNow.
Notification	Determines whether to enable notifications. Select Enable or Disable.
Employee Number	A string identifier, typically numeric or alphanumeric, assigned to a person, often based on order of hire or association with an organization.
Time Zone	The user's time zone. Uses IANA time zone database format. For example, "America/Los_Angeles". **Note** The ServiceNow Provisioner can only set time zones that are already assigned to the user account that you use for provisioning. The time zone might need to be updated in ServiceNow to account for other allowable values in this field.
Time Format	The user's time format. For details on time formatting, see Global date and time field format in the ServiceNow documentation. **Note** The ServiceNow Provisioner can only set time formats that are already assigned to the user account that you use for provisioning.
Manager	The distinguished name (DN) representing the manager for the user. **Note** During the first synchronization between ServiceNow and your data store, you might see errors in `provisioner.log` because the manager for a user has not been provisioned yet. The user is still created without the manager attribute. On the next retry, the manager will exist and the user's `manager` attribute will be updated. **Important** This attribute is not supported in PingFederate 9.0 and 9.0.1.

Copyright © 2025 Ping Identity Corporation

Configuring single sign-on

To allow PingFederate to act as an identity provider for ServiceNow, configure SSO, exchange signing certificates, and create a connection.

About this task

> **Note**
>
> Single sign-on integration is an optional part of this integration. If you only want to use the ServiceNow Provisioner for provisioning, skip these steps.

Steps

1. **In ServiceNow, install the SSO integration plugin.**

   > **Note**
   >
   > For official documentation, see Activate and set up SAML 2.0 in the ServiceNow documentation.

   1. **In your ServiceNow instance, go to System Definition > Plugins.**
   2. **Click Integration - Multiple Provider Single Sign-On Installer.**
   3. **On the Integration - Multiple Provider Single Sign-On Installer detail screen, in the Related Links section, click Activate/Repair.**
   4. **On the Activate Plugin dialog, click Activate.**
   5. **After the installation finishes, click Close & Reload Form.**

2. **Complete the steps in Registering PingFederate as an identity provider in ServiceNow.**
3. **Complete the steps in Exchanging signing certificates.**
4. **Complete the steps in Creating a single sign-on connection.**

Registering PingFederate as an identity provider in ServiceNow

To allow PingFederate to coordinate authentication for ServiceNow, configure the SAML 2.0 properties.

About this task

For official documentation, see External single sign-on (SSO) in the ServiceNow documentation.

Steps

1. **In your ServiceNow instance, go to SAML 2 Single Sign-on > Properties.**
2. **On the SAML 2.0 Single Sign-on properties screen, select Enable external authentication.**
3. **In the Identity Provider properties section, in the Identity Provider URL field, enter your PingFederate URL based on the following:**

   https://*pf_host*:*pf_port*

Copyright © 2025 Ping Identity Corporation

4. In the base URL to the Identity Provider's AuthnRequest service field, enter your PingFederate SSO endpoint based on the following:

   https://*pf_host*:*pf_port*/idp/SSO.saml2

5. Optional: Select Sign AuthnRequest.
6. In the base URL to the Identity Provider's SingleLogOutRequest service field, enter your PingFederate SSO endpoint based on the following:

   https://*pf_host*:*pf_port*/idp/SLO.saml2

7. In the protocol binding for the Identity Provider's SIngleLogoutRequest service field, enter `urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST`.
8. Select Sign LogoutRequest.
9. In the Service Provider (ServiceNow) properties section, update the instance homepage, entity identification, and audience uri fields to point your ServiceNow instance.
10. In the User table field, enter `user_name`.
11. In the NameID policy field, enter `urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified`.
12. Click Save.

Exchanging signing certificates

To allow PingFederate and ServiceNow to communicate securely, exchange the signing certificates between the two systems.

Steps

1. In PingFederate, export your signing certificate. On the PingFederate admin console, go to Security > Signing & Decryption Keys & Certificates.For the certificate that you want to use, in the Action column, click Export.On the Export Certificate screen, click Next.On the Export & Summary screen, click Export.Open the `*.crt` file in a text editor.

2. In ServiceNow, import your PingFederate signing certificate.
   1. On your ServiceNow instance, go to SAML 2 Single Sign-on > Certificate. Click New.
   2. On the New record screen, in the Name field, enter `SAML 2.0`.
   3. Optional: In the Short description field, enter a description. This appears on the Certificate screen.
   4. In the PEM Certificate field, paste the contents of the `*.crt` file that you exported from PingFederate.
   5. Click Submit.
3. In ServiceNow, export your ServiceNow single logout (SLO) certificate.
   1. On your ServiceNow instance, go to SAML 2 Single Sign-on > Certificate. Click SAML 2.0 SP Keystore.
   2. On the SAML 2.0 SP Keystore screen, download the certificate keystore by clicking saml2sp_keystore.

Copyright © 2025 Ping Identity Corporation

3. Extract the certificate from `saml2sp_keystore` as shown in How to print the Public Key of a Certificate using Keytool in the ServiceNow documentation.
4. Copy the output of the command to a text file on your computer, and save it as `sn-certificate.crt`.

4. In PingFederate, import your ServiceNow SLO certificate as a trusted certificate authority (CA).
    1. On the PingFederate admin console, go to Security > Trusted CAs.
    2. On the Trusted CAs screen, click Import.
    3. On the Import Certificate screen, select `sn-certificate.crt`. Click Next.
    4. On the Summary screen, click Save.

5. In ServiceNow, export your SAML 2.0 metadata.
    1. On your ServiceNow instance, go to SAML 2 Single Sign-on > Metadata.
    2. Copy the metadata block to a text file on your computer, and save it as `sn-metadata.xml`.

You will use this in Creating a single sign-on connection.

Creating a single sign-on connection

To allow PingFederate to handle single sign-on (SSO) to ServiceNow, create a service provider (SP) connection.

About this task

> **Note**
> You can follow these steps to create a new SP connection, or you can modify your provisioning connection.

Steps

1. In the PingFederate administrator console, configure an SP connection.
    1. On the Identity Provider tab, in the SP Connections area, click Create new.
    2. On the Connection Template tab, select Use a template for this connection.
    3. In the Connection Template list, select ServiceNow Connector.
    4. Click Choose File, select the `sn-metadata.xml` file that you exported in Exchanging signing certificates, and then click Open. Click Next.
2. On the Connection Type tab, select Browser SSO Profiles. If you don't want provisioning, clear Outbound Provisioning. Click Next.
3. On the Connection Options tab, click Next.
4. On the General Info tab, in the Partner's Entity ID, Connection Name, and Base URL fields, change *yourinstance* to your ServiceNow instance name. Click Next.

Copyright © 2025 Ping Identity Corporation

5. On the Browser SSO tab, configure browser SSO.

   For a complete guide, see Configure IdP Browser SSO in the PingFederate documentation.

   1. On the Browser SSO > SAML Profiles tab, select only IdP-Initiated SSO and SP-Initiated SSO.
   2. On the Browser SSO > Protocol Settings > Allowable SAML Bindings tab, select only POST and Redirect.

6. On the Credentials tab, configure the connection credentials.

   For a complete guide, see Configuring credentials in the PingFederate documentation.

   1. On the Credentials > Digital Signature Settings tab, from the Signing Certificate list, select the certificate that you want to use with ServiceNow.
   2. Select Include the certificate in the signature <KEYINFO> element. Click Done.

7. On the Activation and Summary tab, above the Summary section, click the toggle button to enable the connection. Click Save.

Release notes

Changelog

The following is the change history for the ServiceNow Provisioner.

ServiceNow Provisioner 2.3 – November 2023

- Certified with the Utah and Vancouver versions of ServiceNow.

ServiceNow Provisioner 2.3 – April 2023

- Certified with the Tokyo version of ServiceNow.

ServiceNow Provisioner 2.3 – August 2022

- Certified with the Rome and San Diego versions of ServiceNow.

ServiceNow Provisioner 2.3 – May 2021

- Certified with the Quebec version of ServiceNow.
- Added support for mapping users to locations in ServiceNow.

ServiceNow Provisioner 2.2 – November 2020

- Certified with the Paris version of ServiceNow.
- No software changes.

Copyright © 2025 Ping Identity Corporation

ServiceNow Provisioner 2.2 – June 2020

- Added the ServiceNow URL field and removed the ServiceNow Instance Name field.
- Fixed an issue that caused an error when assigning a role when the provisioning user account did not also have that role.
- Added support for the Orlando version of ServiceNow.

ServiceNow Provisioner 2.1 – November 2019

- Added support for the London, Madrid, and New York versions of ServiceNow.
- Added support for mapping users to departments in ServiceNow.
- Improved user ID validation when updating and deleting users.
- Removed support for the Jakarta and Instanbul versions of ServiceNow.

ServiceNow Provisioner 2.0 – May 2018

- Added support for Istanbul, Jakarta, and Kingston.
- Added configuration options for CRUD capabilities.
- Added configuration options for provisioning disabled users.

ServiceNow Provisioner 1.0.1 – June 2015

- Fixed compatibility issues.

ServiceNow Provisioner 1.0 – January 2015

- Initial release.
- Added support for browser-based SP and IdP-initiated SSO.
- Added support for outbound user provisioning.

Known issues and limitations

The following are known issues or limitations with the ServiceNow Connector.

Known issues

- Avoid naming `department` and `location` objects with the same format as ServiceNow system IDs, such as `972456281bcc3010dbb0dd7ebd4bcbcc`. This might cause users to be provisioned with an incorrect department or location.

Known limitations

- Attributes
  - Due to limitations in PingFederate, user attributes cannot be cleared after they are set.

Copyright © 2025 Ping Identity Corporation

- Due to a limitation with PingFederate 9.0 and 9.0.1, the `manager` attribute cannot be mapped and used. There is no impact to other functionality.
- When using the manager attribute, if the user's manager is not managed by the provisioner, the user will be created with an exception thrown. PingFederate will continue to retry the user indefinitely.
- User role values must contain only URL-safe characters.
- Due to a limitation in the ServiceNow API, the ServiceNow Connector requires additional security permissions to be able to remove roles from users. If you try to remove a role from a user without granting these permissions, your data store will become out of sync with ServiceNow, and an error will appear in `provisioner.log`. You can grant the permissions as shown in Adding the Ping Identity provisioning role in ServiceNow.

• Provisioning

- PingFederate uses the `Username` attribute to synchronize users between the data store and ServiceNow, as described in User management. When attempting to provisioning a user that does not have a `Username`, PingFederate logs an error in provisioner.log and keeps trying until the `Username` is present.
- When provisioning users, the username attribute must only contain URL-safe characters.
- If a new user is created with the same username as an existing user, a duplicate user will not be created. Instead, the existing user will be updated with any information in the create.
- For `department` objects that contain the `^` character in the name, the ServiceNow API causes the creation of multiple departments with the same name.
- For the `department` and `location` objects, the ServiceNow API ignores capitalization. When provisioning a user that matches multiple departments or locations in ServiceNow (such as `Accounting` and `accounting`), PingFederate provisions the user with an empty `department` or `location` attribute and logs an error in provisioner.log.

• Deprovisioning

- When an LDAP user is deleted in a targeted group distinguished name (DN), the provisioning connector does not propagate the deletion until a new user is added to the group. This limitation is compounded when the User Create provisioning option is disabled. For solutions, see SaaS provisioner does not remove the user in the Knowledge Base.

• Performance

- When synchronizing user roles, PingFederate performs multiple calls to the ServiceNow API. This can impact provisioning performance.
- When using the default mapping for the manager attribute, the process for adding a manager involves an Active Directory search, followed by a database lookup to get the ServiceNow manager ID. This can impact provisioning performance. To improve performance, you can use a custom attribute mapping to link the manager attribute to a manager's email.

• Configuration

- When using multiple channels, the same `Username` mapping is required to coordinate manager assignments across different channels.

Copyright © 2025 Ping Identity Corporation

Download manifest

The following files are included in the ServiceNow Provisioner `.zip` archive:

- `legal/Legal.pdf` – copyright and license information.
- `dist` – contains the integration files
  - `pf-servicenow-quickconnection-<version>.jar` – The ServiceNow Provisioner quick connection template.
- `servicenow-saml-metadata.xml` – a modifiable metadata file used to create the connection.

Copyright © 2025 Ping Identity Corporation

SharePoint People Picker Integration Kit

The People Picker control is a central component in Microsoft SharePoint Server that is used to search and select users and groups when a resource owner assigns permissions. By default, when a SharePoint web application is configured to use SAML token-based authentication or "SAML claims", all queries entered in the People Picker are automatically displayed as if they had been resolved, regardless of whether they are valid users or groups. This poses a significant usability problem for SharePoint users and administrators, particularly in collaboration-oriented deployments where potentially every user has the ability to edit file and library permissions using the People Picker.

To address this issue, Microsoft recommends you build a custom claims provider to provide capabilities for custom search and name resolution. The Ping Identity Custom Claims Provider for SharePoint is an implementation capable of connecting to one or more LDAP user stores or domains to fulfill search and name resolution queries.

When this custom claims provider is associated with a Trusted Login Provider (referred to in the Management Shell as a Trusted Identity Token Issuer) in SharePoint, such as might be configured to accept inbound SAML claims from PingFederate for one or more SharePoint web applications, the People Picker will provide functionality similar to that seen with classic-mode authentication where users and groups in Active Directory are available for search, name resolution, and attribute listing.

System requirements

This version of the Ping Identity Custom Claims Provider for SharePoint requires the following:

- Microsoft SharePoint Server 2013, 2016, or 2019. Specifically, an environment that is already configured with PingFederate as the Trusted Identity Token Issuer.
- Connectivity to one or more backing LDAP user stores. Testing has been performed on Microsoft Active Directory 2016.

Assumptions and prerequisites

This document assumes:

- A Trusted Identity Token Issuer, referred to hereon as the Partner STS, has been created and enabled as an Authentication Provider for at least one SharePoint web application.
- SAML token-based authentication has been successfully tested for the SharePoint web application using the Partner STS.
- The Partner STS is configured to send the user identity claim type that will be used by SharePoint (for example, http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress).

> **Tip**
>
> Make note of the name of the Partner STS. You will need it when associating the Custom Claims Provider with the Partner STS. You can view all Trusted Identity Token Issuers and their names by executing the following command from the Management Shell: *Get-SPTrustedIdentityTokenIssuer* and looking at the *Name* attribute.

- Connectivity and trust (for LDAPS connections) exists to all domain controllers that are to be searched.
- A provisioned service account with read access for each domain controller to be searched is available.

Copyright © 2025 Ping Identity Corporation

- The SharePoint Administration service must be running prior to installing the solution (.wsp) file.

Setup

Installation and setup

Use the following sections for instructions to install the solution.

- Upgrade from a previous version
- Add and deploy the solution
- Activate features
- Set the default claims provider

Upgrade from a previous version

Use Update-SPSolution in a Sharepoint Management Shell to upgrade to a newer version:

```
Update-SPSolution -Identity pingidentity.sharepoint.ppclaimsprovider.wsp
     -LiteralPath "C:\path_to_solution\PingIdentity.SharePoint.PPClaimsProvider.wsp"
     –GacDeployment
```

> **Note**
>
> The remaining steps of the Installation and Setup section are not required when upgrading.

Add and deploy the solution

Steps

1. Extract the SharePoint People Picker Integration Kit distribution file, and then go to `dist/sharepoint<yourversion>`.
2. Copy the solution (WSP) file to the SharePoint server.
3. Open the SharePoint Management Shell (PowerShell) as Administrator and use the `Add- SPSolution` command to add the solution to the farm:

   ```
   Add-SPSolution -LiteralPath
   C:\path\PingIdentity.SharePoint.PPClaimsProvider.wsp
   ```

4. Deploy the solution using Central Administrator using the `Install-SPSolution` command:

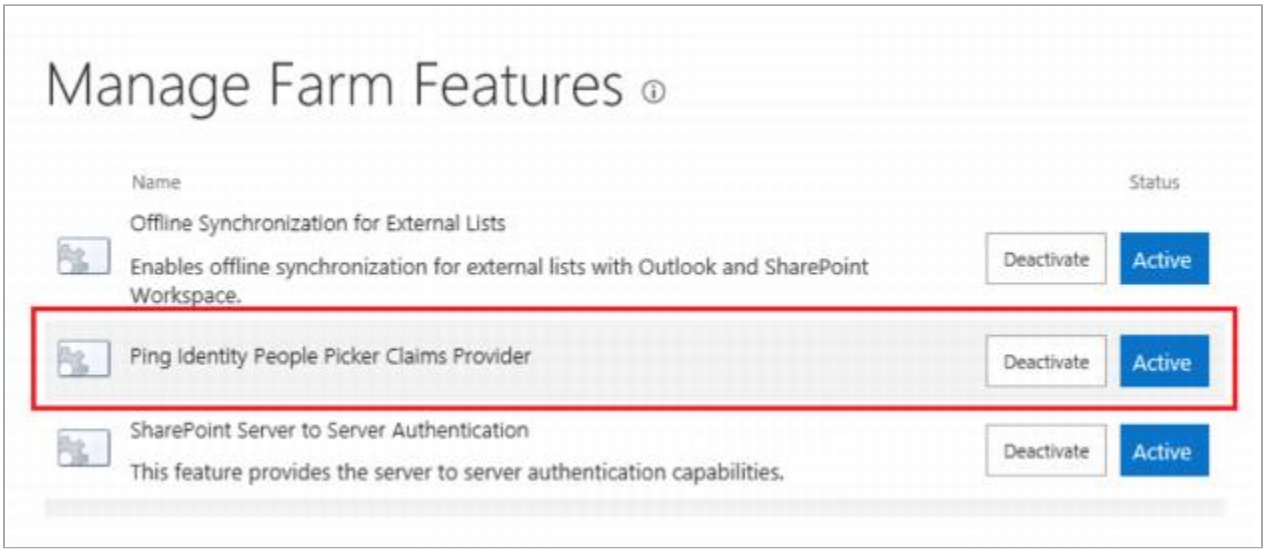
Copyright © 2025 Ping Identity Corporation

```
Install-SPSolution PingIdentity.SharePoint.PPClaimsProvider.wsp
-GACDeployment
```

Activate features

Steps

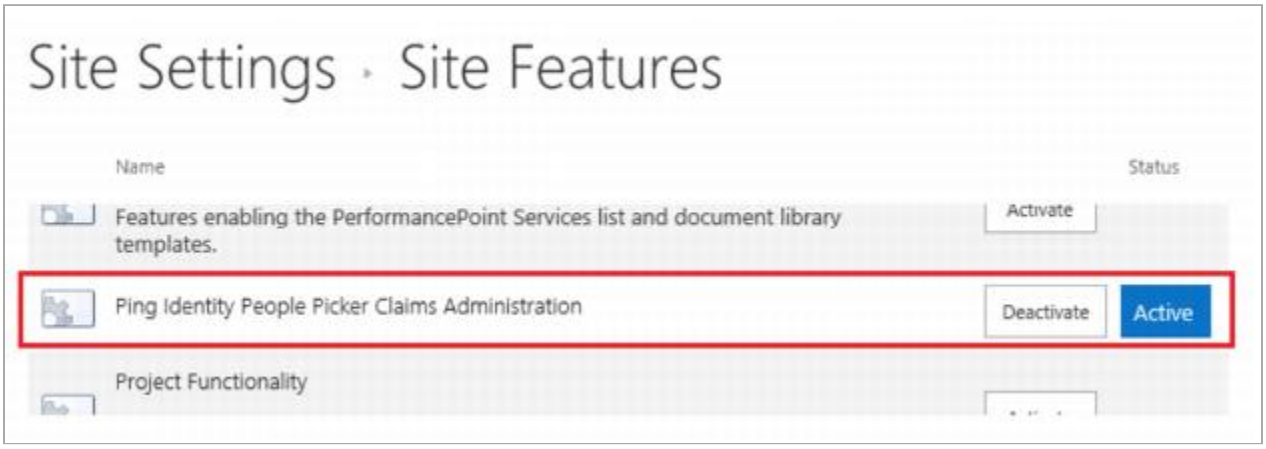
1. Login to your SharePoint Central Administration site.
2. Go to Central Administration > System Settings > Manage farm features.
3. Activate the Ping Identity People Picker Claims Provider feature if it is not already active:



> **Tip**
>
> Activating the feature registers the claim provider in SharePoint.

4. Click on the gear icon on the top right corner of the Central Admin site.
5. Click on Site Settings > Site Actions > Manage site features.
6. Activate the Ping Identity People Picker Claims Administration feature:

Copyright © 2025 Ping Identity Corporation



Set the default claims provider

About this task

Once the claims provider is installed and activated you can set it as the default claims provider for your PingFederate Partner STS by doing the following:

Steps

1. Open SharePoint Management Shell as an Administrator.
2. Execute each command listed below.

```
$ti = Get-SPTrustedIdentityTokenIssuer "<Partner STS Name>"
$ti.ClaimProviderName = "PingIdentityPeoplePickerProvider"
$ti.Update()
```

Replace "*<PartnerSTSName>*" with the name of your configured Trusted Identity Token Issuer.

Configuration

Use the following sections to configure the solution.

- People Picker configuration
- Configuration storage

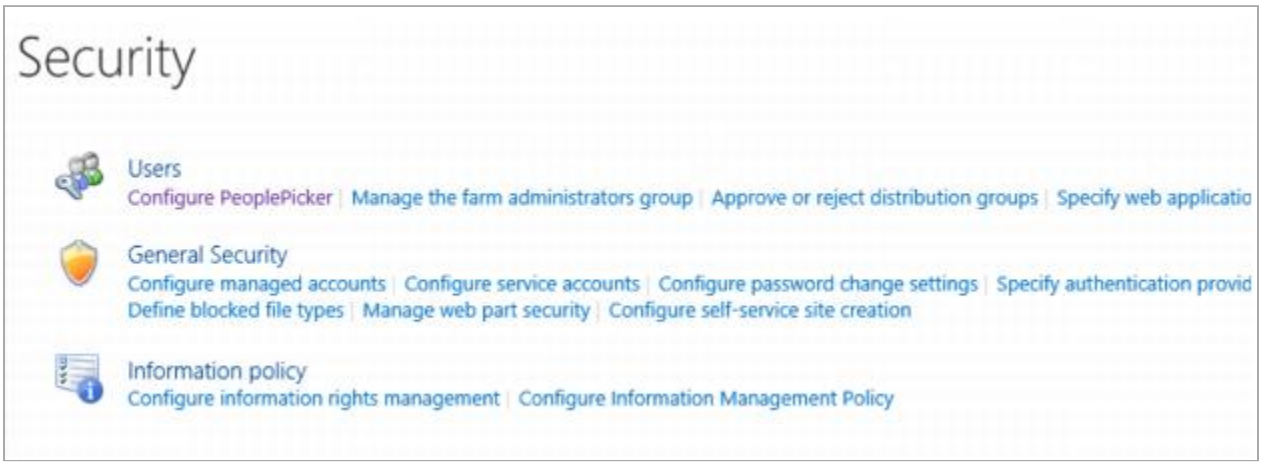
People Picker configuration

Steps

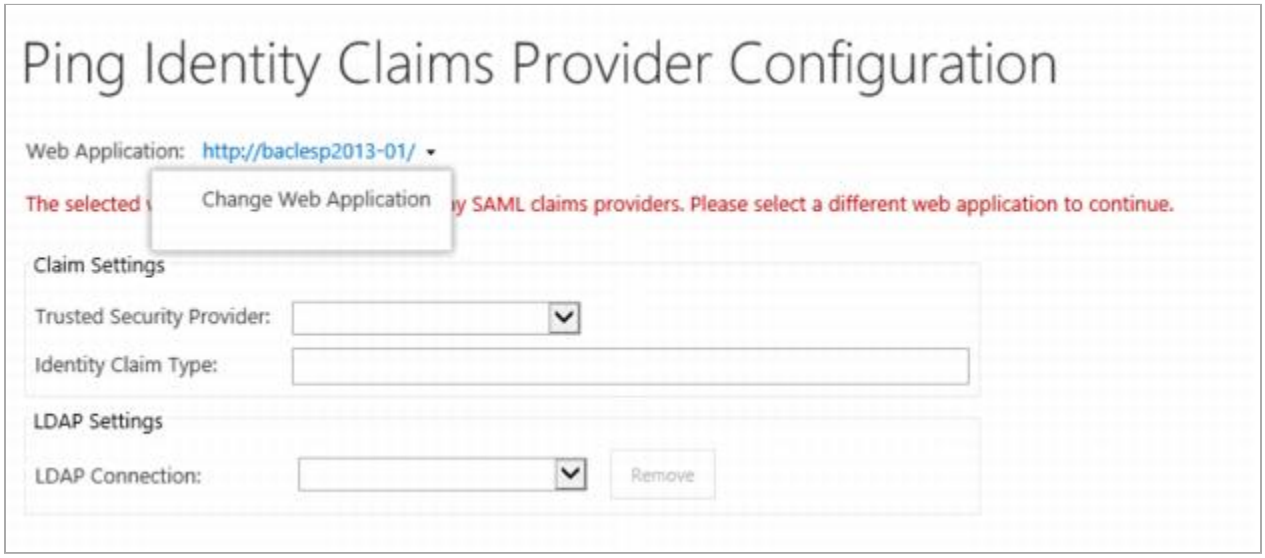
1. Sign on to your SharePoint Central Administration site.
2. Go to Central Administration > Security.

Copyright © 2025 Ping Identity Corporation

3. **Click on Configure People Picker (under Users):**



4. **Select the web application that is configured to use the Partner STS (Trusted Identity Provider) for authentication from the drop down at the top of the page):**



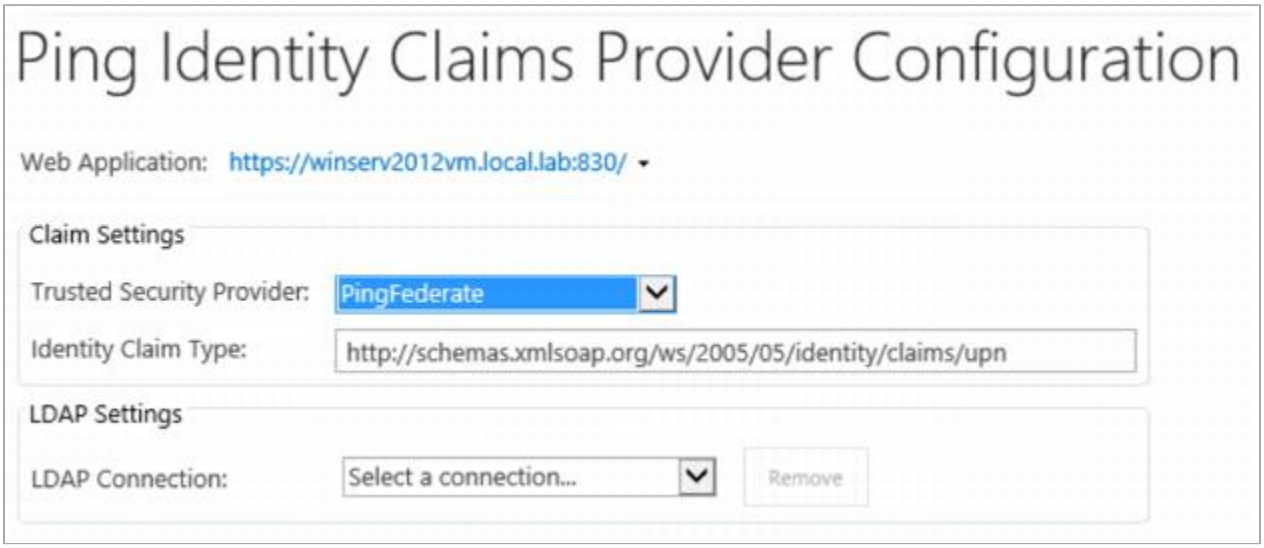
5. **Configure the Claim Settings by selecting the Partner STS and specifying the Identity Claim Type.**

Identity Claim Type examples

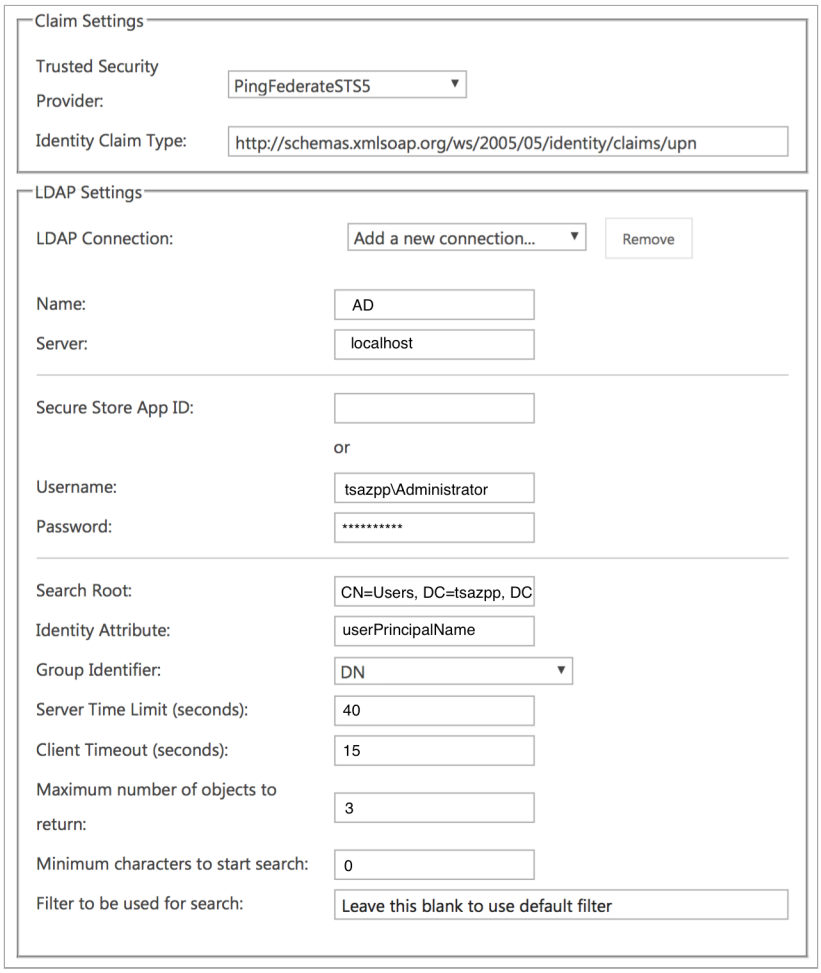
```
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress
```

```
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/upn
```

Copyright © 2025 Ping Identity Corporation



6. **Add LDAP connections by selecting Add a new connection... in the LDAP Connection list, then fill out the LDAP connection settings as described in the following example.**



- **Name**

  The name of this LDAP connection.

Copyright © 2025 Ping Identity Corporation

- Server

  The FQDN or IP address for the LDAP server. If using LDAPS, include the relevant port (for example, `ldap.domain.com:636`)

- Secure Store App Id

  The ID of your saved credentials in SharePoint's Secure Store Service.

- Username

  The username of the LDAP account that will be used to bind to LDAP to query users or groups. If the Username field is left blank the LDAP query will be made using the SharePoint farm account.

- Password

  The password for the LDAP account.

  > **Note**
  >
  > Passwords are stored in plain text with the other configuration data.

- Search Root

  The root (BaseDN) for the LDAP search.

- Identity Attribute

  The Identity Attribute refers to the LDAP attribute used to populate the Identity Claim Type (which you selected in the Claim Settings at the top of the page). The LDAP attribute configured where must match the LDAP attribute used in PingFederate to populate that WS-Fed attribute.

  For example, an Identity Attribute of `userPrincipalName` has an Identity Claim Type of `http://schemas.xmlsoap.org/ws/2005/05/identity/claims/upn`

- Group Identifier

  Select either SID or Distinguished Name (DN) to be used as the group unique identifier.

- Server Time Limit (seconds)

  The maximum number of seconds that the server will wait for a search to complete.

- Client Timeout (seconds)

  The maximum number of seconds that the client will wait for the server to return results.

- Maximum number of objects to return

  This refers to the maximum number of search results you want the People Picker to return.

  - You can set this to a number between 0 and 500.
  - If you set this to 0, it will use the SharePoint default size limit of 1000 entries.

- Minimum characters to start search

Copyright © 2025 Ping Identity Corporation

This refers to the minimum number of characters (letters) an end user must type into the People Picker search window before the search starts to execute.

- You can set this to a number between 4 and 10.
- If you set this to 0, it will use the SharePoint default setting of 3 characters.

> **Tip**
>
> Each LDAP connection you add here will only be enabled for the specific SharePoint web application you selected. If you want to add the same LDAP connection to multiple SharePoint web applications, you will need to repeat the same configuration steps (4-5-6) for each SharePoint web application.

- Filter

  Provide a custom LDAP query to be used for search. Leave this property blank to use the default query:

  ```
  "(&(&(\|(objectCategory=person)(groupType=-2147483646))(\|(displayName={0}*)(sAMAccountName={0}*)(userPrincipalName={0}*)(cn={0}*)(mail={0}*))))"
  ```

Configuration storage

About this task

Configuration settings are stored in the web application properties collection with the key "PingClaimsProviderConfig". The configuration is stored in JSON format in plain text (unencrypted).

> **Note**
>
> If using the *Secure Store Application Id* feature, the credentials will be encrypted (see below).

```
$webApp = Get-SPWebApplication <web app URL>\$webApp.Properties["PingClaimsProviderConfig"]\
```

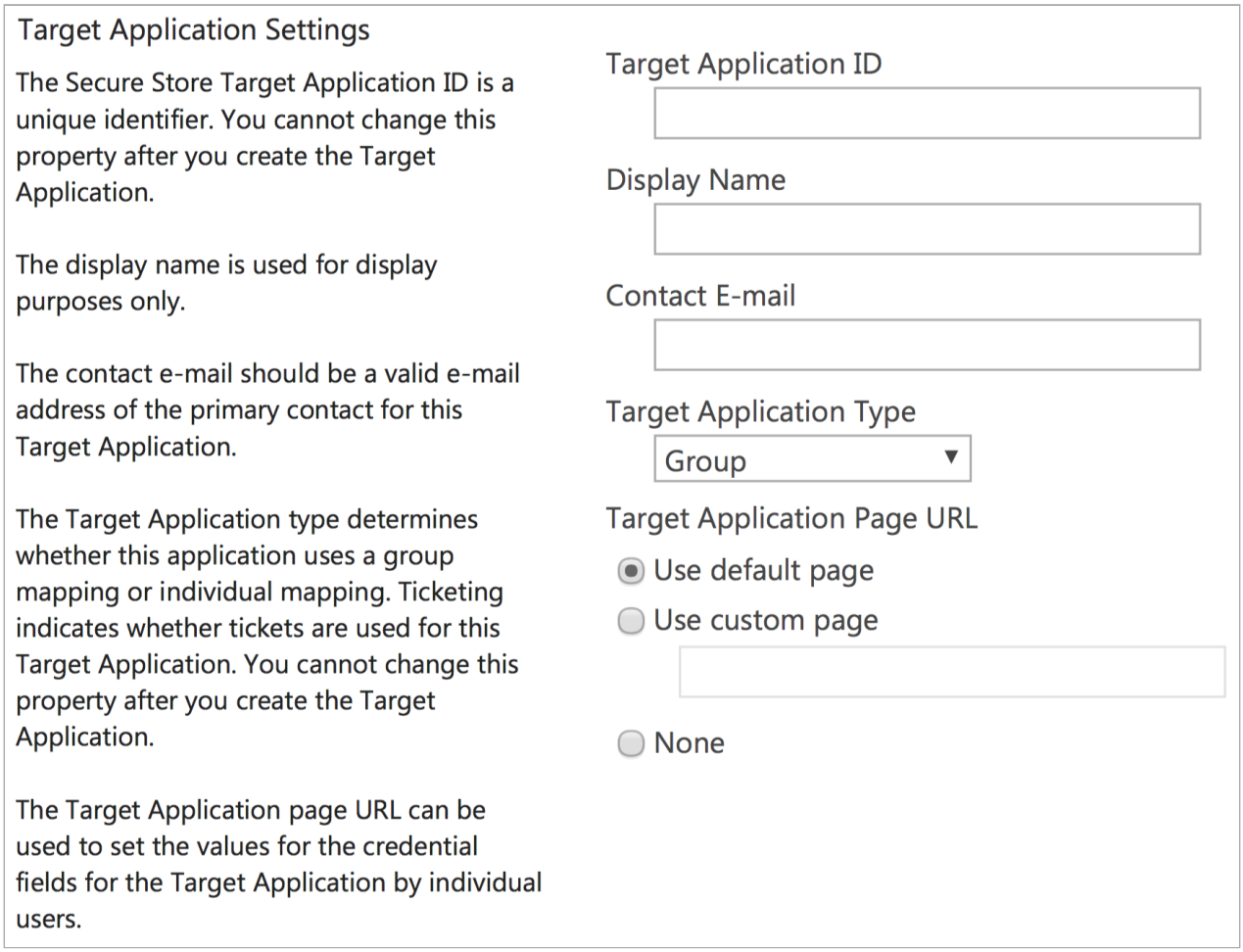
Secure Storage Configuration

Configuring a Secure Store Target Application encrypts the credentials necessary for the LDAP query used in the People Picker. To set up a Secure Store Target Application for the Ping Identity People Picker, use the following steps.

Steps

1. Create a new Secure Store Target Application and populate: Target Application ID, Display Name, and Contact E-mail. Use *Group* for the Target Application Type. Click Next.

Copyright © 2025 Ping Identity Corporation

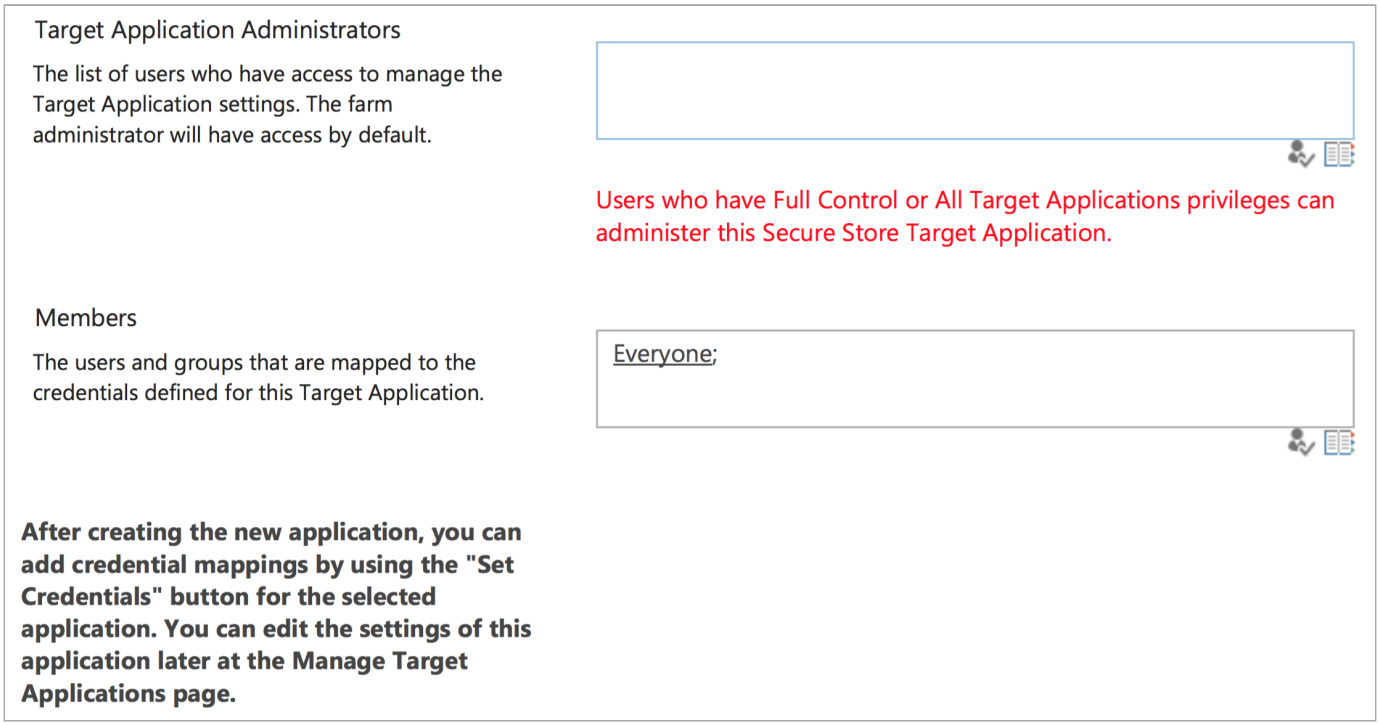


2. Use the field types *Username* and *Password* for the User Name and Password fields, respectively. Click Next.



3. Select an account to administer the application in the Target Application Administrators field. In the Members field, enter a list of users who will have access the target application. Click OK to complete the setup.

Copyright © 2025 Ping Identity Corporation



Supported search attributes with examples

The solution by default enables search on the following LDAP attributes:

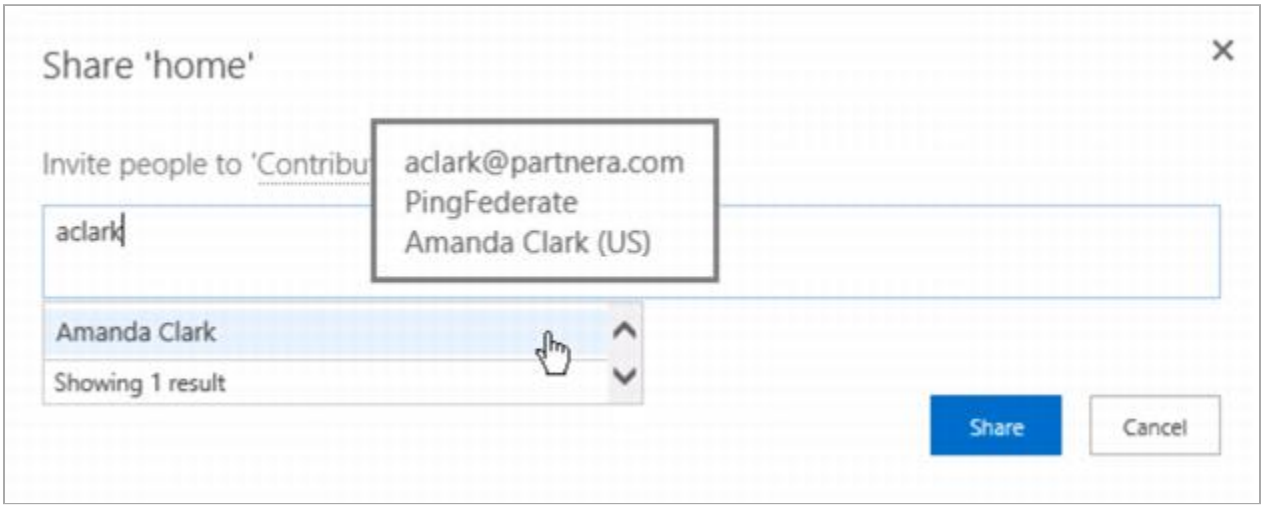
- cn
- displayName
- mail
- sAMAccountName
- userPrincipalName

Search by user

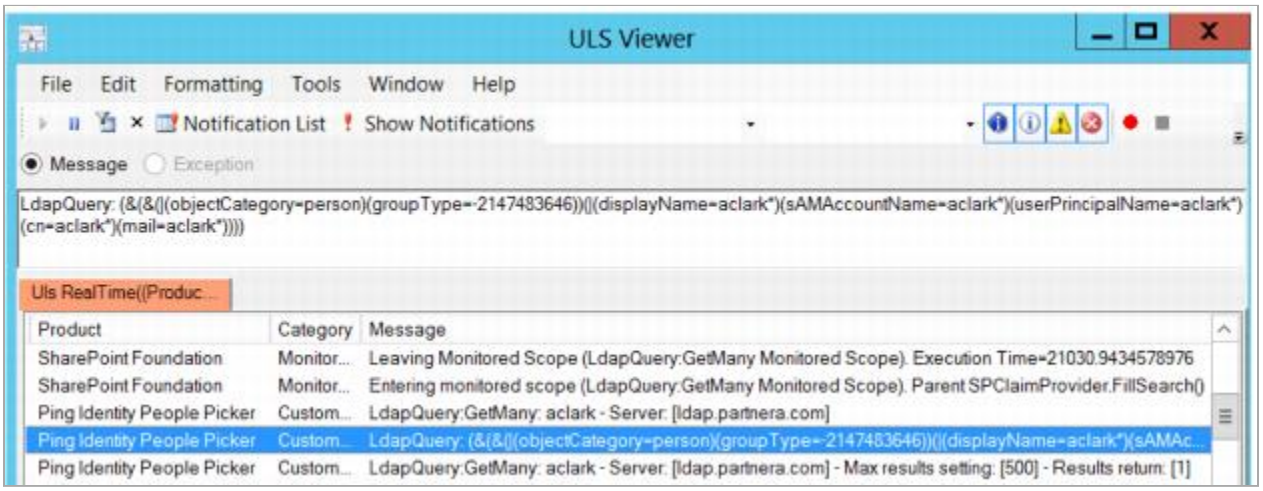
About this task

The following image is an example of a successful search for a user by username (sAMAccountName):

Copyright © 2025 Ping Identity Corporation



The following image is the corresponding log entry for the successful search:



Search by group

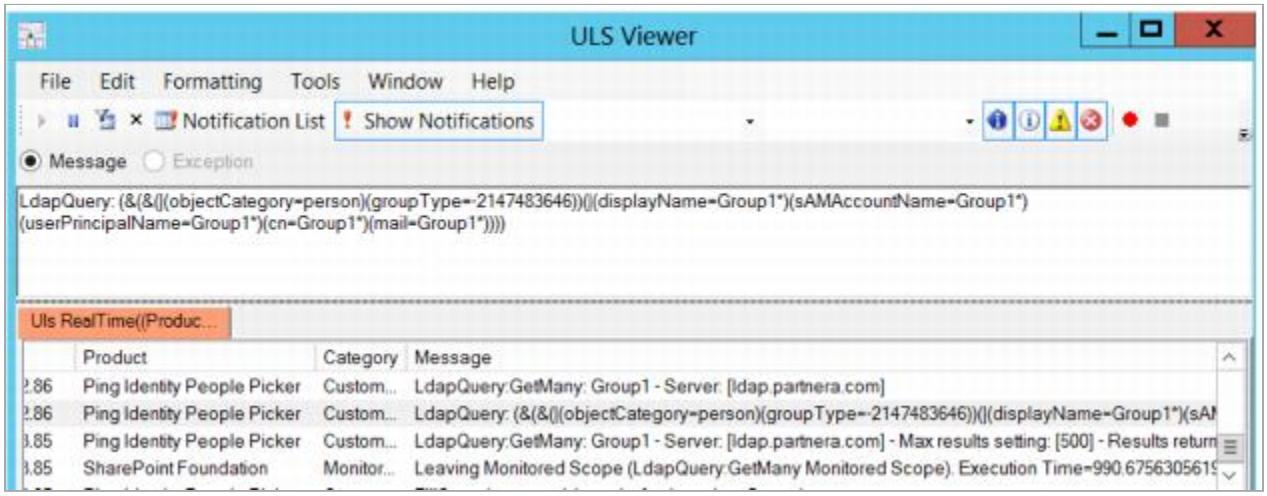
About this task

The following image is an example of a successful search for a group by the group's name (cn, sAMAccountName):

Copyright © 2025 Ping Identity Corporation



The following image is the corresponding log entry for the successful search:



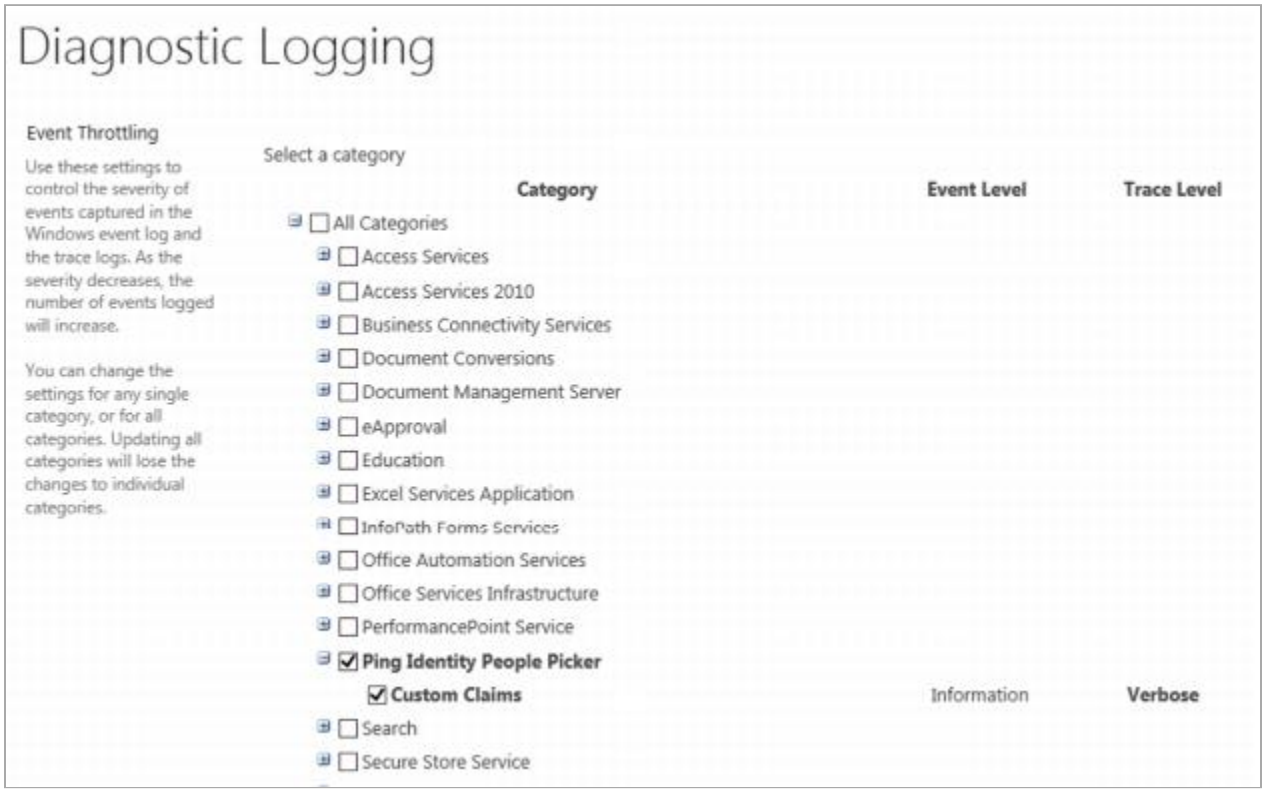
Troubleshooting

Diagnostic logging

Verbose tracing can be enabled for the Ping Identity People Picker on the Diagnostic Logging page of the Monitoring section in Central Administration.

1. **Go to Central Administration > Monitoring > Reporting > Configure diagnostic logging.**
2. **Select the Custom Claims category.**

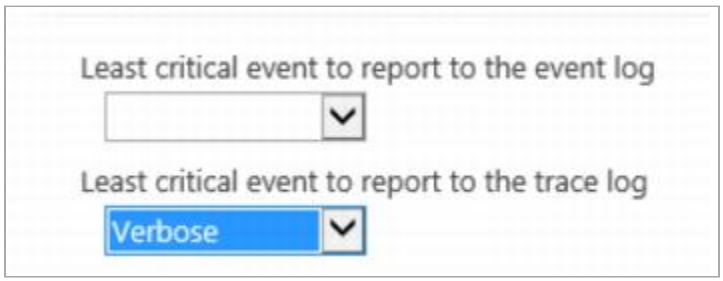
Copyright © 2025 Ping Identity Corporation



3. (Optional) Select SharePoint Foundation >> Monitoring to enable monitored scopes that will show LDAP queries.



4. Select Verbose as the least critical event to report to the trace log.



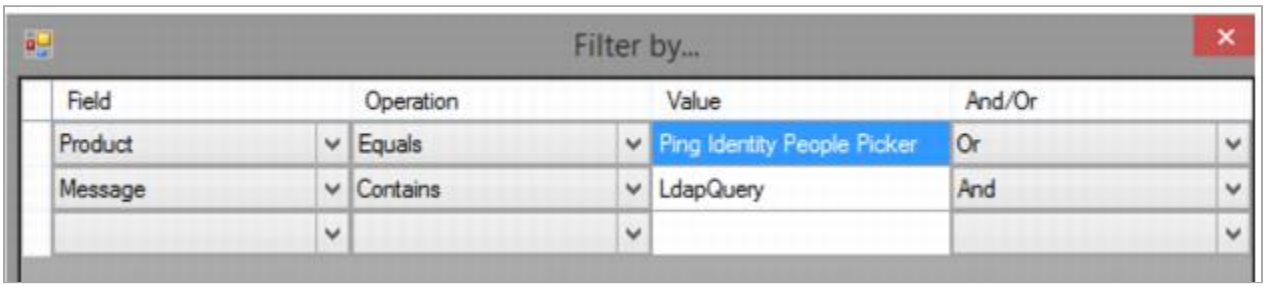
5. The default log directory will be similar to this path:

```
C:\Program Files\Common Files\Microsoft Shared\Web Server Extensions\15\LOGS
```

6. If the SharePoint admin has changed the default log path, you can find it by going to Central Admin > Monitoring > Reporting > Configure diagnostic logging scroll down to Trace Log - the Path configured here is where your Sharepoint Logs are being stored.

Copyright © 2025 Ping Identity Corporation

7. Use the ULS Viewer app to monitor the ULS logs. You can use the following filters to see just the output from the custom claims provider.



You can download the ULS Viewer tool from Microsoft here: https://www.microsoft.com/en-us/download/details.aspx?id=44020

SharePoint log location

The default log directory is: `C:\Program Files\Common Files\Microsoft Shared\Web Server Extensions[.varname]_<version>_\LOGS`.

If the SharePoint administrator has changed the default log path, you can find it by going to Central Admin > Monitoring Reporting > Configure diagnostic logging scroll down to Trace Log - the Path configured here is where your Sharepoint Logs are being stored.

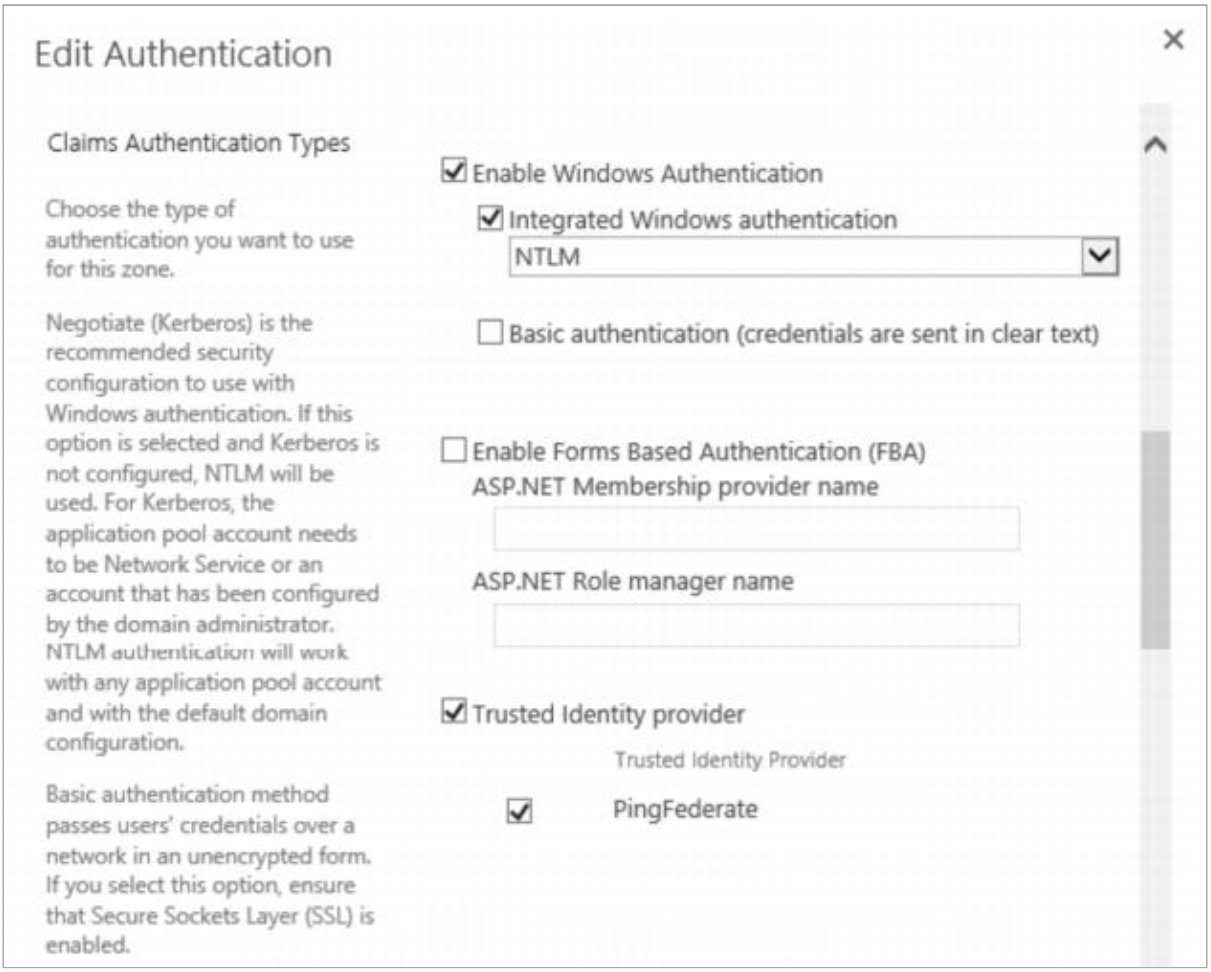
Uninstall the solution

Steps

1. In Central Administration disable the Trusted Identity provider from the web application.

   This step must be completed for each web application that is using that Trusted Identity provider (also known as the Partner STS or Trusted Identity Token Issuer).

Copyright © 2025 Ping Identity Corporation



2. **In Central Administration > System Settings > Farm Features deactivate the Ping Identity People Picker Claims Provider farm feature.**

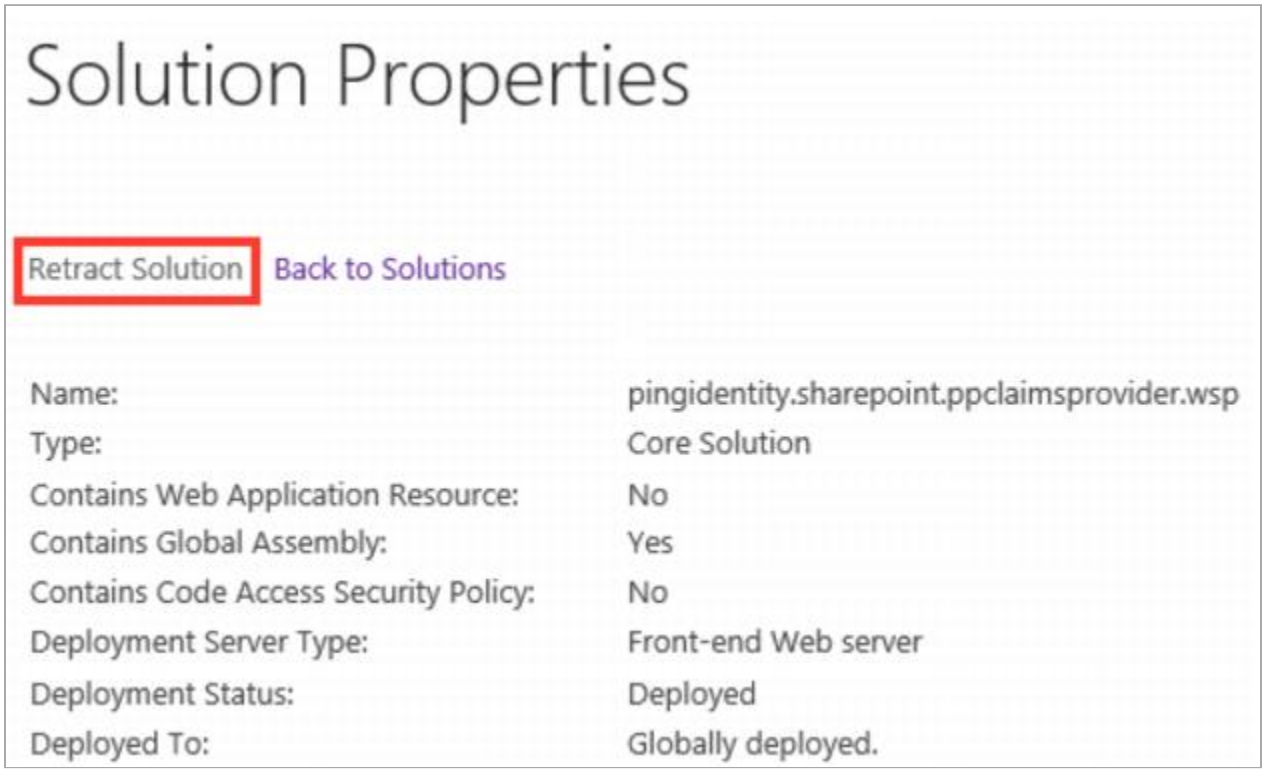


3. **In Central Administration > Site Settings > Site Features deactivate the Ping Identity People Picker Claims Administration site feature.**

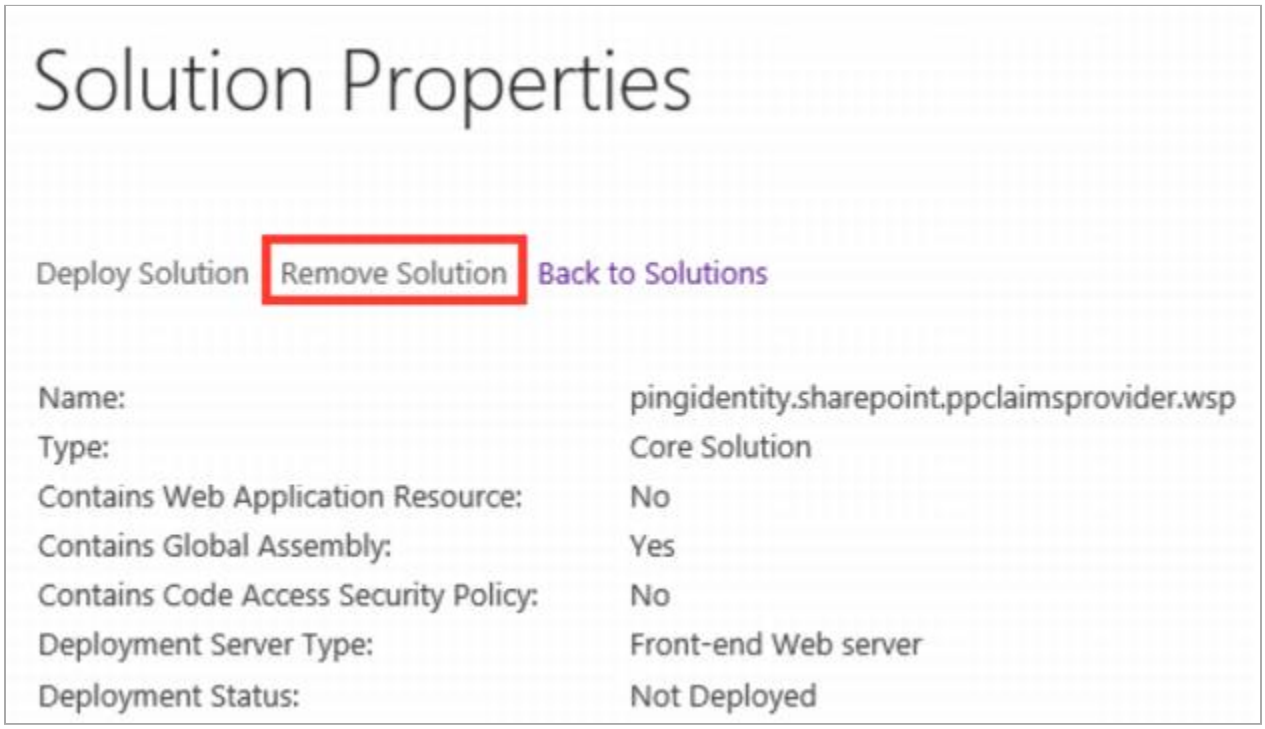


4. **In Central Administration > System Settings > Farm Solutions retract the pingidentity.sharepoint.ppclaimsprovider.wsp solution.**

Copyright © 2025 Ping Identity Corporation



5. **In Central Administration > System Settings > Farm Solutions remove the pingidentity.sharepoint.ppclaimsprovider.wsp solution.**



6. **Open SharePoint Management Shell as an Administrator.**

7. **Make a backup copy of your configuration settings for your existing Trusted Identity Token Issuer. For example: use this command tor produce a file called partnersts.txt that contains your list of token issuers and their settings:**

Copyright © 2025 Ping Identity Corporation

```
Get-SPTrustedIdentityTokenIssuer >partnersts.txt
```

8. Use this command to delete your current Trusted Identity Token Issuer:

```
Remove-SPTrustedIdentityTokenIssuer –Identity “<PartnerSTS>”
```

   Replace `<PartnerSTS>` with the Name of your Trusted Identity Token Issuer.

9. User PowerShell to recreate your SP Trusted Identity Token Issuer (without setting the default claims provider). You can refer to the partnersts.txt to review what settings you used previously.
10. In Central Administration reconfigure the web application to use the newly created SPTrustedIdentityTokenIssuer.
11. (Optional) For a complete cleanup, you may also wish to remove the People Picker configuration settings that were stored for each web application that you configured to use the People Picker. This can be done by running the following commands via the SharePoint Management Shell for each web application:
    - Identify the web application by replacing `<web app URL>` with the SharePoint Web Application’s URL:

```
$webApp = Get-SPWebApplication <web app URL>
```

    - To view the settings associated for the web application:

```
$webApp.Properties[“PingClaimsProviderConfig”]
```

    - To remove the settings associated for the web application:

```
$webApp.Properties.Remove[“PingClaimsProviderConfig”]
```

Release notes

Changelog

SharePoint People Picker Integration Kit 1.4 – December 2020

- Added support for selecting an authentication type for each LDAP connection.
- Added support for SharePoint 2013.

SharePoint People Picker Integration Kit 1.3 – February 2019

- Added support for SharePoint 2019.

Copyright © 2025 Ping Identity Corporation

SharePoint People Picker Integration Kit 1.2.1 – July 2018

- Addressed feature activation failures due to logging and null configuration reference.

SharePoint People Picker Integration Kit 1.2 – May 2017

- Resolved issue when the Ping Identity Custom Claims Provider is enabled for a Trusted Identity Provider in the Intranet Zone.
- Added support for secure storage of LDAP credentials.
- Resolved issue when searching for group names that contain brackets.
- Security fixes.

SharePoint 2013 People Picker Extension Integration Kit 1.1 – September 2016

- Added ability to specify a custom LDAP filter to find users and groups.

SharePoint 2013 People Picker Extension 1.0.1 Integration Kit – August 2016

- SharePoint People Picker can now be used alongside the Windows NTLM Claim Picker.

SharePoint 2013 People Picker Extension 1.0 Integration Kit – Feb 2016

- Initial Release.

Known issues and limitations

- None.

Download manifest

The distribution `.zip` archive for the SharePoint People Picker Integration Kit contains the following:

- `ReadMeFirst.pdf` – contains links to this online documentation.
- `/dist`
  - `PingIdentity.SharePoint.PPClaimsProvider.wsp` – The SharePoint People Picker.

Copyright © 2025 Ping Identity Corporation

Slack Provisioner

The Slack Provisioner allows PingFederate to integrate with Slack for user provisioning and single sign-on (SSO).

Features

- Supports all Slack plans, including Standard, Plus, and Enterprise Grid.
- Manages users in Slack based on changes in an external datastore that is attached to PingFederate.
  - Creates, updates, and disables users.
  - Allows you to enable the create, update, and disable capabilities independently.
  - Allows you to choose to disable users when deprovisioning.
- Browser-based SSO initiated by the service provider (SP) or identity provider (IdP).
- Pre-populates some connection settings with the included quick connection template and SAML metadata file.

Intended audience

This document is intended for PingFederate administrators.

If you need help during the setup process, see the following resources:

- The following sections of the PingFederate documentation:
  - Identity provider SSO configuration
  - Datastores
  - SP connection management
  - Configuring outbound provisioning
  - Configuring outbound provisioning settings
- The following sections of the Slack documentation:
  - Manage members with SCIM Provisioning
  - SAML single sign-on
  - Guide to single sign-on settings

System requirements

- PingFederate 9.0 or later.
- A Slack Workspace Owner account.
- A Slack Plus or Enterprise Grid plan is required to use the optional SSO functionality.

Copyright © 2025 Ping Identity Corporation

- To allow PingFederate to make outbound connections to the Slack API, you might need to allow the following domain in your firewall:
    - https://api.slack.com

User and group management

The Slack Provisioner links users and groups from the data store to Slack. The behavior of each provisioning capability is described below.

Synchronizing existing users

By default, the provisioning connector synchronizes based on the `userName` attribute in Slack. You can change the default mapping in Creating a provisioning connection.

For example:

- In Slack, Janet's `userName` is `jsmith`.
- In your data store, Janet's `sAMAccountName` is `jsmith`.
- On the Attribute Mapping tab of your PingFederate channel configuration, you map the `userName` attribute to `sAMAccountName`.
- When the provisioning connector runs, the data store user is provisioned with a `userName` of `jsmith`. That matches Janet's existing `userName` in Slack, so her information in the data store is synchronized to her Slack account.

User provisioning

Triggered by either of the following:

- A user is added to the data store group or filter that is targeted by the provisioning connector.
- A user with "disabled" status is added to the data store group or filter that is targeted by the provisioning connector, and the Provision disabled users provisioning option is enabled.

The target is determined by the Source Location tab in the provisioning connector configuration.

User updates

Triggered when a change occurs to a user attribute that is mapped in the provisioning connector configuration, such as an attribute change or the user being re-enabled from a disabled status.

User deprovisioning

Triggered by any of the following:

- A user is deleted from the user store.
- A user is disabled in the user store.

Copyright © 2025 Ping Identity Corporation

- A user is removed from the data store group or filter that is targeted by the provisioning connector.

The Remove User Action setting in the connection configuration determines whether the deprovisioning action disables or deletes the user.

Synchronizing existing groups

The provisioning connector synchronizes groups from the data store to Slack based on the group name.

For example:

- In Slack, there is a group is named `Accounting`.
- In your data store, there is a group with a `CN` of `Accounting`.
- When the provisioning connector runs, the two groups are synchronized.

Group provisioning

Triggered when a group is added to the data store filter that is targeted by the provisioning connector.

The target is determined by the Source Location tab in the provisioning connector configuration.

Group name updates

Renaming the group in the data store will trigger PingFederate to rename the group in Slack.

Group membership updates

Changing group memberships through the group's properties or a user's properties will trigger PingFederate to update the group membership in Slack.

Group memberships in the data store overwrite the group memberships in Slack.

Group deletion

Triggered by any of the following:

- Deleting the group in the data store will trigger PingFederate to delete the group in Slack. Group deletions are permanent and cannot be undone.
- The group is removed from the data store group or filter that is targeted by the provisioning connector.

Setup

Upgrading an existing deployment

- If you're upgrading from a previous version of the Slack Provisioner, note your existing service provider (SP) connection configuration and create a new connection.

Copyright © 2025 Ping Identity Corporation

Upgrading an existing deployment in PingFederate 10.1 or later

Steps

1. Back up your current PingFederate configuration as shown in Configuration archive in the PingFederate documentation.
2. In the PingFederate administrative console, go to Applications > Integration > SP Connections and select your connection.

   Note the attribute mappings for your existing SP connection.

   For help, see Mapping attributes in the PingFederate documentation.
3. Delete your existing SP connection.
    1. Go to Applications > Integration > SP Connections.For your existing connection, click Select action, and then click Delete. Click Confirm.
4. Complete the steps in Deploying the integration files.
5. Complete the steps in Creating a provisioning connection.
    - From Outbound Provisioning > Manage Channels > Channel, on the Attribute Mapping tab, configure the attribute mappings based on your notes.
6. Complete the steps in Creating a single sign-on connection.

Upgrading an existing deployment in PingFederate 10.0 or earlier

Steps

1. Back up your current PingFederate configuration as shown in Configuration archive in the PingFederate documentation.
2. In the PingFederate administrative console, go to Identity Provider > SP Connections and select your connection.

   Note the attribute mappings for your existing SP connection.

   For help, see Mapping attributes in the PingFederate documentation.
3. Delete your existing SP connection.
    1. Go to Identity Provider > SP Connections > Manage All.For your existing connection, click Select action, and then click Delete. Click Save.
4. Complete the steps in Deploying the integration files.
5. Complete the steps in Creating a provisioning connection.
    - From Outbound Provisioning > Manage Channels > Channel, on the Attribute Mapping tab, configure the attribute mappings based on your notes.
6. Complete the steps in Creating a single sign-on connection.

Copyright © 2025 Ping Identity Corporation

Deploying the integration files

To get started with the integration, deploy the Slack Provisioner files to your PingFederate directory.

Steps

1. Download the Slack Provisioner `.zip` archive from the .pingidentity.com/s/marketplace-integration-home-page/[Ping Identity Integration Directory].
2. Stop PingFederate.
3. If you're upgrading an existing deployment, delete the `pf-slack-quickconnection-<version>.jar` file from your `<pf_install>/pingfederate/server/default/deploy` directory.
4. Extract the `.zip` archive and merge the contents of the `dist` directory with your `<pf_install>/pingfederate/server/default/deploy` directory.
5. Enable the PingFederate provisioning engine: Open your `<pf_install>/pingfederate/bin/run.properties` file.Change `pf.provisioner.mode` to `STANDALONE`. Save the file.

   > **Note**
   >
   > To configure the `FAILOVER` mode instead, see Deploying provisioning failover in the PingFederate Server Clustering Guide.

6. Start PingFederate.
7. If you operate PingFederate in a cluster, repeat steps 2-4 and step 6 for each engine node.

Enabling provisioning and single sign-on in PingFederate

To use PingFederate for provisioning and single sign-on (SSO), configure an external datastore and set a SAML entity ID.

About this task

Your external datastore acts as the source of data for provisioning. PingFederate also uses an internal datastore to store the state of synchronization between the source datastore and the target datastore.

For more information, see Datastores and Configuring outbound provisioning settings in the PingFederate documentation.

Enabling provisioning and single sign-on in PingFederate 10.1 or later

Steps

1. In the PingFederate administrative console, configure the datastore for PingFederate to use as the source of user data.

   For help, see Datastores in the PingFederate documentation.

2. Go to System > Server > Protocol Settings > Federation Info.
3. In the SAML 2.0 Entity ID field, enter a name for PingFederate to use when identifying itself to SAML partners.

Copyright © 2025 Ping Identity Corporation

4. On the Outbound Provisioning tab, in the Provisioner Data Store list, select the internal database that will store the synchronization state. Click Save.

   For more information, see Configuring outbound provisioning settings in the PingFederate documentation.

Enabling provisioning and single sign-on in PingFederate 10.0 or earlier

Steps

1. In the PingFederate administrative console, configure the datastore for PingFederate to use as the source of user data.

   For help, see Datastores in the PingFederate documentation.

2. Go to System > Protocol Settings > Roles & Protocols.
3. Select the Enable Identity Provider IdP Role and Support the Following check box.
4. Select the SAML 2.0 and Outbound Provisioning check boxes. Click Next.
5. Click the Federation Info tab.
6. In the SAML 2.0 Entity ID field, enter a name for PingFederate to use to identify itself to SAML partners.
7. On the Outbound Provisioning tab, in the Provisioner Data Store list, select the internal database that will store the synchronization state. Click Save.

   For more information, see Configuring outbound provisioning settings in the PingFederate documentation.

Registering PingFederate as an app in Slack

Before PingFederate can access the Slack API, it needs an access token. To get the access token, you need to register PingFederate as an app in Slack.

Steps

1. Sign on to Slack as an Admin or workspace Owner.

   If you don't already have a Slack workspace, create one. For help, see Getting started for workspace creators in the Slack documentation.

2. Go to the Slack Your Apps window.
3. Click Create New App.
4. On the Create a Slack App modal, enter a name and select a workspace. Click Create App.
5. On the Basic Information window, in the App Credentials section, note the Client ID and Client Secret.

   You will use these in Getting a Slack API access token.

6. In the navigation pane, click OAuth & Permissions.

Copyright © 2025 Ping Identity Corporation

7. On the OAuth & Permissions window, in the Redirect URLs section, add the Ping Identity OAuth Configuration Service (OCS).
    1. Click Add New Redirect URL.
    2. In the Redirect URLs field, enter the following:

        ```
        https://oauth.pingone.com/ocs/ppm/rest/v1/oauth/oastempcredresponse/
        ```

    3. Click Add. Click Save URLs.
8. In the Scopes section, add the admin scope.
    1. In the User Token Scopes section, click Add an OAuth Scope.
    2. Select Admin.
9. Submit your app to the Slack App Directory.

    > **Important**
    >
    > This step is required for Slack Enterprise Grid, but optional for other Slack plans.

    1. In the navigation pane, click Manage Distribution.
    2. In the Remove Hard Coded Information section, select the I've reviewed and removed any hard-coded information check box.
    3. Click Activate Public Distribution.

Getting a Slack API access token

Use the OAuth Configuration Service to exchange your Slack client ID and secret for an access token. PingFederate uses this access token to connect to Slack. The process differs slightly depending on your plan.

Getting an access token for Slack Standard or Plus

Exchange your Slack Standard or Plus client ID and secret for an access token.

Steps

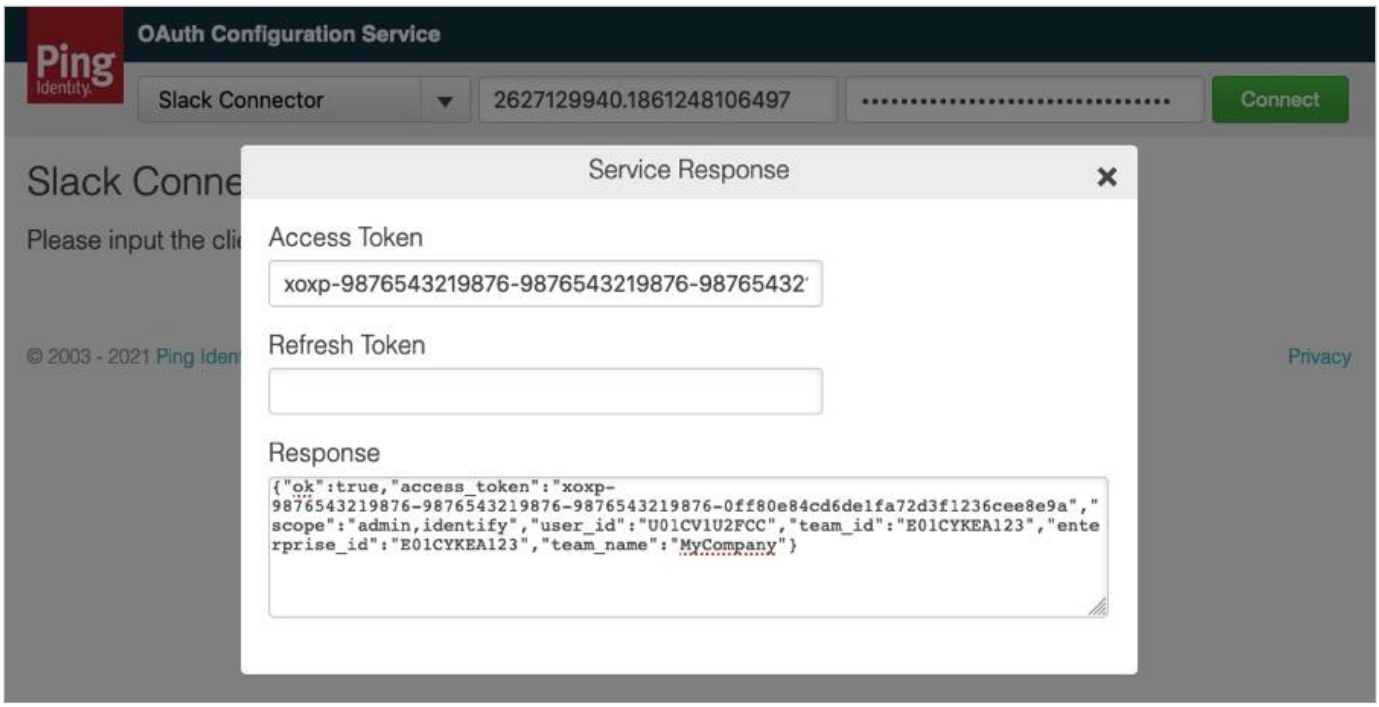
1. Go to the Ping Identity OAuth Configuration Service.
2. From the product list, select Slack Connector.
3. Enter the Client ID and Client Secret that you noted in Registering PingFederate as an app in Slack.

Copyright © 2025 Ping Identity Corporation



4. Click Connect.
5. If you receive a prompt to sign on, sign on with your Workspace Owner account.
6. When you receive a message that your application is requesting permission to administer your Slack team, click Authorize.
7. On the OAuth Configuration Service page, copy the Access Token.

   You will use the access token in Creating a provisioning connection.



Getting an access token for Slack Enterprise Grid

Exchange your Slack Enterprise Grid client ID and secret for an access token.

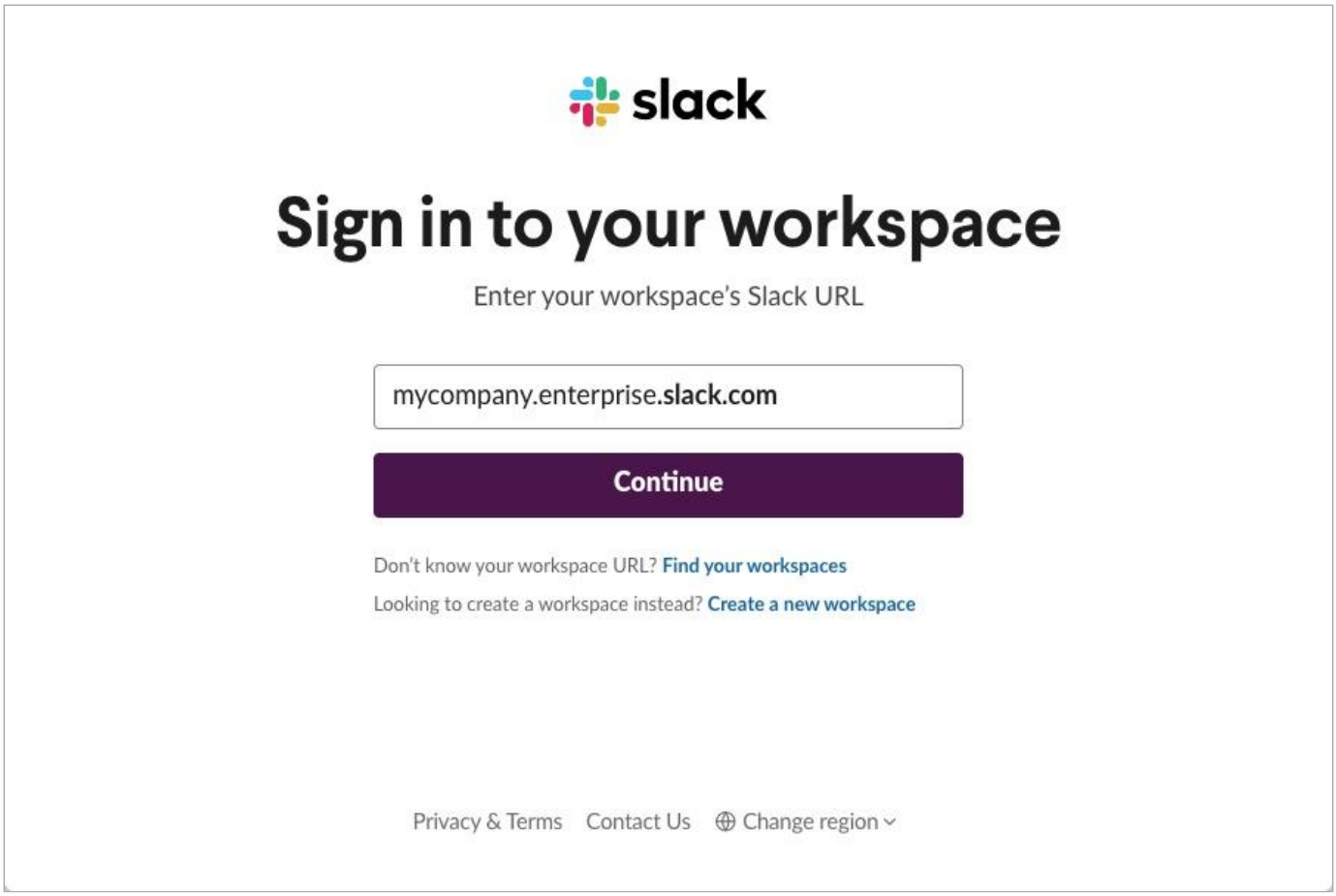
Steps

1. In an "Incognito" (or "Private") browser window, go to the Ping Identity OAuth Configuration Service (OCS).
2. From the product list, select Slack Connector.
3. Enter the Client ID and Client Secret that you noted in Registering PingFederate as an app in Slack.

Copyright © 2025 Ping Identity Corporation

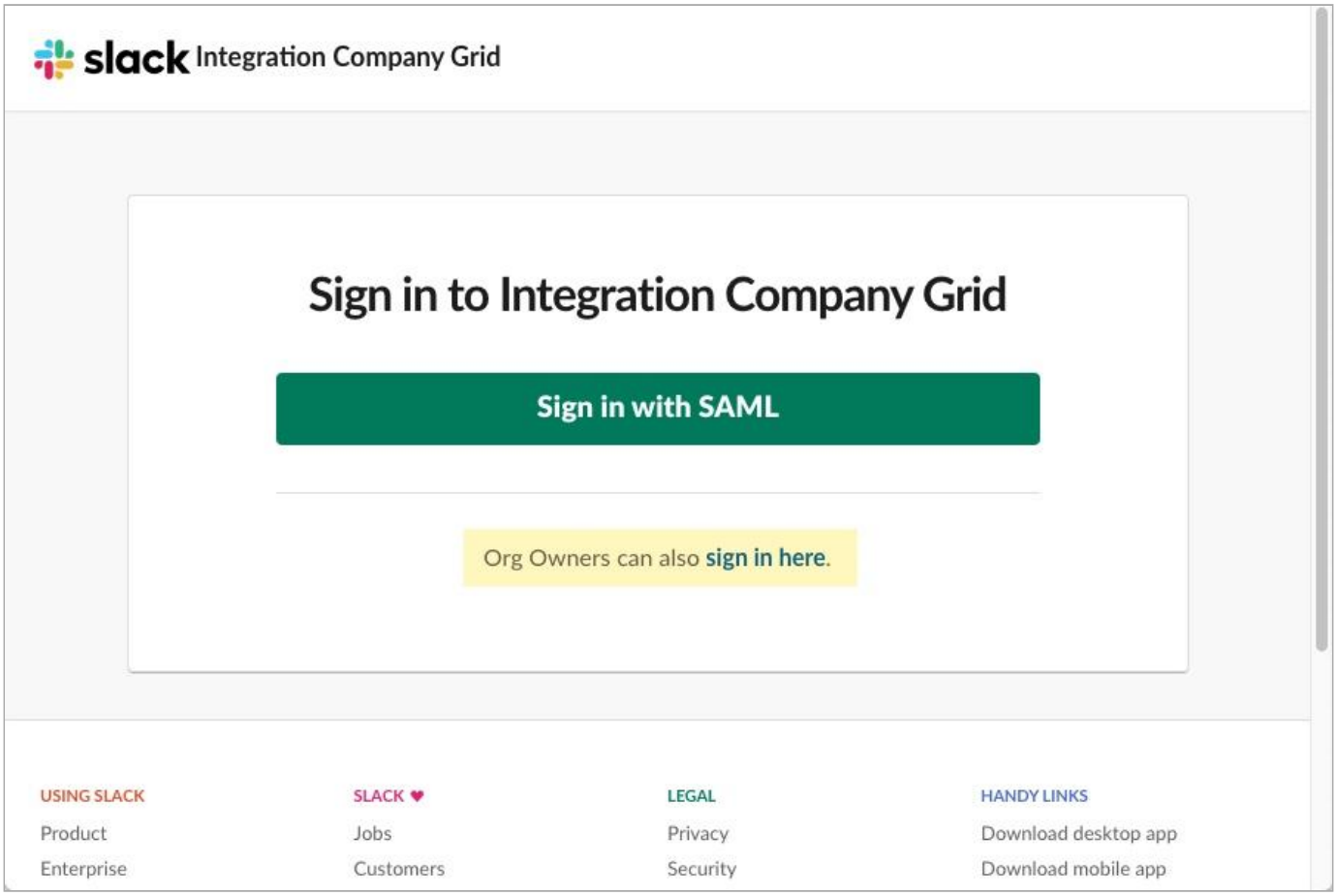


4. Click Connect.
5. In the Sign in to your workspace window, enter the URL for your Slack organization, such as `mycompany.enterprise`.slack.com. Click Continue.



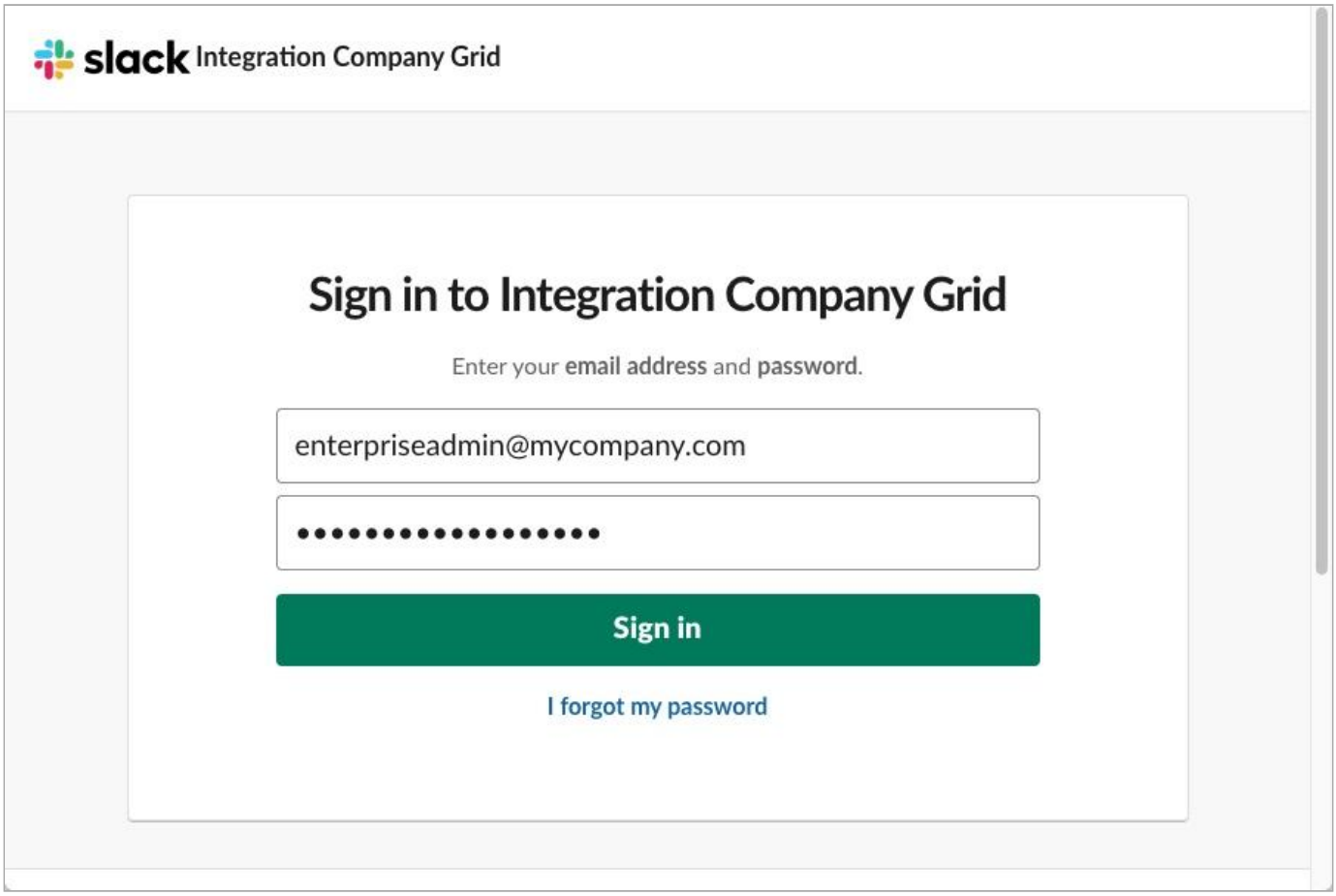
6. Click Org Owners can also sign in here.

Copyright © 2025 Ping Identity Corporation



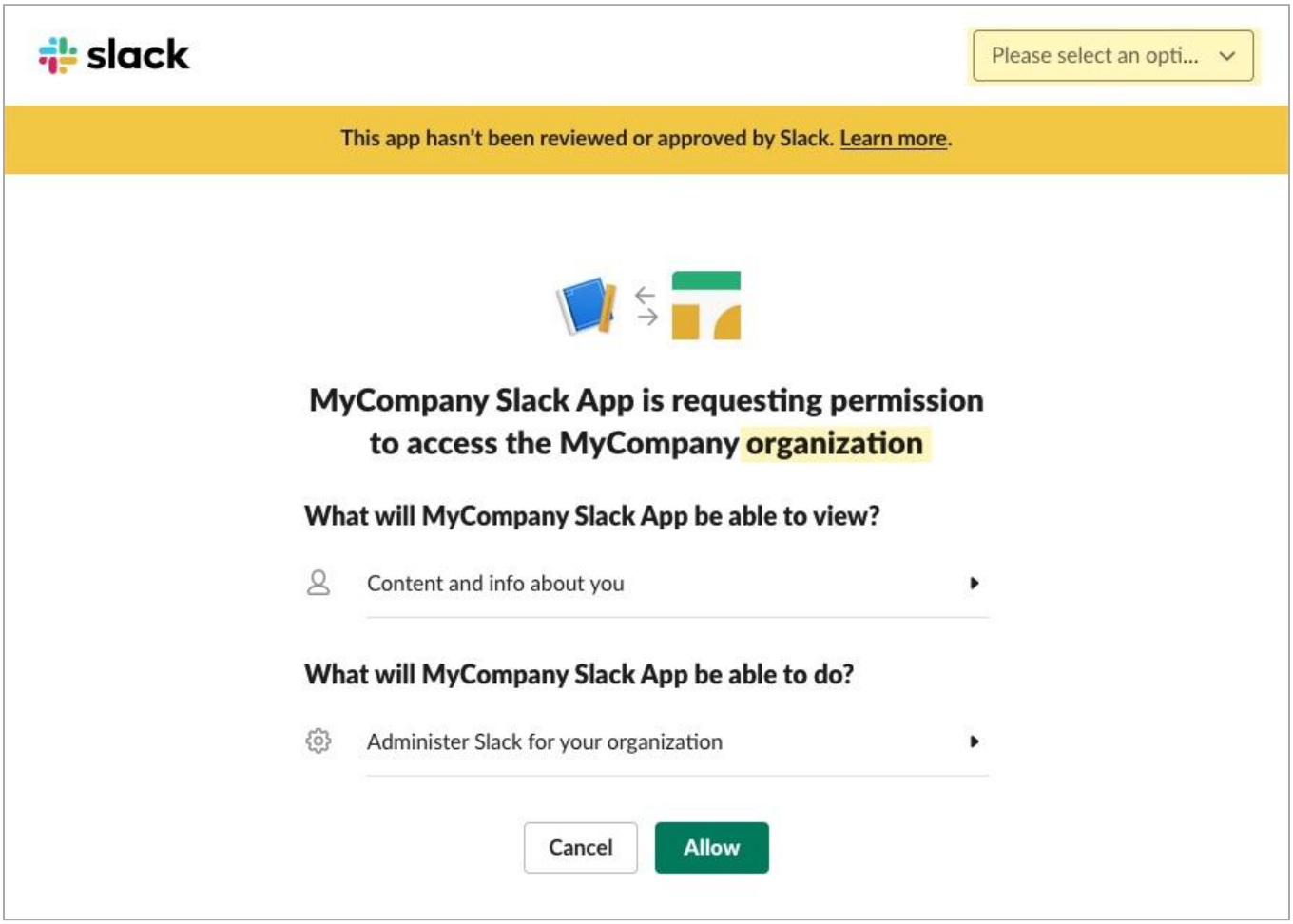
7. Enter your administrator credentials. Click Sign in.

Copyright © 2025 Ping Identity Corporation



8. When you receive a message that your application is requesting permission, check that the request is for the organization. Do NOT select a Workspace from the Please select an option list. Click Allow.

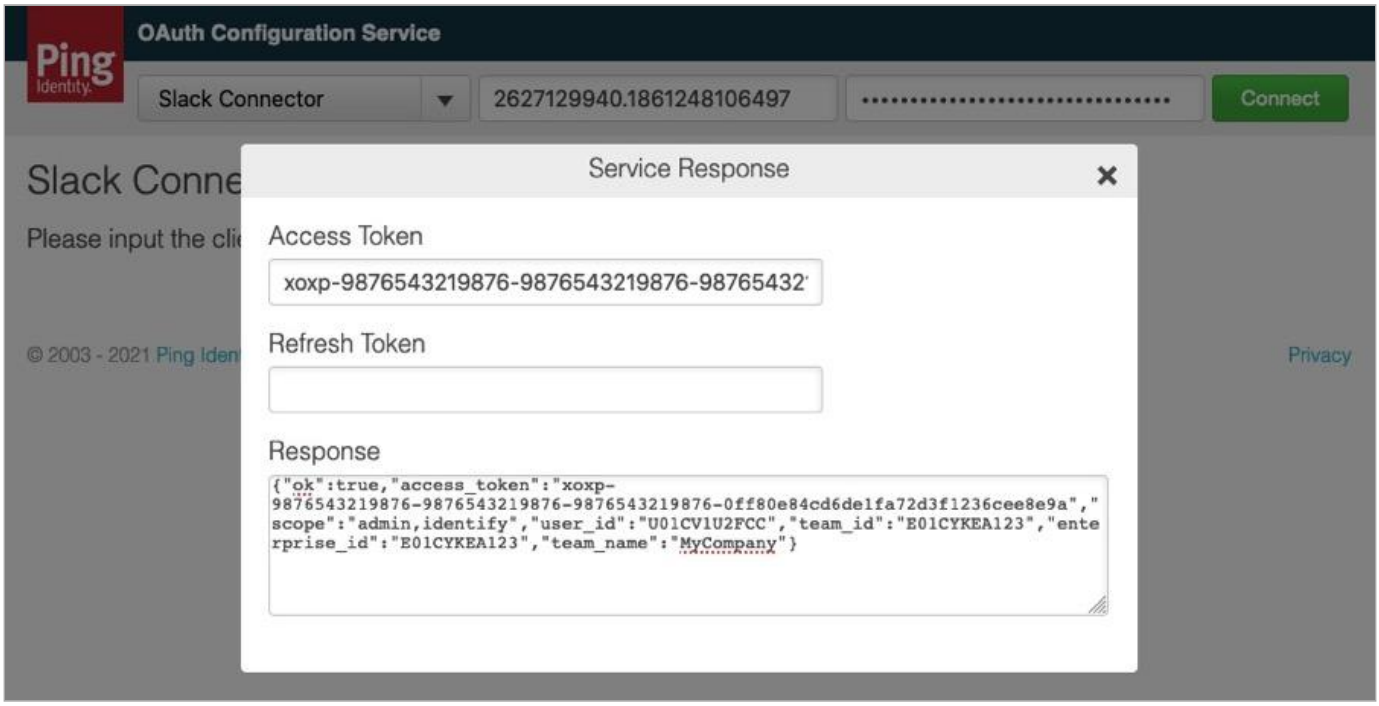
Copyright © 2025 Ping Identity Corporation



9. On the OAuth Configuration Service page, copy the Access Token.

   You will use the access token in Creating a provisioning connection.

Copyright © 2025 Ping Identity Corporation



Preparing your SAML metadata file

Modify the SAML details in the included `saml-metadata.xml` file. This file makes it easier to configure your connection in PingFederate.

Steps

1. In a text editor, open the `saml-metadata.xml` file that is included in the Slack Provisioner `.zip` archive.
2. In the `.xml` file, find the `Location="https://TEAM_NAME.slack.com/sso/saml"` line. Modify it with your Slack Workspace or Enterprise Grid name.

   For example, `Location="https://myworkspace.slack.com/sso/saml"`.
3. Remove the trailing slash (`/`) from `entityID="https://slack.com/"`.

   Result: `entityID="https://slack.com"`
4. Save your changes.

   You'll use this file in Creating a provisioning connection.

Creating a provisioning connection

To allow PingFederate to manage users in Slack, create a service provider (SP) connection.

Steps

1. In the PingFederate administrator console, create a new SP connection:
    1. For PingFederate 10.1 or later: go to Applications > Integration > SP Connections. Click Create Connection.

Copyright © 2025 Ping Identity Corporation

2. For PingFederate 10.0 or earlier: go to Identity Provider > SP Connections. Click Create Connection.

2. Configure the basic connection details with the Slack quick connection template.
    1. On the Connection Template tab, select Use a template for this connection.
    2. From the Connection Template list, select Slack Provisioner.
    3. On the Metadata File row, select the saml-metadata.xml file that you saved in Preparing your SAML metadata file. Click Next.
    4. On the Connection Type tab select only Outbound Provisioning. Click Next.
    5. On the Connection Options tab, click Next.
    6. On the General Info tab, in the Connection Name field, enter a name of your choosing. Click Next.
3. On the Outbound Provisioning tab, configure the provisioning target and channel as shown in Configuring outbound provisioning in the PingFederate documentation.
    1. Click Configure Provisioning.
    2. On the Target tab, enter the OAuth 2 Access Token that you noted in Getting a Slack API access token.

        > **Note**
        >
        > PingFederate verifies the token when you activate the channel and SP connection.

    3. Optional: If your environment requires you to use the `userName` attribute to synchronize users between your datastore and Slack, select it from the Unique User Identifier list. Otherwise, use `primaryEmail`.

        > **Note**
        >
        > You cannot change this selection without deleting the connection. See Known issues and limitations.

    4. Under Provisioning Options, customize the provisioning connector actions as shown in Provisioning options reference. Click Next.
    5. On the Manage Channels tab, create a channel as shown in Managing channels in the PingFederate documentation. Click Done.

        > **Note**
        >
        > For more information about the attributes available in your channel configuration, see Supported attributes reference.

    6. On the Outbound Provisioning tab, click Next.
4. On the Activation and Summary tab, above the Summary section, click the toggle to turn on the connection. Click Save.

Provisioning options reference

The table provided lists the main provisioning capabilities available in the Slack connection configuration.

Copyright © 2025 Ping Identity Corporation

Field Name	Description
User Create	Selected (default) – PingFederate creates users in Slack. Cleared – PingFederate does not create users in Slack.
User Update	Selected (default) – PingFederate updates existing users in Slack. PingFederate can also re-enable disabled users. Cleared – PingFederate does not update existing users in Slack.
User Disable	Selected (default) – PingFederate disables users in Slack. Cleared – PingFederate does not disable users in Slack.

Note
If any of the above options are cleared, PingFederate logs a warning in the user workflow section of `provisioner.log` when the related action fails.

Supported attributes reference

The table provided lists the attributes that can be mapped for user provisioning to Slack.

Attribute	Description
userName	The user's unique identifier in Slack. A user's userName cannot be updated once it is created. This attribute is required.
primaryEmail	The user's email address. This attribute is required.
givenName	The given name of the user, or first name in most Western languages. For example, "Barbara" given the full name "Ms. Barbara Jane Jensen, III".
familyName	The family name of the user, or last name in most Western languages. For example, "Jensen" given the full name "Ms. Barbara Jane Jensen, III".
honorificPrefix	The honorific prefix or prefixes of the user, or title in most Western languages. For example, "Ms." given the full name "Ms. Barbara Jane Jensen, III".
displayName	The name of the user, suitable for display to end-users.
title	The user's title, such as "Vice President".
preferredLanguage	Indicates the user's preferred written or spoken languages and is generally used for selecting a localized user interface.

Copyright © 2025 Ping Identity Corporation

Attribute	Description
userType	Used to identify the relationship between the organization and the user. Typical values used might be "Contractor", "Employee", "Intern", "Temp", "External", and "Unknown", but any value can be used.
locale	Used to indicate the user's default location for purposes of localizing such items as currency, date time format, or numerical representations.
profileUrl	A URI that is a uniform resource locator that points to a location representing the user's online profile, such as a web page.
profilePhotoUrl	A URI that is a uniform resource locator that points to the user's profile photo. The resource MUST be a file (for example, a GIF, JPEG, or PNG image file) rather than a web page containing an image.
timezone	The user's time zone, in IANA Time Zone database format, such as "America/Los_Angeles".
roles	A role for the user that represents who the user is, such as "Student", "Faculty". This attribute can only accept one value.
primaryPhone	The primary phone number for the user, such as "+1-201-555-0123".
mobilePhone	The mobile phone number for the user, such as "+1-201-555-0123".
streetAddress	The work street address for the user, which may include house number, street name, P.O. box, and multi-line extended street address information.
locality	The work locality or locality component for the user's mailing address.
region	The region or state component for the user's mailing address.
postalCode	The work ZIP or postal code component for the user's mailing address.
country	The work country component for the user's mailing address.
employeeNumber	A string identifier, typically numeric or alphanumeric, assigned to a person, often based on order of hire or association with an organization.
costCenter	The cost center for the user.
organization	The organization for the user.
division	The division for the user.
department	The department for the user.

Copyright © 2025 Ping Identity Corporation

Configuring single sign-on

To allow PingFederate to act as an identity provider for Slack, create a connection and enable single sign-on (SSO) in Slack.

About this task

> **Note**
>
> SSO integration is an optional part of this integration. If you only want to use the Slack Provisioner for provisioning, skip these steps.

Steps

1. Complete the steps in Creating a single sign-on connection.
2. Complete the steps in Configuring single sign-on in Slack.

Creating a single sign-on connection

To allow PingFederate to handle single sign-on (SSO) to Slack, create a service provider (SP) SSO connection.

About this task

> **Note**
>
> You can follow these steps to create a new SP connection, or you can modify an existing connection.

> **Note**
>
> SSO isn't required for provisioning. If you only want to use the Slack Provisioner for provisioning, skip these steps.

Steps

1. In the PingFederate administrator console, create a new SP connection:
    1. For PingFederate 10.1 or later: go to Applications > Integration > SP Connections. Click Create Connection.
    2. For PingFederate 10.0 or earlier: go to Identity Provider > SP Connections. Click Create Connection.
2. Configure the basic connection details with the Slack quick connection template.
    1. On the Connection Template tab, select Use a template for this connection.
    2. From the Connection Template list, select Slack Provisioner.
    3. On the Metadata File row, select the `saml-metadata.xml` file that you saved in Preparing your SAML metadata file. Click Next.
    4. On the Connection Type tab select Browser SSO Profiles. Click Next.
    5. On the Connection Options tab, select only Browser SSO. Click Next.
    6. If you see the Metadata URL tab, clear the Enable automatic reloading check box. Click Next.
    7. On the General Info tab, in the Connection Name field, enter a name for your connection. Click Next.

Copyright © 2025 Ping Identity Corporation

3. On the Browser SSO tab, complete the steps in Configuring Browser SSO.
4. On the Credentials tab, configure the connection credentials.

   For a complete guide, see Configuring credentials in the PingFederate documentation.

   1. Click Configure Credentials.
   2. On the Digital Signature Settings tab, select your signing certificate.
   3. Clear the Include the certificate in the signature <KEYINFO> element check box. Click Next.
   4. On the Summary tab, click Done.
   5. On the SP Connection > Credentials tab, click Next.
5. On the Activation and Summary tab, above the Summary section, click the toggle to turn on the connection. Click Save.

Configuring Browser SSO

Use the following details for the Browser SSO part of your single sign-on (SSO) connection to Slack.

About this task

> **Tip**
>
> For all other settings, you can use the default or customize the configuration for your needs. For help, see Configuring IdP Browser SSO in the PingFederate documentation.

For more context on subject format requirements, see SAML NameID Format urn:oasis:names:tc:SAML:2.0:nameid-format:persistent in the Ping Identity Knowledge Center.

Steps

1. On the Browser SSO tab, click Configure Browser SSO.
2. On the Browser SSO > SAML Profiles tab, select only IDP-Initiated SSO and SP-Initiated SSO. Click Next.
3. On the Assertion Lifetime tab, click Next.
4. On the Assertion Creation tab, configure the assertion.

   The following steps only cover the critical settings for the Slack connection. For a complete guide, see Managing authentication source mappings in the PingFederate documentation.

   1. Click Configure Assertion Creation.
   2. On the Attribute Contract tab, set the following name format.

Attribute Contract	Subject Name Format
SAML_SUBJECT	`urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified`

Copyright © 2025 Ping Identity Corporation

3. In the Extended the Contract section, add the following attributes, and then click Next.

Extend the Contract Mappings

Extend the Contract	Attribute name Format
SAML_NAME_FORMAT	`urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified`
User.Email	`urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified`

4. On the Authentication Source Mapping tab, select or create your authentication source.

   For help, see Managing authentication source mappings in the PingFederate documentation.

5. On your authentication source Mapping Methodtab, select Retrieve additional attributes from a data store.

6. On your authentication source Attribute Sources & User Lookuptab, select or create your LDAP datastore.

7. On your authentication source Attribute Sources & User Lookuptab, select or create your LDAP datastore. On the LDAP Directory Search tab, use the following configuration.

   For more information, see Specifying directory properties and attributes in the PingFederate documentation.

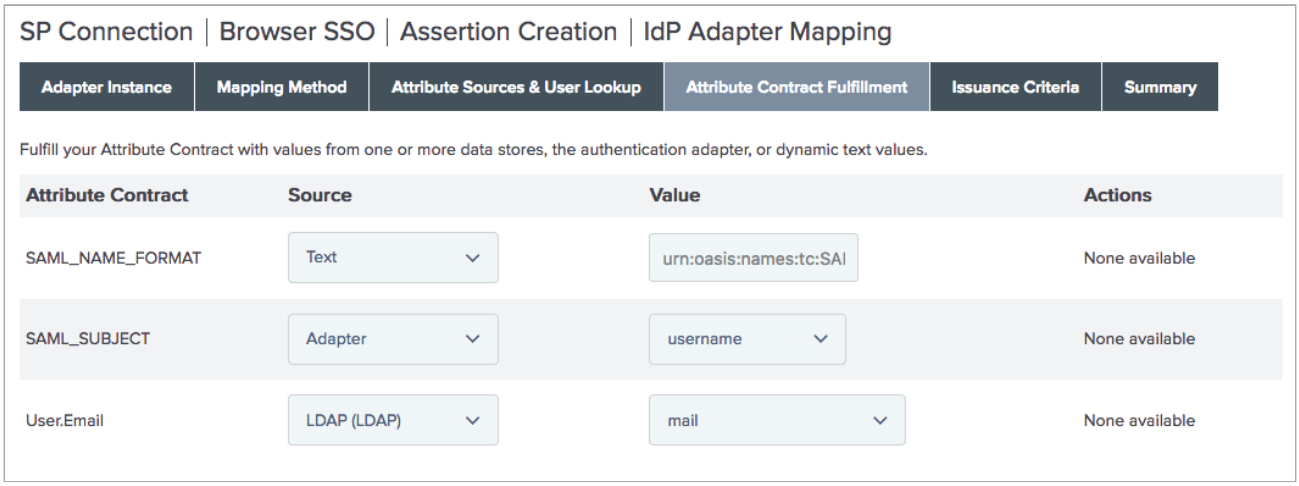
Data Store	
Attribute Source	LDAP
Attribute Source Id	LDAP
Type of Data Store	LDAP
Data Store	10.101.37.243
LDAP Directory Search	
Base DN	OU=Slack,DC=25qizl,DC=global
Search scope	SUBTREE_SCOPE
Attribute	Subject DN
Attribute	mail
LDAP Filter	
Filter	sAMAccountName=${username}

8. On the Browser SSO > Assertion Creation > IdP Adapter Mapping > Attribute Contract Fulfillment tab, map the attributes as follows. Click Next.

Copyright © 2025 Ping Identity Corporation

> **Note**
>
> This allows the connection to provide Slack with the required `urn:oasis:names:tc:SAML:2.0:nameid-format:persistent` name format, which PingFederate does not provide as a default option.

Attribute Contract	Source	Value
SAML_NAME_FORMAT	text	`urn:oasis:names:tc:SAML:2.0:nameid-format:persistent`
SAML_SUBJECT	Adapter	`username`
User.Email	LDAP	`mail`



5. On the Browser SSO > Protocol Settings > Signature Policy tab, clear the Require authn requests... and Always sign assertion check boxes. Click Next.
6. On the Encryption Policy tab, select None. Click Next.
7. On the Browser SSO > Protocol Settings tab, click Next.
8. On the Summary tab, click Done.
9. On the SP Connection > Browser SSO tab, click Next.

Configuring single sign-on in Slack

To allow PingFederate to manage authentication, enable single sign-on (SSO) in Slack.

Configuring SSO in Slack Standard or Plus

If you use Slack Standard or Plus, configure SSO as follows.

About this task

For help, see the Plus plan tab in SAML single sign-on in the Slack documentation.

Copyright © 2025 Ping Identity Corporation

Steps

1. Sign on to Slack as a Workspace Owner.
2. Click your workspace name, and then click Administration > Workspace Settings.
3. In the navigation pane, click Authentication.
4. On the Settings & Permissions page, for SAML Authentication, click Configure.
5. Select Custom SAML 2.0 and click Configure.
6. In the SAML 2.0 Endpoint (HTTP) field, enter `https://pf_host:pf_port/idp/SSO.saml2`.

   For example: `https://pf.example.com:9031/idp/SSO.saml2`
7. In the Identity Provider Issuer field, enter the SAML 2.0 Entity ID that you created in Enabling provisioning and single sign-on in PingFederate.
8. In the Public Certificate field, enter the contents of your PingFederate signing certificate.

   To get your certificate, see Exporting a certificate in Managing digital signing certificates and decryption keys in the PingFederate documentation.
9. Click Save Configuration.
10. Sign out of Slack, and then sign back on using SSO.

    Result:

    This saves your configuration and sends an email to team members inviting them set up SSO with their Slack accounts.

Configuring SSO in Slack Enterprise Grid

If you use Slack Enterprise Grid, configure SSO as follows.

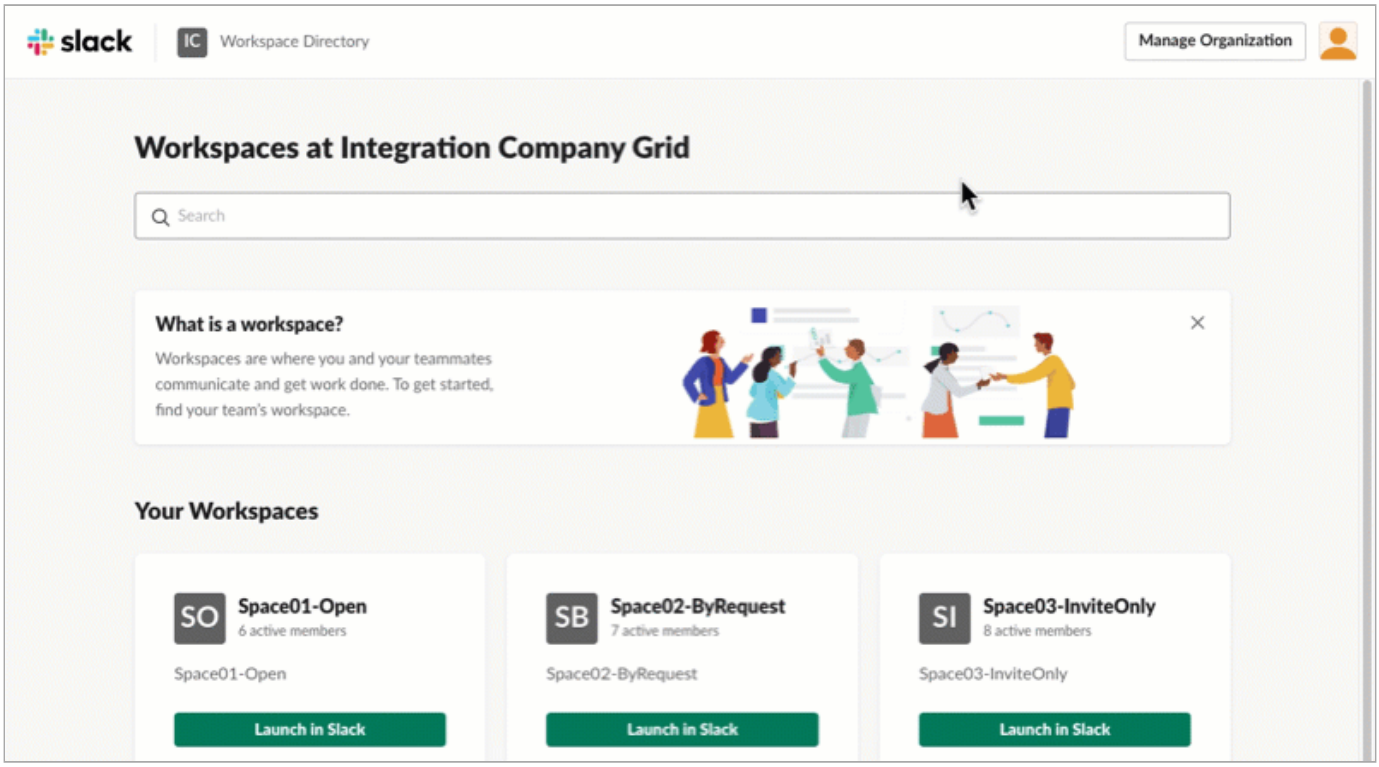
About this task

For help, see the Enterprise Grid plan tab in SAML single sign-on in the Slack documentation.

Steps

1. Sign on to Slack Enterprise Grid as a Workspace Owner.
2. Go to Manage Organization > Security > SSO Settings > Configure SSO.

Copyright © 2025 Ping Identity Corporation



3. In the SAML 2.0 Endpoint (HTTP) field, enter `https://pf_host:pf_port/idp/SSO.saml2`.

   For example: `https://pf.example.com:9031/idp/SSO.saml2`

4. In the Identity Provider Issuer field, enter the SAML 2.0 Entity ID that you created in Enabling provisioning and single sign-on in PingFederate.

5. In the Service Provider Issuer URL, use the default value of https://slack.com.

6. In the Public (X.509) Certificate field, enter the contents of your PingFederate signing certificate.

   To get your certificate, see Exporting a certificate in Managing digital signing certificates and decryption keys in the PingFederate documentation.

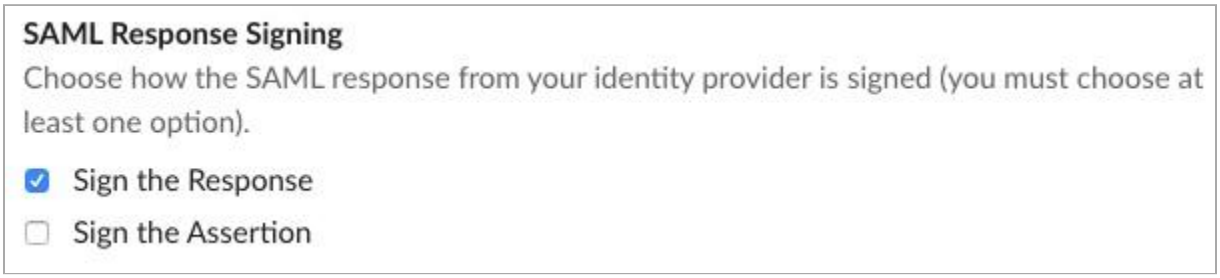
7. Enable authentication request signing.

   1. Select the Sign the AuthnRequest check box.
   2. Copy the certificate text.
   3. Create a new `.crt` file on your computer and paste the certificate text.
   4. In PingFederate, import the `.crt` file as a trusted certificate authority.

   For help, see Manage trusted certificate authorities in the PingFederate documentation.

8. Clear the Sign the Assertion check box.

Copyright © 2025 Ping Identity Corporation



9. Click Test Configuration.
10. Save your configuration.
11. Sign out of Slack, and then sign back on using SSO.

    Result:

    This saves your configuration and sends an email to team members inviting them set up SSO with their Slack accounts.

Release notes

Changelog

Slack Connector 3.0.2 – March 2020

- Added support for handling rate-limiting responses from Slack

Slack Connector 3.0.1 – November 2019

- Fixed an issue that caused the connector to update the wrong phone number attribute as a result of a change in the Slack API

Slack Connector 3.0 – October 2017

- Added deprovisioning configuration options for CRUD capabilities
- Added support for additional user attributes

Slack Connector 2.2.1 – June 2017

- Fixed an issue with deserialization the response due to an API change

Slack Connector 2.2 – May 2017

- Added configuration options for the Unique User Identifier

Slack Connector 2.1.1 – March 2017

- Optimized HTTP calls to Slack

Copyright © 2025 Ping Identity Corporation

- Fixed synchronization on update of users that were previously created with "User Create Enabled" set to false in configurable options

Slack Connector 2.1 – January 2017

- Added configuration options for CRUD capabilities

Slack Connector 2.0 – December 2015

- Added Support for Group Provisioning

Slack Connector 1.0 – October 2015

- Initial Release
- Added Support for User Provisioning

Known issues and limitations

Known issues

There are no known issues.

Known limitations

- Attributes
  - Due to a limitation in Slack, the username attribute cannot be updated after it is set.
  - Due to limitations in PingFederate, user attributes cannot be cleared after they are set.
  - The unique user identifier (`userName` or `primaryEmail`) that you choose when you create your connection to Slack can only be changed be deleting the connection, restarting PingFederate, and then creating a new connection.
  - When using multiple channels, the same unique user identifier mapping is required to coordinate manager assignments across different channels.
- Deprovisioning
  - When an LDAP user is deleted in a targeted group distinguished name (DN), the provisioning connector does not propagate the deletion until a new user is added to the group. This limitation is compounded when the User Create provisioning option is disabled. For solutions, see SaaS provisioner does not remove the user in the Knowledge Base.
- Provisioning
  - Due to a limitation in Slack, PingFederate cannot create users in a disabled state. If a user is disabled in your datastore, it will not be created in Slack by the provisioner.
  - For more information on Slack provisioning limitations, see .slack.com/scim//[SCIM API] in the Slack API documentation.

Copyright © 2025 Ping Identity Corporation

- Configuration
    - In PingFederate 8.1 and earlier, when you configure a second SP connection with the provisioning connector, the second connection may be pre-populated with the channel from the first connection. To avoid conflicts, remove this pre-populated channel and create a unique channel for each connection.

Download manifest

The following files are included in the Slack Provisioner `.zip` archive:

- `legal/Legal.pdf` – copyright and license information.
- `dist` – contains the integration files
    - `pf-slack-quickconnection-<version>.jar` – The Slack Provisioner quick connection template.
- `saml-metadata.xml` – a modifiable metadata file used to create the connection.

Copyright © 2025 Ping Identity Corporation

Symantec VIP Integration Kit

The Symantec VIP integration kit allows PingFederate to use Symantec VIP as a second authentication factor for single sign-on (SSO).

Components

- Symantec VIP IdP adapter
  - Allows PingFederate to contact the Symantec VIP application programming interface (API), which triggers a multi-factor authentication (MFA) request for the user.
  - Accepts a security code from earlier in the authentication flow to allow users to sign on without a prompt.
  - Allows users to add and reset authentication credentials.
- Security code prompt page
  - Allows users to enter a security code from Symantec VIP manually or choose an authentication method.
- Language pack
  - Allows you to customize or localize the messages shown to the user during authentication.

Intended audience

This document is intended for PingFederate administrators.

Before starting, you should familiarize yourself with the following:

- Managing IdP adapters in the PingFederate documentation
- Authentication Policies in the PingFederate documentation

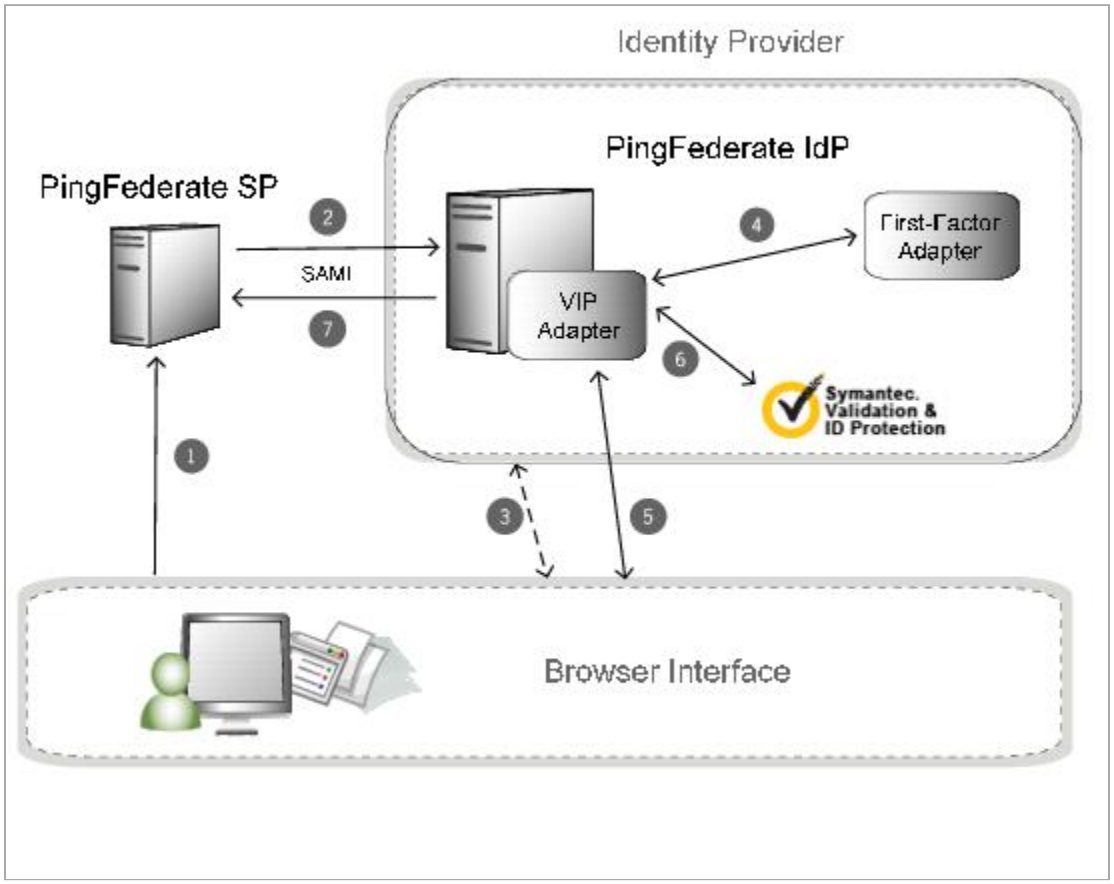
System requirements

- PingFederate 11.3 or later
- A Symantec VIP account

Overview of the SSO flow

With the Symantec VIP integration kit, PingFederate includes the Symantec VIP service in the authentication flow.

Copyright © 2025 Ping Identity Corporation



Description

1. The user initiates single sign-on (SSO) from an service provider (SP) application through a PingFederate SP server.

> **Note**
>
> This SP-initiated scenario represents the optimal use case, one in which both the identity provider (IdP) and SP are using PingFederate. However, PingFederate accepts any valid Security Assertion Markup Language (SAML) authentication request from an SP.
> In addition, you can enable IdP-initiated SSO. In this case, the user attempts SSO to an SP application from the IdP site, and the processing sequence would not include the following step.

2. The PingFederate SP server generates a SAML AuthnRequest to the PingFederate IdP server.
3. If not already signed on at the IdP (using a first-factor adapter such as Lightweight Directory Access Protocol (LDAP) or Integrated Windows authentication (IWA)), the user is challenged to authenticate.
4. The PingFederate IdP server obtains user-session information via the first-factor adapter.
5. The Symantec VIP IdP adapter requests a one-time passcode (OTP) from the user.
6. The Symantec VIP IdP adapter uses the username obtained by the first-factor adapter and the OTP to verify the user and the code via the Symantec VIP application programming interface (API).
7. If the validation succeeds, the PingFederate IdP server generates a SAML assertion with the username as the Subject and passes it to the PingFederate SP server.

Copyright © 2025 Ping Identity Corporation

8. (Not shown) The user is signed on to the SP target application.

PingFederate Authentication API support

You can use the PingFederate Authentication API to integrate the Symantec VIP IdP adapter into your application.

The PingFederate Authentication API provides access to the current state of the authentication flow as a user steps through the PingFederate authentication policy. Learn more in PingFederate Authentication API in the PingFederate documentation.

To integrate the Symantec VIP IdP adapter into your authentication flow, configure your application based on the information in this section.

> **Tip**
>
> You can also explore integrating the Symantec VIP IdP adapter into your application using the PingFederate Authentication API Explorer. Learn more in Exploring the Authentication API in the PingFederate documentation.

Models, objects, and error codes

When using the Symantec VIP integration kit through the PingFederate Authentication API, the adapter uses the following state models, action models, objects, and error codes:

Status	Request model	Action	Description
`VIP_AUTHENTICATION_REQUIRED`	**`username`** The authenticating user's username. **`pushNotificationInfo`** An object with data required to complete push notification. If number matching is enabled, the user must enter the number displayed on their device. **`VIPCredentialInfo`** An object with details about the VIP credential.	• `initiatePushAuthentication` • `selectVIPCredential` • `checkSecurityCode` • `continue`	To continue, the user must authenticate with a credential or approve a push notification.
`VIP_CREDENTIAL_REQUIRED`	**`username`** The authenticating user's username.	• `submitVIPCredential` • `continue`	The user must submit a credential to complete the enrollment process.

Copyright © 2025 Ping Identity Corporation

Status	Request model	Action	Description
`VIP_CREDENTIAL_RESET_REQUIRED`	**`username`** The authenticating user's username. **`resettableVipCredentials`** A list of `VIP Credential Info` objects with details about the resettable credentials.	• `resetVIPCredential` • `continue`	The user must reset their credential to continue.
`VIP_EMAIL_CREDENTIAL_REQUIRED`	**`username`** The authenticating user's username.	• `submitEmailCredential` • `continue`	The user must submit an email credential to complete the enrollment process.

Copyright © 2025 Ping Identity Corporation

Status	Request model	Action	Description
`VIP_EMAIL_CREDENTIAL_VERIFICATION_REQUIRED`	**`username`** The authenticating user's username. **`credentialId`** The email credential ID. **`credentialType`** The type of credential that's currently going through verification. This can be `SMS`, `Email`, or `Voice`. **`errorMessage`** An optional error message indicating the error encountered during validation, if applicable. **`informationMessage`** An optional message containing additional information about the error encountered during validation. **`friendlyName`** The friendly name of the added credential.	• `validateSecurityCode` • `continue`	The user must verify their email credential to complete the enrollment process.

Copyright © 2025 Ping Identity Corporation

Status	Request model	Action	Description
`VIP_ENROLLMENT`	**`username`** The authenticating user's username. **`vipCredentials`** A list of `VIP Credential Info` objects with details about the credential. **`suppressAddCredentials`** Indicates whether the user can add credentials. **`enableAddingSMSCredential`** Indicates whether the user can add an SMS credential. **`enableAddingVoiceCredential`** Indicates whether the user can add a voice credential. **`enableAddingEmailCredential`** Indicates whether the user can add an email credential.	• `addVIPCredential` • `continue` • `cancel`	The user can either add a credential or continue to the authentication process.
`VIP_SMS_CREDENTIAL_REQUIRED`	**`username`** The authenticating user's username.	• `submitSMSCredential` • `continue`	The user must submit an SMS credential to complete the enrollment process.

Copyright © 2025 Ping Identity Corporation

Status	Request model	Action	Description
`VIP_SMS_CREDENTIAL_VERIFICATION_REQUIRED`	**`username`** The authenticating user's username. **`credentialId`** The SMS credential ID. **`credentialType`** The type of credential that's currently going through verification. This can be `SMS`, `Email`, or `Voice`. **`errorMessage`** An optional error message indicating the error encountered during validation, if applicable. **`informationMessage`** An optional message containing additional information about the error encountered during validation. **`friendlyName`** The friendly name of the added credential.	• `validateSecurityCode` • `continue`	The user must verify their SMS credential to complete the enrollment process.
`VIP_VOICE_CREDENTIAL_REQUIRED`	**`username`** The authenticating user's username.	• `submitVoiceCredential` • `continue`	The user must submit a voice credential to complete the enrollment process.

Copyright © 2025 Ping Identity Corporation

Status	Request model	Action	Description
`VIP_VOICE_CREDENTIAL_VERIFICATION_REQUIRED`	**`username`** The authenticating user's username. **`credentialId`** The voice credential ID. **`credentialType`** The type of credential that's currently going through verification. This can be `SMS`, `Email`, or `Voice`. **`errorMessage`** An optional error message indicating the error encountered during validation, if applicable. **`informationMessage`** An optional message containing additional information about the error encountered during validation. **`friendlyName`** The friendly name of the added credential.	• `validateSecurityCode` • `continue`	The user must verify their voice credential to complete the enrollment process.

Status	Request model	Action	Description
`initiatePushAuthentication`	This action has no model.	This action has no errors.	Initiates push notification authentication.

Copyright © 2025 Ping Identity Corporation

Status	Request model	Action	Description
`selectVIPCredential`	**`vipCredentialId`** The credential ID to use. **`type`** The credential type. Possible values are `STANDARD_OTP`, `SMS_OTP`, `EMAIL_OTP`, AND `VOICE_OTP`.	• Error: `VALIDATION_ERROR` ErrorDetail: `INVALID_VIP_CREDENTIAL_ID`: Credential ID isn't valid. `AUTHENTICATION_FAILED`: Authentication failed.	Select the credential to use for authentication.
`checkSecurityCode`	**`securityCode`** The security code. **`type`** The credential type. Possible values are `STANDARD_OTP`, `SMS_OTP`, `EMAIL_OTP`, AND `VOICE_OTP`.	• Error: `VALIDATION_ERROR` ErrorDetail: `AUTHENTICATION_FAILED`: Failed to authenticate successfully with the given security code.	Validates the entered security code.
`cancel`	This action has no model.	This action has no errors.	Cancels the current authentication step.
`submitVIPCredential`	**`vipCredentialId`** The credential ID. **`securityCode`** The security code.	• Error: `VALIDATION_ERROR` ErrorDetail: `SUBMIT_VIP_CREDENTIAL_ERROR`: Error adding credential because of an incorrect security code.	Submit a security code to complete the enrollment process of the VIP credential type.

Copyright © 2025 Ping Identity Corporation

Status	Request model	Action	Description
`resetVIPCredential`	**`vipCredentialId`** The credential ID of the credential to be reset. **`securityCode`** A security code from the credential to be reset. **`nextSecurityCode`** The next security code from the credential to reset.	• Error: `VALIDATION_ERROR` ErrorDetail: `INVALID_VIP_CREDENTIAL_ID` : Invalid credential ID was provided. `RESET_VIP_CREDENTIAL_ERROR` : Error encountered resetting credential.	Process with two consecutive security codes to reset the credential.
`submitEmailCredential`	**`credentialId`** The new credential ID (email address) to add. **`friendlyName`** The friendly name of the email credential.	• Error: `VALIDATION_ERROR` ErrorDetail: `SUBMIT_EMAIL_CREDENTIAL_ERROR` : Error adding email credential.	Enter your email credentials. An OTP will be sent to the provided email address. When the OTP is presented back for verification, this completes the credential addition process.
`validateSecurityCode`	**`securityCode`** A security code received by the new credential.	• Error: `VALIDATION_ERROR` ErrorDetail: `AUTHENTICATION_FAILED` : Authentication failed.	Validate the received security code to prove device ownership and complete the credential addition process.
`addVIPCredential`	**`type`** The credential type. Acceptable values are `VIP Credential` , `SMS` , `VOICE` , and `Email` . If not specified, `VIP Credential` will be used as the default value.	• Error: `VALIDATION_ERROR`	Initiate credential enrollment process.

Copyright © 2025 Ping Identity Corporation

Status	Request model	Action	Description
`continue`	This action has no model.	This action has no errors.	Continues the current authentication flow.
`submitSMSCredential`	**`credentialId`** The new credential ID (SMS mobile device number) to add. **`friendlyName`** The friendly name of the SMS credential.	• Error: `VALIDATION_ERROR` ErrorDetail: `SUBMIT_SMS_CREDENTIAL_ERROR` : Error adding SMS credential.	Enter your mobile device number. An OTP will be sent to the provided SMS address. When the OTP is presented back for verification, this completes the credential addition process.
`submitVoiceCredential`	**`credentialId`** The new credential ID (voice mobile device number) to add. **`friendlyName`** The friendly name of the voice credential.	• Error: `VALIDATION_ERROR` ErrorDetail: `SUBMIT_VOICE_CREDENTIAL_ERROR` : Error adding voice credential.	Enter your mobile device number. An OTP will be sent to the provided phone number through a voice call. When the OTP is presented back for verification, this completes the credential addition process.

Objects

Parameter Name	Type	Description
numberChallengeValue	String	If number matching is enabled, this is the number you must enter on the VIP app.
statusCode	String	The status code of the push notification request.
statusMessage	String	The status message of the push notification request.

Parameter Name	Type	Description
id	String	The credential ID.
maskedCredentialId	String	The masked credential ID for display purposes.

Copyright © 2025 Ping Identity Corporation

Parameter Name	Type	Description
friendlyName	String	The credential's friendly name.
type	String	The credential type. Possible values are: • `STANDARD_OTP` • `SMS_OTP` • `EMAIL_OTP` • `VOICE_OTP`
authenticationInvoked	Boolean	Indicates whether authentication with associated credentials has been initiated: • For SMS, voice, and email credentials this means Symantec VIP sent a security code using the respective channel. • For `STANDARD_OTP` credentials, use the code displayed within the app or approve the push notification if push was initiated for authentication.
pushEnabled	Boolean	Indicates whether the credential is push-enabled.

Error codes

If the call flow state hasn't reached a dead end and the user can still authenticate with a device the PingFederate Authentication API returns an error code.

Error code	Message	HTTP status
`VALIDATION_ERROR`	One or more validation errors occurred.	`400`
`INVALID_ACTION_ID`	A push notification was initiated with no underlying support from the credential.	`400`

Copyright © 2025 Ping Identity Corporation

Error code	Message	userMessageKey	Parent code
`INVALID_VIP_CREDENTIAL_ID`	Credential ID isn't valid.	`vip.message.invalid.security.code`	`VALIDATION_ERROR`
`AUTHENTICATION_FAILED`	Message varies based on the error status code returned by the VIP API.	Various message properties associated with `vip.status.message.*` keys	`VALIDATION_ERROR`
`SUBMIT_EMAIL_CREDENTIAL_ERROR`	Error adding an email credential.		`VALIDATION_ERROR`
`SUBMIT_VIP_CREDENTIAL_ERROR`	Error adding a credential.		`VALIDATION_ERROR`
`RESET_VIP_CREDENTIAL_ERROR`	Error resetting a credential.		`VALIDATION_ERROR`
`SUBMIT_SMS_CREDENTIAL_ERROR`	Error adding an SMS credential.		`VALIDATION_ERROR`
`SUBMIT_VOICE_CREDENTIAL_ERROR`	Error adding a voice credential.		`VALIDATION_ERROR`

Setup

Deploying the integration files

To get started with the integration, deploy the Symantec VIP integration kit files to your PingFederate directory.

Steps

1. Download the Symantec VIP integration kit `.zip` archive from the Add-ons tab of the PingFederate downloads page or the Ping Identity Integration Directory.
2. Stop PingFederate.
3. If you're upgrading an existing deployment, back up your customizations and delete previous versions of the integration files:
    1. Back up any Symantec VIP integration kit files that you customized in `<pf_install>/pingfederate/server/default/conf/`.
    2. Delete the `yep` file from `<pf_install>/pingfederate/server/default/deploy`.
4. Extract the `.zip` archive and merge the contents of the `dist` directory with your `<pf_install>/pingfederate` directory.

Copyright © 2025 Ping Identity Corporation

5. If there is more than one version of the `pf-authn-api-sdk-<version>.jar` file in your `<pf_install>/pingfederate/server/default/lib` directory, delete all but the latest version of the file.
6. If you backed up any customized files, modify the new files with your customizations.
7. Start PingFederate.
8. If you operate PingFederate in a cluster, repeat steps 2 - 7 for each engine node.

Importing the Symantec certificate into PingFederate

To allow PingFederate and Symantec VIP to communicate securely, import the Symantec signing certificate into PingFederate.

Steps

1. In the Symantec VIP Manager portal, export your SSL client certificate in the PKCS#12 format.

   To do so, complete the steps in Managing VIP Certificates in the Symantec VIP Manager documentation.
2. In PingFederate, go to the certificate import page:
   1. Go to Security > Certificate & Key Management > SSL Client Keys & Certificates.
   2. On the SSL Client Keys & Certificates window, click Import.
3. On the Import Certificate tab, select your Symantec certificate and enter the password. Click Next.
4. On the Summary tab, click Save.

Configuring an adapter instance

Configure the Symantec VIP IdP adapter to allow PingFederate to use Symantec VIP as an authentication factor.

Steps

1. Sign on to the PingFederate administrative console.
2. On the Identity Provider > Adapters tab, click Create New Instance.
3. On the Type tab, set the basic adapter instance attributes.
   1. In the Instance Name field, enter a name for the adapter instance.
   2. In the Instance ID field, enter a unique identifier for the adapter instance.
   3. In the Type list, select Symantec VIP Adapter. Click Next.
4. On the IdP Adapter tab, configure the adapter instance by referring to Symantec VIP IdP adapter settings reference. Click Next.
5. On the Extended Contract tab, add any attributes that you expect to retrieve in addition to the core contract attributes. Click Next.
6. Complete the adapter configuration.

Copyright © 2025 Ping Identity Corporation

7. On the Summary tab, check that the configuration is correct. Click Done.
8. On the Manage IdP Adapter Instances tab, click Save.
9. Modify your PingFederate authentication policy to pass attributes to your Symantec VIP IdP adapter instance from earlier in the authentication flow.

   The adapter expects the `username` attribute and accepts the optional security code attribute you defined in the Security Code Attribute Name field.

   Learn more about configuring authentication policies in Authentication policies in the PingFederate documentation.

Symantec VIP IdP adapter settings reference

The following tables show field descriptions for the Symantec VIP Adapter configuration page.

Field	Description
Client Certificate	Select the VIP Manager certificate that you downloaded and imported into PingFederate.
VIP Configuration Type	Selection options are: ***Pilot*** The adapter tests the connection to the VIP application programming interface (API), which is used to verify user-token authentication. ***Production*** The adapter runs as normal. **Note** The standard one-time passcode (OTP) security-code generators for VIP are usable only for production. They don't provide valid codes for a pilot configuration. Instead, you must obtain the VIP Test Drive OTP generator for pilot testing. If you have a specific URL to use for the API, enter it in the Advanced Fields section.

Copyright © 2025 Ping Identity Corporation

Field	Description
Default Authentication Method	Determines whether the adapter defaults to push, SMS, or voice, or prompts the user to select an authentication method for the current session. The selection options are: *None (default)* The adapter prompts the user to choose an authentication method or enter a Symantec VIP security code. This is the default selection. *Security Code* The adapter prompts the user to enter a Symantec VIP security code. *Push Notification* The adapter sends a push notification to the Symantec VIP app on the user's mobile device. *SMS and Voice Call* The adapter sends a text or voice authentication message to the user's phone number. *Email* The adapter sends a security code to the user's email address. **Note** If the default method isn't valid for a user, the adapter prompts the user to select another authentication method.
Override Default Authentication Method	When enabled, the adapter checks the Security Code Attribute Name defined in the following row. • If the user has a security code in the data store, the adapter passes it to Symantec VIP, allowing the user to skip any prompts. • If the user does not have a security code in the data store, the adapter falls back to the default authentication method.
Security Code Attribute Name	The name of the attribute in your data store that contains a user's Symantec VIP security code. **Note** The adapter checks this attribute when **Override Default Authentication Method** is enabled.

Copyright © 2025 Ping Identity Corporation

Field	Description
Create User	Select this checkbox to automatically create a new user in Symantec VIP if the user doesn't already exist. The adapter uses the incoming `User ID` value as the user ID for the new account. This checkbox is cleared by default.
Suppress Add Credential	When selected, users aren't shown the interface to register new credentials, such as phone numbers or email addresses. **Important** If you are using this adapter instance in a password reset flow, select this checkbox. This prevents users from bypassing authentication by adding credentials during the password reset flow.
Enable Adding SMS Credential	Select this checkbox to allow users to add SMS credentials. This checkbox is cleared by default.
Enable Adding Voice Credential	Select this checkbox to allow users to add Voice Call credentials. This checkbox is cleared by default.
Enable Adding Email Credential	Select this checkbox to allow users to add Email credentials. This checkbox is cleared by default.
Enable Number Matching in Push Notifications	Select this checkbox to allow users to enter a number to match the number displayed in the push notification. This checkbox is cleared by default.
Visible Leading Characters Length	The number of leading characters to display in the credential ID. The default value is `3`.
Visible Trailing Characters Length	The number of trailing characters to display in the credential ID. The default value is `3`.
Push Request Timeout	The timeout for push requests, in seconds. The default value is `60`.
Challenge Retries	The maximum number of times that a user can try to authenticate before authentication fails.

Copyright © 2025 Ping Identity Corporation

Field	Description
API URL Override	Overrides the API URL defined by the selected VIP Configuration Type. Use a URL override to connect to the API service if you have a non-standard pilot or production instance of Symantec VIP. By default, the adapter uses the following URL for a production configuration: https://userservices-auth.vip.symantec.com/vipuserservices/AuthenticationService_1_6
Management API URL Override	Overrides the Management API URL defined by the selected VIP Configuration Type. Use a URL override to connect to the Management API service if you have a non-standard pilot or production instance of Symantec VIP. By default, the adapter uses the following URL for a production configuration: https://userservices-auth.vip.symantec.com/vipuserservices/ManagementService_1_6
Query API URL Override	Overrides the Query API URL defined by the selected VIP Configuration Type. Use a URL override to connect to the Query API service if you have a non-standard pilot or production instance of Symantec VIP. By default, the adapter uses the following URL for a production configuration: https://userservices-auth.vip.symantec.com/vipuserservices/QueryService_1_6
VIP API URL Override	Overrides the VIP API URL defined by the selected VIP Configuration Type. Use a URL override to connect to the VIP API service if you have a non-standard pilot or production instance of Symantec VIP. By default, the adapter uses the following URL for a production configuration: https://services-auth.vip.symantec.com/mgmt/soap
HTML Template Prefix	Identifies the set of HTML templates the adapter users during authentication. You can find a description of the template files in Download manifest. If you customize the file names of the templates in the `/server/default/conf/template` directory, enter the new prefix in this field. The default value is `vip`.

The adapter contract returns the following attributes when you make a call to it:

Copyright © 2025 Ping Identity Corporation

Attribute	Description
`subject` (core attribute)	Specifies the username obtained by the first-factor adapter.
`credential_id` (non-core attribute)	Specifies the credential ID used to sign on to Symantec VIP.
`credential_type` (non-core attribute)	Specifies the type of credential used to sign on to Symantec VIP.

Customizing templates

The Symantec VIP integration kit includes templates that can display Symantec VIP authentication challenges and other messages. You can customize the HTML portions of these templates to suit your organization's branding and presentation needs.

The following table lists the available templates and their uses. You can find more information about the templates in Customizable user-facing pages in the PingFederate documentation.

Attribute	Description
`vip.enrollment.html`	Initial form presented to the user. Offers the option to add a new credential (if configured) or proceed with authentication using an existing credential.
`vip.security_code_challenge.html`	Form presented to the user to enter a security code for authentication. Offers the option to choose alternative credential types if those methods are available to the user. For example, push notification, SMS, voice, or email.
`vip.security_code_reset.html`	Form presented to the user when they begin the reset credential flow.
`vip.add_voice_credential.html`	Form presented to the user when they add a new voice credential.
`vip.add_email_credential.html`	Form presented to the user when they add a new email credential.
`vip.add_sms_credential.html`	Form presented to the user when they add a new SMS credential.
`vip.add_credential.html`	Form presented to the user when they verify ownership of a new credential using a security code they received.

The `vip-messages.properties` file contains configurable properties that allow administrators to customize the messages displayed in the previous templates for various scenarios. The properties file has inline comments providing more context. Keys are self-descriptive.

Copyright © 2025 Ping Identity Corporation

Customizing user messages

The Symantec VIP integration kit allows you to customize or translate the messages shown to the user on the authentication prompt.

About this task

Learn more about localization in PingFederate in Localizing messages for end users.

Steps

1. In the `<pf_install>/pingfederate/server/default/conf/language-packs` directory, open the `vip-messages.properties` file for editing.
2. Translate or modify the messages.
3. Save the file with a suffix to indicate the language, such as `vip-messages.properties_fr` for French.
4. Restart PingFederate.
5. If you operate PingFederate in a cluster, repeat steps 1 - 3 for each instance of PingFederate.

User credential management

Users sign on to their Symantec VIP IdP adapter account using a credential that's associated with their Symantec VIP account.

In Symantec VIP, a credential is an authentication method (such as voice or SMS) that is paired with a unique credential ID. The credential ID is shown on the user's VIP Access mobile app, or on their physical VIP security token or card.

Adding credentials

When you create a user account in Symantec VIP, you must add at least one credential for the user. Additional credentials let the user sign on with different devices or authentication methods.

You can add additional credentials on behalf of a user, or allow users to add their own credentials through the Symantec VIP IdP adapter sign-on page. Learn more in:

- Adding credentials as an administrator
- Adding credentials as a user

The Suppress Add Credentials setting in the Symantec VIP IdP adapter instance configuration determines whether users are allowed to add credentials.

Resetting credentials

If a user fails to authenticate twice in a row, the Symantec VIP IdP adapter prompts the user to reset their credentials, as shown in Resetting credentials as a user. This self-service prevents users from locking their accounts and reduces the workload for administrators.

Copyright © 2025 Ping Identity Corporation

Adding credentials as an administrator

As an administrator, you can manually add credentials to a user's account. This allows them to sign on with Symantec VIP using another device or authentication method.

Steps

1. Sign on to the Symantec VIP Manager portal as an administrator.
2. On the Users tab, go to the user that you want to modify and click Edit Details.
3. In the Credentials section, click Add Another.
4. In the Type list, select the desired authentication method.
5. In the Credential ID field, enter the user's credential ID.

   You can find the ID on the user's VIP Access mobile app, or on their physical VIP security token or card.
6. (Optional) If the Extension field appears, enter a phone extension.

   The extension can include hashtags, commas, asterisks, periods, and the letter `x`, but only if numbers precede these characters.
7. In the Name field, enter a name, then click Save.

Adding credentials as a user

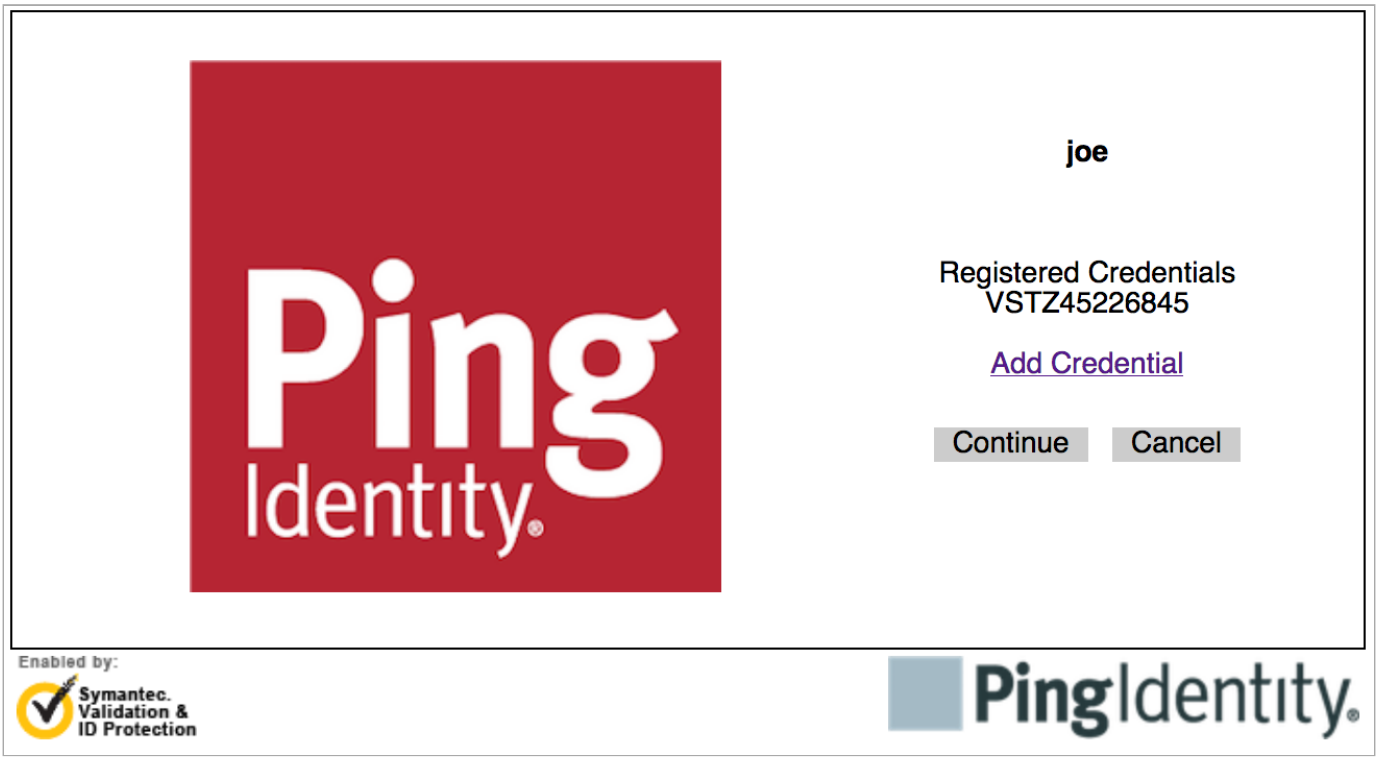
The Symantec VIP IdP adapter allows users to add credentials as a self-service by default.

Steps

1. Open the VIP Access app on your mobile device.
2. Start the PingFederate authentication flow by signing on to the first-factor authentication adapter.

   For example, this could be an HTML Form Adapter or OpenToken Adapter instance.
3. On the Symantec VIP IdP adapter, sign on with the same username and password.
4. Click Add Credential.

Copyright © 2025 Ping Identity Corporation



5. On the Add Credentials page, enter the credential ID and security code shown on your VIP Access mobile app, or on your physical VIP security token or card. Click Submit.
6. On the sign-on screen, select Security Code, and enter the next security code that appears on your device. Click Submit.

Copyright © 2025 Ping Identity Corporation



Resetting credentials as a user

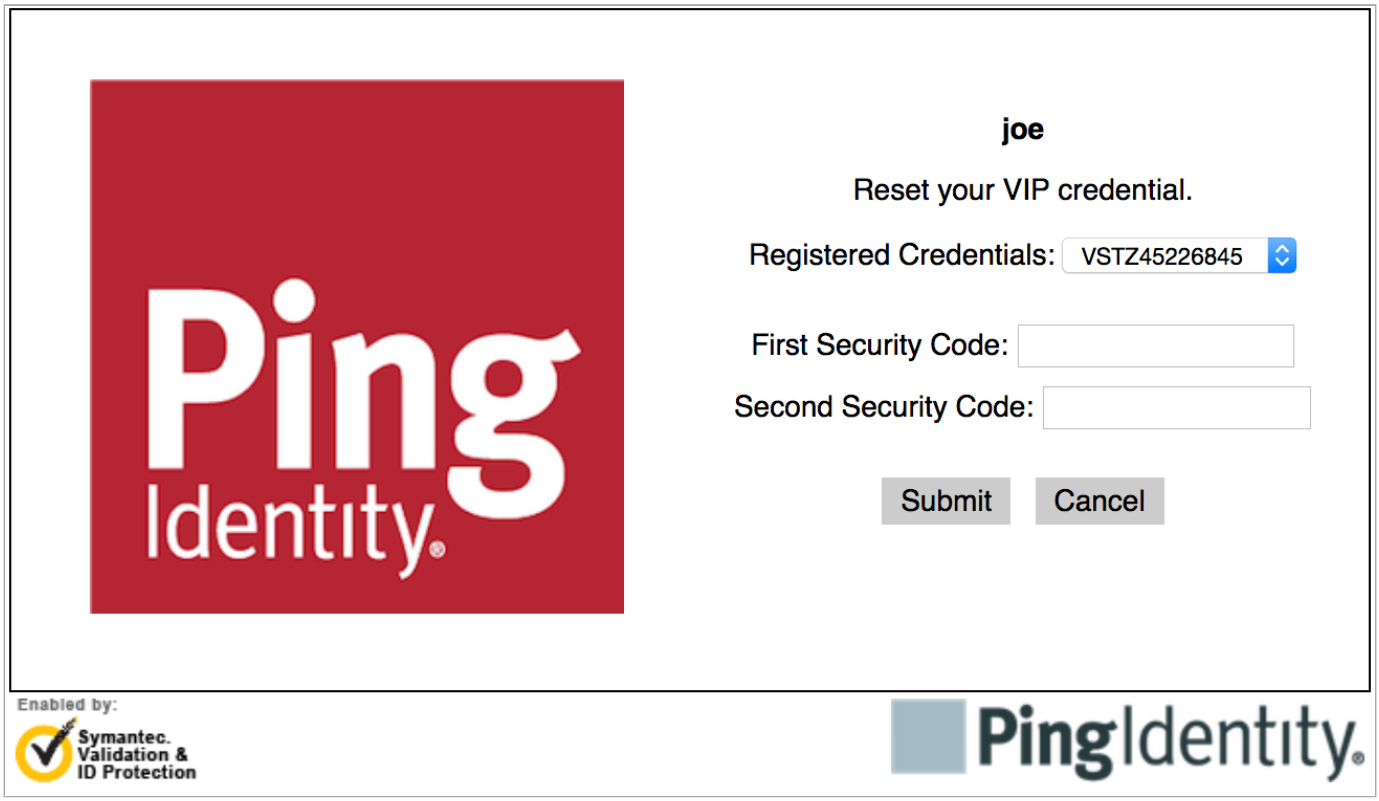
Users can reset their own credentials after failing to authenticate twice in a row.

Steps

1. Open the VIP Access app on your mobile device.
2. Start the PingFederate authentication flow by signing on to the first-factor authentication adapter.

   For example, this could be an HTML Form Adapter or OpenToken Adapter instance.
3. On the Symantec VIP IdP adapter, sign on with the same username and password.
4. On the Reset your VIP credential page, in the First Security Code field, enter the security code shown on your VIP Access mobile app or physical VIP security token or card.

Copyright © 2025 Ping Identity Corporation



5. In the Second Security Code field, enter the next security code that appears. Click Submit.

Troubleshooting

The following solutions address common situations you might encounter with the Symantec VIP integration kit:

Situation	Recommendation
You receive an error in the `server.log` file that says Operation not allowed in current user status.	The user is disabled. Enable the user account in Symantec VIP and try again.
You receive an error in the `server.log` file that says No credentials available for reset, failing authentication.	The user doesn't have any standard credentials available. Have the user install the Symantec VIP mobile app or use an SMS or voice credential instead.

Release notes

Changelog

The following is the change history for the Symantec VIP integration kit.

Copyright © 2025 Ping Identity Corporation

Symantec VIP integration kit 2.0 - July 2025

- Added the ability to obscure personally identifiable information (PII) in displayed user credentials. Learn more in the Visible Leading Characters Length and Visible Trailing Characters Length table entries in Symantec VIP IdP adapter settings reference.
- Added the ability to create users automatically. Learn more in the Create User table entry in Symantec VIP IdP adapter settings reference.
- Added support for number challenges in push notification authentication. Learn more in the Enable Number Matching in Push Notifications table entry in Symantec VIP IdP adapter settings reference.
- Added the ability for users to configure and authenticate with SMS, voice, and email credentials. Learn more in the Enable Adding SMS Credential - Enable Adding Email Credential table entries in Symantec VIP IdP adapter settings reference.
- Added documentation for the PingFederate authentication API. Learn more in PingFederate Authentication API support.
- Removed legacy web application authentication and added support for template-based authentication. Learn more in Customizing templates.

> **Note**
>
> The `vip-adapter-security-code-challenge.war` directory is no longer in use. If you created a `.war` file to make customizations to this directory previously, you can re-implement your changes with the new HTML templates after upgrading the adapter.

- Fixed an issue that allowed users without any configured credentials to attempt to authenticate, leading to a dead end in the authentication flow.

> **Note**
>
> This issue was applicable if the **Suppress Add Credential** checkbox was selected in the adapter configuration.

Symantec VIP integration kit 1.8.1 – February 2024

- Fixed an issue that caused failures if a temporary code was used.

Symantec VIP integration kit 1.8 – July 2023

- Added two new exposed attributes in the adapter contract, `credential_type` and `credential_id`. You can use these attributes to do branching logic in PingFederate authentication policies.
  - `credential_type` specifies the type of credential used to sign on to Symantec VIP.
  - `credential_id` specifies the credential ID used to sign on to Symantec VIP.

Symantec VIP integration kit 1.7.1 – March 2023

- Fixed an issue that caused some sensitive information to display in the JSP.

Copyright © 2025 Ping Identity Corporation

Symantec VIP integration kit 1.7 – July 2022

- Removed adapter-based session management.
- Added the ability to retrieve the targeted application's Application Name from the SP Connection in PingFederate when a push notification is triggered for an account.

Symantec VIP integration kit 1.6 – September 2021

- Added support for the JavaScript Widget for the PingFederate Authentication API.
- Added support for the PingFederate authentication API.

Symantec VIP integration kit 1.5 – May 2020

- Added support for the PingFederate localization framework.
- Added the ability to select a default authentication method or allow users to choose.
- Added the ability to skip the authentication prompt when a one-time password is provided.

Symantec VIP integration kit 1.4 – January 2018

- SMS and voice are now supported for one–time passcode delivery.
- Added ability to configure the maximum number of retry attempts for the one–time passcode.
- Bug fixes.

Symantec VIP integration kit 1.3 – October 2016

- Added support for VIP's Access Push feature, which replaces the need for the user to manually enter a security code, by sending a push verification to the registered mobile device upon sign–in.
- Validated solution against VIP's latest API changes (SHA–1 to SHA–2 migration).
- Updated html templates with the latest UI theme.

Symantec VIP integration kit 1.2 – August 2015

- Added support to suppress the register credentials workflow.

Symantec VIP integration kit 1.1 – May 2014

- Added the ability to override default VIP endpoint URLs.
- Added support to register credentials.
- Added support to reset credentials after two consecutive failed authentication attempts.
- Added support for Symantec VIP API 1.1

Symantec VIP integration kit 1.0.1 – July 2011

- Corrected malformed HTML which would prevent the Submit button from rendering on Internet Explorer 8 and Firefox 5 browsers

Copyright © 2025 Ping Identity Corporation

Symantec VIP integration kit 1.0 – January 2011

- Initial release

Known issues and limitations

The following are known issues or limitations for the Symantec VIP integration kit:

Known issues

There aren't any known issues.

Known limitations

There aren't any known limitations.

Download manifest

The following items are included in the Symantec VIP integration kit:

- `Legal.pdf` : A file that contains copyright and license information.
- `dist/pingfederate/server/default` : A directory that contains the Java libraries needed to run the adapter.
  - `deploy` : A directory that contains the integration files.
    - `pf-vip-idp-adapter-<version>.jar` : A JAR file that contains the Symantec VIP IdP adapter.
  - `lib` : Contains common JAR files.
    - `pf-authn-api-sdk-<version>.jar` : A JAR file that contains the PingFederate Authentication API SDK.
  - `conf` : A directory that contains language template files.
    - `language-packs/vip-messages.properties` : A variable file that customizes the messages that appear on the template file.
    - `template` : A directory that contains language template files.
      - `assets` : A directory that contains style, font, and image files for the template files.
        - `css/vip.css` : A CSS file that customizes the appearance of the template file.
        - `end-user/<version>/end-user.css` : A CSS file that customizes the appearance of the template files.
        - `fonts/end-user/icons` : Contains template icons.
        - `images` : Contains template image files.
          - `logo.png` : An image file with company branding.
          - `ping-logo-horizontal.png` : An image file with company branding.

Copyright © 2025 Ping Identity Corporation

        - `ping-logo.svg`: An image file with company branding.
        - `product.png`: An image file with company branding.
        - `spinner.svg`: An image file used in a spinner animation.
        - `VIP_logo_RGB.png`: An image file with company branding.
    - `scripts`: Contains script files used to collect and send information.
        - `jquery-<version>.min.js`: A JavaScript file that contains the jQuery library.
        - `vip.js`: A JavaScript file that contains common functions used by templates.
- `vip.add_credential.html`: Template displayed to the user during verification of a new credential using the received security code.
- `vip.add_email_credential.html`: Template displayed to the user when the add email credential process is initiated.
- `vip.add_sms_credential.html`: Template displayed to the user when the add SMS credential process is initiated.
- `vip.add_voice_credential.html`: Template displayed to the user when the add voice credential process is initiated.
- `vip.enrollment.html`: Template displayed to the user to add a new credential (if configured on the adapter) or proceed with authentication using preexisting credentials.
- `vip.security_code_challenge.html`: Template displayed to the user for entering a security code for authentication, with the option to choose alternative credential types such as Push Notification, SMS, Voice, or Email if those methods are available to the user.
- `vip.security_code_reset.html`: Template displayed to the user when the user goes through the reset credential flow.

Copyright © 2025 Ping Identity Corporation

ThreatMetrix Integration Kit

The ThreatMetrix Integration Kit allows PingFederate to communicate with ThreatMetrix for risk-based authentication.

By sending a device profile and optional user attributes to ThreatMetrix when a user signs on, PingFederate can get a security risk assessment for the sign-on event. You can use this to dynamically adjust the authentication requirements each time a user signs on. For example, by configuring policies in PingFederate and ThreatMetrix, you could require multi-factor authentication (MFA) when a user signs on with a new device.

Features

- Supports the ThreatMetrix "web" and "SDK" device profiling scripts. For details, see Introduction to Profiling in the ThreatMetrix documentation.
- Supports device profiling in any browser-based authentication source or web application. For details, see Device profiling methods.
- Supports the PingFederate PingFederate authentication API.
- Supports the JavaScript Widget for the PingFederate Authentication API.

Components

- Template and script files
  - When a user signs on through PingFederate, these files create the device profile and send it to ThreatMetrix. There are several files to accommodate a variety of device profiling methods.
- ThreatMetrix IdP Adapter
  - When a user signs on through PingFederate, the adapter sends the user attributes to ThreatMetrix.
  - The adapter receives the result of the risk assessment as well as other attributes and sign-on event data. The adapter makes this information available in the PingFederate authentication policy.

Intended audience

This document is intended for PingFederate administrators.

If you need help during the setup process, see the following resources:

- The following sections of the PingFederate documentation:
  - Managing IdP adapters
  - Authentication policies
- The following sections of the ThreatMetrix documentation:
  - Introduction to ThreatMetrix Platform
  - Architecture Overview
  - Implementation Resources

Copyright © 2025 Ping Identity Corporation

- Implementation Steps

System requirements

- PingFederate 9.3 or later
- A ThreatMetrix account

Overview of ThreatMetrix

ThreatMetrix evaluates the level of security risk for a user sign-on event based on a device profile and user attributes.

Device profile

The device profile is collected by a JavaScript script that runs during the sign-on flow. There are various ways to collect the device profile, as described in Device profiling methods.

ThreatMetrix also provides two script options. The ThreatMetrix SDK script runs locally, and the ThreatMetrix Web script fetches the latest device profiling script from ThreatMetrix each time. For more information, see Introduction to Profiling in the ThreatMetrix documentation.

Session ID

ThreatMetrix assigns a session ID to every authentication session. This session ID is associated with the device profile, and can be used to send in additional (optional) user attributes from the ThreatMetrix IdP Adapter. It also allows the adapter to get the resulting review status and reason code from the risk assessment.

In some device profiling methods, the session ID is passed to or from the ThreatMetrix IdP Adapter to coordinate sending information to ThreatMetrix from multiple sources for the same authentication session.

User attributes

When sending the device profile to ThreatMetrix, you can also provide user attributes such as name, address, and email. Use these in your ThreatMetrix policies to affect risk assessments.

For the complete list of attributes that ThreatMetrix can collect, see Session Query API in the ThreatMetrix documentation.

Review statuses

ThreatMetrix evaluates risk for a sign-on event by using configurable policies. The result is a "review status" value of "pass", "review", "challenge", or "reject".

You can configure your PingFederate authentication policy to determine how each of the "pass", "review", "challenge", and "reject" results affects a user's ability to sign on. For example, you can prompt a user for a second authentication factor if their review status is "review".

Attributes and sign-on event data in the response from ThreatMetrix

The response from ThreatMetrix also contains attributes and sign-on event data.

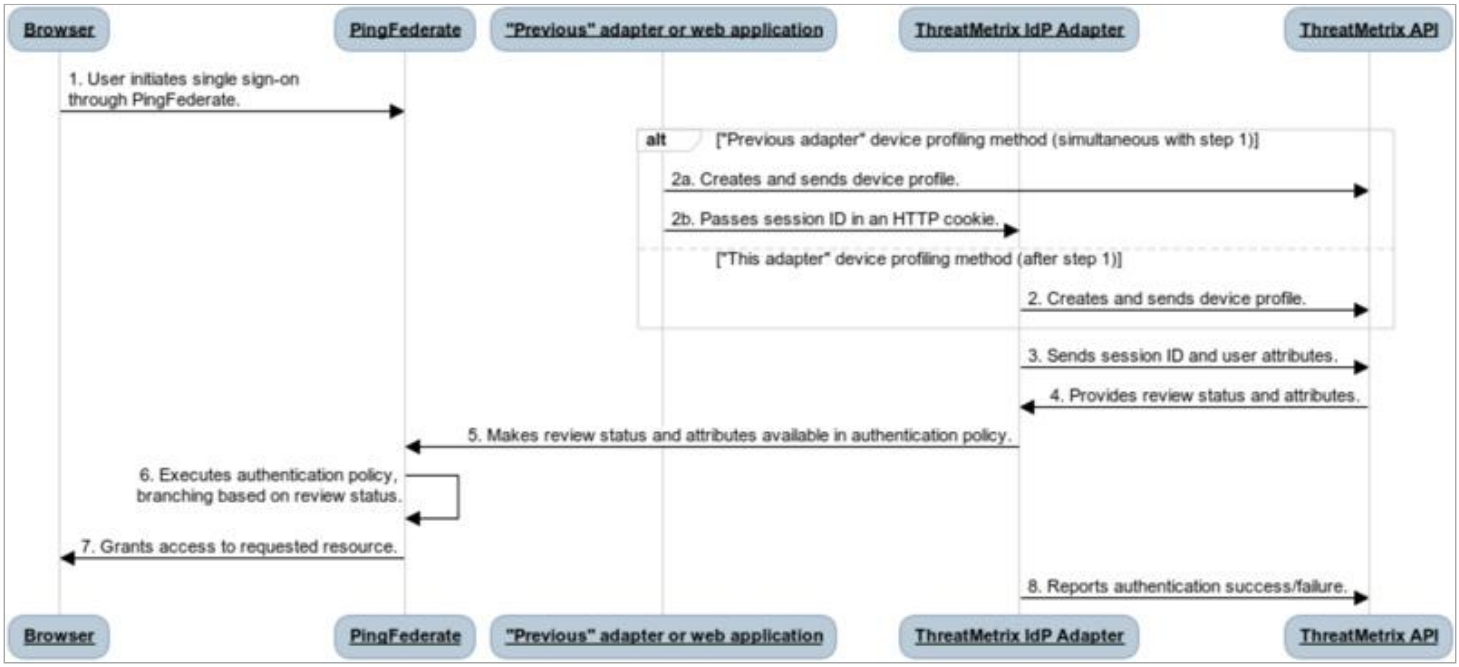
Copyright © 2025 Ping Identity Corporation

In your ThreatMetrix IdP Adapter instance configuration, you can capture this information and make it available to other adapters and contracts in the PingFederate authentication policy.

Overview of the SSO flow

With the ThreatMetrix Integration Kit, PingFederate includes ThreatMetrix in the sign-on flow.

The following figure shows how ThreatMetrix is integrated into the sign-on process:



Description

1. A user initiates the sign-on process by requesting access to a protected resource.
2. Depending on the device profiling method, the ThreatMetrix IdP Adapter or a previous authentication adapter collects the device profile and sends it back to ThreatMetrix with a session ID.

   For the "captured by a previous adapter" device profiling method, this step takes place at the same time as step 1.
3. The ThreatMetrix IdP Adapter sends the session ID and any optional user attributes to ThreatMetrix.
4. ThreatMetrix responds with the review status("pass", "review", "challenge", or "reject") as well as additional attributes and sign-on event data.
5. The ThreatMetrix IdP Adapter makes the review status and attributes and sign-on event data available in the PingFederate authentication policy.
6. PingFederate continues executing the authentication policy, which branches based on the review status provided by the adapter.
7. If the user authenticates successfully, PingFederate returns the resource that the user requested.

Copyright © 2025 Ping Identity Corporation

8. Optional: If the review status was ""review"" and authentication ultimately succeeded, the adapter notifies ThreatMetrix. This allows ThreatMetrix to train models and tune policies for future sign-on attempts.

Device profiling methods

ThreatMetrix requires a device profile to determine a review status. There are several methods for capturing the device profile and sending it to ThreatMetrix.

There are several device profiling methods to accommodate:

- Where your users sign on, such as:
  - A PingFederate adapter template, such as the HTML Form Adapter or ThreatMetrix IdP Adapter
  - A web app that uses the PingFederate authentication API
  - A mobile or native app that uses the PingFederate authentication API
- Whether you are able to modify the sign-on page or app to:
  - Run a device profiling script
  - Pass a session ID to the ThreatMetrix IdP Adapter through an HTTP cookie
  - Pass a session ID to the ThreatMetrix IdP Adapter through the PingFederate authentication API

Choosing a device profiling method

The "basic" methods below are simpler to set up, but the user has to wait during the device profiling process.

The "enhanced" methods are more complex to set up, but the device profiling process happens in the background before the ThreatMetrix IdP Adapter is triggered by the PingFederate authentication API. This eliminates the perceived wait time for the users.

With all device profiling methods, the ThreatMetrix IdP Adapter uses a session ID to pass additional (optional) attributes to ThreatMetrix.

Compare the device profiling methods in the table and descriptions below, and decide which is the best fit for your environment. Detailed instructions are provided in the setup steps.

Comparison of device profiling methods

Method	Authentication mode	Session ID created by	Profile captured by	Profile submitted by	Device Profilingsetting	Notes
Built-in (basic)	PingFederate template	ThreatMetrix IdP Adapter	ThreatMetrix IdP Adapter	ThreatMetrix IdP Adapter	Create a new device profile	User waits while device is profiled

Copyright © 2025 Ping Identity Corporation

Method	Authentication mode	Session ID created by	Profile captured by	Profile submitted by	Device Profilingsetting	Notes
Web app (basic)	Authentication API	ThreatMetrix IdP Adapter	Web app	Web app	Create a new device profile	User waits while device is profiled Requires script setup
Web app (enhanced)	PingFederate template or Authentication API	Web app	Web app	Web app	Use existing session ID	Requires script setup Passes session ID in cookie
Mobile or native app (enhanced)	Authentication API	Mobile or native app	Mobile or native app	Mobile or native app	Use existing session ID	Requires ThreatMetrix SDK Passes session ID via API call

Built-in (basic)

When the ThreatMetrix IdP Adapter is triggered in the sign-on flow, it inserts a page that runs the device profiling script.

> **Note**
>
> This method does not require modifications to any other pages, but users must wait while the device profile is captured. The length of the wait depends on your environment.

Sequence:

1. The user arrives at the first-factor sign-on page and enters their credentials.
2. The ThreatMetrix IdP Adapter is triggered by the PingFederate authentication policy. It presents a "spinner" animation page that runs the device profiling script. The script sends the device profile to ThreatMetrix with a unique session ID.
3. The ThreatMetrix IdP Adapter requests a review status by sending the session ID and any user attributes to ThreatMetrix.

Web app (basic)

If you have a web app that uses the PingFederate authentication API, you can add the device profiling script to your existing sign-on page.

> **Note**
>
> This method requires you to add a device profiling script to your existing page, and users must wait while the device profile is captured. The length of the wait depends on your environment.

Copyright © 2025 Ping Identity Corporation

Sequence:

1. The user arrives at the sign-on page and enters their credentials.
2. The ThreatMetrix IdP Adapter is triggered by the PingFederate authentication policy.
3. The web app gets the session ID from the authentication API and runs the device profiling script. The script sends the device profile to ThreatMetrix with the session ID.
4. The web app tells PingFederate to continue the authentication flow.
5. The ThreatMetrix IdP Adapter requests a review status by sending the session ID and any user attributes to ThreatMetrix.

This method is recommended if you have a web app and cannot accommodate HTTP cookies.

Web app (enhanced)

To reduce perceived wait times for the user, you can run the device profiling script while the user interacts with a web page that is already part of the sign-on flow.

You can integrate the device profiling script into any web page that meets the following criteria:

- The user sees your sign-on page before the ThreatMetrix IdP Adapter is triggered in your PingFederate authentication policy.
- The page is hosted in the same domain as your PingFederate server.

  This is required to accommodate the HTTP cookies that passes the ThreatMetrix session ID to the ThreatMetrix IdP Adapter.

  You might be able to work around this requirement by consolidating your domains with a reverse proxy server.

For example, you can use this method with the following:

- The HTML Form Adapter, or another PingFederate adapter that presents a web page
- A web app that uses the PingFederate authentication API

Sequence:

1. The user arrives at a first-factor sign-on page presented by the HTML Form Adapter or your web app.
2. While the user interacts with the page (for example, entering their username and password), the device profiling script sends the device profile to ThreatMetrix with a unique session ID. The script also stores the session ID in an HTTP cookie.
3. The ThreatMetrix IdP Adapter gets the session ID from the HTTP cookie.
4. The ThreatMetrix IdP Adapter sends the session ID and any user attributes to ThreatMetrix and requests the review status.

Mobile or native app (enhanced)

If your users authenticate through a mobile or native app, you can use the ThreatMetrix SDK to capture the device profile. Your app can then provide the ThreatMetrix session ID to PingFederate to continue the authentication flow.

Copyright © 2025 Ping Identity Corporation

ThreatMetrix provides SDKs for Android, iOS, OSX, Windows, and Java. For details, see Introduction to ThreatMetrix SDK and FAQ in the ThreatMetrix documentation.

Sequence:

1. The user starts the authentication process in your mobile or native app.
2. While the user interacts with the app (for example, entering their username and password), the ThreatMetrix SDK captures the device profile and sends it to ThreatMetrix. The app then sends the ThreatMetrix session ID to PingFederate.
3. The ThreatMetrix IdP Adapter sends the session ID and any user attributes to ThreatMetrix and requests the review status.

Setup

Deploying the integration files

To get started with the integration, deploy the ThreatMetrix Integration Kit files to your PingFederate directory.

Steps

1. Download the ThreatMetrix Integration Kit `.zip` archive from the Ping Identity Integration Directory.
2. Stop PingFederate.
3. If you're upgrading an existing deployment, back up your customizations and delete earlier versions of the integration files:Back up any ThreatMetrix Integration Kit files that you customized in the `<pf_install>/pingfederate/server/default/conf/` directory.Delete the `pf-threatmetrix-adapter-<version>.jar` file from your `<pf_install>/pingfederate/server/default/deploy` directory.
4. Extract the `.zip` archive and merge the contents of the `dist` directory with your `<pf_install>/pingfederate` directory.
5. If there is more than one version of the `pf-authn-api-sdk-<version>.jar` file in your `<pf_install>/pingfederate/server/default/lib` directory, delete all but the latest version of the file.
6. If you backed up any customized files, modify the new files with your customizations.
7. Start PingFederate.
8. If you operate PingFederate in a cluster, repeat steps 2-8 for each engine node.

Configuring ThreatMetrix

To complete the integration, you need your ThreatMetrix organization ID, API key, and optional device profiling domain.

About this task

Copyright © 2025 Ping Identity Corporation

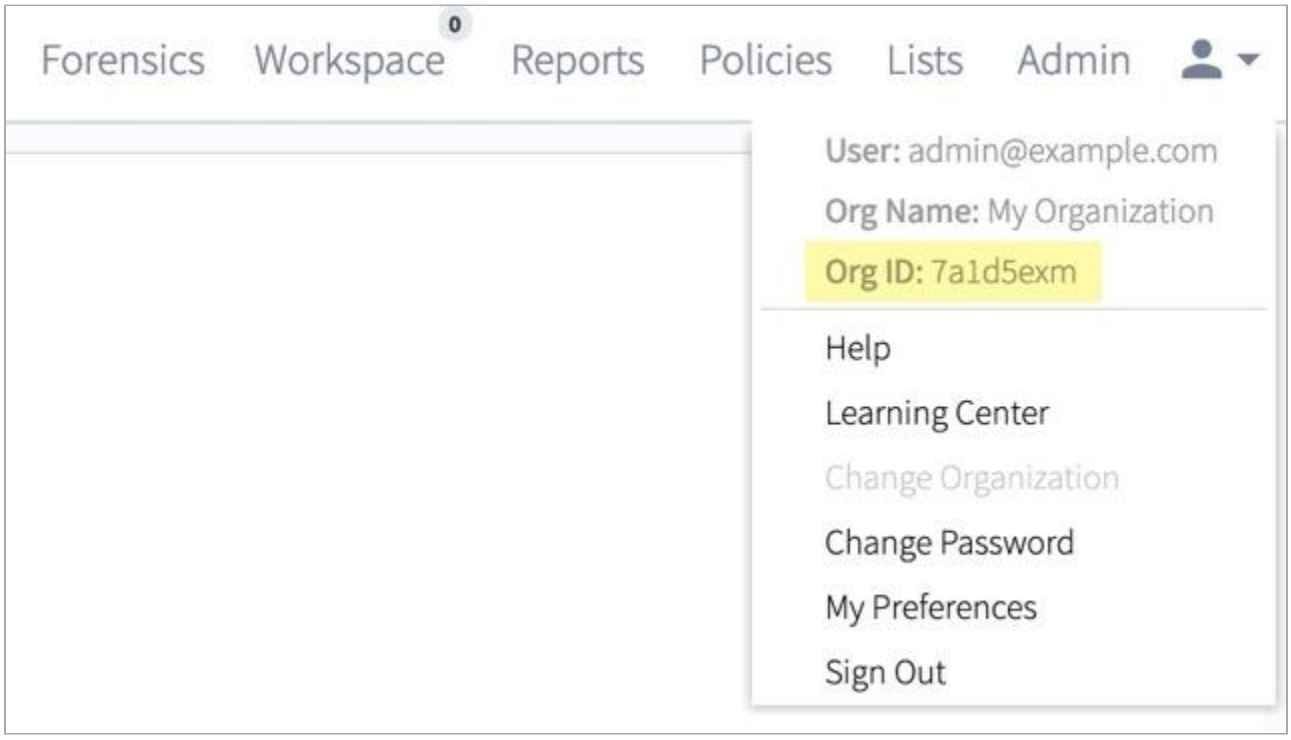
For more information about these values, see the following sections of the ThreatMetrix documentation.

- Org ID
- API Key
- Enhanced Profiling via Hosted SSL

Steps

1. Sign on to the ThreatMetrix Portal as an administrator.
2. Get your organization ID:
    1. In the upper-right corner, click your user avatar.
    2. Note the Org ID.

    You will use this in Integrating device profiling and Configuring an adapter instance.



3. Get your API key:
    1. Go to Admin > API Keys.
    2. If you want a new API key, click Create New API Key. On the Create New API Key modal, click Save.
    3. Note the API Key.

    You will use this in Configuring an adapter instance.
4. Optional: Request a custom device profiling domain to make ThreatMetrix requests appear to come from your domain:
    1. Follow the Enhanced Profiling via Hosted SSL guide in the ThreatMetrix documentation.

Copyright © 2025 Ping Identity Corporation

2. Note your device profiling domain.

You will use this in Integrating device profiling and Configuring an adapter instance.

Integrating device profiling

Depending on your authentication mode and preferences, complete one of the steps below to capture the device profile.

About this task

For description of each method, see Device profiling methods.

Steps

1. Do one of the following:

Method	Steps
Built-in (basic)	No extra steps are needed. Continue to Configuring an adapter instance.
Web app (basic)	Complete the steps in Integrating device profiling - Web app (basic).
Web app (enhanced)	Complete the steps in Integrating device profiling - Web app (enhanced).
Mobile or native app (enhanced)	Complete the steps in Integrating device profiling - Mobile or native app (enhanced).

Integrating device profiling - Web app (basic)

If you have a web application that uses the PingFederate authentication API and cannot accommodate HTTP cookies, modify your web application to run the device profiling script.

About this task

For more information about the authentication API, see PingFederate authentication API in the PingFederate documentation.

There are two device profiling scripts to choose from, including the "SDK" and "Web" scripts. For a description of the differences, see Introduction to Profiling in the ThreatMetrix documentation.

Steps

1. If you want to use the ThreatMetrix SDK script, do the following:
    1. Copy the `tmx_sdk_profiling.js` file from the integration `.zip` archive to a location that your web application page can access.
    2. Add the following code to the sign-on page.

Copyright © 2025 Ping Identity Corporation

```
<script type="text/javascript" src="tmx_sdk_profiling.js"></script>
<script type="text/javascript">pinghelper.run_sid_provided("https://h-api.online-metrix.net",
"<orgId>", "<sessionId>");</script>
```

Substitute your organization ID and adjust the path to the script file.

If you requested a custom device profiling domain in Configuring ThreatMetrix, substitute it here.

For each sign-on event, insert the session ID provided by PingFederate.

2. If you want to use the ThreatMetrix Web script, add the following code to the sign-on page.

```
<script type="text/javascript"
src="https://h-api.online-metrix.net/fp/tags.js?org_id=<orgId>&session_id=<sessionId>"></script>
```

Substitute your organization ID.

If you requested a custom device profiling domain in Configuring ThreatMetrix, substitute it here.

For each sign-on event, insert the session ID from PingFederate.

3. Configure your application to complete the following actions in sequence:

    1. Get the session ID from PingFederate.

    This is an attribute of the `DEVICE_PROFILE_REQUIRED` state.

    1. Run the device profiling script with the session ID.
    2. POST `continueAuthentication` to the authentication API.

4. When you complete the steps in Configuring an adapter instance, set Device Profiling to Create new device profile.

Integrating device profiling - Web app (enhanced)

Instead of using the ThreatMetrix IdP Adapter to collect the device profile, you can capture the device profile using an existing sign-on page. This can reduce perceived wait times for the user.

About this task

You can adapt these instructions to add device profiling to any page, such as the HTML Form Adapter or your external web app. The page must meet the criteria listed in Device profiling methods.

There are two device profiling scripts to choose from, including the "SDK" and "Web" scripts. For a description of the differences, see Introduction to Profiling in the ThreatMetrix documentation.

Steps

1. If you want to use the ThreatMetrix SDK script, do the following:

    1. If you are modifying an external web application, copy the `tmx_sdk_profiling.js` file from the integration `.zip` archive to a location that your page can access.

Copyright © 2025 Ping Identity Corporation

2. Add the following code to the sign-on page:

```
<script type="text/javascript" src="tmx_sdk_profiling.js"></script>
<script type="text/javascript">pinghelper.run("https://h-api.online-metrix.net", "<orgId>");</
script>
```

Substitute your organization ID and adjust the path to the script file.

> **Tip**
>
> If you are modifying a PingFederate template, the script path is `../assets/scripts/tmx_sdk_profiling.js`.

If you requested a custom device profiling domain in Configuring ThreatMetrix, substitute it here.

2. If you want to use the ThreatMetrix Web script, do the following:

    1. If you are modifying an external web application, copy the `tmx_web_profiling.js` file from the integration `.zip` archive to a location that your page can access.

    2. In the `tmx_web_profiling.js` file, substitute your organization ID:

```
var deviceProfilingDomain = "h.online-metrix.net";
var orgId = "orgId";
```

If you requested a custom device profiling domain in Configuring ThreatMetrix, substitute it here.

    1. Add the following to the sign-on page and adjust the path to the script file:

```
<script type="text/javascript" src="tmx_web_profiling.js"></script>
```

> **Tip**
>
> If you are modifying a PingFederate template, `../assets/scripts/tmx_web_profiling.js` is the script path.

3. Optional: In the script file, customize the name prefix for the device profile cookie to suit your environment.

```
var cookieNamePrefix = "tmxSessionID";
```

4. When you complete the steps in Configuring an adapter instance, do the following:

    1. Set Device Profiling to Use existing ThreatMetrix session ID.

    2. Update the Cookie Name Prefix field if you customized it above.

Copyright © 2025 Ping Identity Corporation

Integrating device profiling - Mobile or native app (enhanced)

Instead of using the ThreatMetrix IdP Adapter to collect the device profile, you can capture the device profile using your mobile or native app.

Steps

1. Integrate the ThreatMetrix SDK device profiling functionality into your mobile app.

   ThreatMetrix provides SDKs for Android, iOS, OSX, Windows, and Java. For details, see .threatmetrix.com/kb/index.htm//[Introduction to ThreatMetrix SDK and FAQ] in the ThreatMetrix documentation.

2. Configure your mobile app to communicate with PingFederate via the authentication API.

   To get started, see Authentication API Support.

3. Configure your mobile app to provide the ThreatMetrix session ID when the authentication API reaches the `[.codeph]` DEVICE_PROFILE_SESSION_ID_REQUIRED```` state.

   ```
   submitDeviceProfileSessionId{
       "sessionId":"8de77938-65bf-4578-9d1d-cd14a2f87042"
   }
   ```

4. When you complete the steps in Configuring an adapter instance, set Device Profiling to Use existing ThreatMetrix session ID.

Configuring an adapter instance

Configure the ThreatMetrix IdP Adapter to determine how PingFederate communicates with ThreatMetrix.

Steps

1. In the PingFederate administrative console, create a new IdP adapter instance:
   1. For PingFederate 10.1 or later: go to Authentication > Integration > IdP Adapters. Click Create New Instance.
   2. For PingFederate 10.0 or earlier: go to Identity Provider > Adapters. Click Create New Instance.
2. On the Type tab, set the basic adapter instance attributes:
   1. In the Instance Name field, enter a name for the adapter instance.
   2. In the Instance ID field, enter a unique identifier for the adapter instance.From the Type list, select ThreatMetrix IdP Adapter. Click Next.
3. Optional: On the IdP Adapter tab, in the Additional User Attributes section, configure user attributes to send to ThreatMetrix.

Copyright © 2025 Ping Identity Corporation

> **Note**
>
> For the complete list of user attributes that ThreatMetrix can collect, see Session Query API in the ThreatMetrix documentation.

1. Click Add a new row to 'Additional User Attributes'.
2. In the Incoming Attribute Name field, enter the name of an attribute from any authentication source that appears earlier in your PingFederate authentication policy than the ThreatMetrix IdP Adapter.
3. From the ThreatMetrix Attribute list, select the ThreatMetrix attribute that you want to populate.
4. In the Action column, click Update.
5. To add more attributes, repeat steps a-d.

4. Optional: On the IdP Adapter tab, in the ThreatMetrix API Response Mappings section, map attributes and sign-on event data from the ThreatMetrix response to the attribute contract.

   These attributes become available in your PingFederate authentication policy.

   1. Click Add a new row to 'ThreatMetrix API Response Mappings'.
   2. In the Local Attribute field, enter a name of your choosing for an attribute.
   3. In the ThreatMetrix API Attribute Mapping field, enter the JSON Pointer syntax for the value of the matching ThreatMetrix attributes. Alternately, leave the field blank to include the entire response as the value.
   4. In the Action column, click Update.
   5. To add more attributes, repeat steps a-d.

5. On the IdP Adapter tab, configure the adapter instance by referring to ThreatMetrix IdP Adapter settings reference. Click Next.
6. On the Extended Contract tab, add any attributes that you included in the ThreatMetrix API Response Mappings section of the IdP Adapter tab. Click Next.
7. On the Adapter Attributes tab, set pseudonym and masking options as shown in Set pseudonym and masking options in the PingFederate documentation. Click Next.
8. On the Adapter Contract Mapping tab, configure the contract fulfillment details for the adapter as shown in Define the IdP adapter contract in the PingFederate documentation. Click Next.
9. On the Summary tab, check and save your configuration:
   1. For PingFederate 10.1 or later: click Save.
   2. For PingFederate 10.0 or earlier: click Done. On the Manage IdP Adapter Instances tab, click Save.

ThreatMetrix IdP Adapter settings reference

The following are setting descriptions for the ThreatMetrix IdP Adapter.

Copyright © 2025 Ping Identity Corporation

Standard fields

Field	Description
Org ID	The org ID that you noted in Configuring ThreatMetrix. This field is blank by default.
API Key	The API key that you noted in Configuring ThreatMetrix. This field is blank by default.
Policy Name	The name of the policy to use when requesting a review status. The default value is `default`.
ThreatMetrix Base URL	The ThreatMetrix API URL. If ThreatMetrix changes this URL, enter the new URL. The default value is `https://h-api.online-metrix.net`.
Device Profiling	Determines whether the adapter creates a new ThreatMetrix session ID, or receives one from another source. Your selection depends on which of the Device profiling methods you set up. ***Create new device profile*** Select this if you used one of the "basic" device profiling methods. The ThreatMetrix IdP Adapter creates a new ThreatMetrix session ID. In authentication API mode, the adapter provides a session ID to your web app. Otherwise, the adapter shows the built-in device profiling page that runs the device profiling script. ***Use existing ThreatMetrix session ID*** Select this if you used one of the "enhanced" device profiling methods. In authentication API mode, the adapter looks for a session ID provided in response to the `SESSION_ID_REQUIRED`. Otherwise, the adapter looks for a session ID provided in an HTTP cookie. The default value is Create new device profile.

Copyright © 2025 Ping Identity Corporation

Field	Description
Device Profiling Script Source	Determines the script used to create the device profile. The ThreatMetrix SDK script runs locally, and the ThreatMetrix Web script fetches the latest device profiling script from ThreatMetrix each time. For details, see Introduction to Profiling in the ThreatMetrix documentation. Applies only with the "Built-in (basic)" device profiling method. The default value is ThreatMetrix Web.

Advanced fields

Field	Description
Device Profiling Domain	The domain used for device profiling. Applies when Device Profiling is set to Create new device profile. If you requested a custom device profiling domain in Configuring ThreatMetrix, enter it here. The default value is `h.online-metrix.net`.
Device Profiling Timeout	The amount of time in milliseconds that PingFederate waits for the device profiling script to collect device details. Applies only when Device Profiling is set to Create new device profile. The minimum value is `3000`. The default value is `5000`.
Cookie Name	The name of the cookie that contains the device profile. Applies only when Device Profiling is set to Use existing ThreatMetrix session ID. If you customized the name for the cookie in the optional Integrating device profiling - Web app (enhanced) steps, enter the same name in this field. The default value is `tmxSessionID`.
Service Type	Determines the attributes and sign-on event data that ThreatMetrix provides in the response. For details, see the `service_type` parameter in .threatmetrix.com/kb/index.htm//[Session Query Parameters] in the ThreatMetrix documentation. The default value is `session-policy`.

Copyright © 2025 Ping Identity Corporation

Field	Description
Failure Mode	When ThreatMetrix is unavailable or an error occurs, this setting determines the default review status. To allow users to continue to sign on by satisfying stricter authentication requirements, select "review". **Note** Setting this field to **"pass"** is not recommended outside a test environment.
Unknown Session Mode	When ThreatMetrix returns an unknown session, this setting determines the review status used. **Note** Setting this field to **"pass"** is not recommended outside a test environment.
Session Query API Endpoint	The ThreatMetrix Session Query API endpoint. If ThreatMetrix changes this endpoint, enter the new endpoint. The default value is `/api/session-query`.
Update API Endpoint	The ThreatMetrix Update API endpoint. If ThreatMetrix changes this endpoint, enter the new endpoint. The default value is `/api/update`.
Update API Enabled	After a user with a "review" status moves through the PingFederate authentication policy, the adapter informs ThreatMetrix whether authentication ultimate succeeded. This helps improve future risk assessments. If your authentication policy doesn't require users with a "review" status to pass any other authentication challenges, clear this check box to skip the update step. This check box is selected by default.
API Request Timeout	The amount of time in milliseconds that PingFederate allows when establishing a connection with ThreatMetrix or waiting for a response to a request. A value of 0 disables the timeout. The default value is `2000`.
Proxy Settings	Defines proxy settings for outbound HTTP requests. The default value is System Defaults.
Custom Proxy Host	The proxy server host name to use when Proxy Settings is set to Custom. This field is blank by default.

Copyright © 2025 Ping Identity Corporation

Field	Description
Custom Proxy Port	The proxy server port to use when Proxy Settings is set to Custom. This field is blank by default.

Adding review statuses to your authentication policy

By modifying your PingFederate authentication policy to include the review status from ThreatMetrix, you can dynamically change authentication requirements based on security risk level.

About this task

These steps are designed to help you add to an existing authentication policy. For general information about configuring authentication policies, see PingFederate authentication API in the PingFederate documentation.

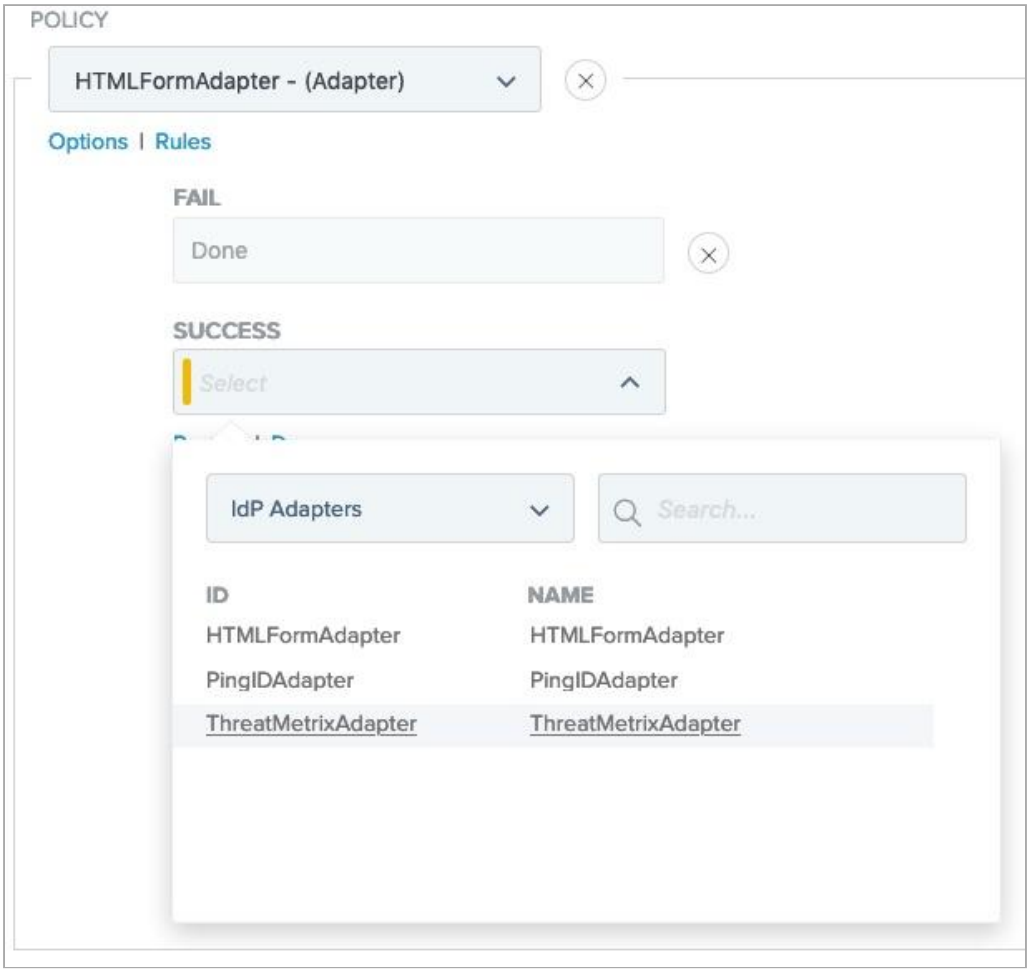
> **Note**
>
> ThreatMetrix automatically tunes its rules and policies based on user behavior. To accommodate an initial training period, we recommend that you allow all transactions to succeed for a period of time, regardless of review status.

Steps

1. In the PingFederate administrative console, go to the Policies tab.
    - For PingFederate 10.1 or later: go to Authentication > Policies > Policies.
    - For PingFederate 10.0 or earlier: go to Identity Provider > Authentication Policies > Policies.
2. Select the IdP Authentication Policies check box.
3. Open an existing authentication policy, or click Add Policy.

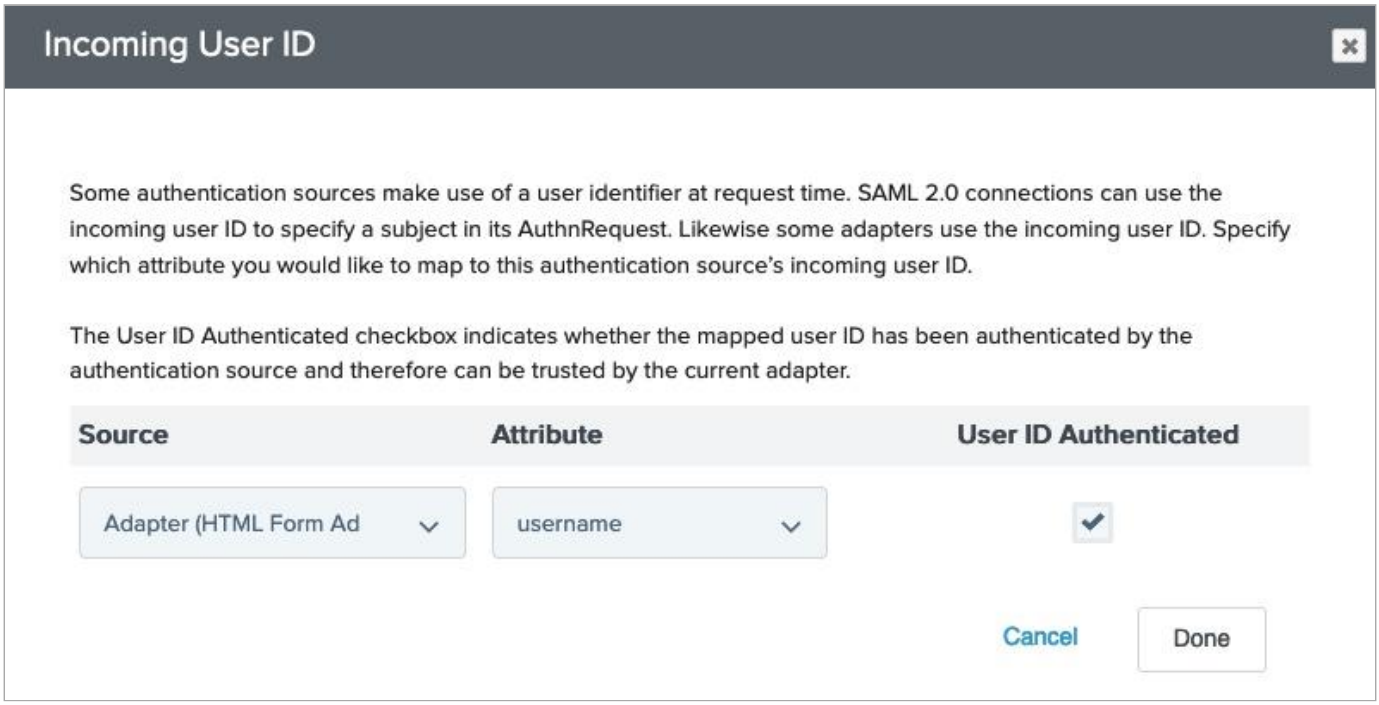
   For help, see Defining authentication policies in the PingFederate documentation.
4. In the Policy area, select your ThreatMetrix IdP Adapter instance.

Copyright © 2025 Ping Identity Corporation



5. Map the user ID into the ThreatMetrix IdP Adapter instance:

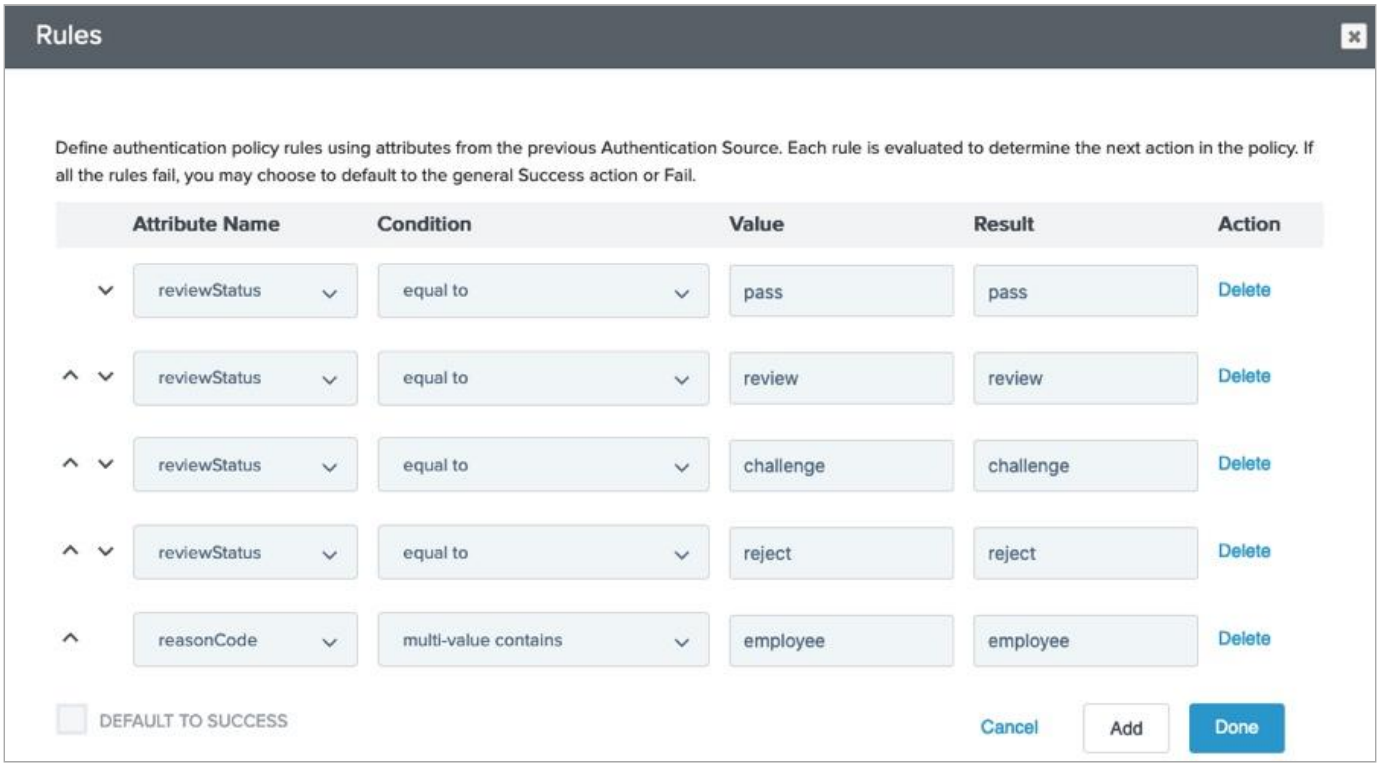
Copyright © 2025 Ping Identity Corporation



1. Under the ThreatMetrix IdP Adapter instance, click Options.
2. On the Options dialog, from the Source list, select a previous authentication source that collects the user ID.
3. From the Attribute list, select the user ID.
4. Click Done.For PingFederate 10.2 and later, select the User ID Authenticated check box.

6. Define policy paths based on the information provided by ThreatMetrix:

Copyright © 2025 Ping Identity Corporation



1. Under the ThreatMetrix IdP Adapter instance, click Rules.
2. On the Rules dialog, from the Attribute Name list, select reviewStatus.
3. From the Condition list, do one of the following:
   - For reviewStatus, select equal to.
   - For reasonCode, select multi-value contains.
4. In the Value field, do one of the following:
   - For reviewStatus, enter "pass", "review", "challenge", or "reject".
   - For reasonCode, enter a ThreatMetrix reason code.

   > **Tip**
   >
   > Reason codes reflect the policy or policies that ThreatMetrix used to determine the review status. In the ThreatMetrix admin console, set meaningful names for your policies and enter the names here.

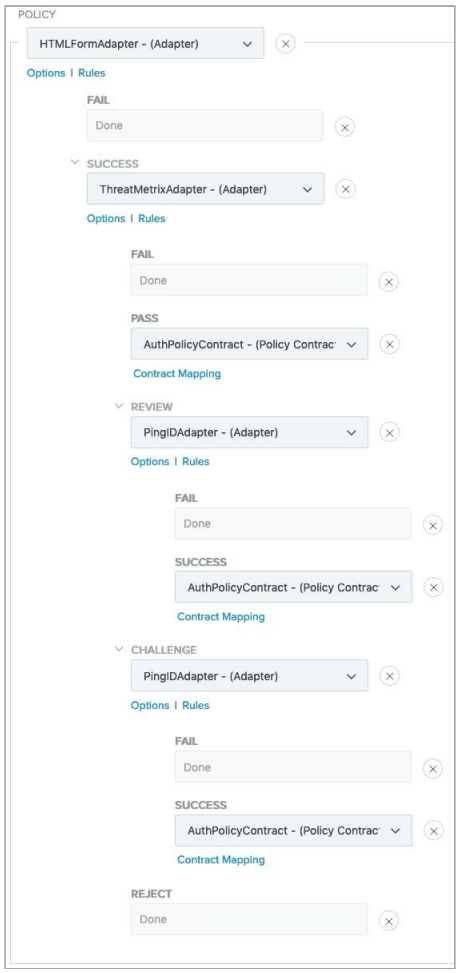
5. In the Result field, enter a name.

   This appears as a new policy path that branches from the ThreatMetrix IdP Adapter.
6. If you want to add more authentication paths, click Add and repeat steps b-e.
7. Optional: Clear the Default to success check box.
8. Click Done.

Copyright © 2025 Ping Identity Corporation

7. Configure each of the authentication paths.

> **Note**
>
> In case ThreatMetrix is unreachable or returns an error, the **Failure mode** setting in Configuring an adapter instance will assign a default review status. As a result, the **Fail** outcome of the ThreatMetrix IdP Adapter instance is not used.



8. Click Done. In the Policies window, click Save.

Authentication API Support

You can use the PingFederate authentication API to integrate the ThreatMetrix IdP Adapter into your application.

The PingFederate Authentication API provides access to the current state of the authentication flow as a user steps through the PingFederate authentication policy. For more information, see PingFederate authentication API in the PingFederate documentation. To integrate the ThreatMetrix IdP Adapter into your authentication flow, configure your application based on the states, actions, and models available in the PingFederate Authentication API Explorer. For help, see Exploring the Authentication API in the PingFederate documentation.

Copyright © 2025 Ping Identity Corporation

Troubleshooting

Enabling debug logging

To help with troubleshooting or monitoring, you can turn on activity logging for PingFederate, the ThreatMetrix IdP Adapter, or both.

About this task

This task is optional. You can use logging for troubleshooting or analytics.

For general information about logging, see Enabling debug messages and console logging in the PingFederate documentation.

Steps

1. Open the `<pf_install>/pingfederate/server/default/conf/log4j2.xml` file for editing.
2. To log activity for PingFederate and all adapters:
    1. Find the following section in the file.

    ```
    <AsyncRoot level="INFO" includeLocation="false">
            <!-- <AppenderRef ref="CONSOLE" /> -->
            <AppenderRef ref="FILE" />
    </AsyncRoot>
    ```

    2. Change `INFO` to `DEBUG`.

    The following code snippet shows `DEBUG` in bold for visibility.

    ```
    <AsyncRoot level="DEBUG" includeLocation="false">
            <!-- <AppenderRef ref="CONSOLE" /> -->
            <AppenderRef ref="FILE" />
    </AsyncRoot>
    ```

    3. Optional: To see the adapter activity in the console as well as the log file, remove the comment tags (`<!--` and `-->`) that surround the `CONSOLE` line.

    ```
    <AsyncRoot level="INFO" includeLocation="false">
            <AppenderRef ref="CONSOLE" />
            <AppenderRef ref="FILE" />
    </AsyncRoot>
    ```

3. If you want to log activity relating to the ThreatMetrix IdP Adapter, do one of the following.

Copyright © 2025 Ping Identity Corporation

> **Tip**
>
> You can use this information with a third-party log analysis tool to monitor for important events, such as when a sign-on event has a high-risk review status.

Choose from:

- To log activity for the ThreatMetrix IdP Adapter as well as its HTTPS and component activity, add the following line.

```
<Logger name="com.pingidentity.adapters.threatmetrix" level="DEBUG"/>
```

- To log activity for the adapter's HTTPS activity and other components but not the adapter itself, add the following line.

```
<Logger name="{logging-class}.shade" level="DEBUG"/>
```

- To log activity for the ThreatMetrix IdP Adapter but not its HTTPS or component activity, add the following lines.

```
<Logger name="com.pingidentity.adapters.threatmetrix" level="DEBUG"/>
<Logger name="com.pingidentity.adapters.threatmetrix.shade" level="INFO"/>
```

4. Save the file.

Release notes

Changelog

- The following is the change history for the ThreatMetrix Integration Kit.

ThreatMetrix Integration Kit 1.4 - September 2021

- Add the ability to configure how unknown sessions are handled.
- Added the ability to disable update API calls to ThreatMetrix.

ThreatMetrix Integration Kit 1.3 - June 2021

- Updated the `tmx_sdk_profiling.js` script to use the latest code from ThreatMetrix.
- Fixed an issue that, after upgrading the adapter, caused an error when using the administrative API to bulk import an earlier version of the adapter.

Copyright © 2025 Ping Identity Corporation

ThreatMetrix Integration Kit 1.2 - May 2021

- Added support for the transaction failure callback added in PingFederate 10.2. PingFederate can now signal ThreatMetrix when a transaction fails, which can affect future risk evaluations. Requires PingFederate 10.2 or later.
- Added support for ThreatMetrix reason codes. After a transaction, the reason codes provided by ThreatMetrix are now available in the PingFederate authentication policy in the `reasonCode` attribute.
- Added support for authentication via mobile and native apps. See Device profiling methods.
- Updated the Device Profiling setting labels for clarity and to accommodate mobile and native apps.

1.1 Device Profiling Settings	1.2 Device Profiling Settings
Captured by this adapter	Create new device profile
Captured by a previous adapter	Use existing ThreatMetrix session ID

- Fixed an issue that caused an error when an attribute in the Additional User Attributes table had no value. Attributes with empty values are now ignored.

ThreatMetrix Integration Kit 1.1 - October 2020

- Changed the adapter to retrieve a JSON-encoded (instead of URL-encoded) response from ThreatMetrix. The adapter can now make the entire response available in a single attribute in the PingFederate authentication policy.
- Combined the API connection and request timeout settings.
- Added support for the JavaScript Widget for the PingFederate Authentication API.

ThreatMetrix Integration Kit 1.0 - July 2020

- Initial release.
- Added support for getting risk assessments and additional data from ThreatMetrix.
- Added support for the ThreatMetrix "Web" and "SDK" device profiling scripts.
- Added the ability to add device profiling into any browser-based authentication source or web application.
- Added the ability to send user attributes to ThreatMetrix.
- Added the ability to map any ThreatMetrix API response attribute to an attribute in the PingFederate authentication policy.
- Added support for the PingFederate authentication API.
- Added a setting to handle failed risk assessment requests.
- Added settings for API connection and request timeouts.
- Added settings to override the PingFederate system-default proxy settings.

Copyright © 2025 Ping Identity Corporation

Known issues and limitations

- The following are known issues or limitations for the ThreatMetrix Integration Kit.

Known issues

There are no known issues.

Known limitations

- When a user completes authentication after receiving a review status of ""review"", the ThreatMetrix IdP Adapter reports a success to the ThreatMetrix Update API. However, when the user fails to authenticate, the ThreatMetrix IdP Adapter is not able to send a failure report.

Download manifest

- The following files are included in the ThreatMetrix Integration Kit `.zip` archive.
- `Legal.pdf` – copyright and license information
- `dist/pingfederate/server/default` – contains the integration files
    - `deploy` – contains the Java libraries
        - `pf-threatmetrix-adapter-<version>.jar` – JAR file that contains the ThreatMetrix IdP Adapter.
- `conf` – contains the HTML template that presents the ThreatMetrix sign-on form
    - `language-packs/threatmetrix-messages.properties` – a variable file that customizes the messages that appear on the template file.
        - `template` – contains user-facing HTML template files
            - `threatmetrix.adapter.template.html` – a sign-on redirect page used with the "captured by this adapter" device profiling method. Runs scripts to show a spinner animation and create the device profile.
    - `assets` – contains functional scripts and files used by the template
        - `css` – contains CSS files for the templates.
            - `threatmetrix.css` – a CSS file that customizes the appearance of the template files.
            - `end-user/<version>/end-user.css` – a CSS file that customizes the appearance of the template files.
                - `fonts/end-user/icons` – contains template icons
                - `images` – contains template image files
        - `ping-logo.svg` – an image file with company branding

Copyright © 2025 Ping Identity Corporation

        - `spinner.svg` – an image file used in a spinner animation
        - `scripts` – contains script files used to collect and send information
            - `tmx_sdk_profiling.js` – a JavaScript script that can be embedded in any authentication page. Initiates the ThreatMetrix SDK device profiling script and stores the device profile in a cookie. Used with the "captured by a previous adapter" device profiling method.
            - `tmx_web_profiling.js` – a JavaScript script that can be embedded in any authentication page. Initiates the ThreatMetrix Web device profiling script and stores the device profile in a cookie. Used with the "captured by a previous adapter" device profiling method.
- `lib/pf-authn-api-sdk-<version>.jar` – a JAR file that contains the PingFederate Authentication API SDK

Copyright © 2025 Ping Identity Corporation

Twitter Login Integration Kit

The Twitter Login Integration Kit allows PingFederate to use Twitter as an identity provider (IdP). This allows users to access service provider (SP) applications by signing into Twitter.

Features

- Supports the PingFederate authentication API.
- Supports the JavaScript Widget for the PingFederate Authentication API.

Components

Twitter IdP Adapter

- Allows PingFederate to communicate with the Twitter API to process sign-on requests and get user information.

Templates

- Allows the adapter to prompt the user to sign on. The template can be presented with a browser redirect or as a pop-up window.
- Allows you to modify the appearance of the sign-on prompt.

Language packs

- Allows you to customize or localize the messages returned by the PingFederate authentication API and shown on the templates during authentication. You can find help in Localizing messages for end users in the PingFederate documentation.

Intended audience

This document is intended for PingFederate administrators.

If you need help during the setup process, see the following resources:

- The following sections of the PingFederate documentation:
  - Identity provider SSO configuration
  - Managing IdP adapters
  - Authentication policies
  - SP connection management
- The following sections of the Twitter documentation:
  - Developer portal
  - Developer Apps

Copyright © 2025 Ping Identity Corporation

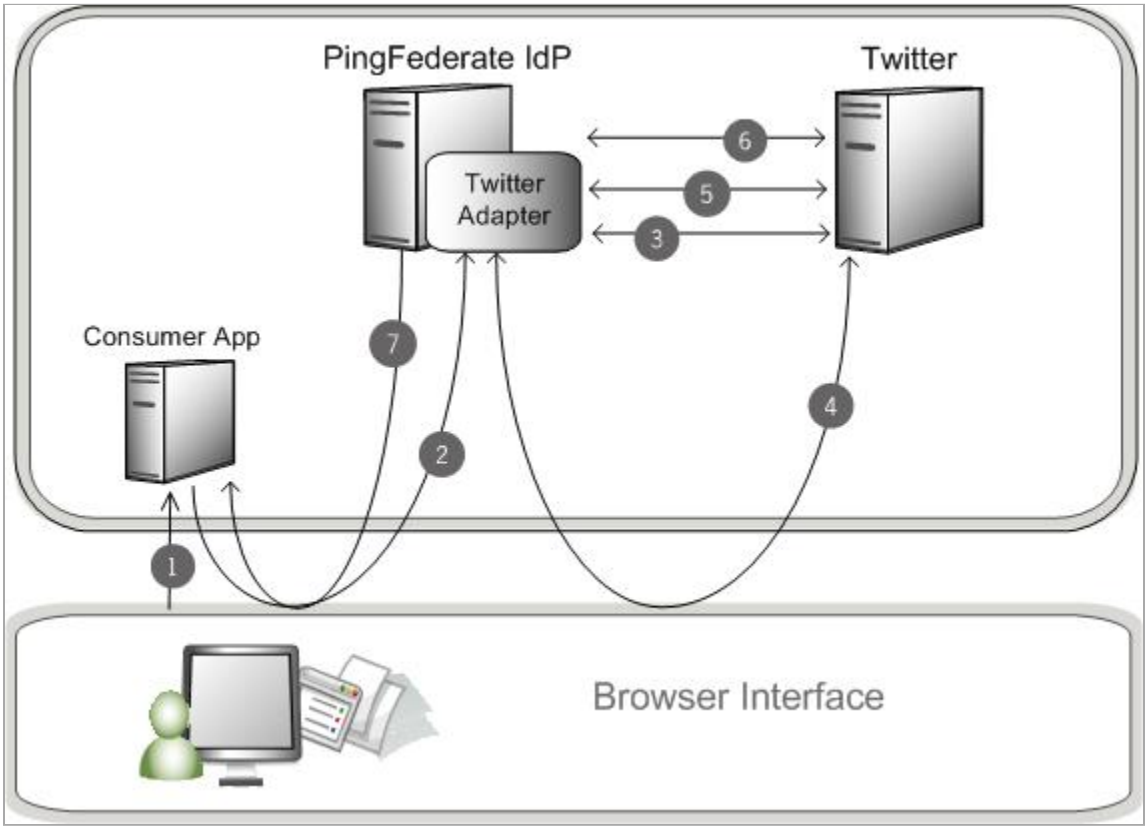
System requirements

- PingFederate 9.0 or later, or 9.3 to use the authentication API
- A Twitter developer account

Overview of the SSO flow

With the Twitter Login Integration Kit, PingFederate includes the Twitter authentication API in the sign-on flow.

The following figure illustrates a service provider (SP)-initiated single sign-on (SSO) scenario in which PingFederate authenticates users to an SP application using the Twitter IdP Adapter.



Description

1. The user opens a web application and chooses the Twitter sign-on option.
2. The sign-on link points to the Twitter IdP Adapter.
3. The Twitter IdP Adapter requests a request token from Twitter and provides the callback URL. Twitter returns the request token.
4. The PingFederate server redirects the user to Twitter with the request token and alist of requested permissions. On Twitter, the user authenticates their identity and then authorizes the requested permissions.

Copyright © 2025 Ping Identity Corporation

Twitter redirects the browser to the Twitter IdP Adapter callback URL with a verification code.

If the user fails to authenticate or does not authorize the request, the response includes an error code instead.

5. PingFederate sends Twitter the request token and verification code. Twitter validates these components and returns an access token to the PingFederate callback URL.
6. PingFederate sends Twitter a request for user attributes and presents the access token.
7. PingFederate redirects the user to the web application with the user attributes.

Authentication API Support

You can use the PingFederate authentication API to integrate the Twitter IdP Adapter into your application.

The PingFederate Authentication API provides access to the current state of the authentication flow as a user steps through the PingFederate authentication policy. For more information, see PingFederate authentication API in the PingFederate documentation. To integrate the Twitter IdP Adapter into your authentication flow, configure your application based on the states, actions, and models available in the PingFederate Authentication API Explorer. For help, see Exploring the Authentication API in the PingFederate documentation.

Setup

Deploying the integration files

To get started with the integration, deploy the Twitter Login Integration Kit files to your PingFederate directory.

About this task

> **Note**
>
> If you operate PingFederate in a cluster, the following steps refer to the console node.

Steps

1. Download the Twitter Login Integration Kit `.zip` archive from the Ping Identity Integration Directory.
2. Stop PingFederate.
3. If you're upgrading an existing deployment, back up your customizations and delete earlier versions of the integration files:Back up any Twitter Login Integration Kit files that you customized in the `<pf_install>/pingfederate/server/default/conf/` directory.Delete the `pf-twitter-adapter-<version>.jar` file from your `<pf_install>/pingfederate/server/default/deploy` directory.
4. Extract the `.zip` archive and merge the contents of the `dist` directory with your `<pf_install>/pingfederate` directory.
5. If there is more than one version of the `pf-authn-api-sdk-<version>.jar` file in your `<pf_install>/pingfederate/server/default/lib` directory, delete all but the latest version of the file.

Copyright © 2025 Ping Identity Corporation

6. If you backed up any customized files, modify the new files with your customizations.
7. Start PingFederate.
8. If you operate PingFederate in a cluster, repeat steps 2-8 for each engine node.

Registering PingFederate in Twitter

To allow PingFederate to process social sign-on requests with Twitter, add PingFederate as an application in the Twitter administrative console.

Steps

1. Sign on to the Twitter developer portal Apps page using a Twitter developer account.
2. On the My Applications Dashboard, create a new Twitter application.
3. In the Name field, enter the name of the application, such as `MyPingFederate`.
4. In the Website field, enter the domain for your web application, such as `http://www.example.com`.
5. In the Callback URL field, enter the hostname (or virtual hostname) and port of your PingFederate server followed by the Twitter IdP Adapter instance endpoint, such as `https://pf.example.com:9031/ext/twitter-authn`.

   > **Note**
   >
   > The default endpoint is `/twitter-authn`. If you set a custom endpoint here, enter the matching value in the **Authorization Callback Endpoint** field of your adapter instance configuration in Configuring an adapter instance.

6. If your web application will use the access token for subsequent Twitter API calls that require read and write access, set the Default Access Type to Read & Write.
7. Complete the configuration based on the needs of your specific environment. You can find help in Developer Apps in the Twitter documentation.
8. Click Create your Twitter application.
9. On the application details page, note the Consumer Key and Consumer Secret. You'll use these in Configuring an adapter instance.

Configuring an adapter instance

Configure the Twitter IdP Adapter to determine how PingFederate communicates with Twitter.

Steps

1. In the PingFederate administrative console, create a new IdP adapter instance:
   1. For PingFederate 10.1 or later: go to Authentication > Integration > IdP Adapters. Click Create New Instance.
   2. For PingFederate 10.0 or earlier: go to Identity Provider > Adapters. Click Create New Instance.

Copyright © 2025 Ping Identity Corporation

2. On the Type tab, set the basic adapter instance attributes.
    1. In the Instance Name field, enter a name for the adapter instance.
    2. In the Instance ID field, enter a unique identifier for the adapter instance.
    3. From the Type list, select Twitter IdP Adapter. Click Next.
3. Optional: On the IdP Adapter screen, in the Attribute Selector section, select or define attributes to request from Twitter.

    For a list of available attributes, see User Object in the Twitter documentation.

    1. Click Add a new row to 'Attribute Selector'.
    2. Select an attribute from the Twitter Attribute list or enter an attribute in the Additional Twitter Attribute field.

    You must use the correct syntax for manual entries. For an example of the data structure, see GET account/verify_credentials in the Twitter documentation.

    1. In the Action column, click Update.
    2. To add more attributes, repeat steps a-c.
4. On the IdP Adapter screen, configure the adapter instance by referring to Twitter IdP Adapter settings reference. Click Next.
5. On the Extended Contract tab, add any attributes that you want to include in the contract. Click Next.
6. On the Adapter Attributes tab, set pseudonym and masking options as shown in Set pseudonym and masking options in the PingFederate documentation. Click Next.
7. On the Adapter Contract Mapping tab, configure the contract fulfillment details for the adapter as shown in Define the IdP adapter contract in the PingFederate documentation. Click Next.
8. On the Summary tab, check and save your configuration:
    1. For PingFederate 10.1 or later: click Save.
    2. For PingFederate 10.0 or earlier: click Done. On the Manage IdP Adapter Instances tab, click Save.

Twitter IdP Adapter settings reference

Field descriptions for the Twitter IdP Adapter configuration screen.

Standard fields

Field	Description
Consumer Key	The consumer key that you noted in Registering PingFederate in Twitter.
Consumer Secret	The consumer secret that you noted in Registering PingFederate in Twitter.

Copyright © 2025 Ping Identity Corporation

Field	Description
Callback Endpoint	The PingFederate endpoint that Twitter uses to respond to authorization requests. If you set a custom endpoint in the Authorization callback URL field in Registering PingFederate in Twitter, change this field to match. This default value is `/twitter-authn`.
Error Redirect URL	When an error occurs in the adapter, PingFederate redirects the browser to this URL instead of the default error page. This field is blank by default.
Unauthorized Redirect URL	When a user denies the requested Twitter permissions, PingFederate redirects the browser to this URL instead of the default error page. This field is blank by default.

Advanced fields

Field	Description
PingFederate Base URL	The fully-qualified host name, port, and path of the PingFederate server. This URL is used to construct the callback URL sent to Twitter. This field is blank by default.
Twitter Authentication URL	The URL that PingFederate uses to send authentication requests to Twitter. If Twitter changes this endpoint, enter the new URL. The default value is `https://api.twitter.com/oauth/authenticate?oauth_token=`.
Twitter User Data URL	The URL that PingFederate uses to retrieve user data from Twitter. The default value is `https://api.twitter.com/1.1/account/verify_credentials.json`.
Twitter Access Token Invalidation URL	The URL that PingFederate uses to signal Twitter to invalidate an access token. The default value is `https://api.twitter.com/1.1/oauth/invalidate_token`.

Copyright © 2025 Ping Identity Corporation

Field	Description
Twitter Login Presentation	Determines how the adapter presents the Twitter sign-on form. *Redirect (default)* The adapter redirects the browser to the Twitter sign-on form. *Pop-up window* The adapter opens a new window with the Twitter sign-on form on a PingFederate template. Use this option if automatic redirects are blocked by your users' browsers. This setting has no effect when using the adapter through the PingFederate authentication API.
Twitter Pop-Up Template	The template file that presents the Twitter sign-on form. Applies only when Twitter Login Presentation is set to Pop-up window. The default value is `twitter-pop-up-template.html`.
Twitter Post-Auth Template	The template file that the adapter presents after the user signs on. Applies only when Twitter Login Presentation is set to Pop-up window. The default value is `twitter-post-auth-template.html`.
Twitter Messages File	The language-pack file associated with the Twitter pop-up template. The default value is `pingfederate-twitter-adapter-messages`.
Retry Request	Determines whether PingFederate will retry requests after it receives a response with a failure code. This check box is selected by default.
Maximum Retries Limit	Determines how many times PingFederate retries a request. The default value is `5`.
Retry Error Codes	A list of response codes that you want to trigger a retry. Separate response codes with a comma. The default value is `429`.
API Request Timeout	The amount of time in milliseconds that PingFederate allows when establishing a connection with Twitter or waiting for a response to a request. A value of 0 disables the timeout. The default value is `2000`.

Copyright © 2025 Ping Identity Corporation

Field	Description
Proxy Settings	Defines proxy settings for outbound HTTP requests. The default value is System Defaults.
Custom Proxy Host	The proxy server host name to use when Proxy Settings is set to Custom. This field is blank by default.
Custom Proxy Port	The proxy server port to use when Proxy Settings is set to Custom. This field is blank by default.

Integrating social sign-on into your application

To complete your Twitter sign-on integration, add a sign-on hyperlink to your application.

Steps

1. Create an SP connection to your application as shown in SP connection management in the PingFederate documentation. Use your IdP adapter instance and Mapping an adapter instance in the PingFederate documentation.
    - Use the Twitter IdP Adapter instance as an authentication source. You can find help in Mapping an adapter instance in the PingFederate documentation.
2. If your application is outside the PingFederate domain, configure a service provider (SP) connection.
    1. Create an SP connection that uses your IdP adapter instance as shown in SP connection management and Mapping an adapter instance in the PingFederate documentation.
    2. In your web application, create a hyperlink to allow users to sign on to the SP application. Use the following URL and replace the variables based on the descriptions in the following table.

        ```
        https://<pf_host>:<pf_port>/idp/startSSO.ping?PartnerSpId=<ConnectionId>
        ```

Variable	Description
<pf_host>	The host name or IP address of the PingFederate server.
<pf_port>	The port number for PingFederate.
<ConnectionId>	The federation identifier of the SP for the connection that uses the GitHub IdP Adapter instance.

Copyright © 2025 Ping Identity Corporation

3. If your application is inside the PingFederate domain, configure an adapter-to-adapter mapping.
    1. On the System > Protocol Settings > Roles & Protocols page, select Enable Identity Provider (IdP) role and support for the following and Enable Service Provider (SP) role and support for the following.
    2. In both the Enable Identity Provider and Enable Service Provider sections, select any protocol, such as SAML 2.0. Click Save.

        > **Note**
        >
        > PingFederate requires a protocol selection to activate the roles. The protocol that you select is not used for this integration.

    3. On the Service Provider > Adapters page, create or select an adapter instance integrated with the application as shown in SP application integration settings in the PingFederate documentation.
    4. On the Identity Provider > Adapter-to-Adapter Mappings page, configure the IdP-to-SP adapter mapping as shown in Adapter-to-adapter mappings in the PingFederate documentation.
    5. In your web application, create a hyperlink to allow users to sign on to the SP application. Use the following URL and replace the variables based on the descriptions in the table below:

        ```
        https://<pf_host>:<pf_port>/pf/adapter2adapter.ping?IdpAdapterId=<IdpAdapterId>&SpSessionAuthnAdapterId=<SpAdapterId>
        ```

Variable	Description
<pf_host>	The host name or IP address of the PingFederate server.
<pf_port>	The port number for PingFederate.
<IdpAdapterId>	The instance ID of the GitHub IdP Adapter instance.
<SpAdapterId>	The instance ID of the SP adapter instance that has been integrated with the application.

Troubleshooting

Enabling debug logging

To help with troubleshooting or monitoring, you can turn on activity logging for the adapter.

About this task

These steps are optional. For general information about logging, see Enabling debug messages and console logging in the PingFederate documentation.

Copyright © 2025 Ping Identity Corporation

Steps

1. Open the `<pf_install>/pingfederate/server/default/conf/log4j2.xml` file for editing.
2. If you want to log activity for PingFederate and all adapters, do the following.
    1. Find the following section.

```
<AsyncRoot level="INFO" includeLocation="false">
        <!-- <AppenderRef ref="CONSOLE" /> -->
        <AppenderRef ref="FILE" />
</AsyncRoot>
```

    2. Change `INFO` to `DEBUG`.

```
<AsyncRoot level="DEBUG" includeLocation="false">
        <!-- <AppenderRef ref="CONSOLE" /> -->
        <AppenderRef ref="FILE" />
</AsyncRoot>
```

    3. If you want to see the adapter activity in the console, remove the comment tags.

```
<AsyncRoot level="INFO" includeLocation="false">
        <AppenderRef ref="CONSOLE" />
        <AppenderRef ref="FILE" />
</AsyncRoot>
```

3. If you want to log activity just for the Twitter IdP Adapter, add the following line.

```
<Logger name="com.pingidentity.adapter.idp.twitter" level="DEBUG"/>
```

4. Save the file.

Release notes

Changelog

Cloud Identity Connector for Twitter 1.4 – November 2020

- Added support for single logout (SLO).
- Added a setting to use browser redirect or pop-up window for the sign-on presentation.
- Added customizable sign-on templates for the pop-up window presentation.

Copyright © 2025 Ping Identity Corporation

- Added customizable user-facing language-pack messages.
- Added support for the PingFederate authentication API.
- Added support for the JavaScript Widget for the PingFederate Authentication API.
- Improved security by updating bundled components.
- Improved configuration field validation.

Cloud Identity Connector for Twitter 1.3 – June 2018

- Added support to configure callback URL to resolve Twitter enforcement of whitelisting callback URLs.

Cloud Identity Connector for Twitter 1.2 – January 2014

- Improved support for Twitter REST API version 1.1.
- Added support for Twitter API restrictions that allow only TLS/SSL connections.

Cloud Identity Connector for Twitter 1.1.1 – July 2013

- Added Masking to Consumer Id and Consumer Secret fields in the adapter GUI.
- Added support for Scribe 1.3.3.

Cloud Identity Connector for Twitter 1.1 – April 2013

- Added support for the Twitter REST API version 1.1.

Cloud Identity Connector for Twitter 1.0 – February 2011

- Initial Release.

Known issues and limitations

- Twitter does not support single logout.
- Twitter has rate limiting for OAuth calls. See Rate limits in the Twitter documentation.

Download manifest

The following files are included in the Twitter Login Integration Kit `.zip` archive:

- `ReadMeFirst.pdf` – contains links to this online documentation
- `legal/Legal.pdf` – copyright and license information.
- `dist` – contains the integration files
  - `deploy` – contains the Java libraries
    - `pf-twitter-adapter-<version>.jar` – JAR file that contains the Twitter IdP Adapter.

Copyright © 2025 Ping Identity Corporation

- `conf` – contains the HTML template that presents the Twitter sign-on form.
- `language-packs` – contains files with customizable user-facing messages
  - `authn-api-messages.properties` – a version of the common PingFederate authentication API variable file that includes adapter-specific messages
    - `pingfederate-twitter-adapter-messages.properties` – a variable file that customizes the messages that appear on the template files.
- `template` – contains user-facing HTML template files
  - `twitter-pop-up-template.html` – a template that opens a pop-up window to prompt the user to sign on.
  - `twitter-post-auth-template.html` – a template that returns the user to the PingFederate sign-on flow after they sign on with Twitter.
    - `assets` – contains functional scripts and files used by the template
      - `css/twitter.css` – a CSS file that customizes the appearance of the template files.
  - `fonts/end-user` – contains template fonts and icons
    - `icons` – contains template icons
    - `proxima-nova` – contains template fonts
  - `images` – contains template image files
    - `ping-logo.svg` – an image file with company branding

- `lib/pf-authn-api-sdk-<version>.jar` – a JAR file that contains the PingFederate Authentication API SDK

Copyright © 2025 Ping Identity Corporation

Overview of Web Access Management integrations

PingFederate Web Access Management Integration Kit

The PingFederate Web Access Management (WAM) Integration Kit allows developers to integrate their applications with a PingFederate server acting as either an identity provider (IdP) or a Service Provider (SP).

PingFederate Web Access Management Token Translator

The PingFederate Web Access Management (WAM) Token Translator provides a Token Processor and a Token Generator for use with the PingFederate WS-Trust Security Token Service (STS).

Web Access Management (WAM) Integration Kit

The PingFederate WAM Integration Kit allows developers to integrate their applications with a PingFederate server acting as either an identity provider (IdP) or a Service Provider (SP).

The WAM IdP Adapter allows an IdP enterprise to extend an existing investment by using the SAML or WS-Federation protocols to expand the reach of the WAM domain to partner applications. The WAM SP Adapter allows an SP enterprise to accept SAML or WS-Federation assertions and provide secure internet single sign-on (SSO) to applications protected by a supported WAM system.

> **Important**
>
> This kit is designed to work with WAM products from multiple vendors. A WAM plug-in is required to connect the integration kit with each third-party system. This kit ships with WAM plug-ins compatible with Oracle Access Manager (OAM) 11g R2, and with RSA Access Manager 6.1.

> **Important**
>
> The current RSA plugin does not support Adaptive Authentication. It is only qualified against Authentication Manager.

A simple software development kit (SDK) is also included to create custom WAM plug-ins for other systems. If you are creating a WAM plug-in for any third-party product other than OAM and RSA Access Manager, you must complete the tasks in the WAM plug-in SDK `README.txt` file located in the `<integration_kit_install_dir>/sdk` directory.

Intended audience

This document is intended for PingFederate administrators with experience in the configuration and maintenance of the OAM Access Server or RSA Access Manager and other WAM tools, and developers with experience using JAVA SDKs

Before you start, you should be familiar with the following parts of the PingFederate documentation:

- Identity provider SSO configuration
- Managing IdP adapters

Please consult the WAM tool documentation if you encounter any difficulties in areas not directly associated with PingFederate or the WAM Integration Kit.

Copyright © 2025 Ping Identity Corporation

System requirements

- PingFederate 6.x or later
- WAM plug-in for the desired third-party system, built and deployed per the WAM plug-in SDK documentation
- Associated vendor-supplied libraries to support the WAM plug-in you are using
- Fully functional WAM plug-ins for OAM and RSA are included in the WAM Integration Kit package
- Separate third-party Web Agent configured using the WAM server administrative software

> **Important**
>
> PingFederate must be running in the same domain as the third-party WAM Web Agent for the applicable WAM Server.

Setup

WAM plug-in installation for RSA

About this task

This section describes how to deploy the pre-built RSA-compatible WAM plug-in for both IdP and SP adapters.

Steps

1. The additional RSA API libraries for creating a WAM plug-in to interact with PingFederate are included in the `<integration_kit_install_dir>/dist/rsa` directory.
    - `axm-runtime-api-6.1.4.jar`
    - `jsafeFIPS-6.1.jar`
    - `jsafeJCEFIPS-6.1.jar`
2. Copy the RSA API libraries from the `<integration_kit_install_dir>/dist/rsa` directory into the `<PF_install>/pingfederate/server/default/deploy` directory.
3. Copy the `pf-rsa-plugin.jar` from the `<integration_kit_install_dir>/dist/rsa` directory into the `<PF_install>/pingfederate/server/default/deploy`.
4. Complete the Install or upgrade the adapter prior to restarting the PingFederate server.

WAM plug-in installation for OAM

About this task

Deploy the pre-built OAM-compatible WAM plug-in for both IdP and SP adapters.

You can find more information on configuring the WAM plug-in for OAM in OAM-specific configuration.

Copyright © 2025 Ping Identity Corporation

Steps

1. Get the necessary OAM API library (`oamasdk-api.jar`) from the Oracle Identity Management Download site and save it to the `<PF_install>/pingfederate/server/default/deploy` directory.
2. Copy the applicable plugin file to the `<PF_install>/pingfederate/server/default/deploy` directory:
    - If you are using OAM 10g, copy `<integration_kit_install_dir>/dist/oam/pf-oam-plugin.jar` .
    - If you are using OAM 11g, copy `<integration_kit_install_dir>/dist/oam11g/pf-oam-11g-plugin.jar` .
3. Complete the Install or upgrade the adapter procedure before restarting the PingFederate server (refer to the next section).

OAM-specific configuration

When configuring the OAM adapter, the following values are needed:

Field	Description	Example Value
Cookie Path	Relative path in the URL where the cookie is active.	`/`
Protected Resource	The path (and optionally, the hostname) that defines the protected resource. This value comes from your OAM configuration.	`http://<OAM Host Identifier>/<Resource Path>`
Error URL	Optional field containing a URL used as a redirection target in the event of an error during SSO when using this adapter.	
User Identifier	HTTP header used to identify the end userID.	`OAM_REMOTE_USER`
Session Token Name	The name of the encrypted cookie used for SSO.	`ObSSOCookie`
Session Token Loggedoff Value	The value the Session Token should be set to when the user has logged out of OAM.	`loggedoutcontinue`

> **Note**
>
> The above values are examples and are dependent on the OAM environment. Ask your Oracle administrator for the values required in your environment.

Learn more about this configuration in the Oracle Access Manager documentation.

Custom WAM plug-in installation

About this task

This section describes how to deploy a custom WAM plug-in for both IdP and SP adapters.

Copyright © 2025 Ping Identity Corporation

Steps

1. If you are creating a WAM plug-in for a third-party WAM product not bundled with this kit, you must complete the tasks in the WAM plug-in SDK `<integration_kit_install_dir>/sdk/README.txt` file.

   > **Note**
   >
   > Contact the third-party vendor support department to obtain required third-party API libraries for creating a WAM plug-in to interact with PingFederate.

2. After completing the tasks in the WAM plug-in SDK `README.txt` file, copy the resultant WAM plug-in output JAR file (`<integration_kit_install_dir>SDK/lib/pf-<WAM_TYPE>-plugin.jar`) to the `<PF-install>/pingfederate/server/default/deploy` directory.

Result

The WAM Integration Kit requires a plug-in to connect with a specific WAM product (see the WAM plug-in SDK in the distribution package for sample code and more details on building the plug-in). The SDK consists of build scripts, libraries, and sample code.

> **Note**
>
> The WAM plug-in SDK is designed specifically to connect the WAM Integration Kit with a third-party WAM product, using an API provided by the vendor.

Install or upgrade the adapter

About this task

This topic describes how to install the WAM Integration Kit for both the IdP and the SP adapters.

> **Note**
>
> If you have already deployed version 2.5.1 or later of the OpenToken Adapter, skip steps 1-3 in the following procedure.

Steps

1. Stop the PingFederate server if it is running.
2. Delete the following files from the `<PF_install>/pingfederate/server/default/deploy` directory:
   - The WAM adapter JAR file (`pf-wam-adapter-<version>.jar`)
   - Any existing OpenToken Adapter files (`opentoken*.jar`)
     - If the adapter JAR filename indicates version 2.1 or earlier, delete the supporting library `opentoken-java-1.x.jar` .
3. Unzip the integration-kit distribution file and copy the following files from the `/dist` directory to the `<PF_install>/pingfederate/server/default/deploy` directory.
   - `opentoken-adapter-2.5.1.jar`

Copyright © 2025 Ping Identity Corporation

- `pf-wam-adapter-2.0.0.jar`

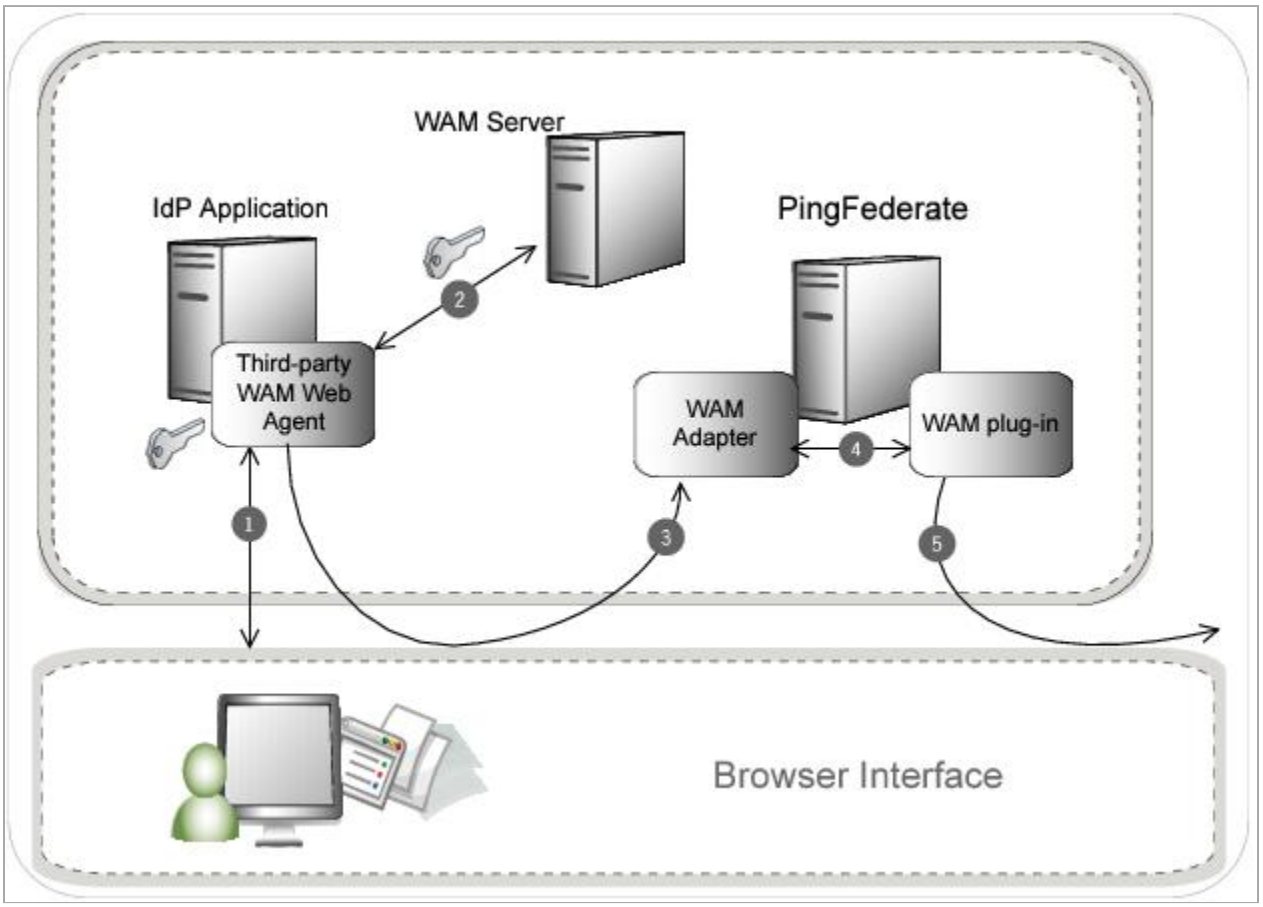
4. If you are running PingFederate 6.0 as a Windows service, add the following line to the `Java Library Path` section of the `pingfederate/sbin/wrapper/PingFederateService.conf` file:

```
wrapper.java.library.path.append_system_path=true
```

5. Start the PingFederate server.

IdP process overview

The following figure illustrates the request flow and how the WAM IdP Adapter is leveraged in generating a SAML/WS-Federation assertion using a WAM session cookie.



Processing Steps

1. The user's browser attempts to access the IdP application. The third-party WAM Web Agent intercepts the request and asks for the user's identity. The user enters the requested credentials and submits the login page.
2. The WAM Server validates the user's credentials and creates a WAM session cookie. The user now has access to the application.

Copyright © 2025 Ping Identity Corporation

3. The user clicks a link that initiates an SSO transaction to the partner application. The request is redirected to the PingFederate IdP Server. The WAM session cookie generated in step 2 is included in the request.
4. The PingFederate WAM IdP Adapter uses the WAM plug-in to decrypt the WAM session cookie and then transfers the attributes to the PingFederate IdP Server. You can create an attribute contract to map the WAM session cookie and response attributes. For more information, see Defining an attribute contract in the PingFederate documentation.
5. The PingFederate IdP server generates a SAML/WS-Federation assertion and redirects the request, with the assertion, back through the user's browser to the SP site.

Configuring an IdP adapter instance

Configure the WAM Integration Kit for an identity provider (IdP).

Before you begin

> **Important**
>
> You must create a third-party WAM Web Agent in your WAM tool before starting the following procedure. Several properties that you use to configure the agent are required to fill out the **IdP Adapter** tab in the following procedure. You can find more details on agent configuration in your WAM Server documentation.

Steps

1. In the PingFederate administrative console, create a new IdP adapter instance:

   Choose from:

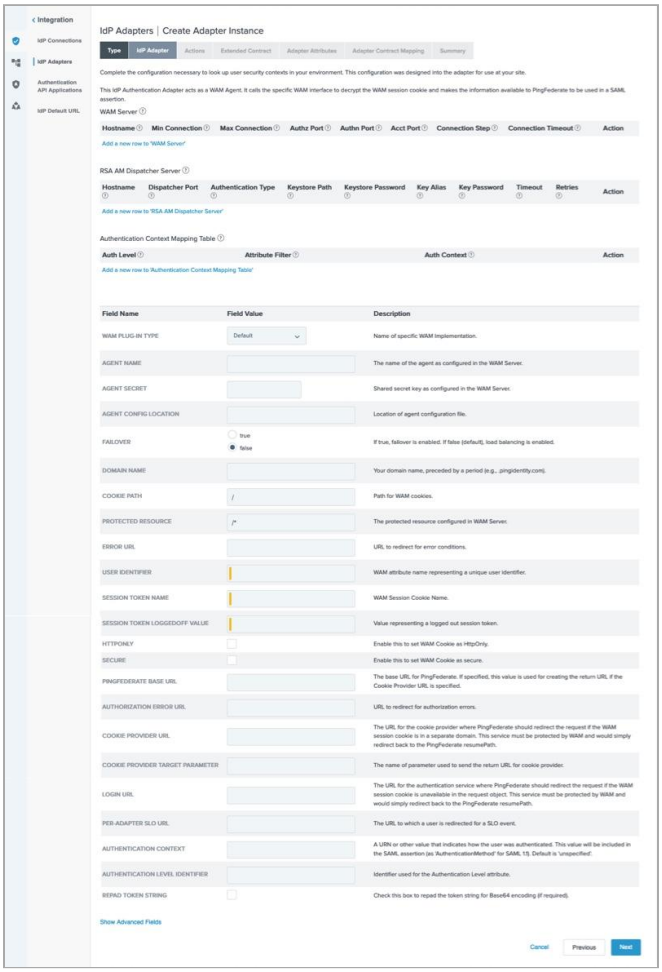
   - For PingFederate 10.1 or later: go to Authentication > Integration > IdP Adapters. Click Create New Instance.
   - For PingFederate 10.0 or earlier: go to Identity Provider > Adapters. Click Create New Instance.

2. On the Type tab, set the basic adapter instance attributes:

   1. In the Instance Name field, enter a name for the adapter instance.
   2. In the Instance ID field, enter a unique identifier for the adapter instance.
   3. In the Type list, select WAM IdP Adapter. Click Next.

   > **Note**
   >
   > - If you are configuring the adapter for a custom plug-in (not bundled with this kit), continue to step 5.
   > - If you are configuring the RSA AM Dispatcher server, continue with step 6.
   > - If you are configuring OAM, continue at step 7.

Copyright © 2025 Ping Identity Corporation



3. (Only for custom plug-ins for WAM servers other than OAM or RSA) On the IdP Adapter tab, click Add a new row to 'WAM Server' and enter the following information:

    1. Enter the hostname or the IP address where the WAM server is running.
    2. Specify the remaining WAM server values required for your configuration.
    3. In the Action column, click Update.
    4. Repeat this step as needed for additional custom WAM plug-ins.

    Skip the next step.

4. (Only for the RSA bundled plug-in) On the IdP Adapter tab, click Add a new row to 'RSA AM Dispatcher Server' and enter the following information:

> **Note**
>
> You must specify at least one RSA AM Dispatcher Server.

    1. Enter the hostname or the IP address and the (optional) dispatcher port where the RSA AM Dispatcher server is running.

Copyright © 2025 Ping Identity Corporation

> **Note**
>
> You must specify the authentication method used by the dispatcher server. If you have specified multiple dispatcher servers, each server can have individual authentication methods.

2. Specify the Authentication Type used by the RSA Dispatcher Server.
    - Clear: Clear text, no encryption.
    - Anon: Anonymous SSL, SSL encryption only.
    - Auth: Mutually authenticated SSL, SSL encryption with certificate-based encryption.
3. If you select Auth as the Authentication Type, you must specify the following RSA server values:
    - Keystore Path: The string filename of the private keystore file (PKCS12 only).
    - Keystore Password: The password for the private keystore.
    - Key Alias: The alias to your private key in the keystore.
    - Key Password: The private key password for keystore.
4. (Optional) Specify the Timeout value required for your configuration.
5. In the Action column, click Update.
6. Repeat this step as needed for additional RSA Servers.

5. (Only for custom plug-ins for WAM servers and the OAM bundled plug-in) On the IdP Adapter tab, click Add a new row to 'Authentication Context Mapping Table' and enter the following information:
    - Authentication Level: A specific value for a WAM system indicating the level of authentication an end-user has gone through.
    - Authentication Context: This is part of the SAML assertion.
    1. In the Action column, click Update. Repeat this step as needed.
6. Provide entries on the IdP Adapter tab, as described on the tab and in the following table.

> **Important**
>
> The selected WAM Plug-in Type may override optional and required fields.
> For example, if the selected **WAM Plug-in Type** is OAM, the **Agent Config Location** becomes a required field. Leaving this field blank generates an error message.

Field	Description
WAM Plug-in Type	The class name for the specific WAM implementation. **Note** The **WAM Plug-in Type** determines optional and required fields.

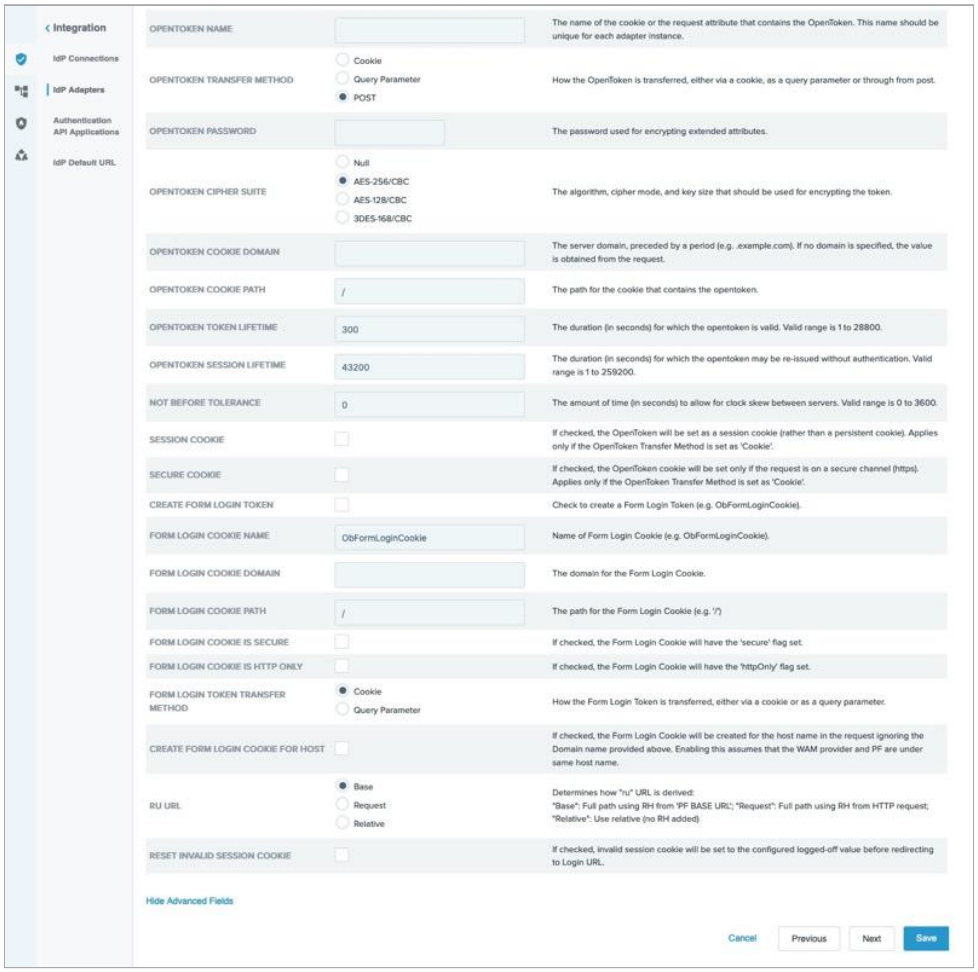
Copyright © 2025 Ping Identity Corporation

Field	Description
Agent Name	This value must match the value used when the third-party WAM Web Agent was configured.
Agent Secret	This value must match the value used when the third-party WAM Web Agent was configured.
Agent Config Location	Required for OAM. This value must contain the full path to an XML network-configuration file generated by the access-management system.
Failover	The default is false, indicating load balancing is enabled and user-session states and configuration data are shared among multiple WAM servers. Select true to enable failover, indicating that when one server fails, the next server is used.
Domain Name	Enter the fully-qualified domain name for the Cookie Domain where the WAM session cookie is stored, preceded by a period. For example, `.pingidentity.com`.
Cookie Path	(Required) The default value is the root directory (`/`). Specify a different path for the WAM session cookie location, if necessary. You can find more details in WAM Server documentation.
Protected Resource	(Required) The default value is all files in the root directory (`/*`). Specify a different path to the resources in the protected realm, if necessary.
Error URL	Enter a URL for redirecting the user if there are errors. For example, incorrect parameters in the link. This URL can contain query parameters. The URL has an `errorMessage` query parameter appended to it, which contains a brief description of the error that happened. The error page can optionally display this message on the screen to provide guidance on remedying the problem. When using the `errorMessage` query parameter in a custom error page, adhere to Web application security best practices to guard against common content injection vulnerabilities. If no URL is specified, the appropriate default error landing page appears. Learn more in Customizable user-facing pages in the PingFederate documentation. **Note** If you define an error redirect URL, errors are sent to the error URL and logged in the PingFederate server log, but not the PingFederate audit log.
User Identifier	(Required) Defines which attribute parsed from the WAM session cookie is the user identifier for use in the assertion.
Session Token Name	(Required) The WAM session cookie name.

Copyright © 2025 Ping Identity Corporation

Field	Description
Session Token LOGGEDOFF Value	(Required) The value representing a logged-out session token.
HTTP Only	Enable this option to set the WAM Session Cookie as HTTP Only. If this option is enabled, the browser will send the WAM Session Cookie through HTTP or HTTPS.
Secure	Enable this option to set the WAM Session Cookie as secure. If this option is enabled, the browser will send the WAM Session Cookie via HTTPS only.
PingFederate Base URL	The base URL for PingFederate. If specified, this value is used for creating the return URL if the Cookie Provider URL is used.
Authorization Error URL	URL to redirect for authorization errors.
Cookie Provider URL	The URL for the cookie provider where PingFederate should redirect the request if the WAM session cookie is in a separate domain. This service must be protected by the WAM server and would simply redirect back to the PingFederate `resumePath`.
Cookie Provider Target Parameter	The name of the parameter used to send the return URL for the cookie provider.
Login URL	An optional URL for the authentication service. If the WAM session cookie is not found in the request, PingFederate redirects the request to the URL page along with the relative `resumePath`. This service must be protected by the WAM Agent.
Per-Adapter SLO URL	If a URL is entered into this field, it will be used as the redirect target during SLO for this adapter instance, instead of the default value from PingFederate.
Authentication Context	This may be any value agreed upon with your SP partner that indicates how the user was authenticated. The value is included in the SAML assertion. Standard URIs are defined in the SAML specifications. You can find more information in the Authentication Context for the OASIS Security Assertion Markup Language(SAML) V2.0 PDF on the OASIS site.
Authentication Level Identifier	Identifier used for the Authentication Level attribute.
Repad Token String	Enable this to pad the incoming session token (if required).

Copyright © 2025 Ping Identity Corporation



7. **(Optional) Click Show Advanced Fields to specify OpenToken configuration values or settings, depending on your network configuration and other requirements at your site.**

   This section also contains fields for configuring tokens capturing the original request information if necessary. This functionality is based on the ObFormLoginCookie from OAM.

> **Note**
>
> If you want to configure the use of OpenToken as part of the WAM adapter configuration, complete the fields as described on the **IdP Adapter** tab and in the table below.

Field	Description
OpenToken Name	**The name of the cookie or request attribute that contains the OpenToken. This name should be unique for each adapter instance.**
OpenToken Transfer Method	**Specifies how the OpenToken is transferred. Select one of the following options:** **◦ Through a cookie** **◦ As a query parameter** **◦ Through Form Post**

Copyright © 2025 Ping Identity Corporation

Field	Description
OpenToken Password	The password used for encrypting extended attributes. **Note** This password is also used to generate the configuration file that the OpenToken agent uses. You must configure a value for the **OpenToken Password** even if the **OpenToken Cipher Suite** is set to `NULL`.
OpenToken Cipher Suite	The algorithm, cipher mode, and key size to use for token encryption.
OpenToken Cookie Domain	The server domain, preceded by a period. For example, `.example.com`. If you don't specify a domain, the value is obtained from the request.
OpenToken Cookie Path	The path for the cookie that contains the OpenToken.
OpenToken Token Lifetime	The duration (in seconds) for which the OpenToken is valid. Range is 1 - 28800.
OpenToken Session Lifetime	The duration (in seconds) during which the OpenToken can be re-issued without authentication. Range is 1 - 259200.
Not Before Tolerance	The amount of time (in seconds) to allow for clock skew between servers. Range is 0 - 3600.
Session Cookie	If selected, the OpenToken cookie is set as a session cookie (rather than a persistent cookie). Applies only if the OpenToken Transfer Method is set as `Cookie`.
Secure Cookie	If selected, the OpenToken cookie is set only if the request is on a secure channel (HTTPS). Applies only if the OpenToken Transfer Method is set as `Cookie`.
Create Form Login Token	If selected, a Token is created containing the information needed to retain the original request information if a redirect to a form authentication page is required. The token contents are implemented based on the requirements of the ObFormLoginCookie from OAM.
Form Login Cookie Name	The name given to the created Form Login Cookie For example, `ObFormLoginCookie`.
Form Login Cookie Domain	The server domain, preceded by a period. For example, `.example.com`.
Form Login Cookie Path	The path for the cookie that contains the Form Login Cookie.
Form Login Cookie is Secure	Sets the `secure` flag on the Form Login Cookie.

Copyright © 2025 Ping Identity Corporation

Field	Description
Form Login Cookie is HTTP Only	Sets the `httpOnly` flag on the Form Login Cookie.
Form Login Token Transfer Method	Specifies how the Form Login Token is transferred. Select one of the following options: ◦ Through a cookie ◦ As a query parameter
Create Form Login Cookie for Host	If selected, the Form Login Cookie is created for the hostname in the request, ignoring the Form Login Cookie Domain.
RU URL	Determines how the ru URL is derived: ***Base*** Full path using the RH from the PF BASE URL. ***Request*** Full path using the RH from the HTTP request. ***Relative*** Use relative (do not add RH).
Reset Invalid Session Cookie	If selected, the invalid session cookie is set to the configured logged-off value before redirecting to the Login URL.

8. Click Next.
9. On the Actions tab, click the Test Connection link to validate the WAM configuration.

> **Note**
>
> If you're using an OpenToken Adapter Configuration, click the **Invoke Download** link and then click **Export** to download the `agent-config.txt` properties to a directory that the WAM Web Agent can read.

10. On the Extended Contract tab, add any attributes that you want to include in the contract. Click Next.
11. On the Adapter Attributes tab, select userId or wamSessionToken under Pseudonym. You can also select any extended attributes specified on the previous screen. Click Next.

    Learn more in Set pseudonym and masking options in the PingFederate documentation. Click Next.
12. On the Summary tab, check and save your configuration:

    Choose from:

    - For PingFederate 10.1 or later: Click Save.
    - For PingFederate 10.0 or earlier: Click Done. On the Manage IdP Adapter Instances tab, click Save.

Copyright © 2025 Ping Identity Corporation

IdP deployment note

The adapter configuration supports a "login URL" parameter. If the WAM session cookie is not found in the request, then the PingFederate server redirects the request to the URL page along with the relative `resumePath`, which is generated from PingFederate and intended for asynchronous communication between the adapter and the external application. (The state is saved in PingFederate, and processing is resumed when the application redirects to the `resumePath`.)

The login URL page can authenticate the user and redirect the request back to PingFederate. An example of a JSP code snippet for redirecting the request is shown below.

```
<%
  String resumePath = request.getParameter("resumePath");
    if(resumePath != null) {
      resumePath = <PingFed_URL> + resumePath; (1)
      response.sendRedirect(resumePath);
    }
%>
```

1 **`<PingFed_URL>`** is the fully-qualified URL of the PingFederate server.

Test the IdP adapter

About this task

You can test this adapter using the samples applications that are included in the Java Integration Kit. Follow this procedure to verify adapter functions:

Steps

1. Download the Java Integration kit from the PingFederate server add-ons page.
2. Complete the steps in Sample application setup in the Java Integration Kit documentation to set up an IdP application.
3. Configure an instance of the WAM Adapter.
4. Reconfigure the SP connection to use the WAM Adapter instance.

   Delete the existing adapter instance and map the WAM Adapter instance in its place. For details, see Managing mappings in the PingFederate documentation.
5. On a web page protected by the third-party WAM web Agent, create an "SSO" link to the PingFederate `startSSO` endpoint, including the sample SP's connection ID, in the following format:

   ```
   http[s]://<PF_host>:<port>/IdP/startSSO.ping?PartnerSpId=<connection_id>
   ```

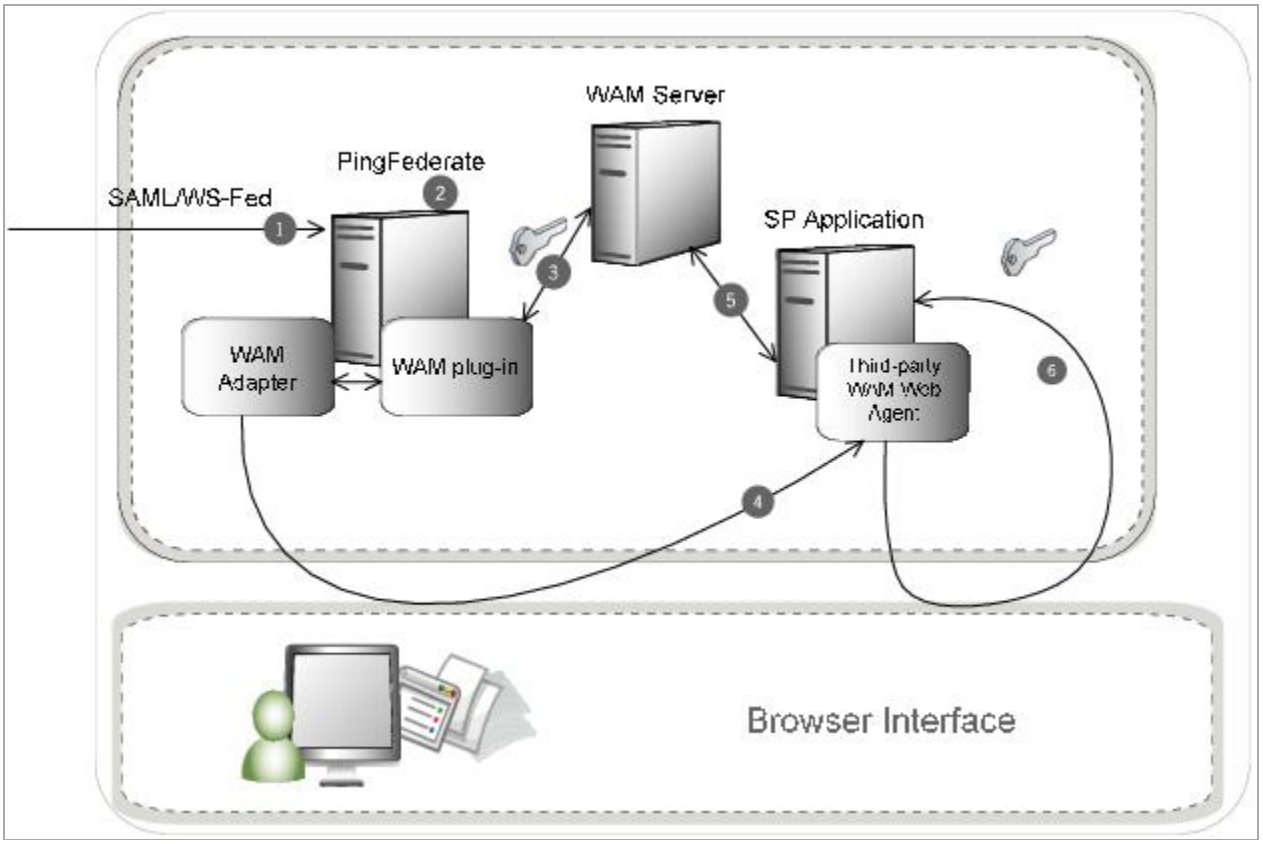
   - `<PF_host>` is the machine running the PingFederate server.
   - `<port>` is the PingFederate port (default value: `9031`).
   - `<connection_id>` is the Connection ID of the SP connection.

Copyright © 2025 Ping Identity Corporation

6. Access the protected web page by authenticating through the WAM web Agent and click the SSO link.
7. You are logged on to the Java sample application.

SP process overview

The following figure illustrates the request flow and how the WAM SP Adapter leverages a SAML/WS-Federation assertion to create a WAM session cookie.



Processing Steps

1. The PingFederate SP server receives a SAML/WS-Federation assertion from the IdP.
2. PingFederate parses the assertion.
3. The WAM SP Adapter uses the WAM plug-in to create a WAM session cookie and embeds the cookie in the response.
4. A request containing the WAM session cookie is redirected to the browser.
5. The request is then redirected to the SP Application, which is protected by the third-party WAM Web Agent.
6. The third-party WAM Web Agent intercepts the request, extracts and validates the WAM session cookie, and allows access to the application.

Copyright © 2025 Ping Identity Corporation

Configuring an SP adapter instance

Configure the WAM Integration Kit for a service provider (SP).

If you are using OAM, you can find configuration information in Creating a Custom Authentication Scheme for OAM.

Before you begin

> **Important**
>
> You must create a third-party WAM Web Agent in your WAM tool before starting the following procedure. Several properties that you use to configure the agent are required to fill out the **Instance Configuration** tab in the following procedure.
> You can find more details on configuring an agent in your WAM Server documentation.

Steps

1. In the PingFederate administrative console, create a new SP adapter instance.

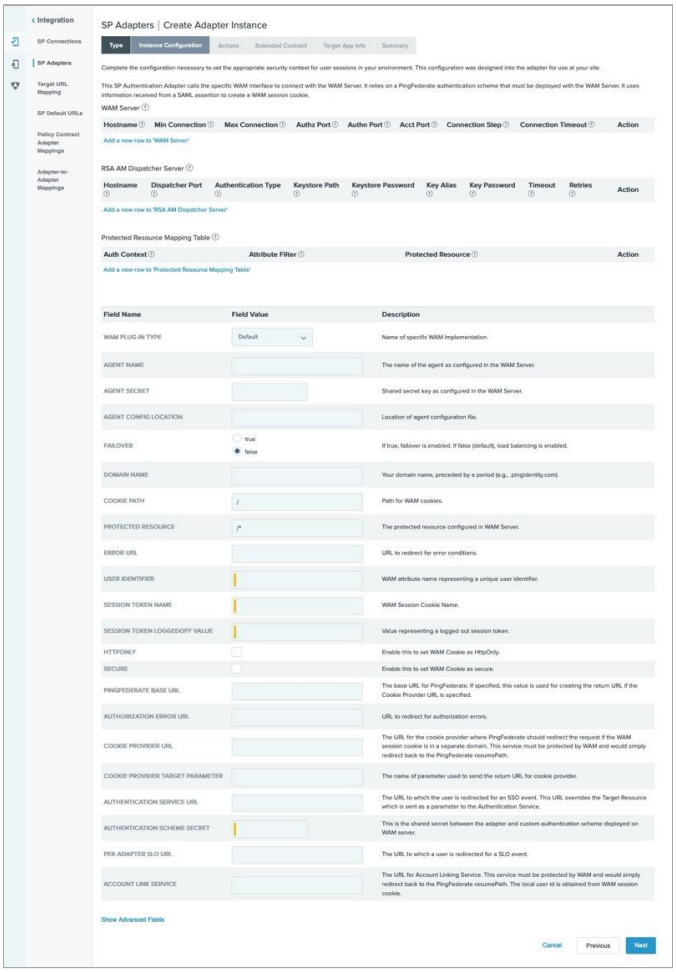
   Choose from:

   - For PingFederate 10.1 or later: go to Applications > Integration > SP Adapters. Click Create New Instance.
   - For PingFederate 10.0 or earlier: go to Service Provider > Adapters. Click Create New Instance.

2. On the Type tab, set the basic adapter instance attributes:
   1. In the Instance Name field, enter a name for the adapter instance.
   2. In the Instance ID field, enter a unique identifier for the adapter instance.
   3. In the Type list, select WAM SP Adapter. Click Next.

      > **Note**
      >
      > - If you are configuring the adapter for a custom plug-in (not bundled with this kit), continue to step 5.
      > - If you are configuring the RSA AM Dispatcher server, continue with step 6.
      > - If you are configuring OAM, continue at step 7.

Copyright © 2025 Ping Identity Corporation



3. **(Only for custom plug-ins for WAM servers other than OAM or RSA) On the Instance Configuration tab, click Add a new row to 'WAM Server' and enter the following information:**

    1. **Enter the Hostname or the IP address where the WAM server is running.**
    2. **Specify the remaining WAM server values required for your configuration.**
    3. **Click Update in the Action column.**
    4. **Repeat this step as needed for additional WAM plug-ins.**

    Skip the next step.

4. **(Only for the RSA bundled plug-in) On the Instance Configuration tab, click Add a new row to 'RSA AM Dispatcher Server' and enter the following information:**

> **Note**
>
> You must specify at least one RSA AM Dispatcher Server.

    1. **Enter the Hostname or the IP address and the (optional) Dispatcher Port where the RSA AM Dispatcher server is running.**

Copyright © 2025 Ping Identity Corporation

> **Note**
> You must specify the authentication method used by the dispatcher server. If you specify multiple dispatcher servers, each server can have individual authentication methods.

2. Specify the Authentication Type used by the RSA Dispatcher Server:
    - Clear: Clear text, no encryption.
    - Anon: Anonymous SSL, SSL encryption only.
    - Auth: Mutually authenticated SSL, SSL encryption with certificate-based encryption.
3. If you select Auth as the Authentication Type, you must specify the following RSA server values:
    - Keystore Path: String filename of the private keystore file (PKCS12 only).
    - Keystore Password: The password for the private keystore.
    - Key Alias: The alias to your private key in the keystore.
    - Key Password: The private key password for the keystore.
4. (Optional) Specify the Timeout value required for your configuration.
5. In the Action column, click Update.
6. Repeat this step as needed for additional RSA Servers.

5. (Optional) Only for custom plug-ins for WAM servers and the OAM bundled plug-in) On the SP Adapter tab, click Add a new row to 'Protected Resource Mapping Table' and enter the following information:
    - Authentication Context: This is part of the SAML assertion.
    - Attribute Filter: The names and values of attributes that the assertion must contain for this protected resource.
    - Protected Resource: The protected resource to be accessed if the authentication context and attribute filters in the assertion match the provided values.

   Click Update in the Action column. Repeat this step as needed.
6. Provide entries on the Instance Configuration tab, as described on the tab and in the table below.

> **Note**
> The selected WAM Plug-in Type might override optional andrequired fields.
> For example, if the selected **WAM Plug-in Type** is OAM, the **Agent Config Location** becomes a required field. Leaving this field blank generates an error message.

Field	Description
WAM Plug-in Type	The class name for the specific WAM implementation. **Note** The **WAM Plug-in Type** determines optional and required fields.

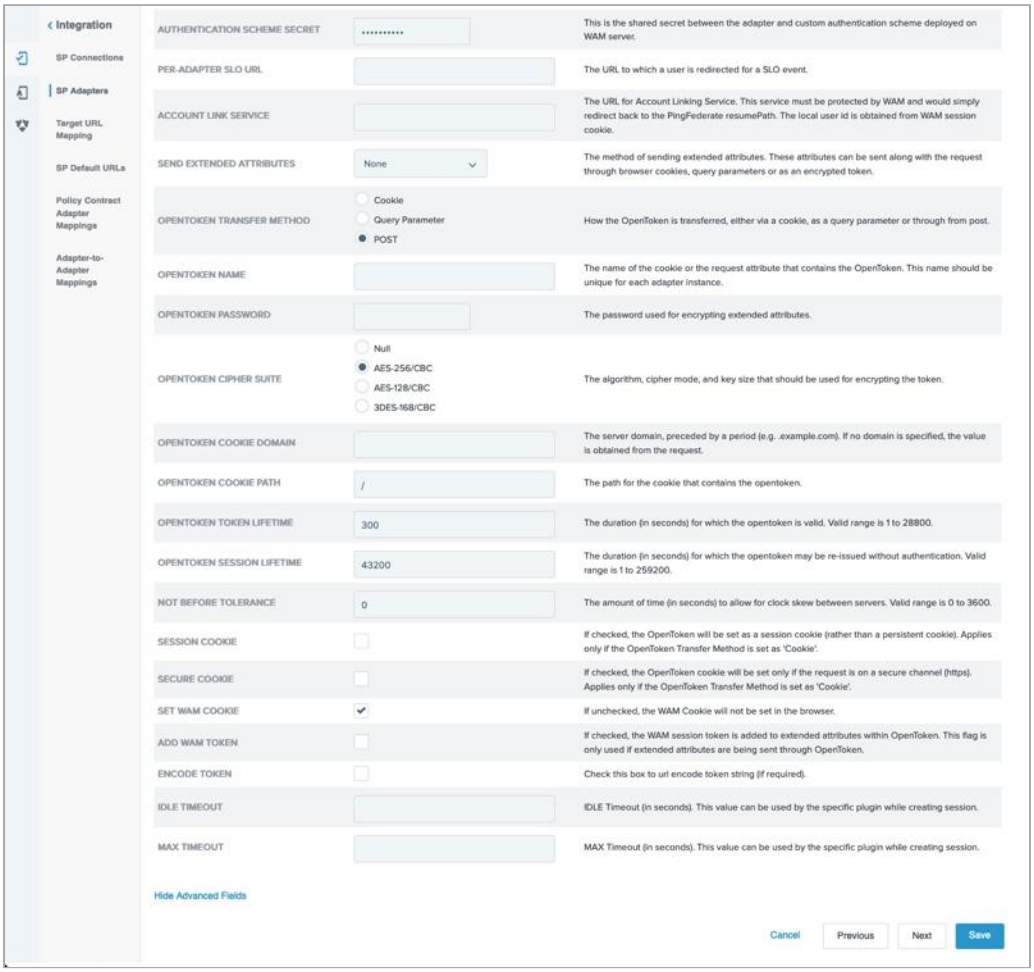
Copyright © 2025 Ping Identity Corporation

Field	Description
Agent Name	This value must match the value used when the third-party WAM Web Agent was configured.
Agent Secret	This value must match the value used when the third-party WAM Web Agent was configured.
Agent Config Location	Required for OAM. This value must contain the full path to an XML network-configuration file generated by the access-management system.
Failover	The default is false, indicating load balancing is enabled and user-session states and configuration data are shared among multiple WAM servers. Select true to enable failover, indicating that when one server fails, the next server is used.
Domain Name	Enter the fully-qualified domain name for the Cookie Domain where the WAM session cookie is stored, preceded by a period. For example, `.pingidentity.com`
Cookie Path	(Required) The default value is the root directory (`/`). Specify a different path for the WAM session cookie location, if necessary. You can find more details in WAM Server documentation.
Protected Resource	(Required) The default value is all files in the root directory (`/*`). Specify a different path to the resources in the protected realm, if necessary.
Error URL	Enter a URL for redirecting the user if there are errors. For example, incorrect parameters in the link. This URL can contain query parameters. The URL has an `errorMessage` query parameter appended to it, which contains a brief description of the error that happened. The error page can optionally display this message on the screen to provide guidance on remedying the problem. When using the `errorMessage` query parameter in a custom error page, adhere to Web application security best practices to guard against common content injection vulnerabilities. If no URL is specified, the appropriate default error landing page appears. Learn more in Customizable user-facing pages in the PingFederate documentation. **Note** If you define an error redirect URL, errors are sent to the error URL and logged in the PingFederate server log, but not the PingFederate audit log.
User Identifier	(Required) Defines which attribute parsed from the WAM session cookie is the user identifier for use in the assertion.
Session Token Name	(Required) The WAM session cookie name.

Copyright © 2025 Ping Identity Corporation

Field	Description
Session Token LOGGEDOFF Value	(Required) The value representing a logged-out session token.
HTTP Only	Enable this option to set the WAM Session Cookie as HTTP Only. If this option is enabled, the browser will send the WAM Session Cookie through HTTP or HTTPS.
Secure	Enable this option to set WAM Session Cookie as secure. If this option is enabled, the browser sends the WAM Session Cookie through HTTPS only.
PingFederate Base URL	The base URL for PingFederate. If specified, this value is used for creating the return URL if the Cookie Provider URL is being used.
Authorization Error URL	The URL to redirect for authorization errors.
Cookie Provider URL	The URL for the cookie provider where PingFederate should redirect the request if the WAM session cookie is in a separate domain. This service must be protected by WAM and redirect back to the PingFederate `resumePath`.
Cookie Provider Target Parameter	The name of the parameter used to send the return URL for the cookie provider.
Authentication Service URL	The URL to which the user is redirected for an SSO event. This URL overrides the Target Resource, which is sent as a parameter to the Authentication Service.
Authentication Scheme Secret	(Required, except for RSA) The shared secret between the adapter and the custom authentication scheme deployed on the WAM server.
Per-Adapter SLO URL	Enter a URL to use as the redirect target during SLO for this adapter instance, instead of the default value from PingFederate.
Account Link Service	The URL for the Account Linking Service. This service must be protected by the WAM Web Agent and redirect back to the PingFederate `resumePath`. The local user id is obtained from the WAM session cookie.

Copyright © 2025 Ping Identity Corporation



7. **(Optional) Click Show Advanced Fields to configure how to send extended attributes or to specify OpenToken configuration values or settings.**

   Learn more in Configuring an OpenToken SP Adapter instance in the PingFederate documentation.

   > **Note**
   >
   > If you want to configure the use of OpenToken as part of the WAM adapter configuration, complete the fields as described on the **IdP Adapter** tab and in the following table.

   You can change default values or settings, depending on your network configuration and other requirements at your site.

Field	Description
Send Extended Attributes	**The method of sending extended attributes. These attributes can be sent along with the request through browser cookies, query parameters, or as an encrypted token.** **Note** Learn more about defining the attributes on the **Extended Contract** tab in step 10.

Copyright © 2025 Ping Identity Corporation

Field	Description
OpenToken Transfer Method	Specifies how the OpenToken is transferred. Select one of the following options: ◦ Through a cookie ◦ As a query parameter ◦ Through Form Post
OpenToken Name	The name of the cookie or request attribute that contains the OpenToken. This name should be unique for each adapter instance.
OpenToken Password	The password used for encrypting extended attributes. **Note** This password is also used to generate the configuration file that the OpenToken agent uses. You must configure a value for the **OpenToken Password** even if the **OpenToken Cipher Suite** is set to `NULL`.
OpenToken Cipher Suite	The algorithm, cipher mode, and key size to use for token encryption.
OpenToken Cookie Domain	The server domain, preceded by a period. For example, `.example.com`. If you don't specify a domain, the value is obtained from the request.
OpenToken Cookie Path	The path for the cookie that contains the OpenToken.
OpenToken Token Lifetime	The duration (in seconds) during which the OpenToken is valid. Range is 1 - 28800.
OpenToken Session Lifetime	The duration (in seconds) during which the OpenToken can be re-issued without authentication. Range is 1 - 259200.
Not Before Tolerance	The amount of time (in seconds) to allow for clock skew between servers. Range is 0 - 3600.
Session Cookie	If selected, OpenToken is set as a session cookie (rather than a persistent cookie). Applies only if the OpenToken Transfer Method is set as `Cookie`.
Secure Cookie	If selected, the OpenToken cookie is set only if the request is on a secure channel (https). Applies only if the OpenToken Transfer Method is set as 'Cookie'.
Set WAM Cookie	If cleared, the WAM Cookie isn't set in the browser.
Add WAM Token	If selected, adds the WAM session token to extended attributes in OpenToken. You can only use this flag if you're sending extended attributes through OpenToken.

Copyright © 2025 Ping Identity Corporation

Field	Description
Encode Token	Select this checkbox to URL encode the token string if the WAM provider requires it.
Idle Timeout	The idle timeout (in seconds). This value can be used by the specific plugin while creating the session if required.
Max Timeout	The max timeout (in seconds). This value can be used by the specific plugin while creating the session.

8. Click Next.
9. On the Actions tab, click Test Connection to validate the WAM configuration.

> **Note**
>
> If you're using an OpenToken Adapter Configuration, click the **Invoke Download** link and then click **Export** to download the `agent-config.txt` properties to a directory that the WAM Web Agent can read.

10. On the Extended Contract tab, add any attributes you want to include in the request header. Click Next.
11. On the Summary tab, check and save your configuration.

    Choose from:

    - For PingFederate 10.1 or later: click Save.
    - For PingFederate 10.0 or earlier: click Done. On the Manage SP Adapter Instances tab, click Save.

Creating a Custom Authentication Scheme for OAM

About this task

The SP Adapter uses a custom authentication scheme when creating a WAM session and validates authentication requests coming from PingFederate. This section describes how to deploy the OAM-compatible Java-based PingFederate Custom Authentication Scheme.

Steps

1. Import the `<integration_kit_install_dir>/dist/PingCustomAuthPlugin.jar` into the OAM.

   The `PingCustomAuthPlugin.jar` file is a custom authentication scheme that supports OAM.

2. Configure your Access Server to use the custom authentication plug-in by creating or modifying a custom authentication scheme.

   To learn more, refer to Creating Custom Authentication Plug-ins in the Oracle documentation.

Copyright © 2025 Ping Identity Corporation

> **Note**
>
> The secret you specify when creating the custom authentication scheme must match the secret stored in the PingFederate SP Adapter.

SP deployment notes

The following notes provide additional information for using the WAM Integration Kit as an SP:

- The WAM SP Adapter relies on a custom authentication scheme to validate the authentication request coming from the PingFederate SP Adapter. The secret specified in the SP Adapter is verified against the one configured with the scheme. You can create custom authentication schemes for specific WAM systems using their API.

  The authentication scheme for OAM is included in the samples folder at the following location: `<integration_kit_install_dir> /sdk/samples/oam/PingCustomAuthPlugin.java`

- To support Account Linking, the Account Linking Service has to be implemented and then protected by the WAM Web Agent. This could be done as a `JSP` page that redirects back to PingFederate. The relative `resumePath` is sent as part of the request and the `JSP` page needs to create the absolute URL and redirect, as shown below.

```
<%
  String resumePath = request.getParameter("resumePath");
    if(resumePath != null) {
      resumePath = <PingFed_URL> + resumePath; (1)
      response.sendRedirect(resumePath);
    }
%>
```

1 `<PingFed_URL>` is the fully-qualified URL of the PingFederate server. `resumePath` is generated from PingFederate and intended for asynchronous communication between the adapter and the external application. The state is saved in PingFederate and processing is resumed when the application redirects to the `resumePath`.

The WAM SP Adapter retrieves the user information from the WAM session cookie and resumes SSO.

Test the SP adapter

About this task

You can test this adapter using the samples applications that are included in the Java Integration Kit.

Steps

1. Download the Java Integration kit from the PingFederate server add-ons page.
2. Complete the steps in Sample application setup in the Java Integration Kit documentation to set up an SP application.
3. Configure an instance of the WAM Adapter as shown in Setting Up the SP Adapter.
4. Reconfigure the IdP connection to use the WAM Adapter instance.

Copyright © 2025 Ping Identity Corporation

Delete the existing adapter instance for the connection and map the WAM Adapter instance in its place.

5. From the Main Menu, click Adapters under My SP Configuration.
6. Protect a web page using the WAM web Agent.
7. On the same web server, create an unprotected web page with a hyperlink to PingFederate's SP-initiated SSO endpoint in the following format:

```
http[s]://<PF_host>:<port>/sp/startSSO.ping?TargetResource=<protected_resource>&PartnerIdpId=<connection_id>
```

   - `<PF_host>` is the machine running the PingFederate server.
   - `<port>` is the port (default value: `9031`).
   - `<protected_resource>` is the web page protected in the previous step.
   - `<connection_id>` is the Connection ID of the IdP connection.
8. Click the SSO link on the unprotected web page.

   You should arrive at the IdP application's login page.
9. Add at least one of the users in the username drop-down list to the WAM Server.

   Refer to your WAM platform documentation for more information.
10. On the IdP application's login page, log in with a username managed by your WAM platform.

    You should be redirected to a WAM platform-protected web page. Independently, you can view cookies from your browser to see that a WAM session cookie has been created.

Release notes

Changelog

Web Access Management Integration Kit 2.0 – August 2014 (Current Release)

- Added a per-adapter SLO URL field
- Added an Authentication Mapping Table with an attribute filter for the IdP adapter
- Added a Protected Resource Mapping Table to the SP adapter with an attribute filter
- Enabled cookie provider functionality during SLO
- Added an encode token field to the SP adapter
- Added support for ObFormLoginCookie with customizable rh value

Web Access Management Integration Kit 1.2 – August 2013

- Added support for Cookie Provider

Copyright © 2025 Ping Identity Corporation

- Added OpenToken support within WAM Adapter
- Added Authentication Level to Authentication Context mapping table

Web Access Management Integration Kit 1.1 – March 2013

- WAM kit now includes WAM plug-in compatible with RSA Access Manager

Web Access Management Integration Kit 1.0.1 – December 2012

- Updated to address security issue found since the previous release.
- Added support for OpenToken 2.5.1 Adapter

Web Access Management Integration Kit 1.0 – November 2012

- Initial Release
- Redesigned former product-specific kits to provide consistent functionality across multiple WAM products

Download manifest

The following files are included in the `.zip` archive:

- `ReadMeFirst.pdf` – contains links to this online documentation
- `/legal` – contains this document:
  - `Legal.pdf` – copyright and license information
- `/dist` – contains libraries needed to run the adapter:
  - `pf-wam-adapter-2.0.jar` – the WAM Adapter JAR file
  - `opentoken-adapter-2.5.1.jar` – OpenToken Adapter JAR file
- `/dist/oam` – contains Oracle Access Manager libraries needed to run the adapter:
  - `pf-oam-plugin.jar` – Pre-built OAM-compatible WAM plug-in JAR file
  - `PingCustomAuthPlugin.jar` – a Java-based PingFederate Custom Authentication Scheme
- `/dist/rsa` – contains RSA libraries needed to run the adapter:
  - `pf-rsa-plugin.jar` – Pre-built RSA-compatible WAM plug-in JAR file
  - `axm-runtime-api-6.1.4.jar` - RSA API library
  - `jsafeFIPS-6.1.jar` – RSA API library
  - `jsafeJCEFIPS-6.1.jar` – RSA API library
- `/sdk` – contains build scripts, documents, libraries, and sample code to build a WAM plug-in:
  - `README.txt` – contains instructions for creating a third-party WAM plug-in to interact with PingFederate.
  - `/docs` – contains documentation on how to build a WAM plug-in.

Copyright © 2025 Ping Identity Corporation

- `/lib` – contains libraries and supporting files needed to build a WAM plug-in.
- `/samples` – contains sample code used to build a WAM plug-in.

Web Access Management (WAM) Token Translator

The PingFederate Web Access Management (WAM) Token Translator provides a Token Processor and a Token Generator for use with the PingFederate WS-Trust Security Token Service (STS).

The Token Processor allows an identity provider (IdP) STS to accept and validate a WAM session token from a Web Service Client (WSC) and then map user attributes into a SAML token for the WSC to send to a Web Service Provider (WSP). The Token Generator allows a Service Provider (SP) STS to issue a WAM session token for a WSP, including mapped attributes from an incoming SAML token.

> **Important**
>
> The Token Translator is designed to work with WAM products from multiple vendors. A WAM plug-in is required to connect the Token Translator with each third-party system. This kit ships with WAM plug-ins compatible with Oracle Access Manager (OAM) 10g and 11g, and with RSA Access Manager 6.1. A simple software development kit (SDK) is also included to create custom WAM plug-ins for other systems.

If you are creating a WAM plug-in for any third-party product other than OAM and RSA Access Manager, you must complete the tasks in the WAM plug-in SDK `README.txt` file located in the `<token_translator_install_dir>/sdk` directory.

> **Important**
>
> Ping Identity provides an SDK for enabling Web Service applications (Client or Provider) to interact with the PingFederate STS. The SDK is available for download on the PingFederate server add-ons page.

Intended audience

This document is intended for PingFederate administrators.

If you need help during the setup process, see the following resources:

- The following sections of the PingFederate documentation:
  - WS-Trust STS configuration

Please consult the WAM documentation tool if you encounter any difficulties in areas not directly associated with PingFederate or the WAM Token Translator.

System requirements

- PingFederate 6.x or later
- WAM plug-in for the desired third-party system, built and deployed per the WAM plug-in SDK documentation
- Associated vendor-supplied libraries to support the WAM plug-in you are using
- Fully functional WAM plug-ins for OAM and RSA are included in the WAM Token Translator package.

Copyright © 2025 Ping Identity Corporation

- Separate third-party Web Agent configured using the WAM server administrative software

Setup

WAM plug-ins

This kit ships with WAM plug-ins compatible with OAM 10g and 11g, RSA Access Manager 6.1, and a simple SDK to create custom WAM plug-ins for other systems.

Custom WAM plug-in installation

About this task

This section describes how to deploy a custom WAM plug-in for both Token Processors and Token Generators.

Steps

1. If you are creating a WAM plug-in for any third-party WAM product not bundled with this kit, you must complete the tasks in the WAM plug-in SDK `README.txt` file located in the `<token_translator_install_dir>/sdk` directory.

   > **Note**
   >
   > Contact the third-party vendor support department to obtain required third-party WAM API libraries for creating a WAM plug-in to interact with PingFederate.

2. If you are deploying for a third-party WAM product, copy the resultant WAM plug-in output JAR file `pf-<WAM_TYPE>-plugin.jar` from the `<token_translator_install_dir>/sdk/samples/<WAM_TYPE>` directory into the `<PF_install>/pingfederate/server/default/deploy` directory.

Result

The WAM Token Translator requires a plug-in to connect with a specific WAM product (see the WAM plug-in SDK in the distribution package for sample code and more details on building the plug-in).

> **Note**
>
> The WAM plug-in SDK is designed specifically to connect the WAM Token Translator with a third-party WAM product, using an API provided by the vendor.

WAM plug-in installation for RSA

About this task

This section describes how to deploy the pre-built RSA-compatible WAM plug-in for both Token Processor and Token Generators.

Copyright © 2025 Ping Identity Corporation

Steps

1. Copy the following API libraries from the `<token_translator_install_dir>/dist/rsa` directory to `<PF_install>/pingfederate/server/default/deploy`:
    - `axm-runtime-api-6.1.4.jar`
    - `jsafeFIPS-6.1.jar`
    - `jsafeJCEFIPS-6.1.jar`
    - `pf-rsa-plugin.jar`
2. Complete the Token translator installation prior to restarting the PingFederate server.

WAM plug-in installation for OAM

About this task

This section describes how to deploy the pre-built OAM-compatible WAM plug-in for both Token Processors and Token Generators.

Steps

1. Get the necessary OAM API library (`oamasdk-api.jar`) from the Oracle Identity Management Download site and save it to the `<PF_install>/pingfederate/server/default/deploy` directory.
2. Copy the `<token_translator_install_dir>/dist/pf-oam-plugin.jar` file to the `<PF_install>/pingfederate/server/default/deploy` directory.
3. Complete the Token translator installation prior to restarting the PingFederate server (see next section).

Token translator installation

About this task

This topic describes how to install the WAM Token Translator to configure the Token Processor, Token Generator, or both.

> **Note**
>
> If you have already deployed version 2.5.1 or later of the OpenToken Adapter, skip steps 1 through 3 in the following procedure.

Steps

1. Stop the PingFederate server if it is running.
2. Delete any existing OpenToken Adapter files (`opentoken*.jar`) from the `<PF_install>/pingfederate/server/default/deploy` directory.
    - If the adapter JAR filename indicates version 2.1 or earlier, delete the supporting library `opentoken-java-1.x.jar`.

Copyright © 2025 Ping Identity Corporation

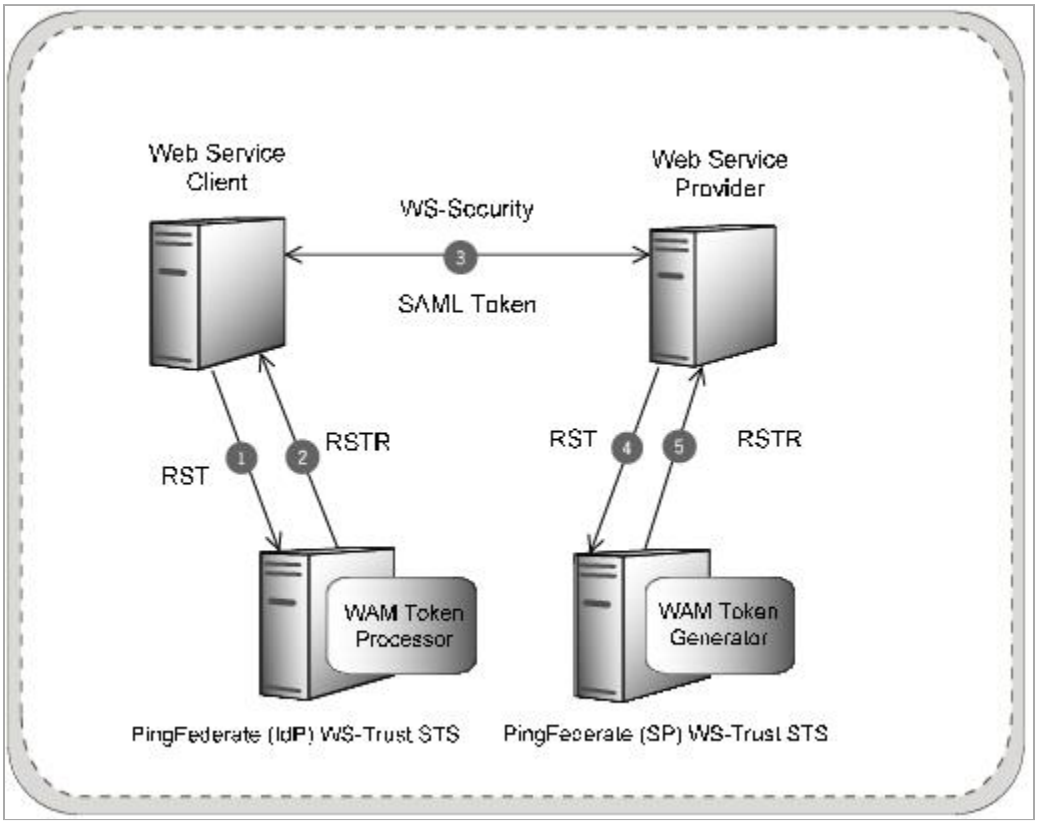
3. Unzip the token translator distribution file and copy the following files from the `/dist` directory to the `<PF_install>/pingfederate/server/default/deploy` directory.
    - `opentoken-adapter-2.5.1.jar`
    - `pf-wam-token-translator-2.0.jar`
4. If you are running PingFederate 6.0 as a Windows service, add the following line to the `Java Library Path` section of the `pingfederate/sbin/wrapper/PingFederateService.conf` file:

```
wrapper.java.library.path.append_system_path=true
```

5. Start the PingFederate server.

WS-Trust STS processing

The following illustration displays a basic Web Services scenario using the PingFederate WS Trust STS in the role of both IdP and SP:



Processing Steps

1. A WSC sends a Request Security Token (RST) message containing a WAM session token to the PingFederate STS IdP endpoint.

Copyright © 2025 Ping Identity Corporation

2. The PingFederate WAM Token Processor extracts, decrypts, parses, and validates the WAM session token. If the WAM session token is valid, PingFederate maps attributes from the WAM session token into a SAML token. PingFederate issues the SAML token based on the SP connection configuration and embeds the token in a Request Security Token Response (RSTR), which is returned to the WSC.
3. The WSC binds the issued SAML token into a Web Service Security (WSS) header and sends it via a SOAP request to the WSP.
4. The WSP sends an RST Issue request containing the SAML token to the PingFederate STS SP endpoint. PingFederate validates the SAML token and, if valid, maps attributes from the SAML token into a WAM session. PingFederate issues the WAM session token based on the WAM Token Generator configuration and embeds the token in an RSTR, which is returned to the WSP.
5. The WSP receives the WAM session token in the RSTR for local domain processing.

Configuring the IdP token processor

About this task

If you are using PingFederate as an IdP server, configure the Token Processor using the following steps:

> **Important**
>
> You must first create a third-party WAM Web Agent within your WAM tool. Several properties used to configure the agent are then used on the Instance Configuration screen. Refer to your WAM documentation for details on agent configuration.

Steps

1. Log on to the PingFederate administrative console and click Token Processors under Application Integration Settings in the IdP Configuration section of the Main Menu.

   If you don't see Token Processors on the Main Menu, enable WS-Trust under Server Settings on the Roles & Protocols screen by selecting WS-Trust for the IdP role.

   > **Note**
   >
   > To enable token exchange, you may be prompted to provide SAML 1.x and SAML 2.0 federation identifiers for the STS on the Federation Info screen. Refer to the Federation Info screen's **Help** page for more information.

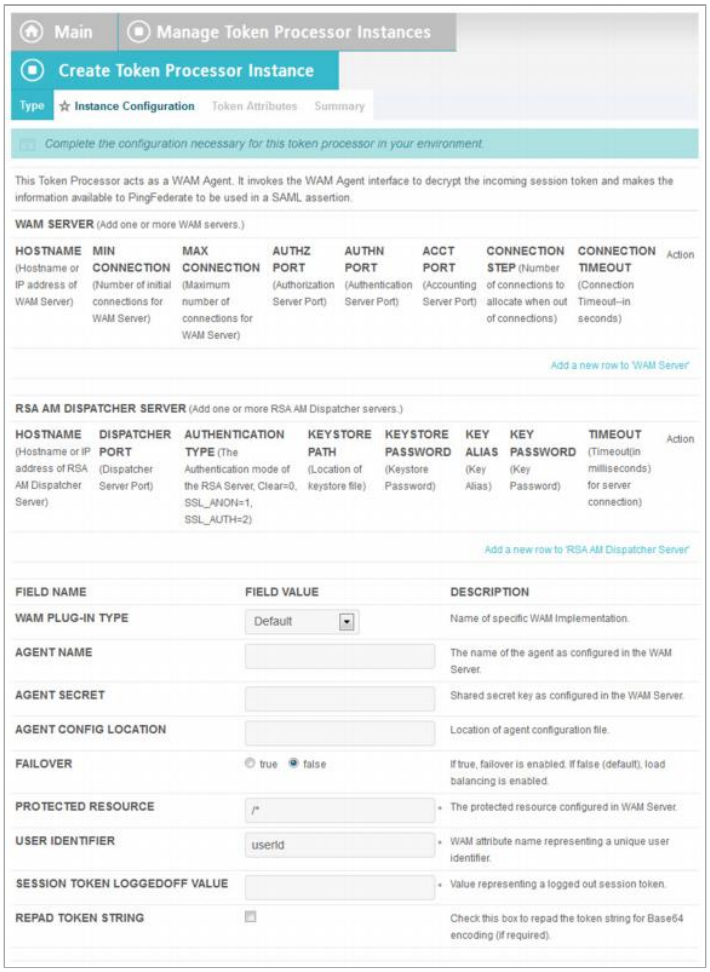
2. On the Manage Token Processor Instances screen, click Create New Instance.
3. On the Type screen, enter an Instance Name and Instance Id.

   The Name is any you choose for identifying this instance. The ID is used internally and may not contain spaces or non-alphanumeric characters.

4. Select WAM Token Processor 2.0 as the Type and click Next.

Copyright © 2025 Ping Identity Corporation

> **Note**
>
> If you are configuring the adapter for a custom plug-in (not bundled with this kit), then continue to step 5. If you are configuring the RSA Dispatcher server, then continue with step 6. If you are configuring OAM, continue at step 7.



5. **(Only for custom plug-ins for WAM servers other than OAM or RSA) On the Instance Configuration screen, click Add a new row to 'WAM Server' and provide the following information into the table:**

    1. **Enter the Hostname or the IP address where the WAM server is running.**
    2. **Specify the remaining WAM server values required for your configuration.**
    3. **Click Update in the Action column.**
    4. **Repeat this step as needed, for additional WAM plug-ins.**

    Skip the next step.

6. **(Only for the RSA bundled plug-in) On the Instance Configuration screen, click Add a new row to 'RSA AM Dispatcher Server' and provide the following information in the table:**

Copyright © 2025 Ping Identity Corporation

> **Note**
>
> You must specify at least one RSA AM Dispatcher Server

1. **Enter the Hostname or the IP address and the (optional) Dispatcher Port where the RSA AM Dispatcher server is running.**

   > **Note**
   >
   > You must specify the authentication method that is used by the dispatcher server. If you have specified multiple dispatcher servers, each server can have individual authentication methods.

2. **Specify the Authentication Type used by the RSA Dispatcher Server.**
   - **Clear – clear text, no encryption**
   - **Anon – anonymous SSL, SSL encryption only**
   - **Auth – mutually authenticated SSL, SSL encryption with certificate-based encryption**
3. **If the selected Authentication Type is Auth, you must specify the following RSA server values:**
   - **Keystore Path – String filename of the private Keystore file (PKCS12 only)**
   - **Keystore Password – password for the private Keystore**
   - **Key Alias – the alias to your private key in the Keystore**
   - **Key Password – private Key Password for Keystore**
4. **(Optional) Specify the Timeout value required for your configuration.**
5. **Click Update in the Action column.**
6. **Repeat this step as needed for additional RSA Servers.**

7. **Provide entries on the Instance Configuration screen, as described on the screen and in the following table.**

> **Note**
>
> selected WAM Plug-in Type may override optional/required fields. For example, if the selected WAM Plug-in Type is `OAM`, the Agent Config Location becomes a required field. Leaving this field blank generates an error message.

Field	Description
WAM Plug-in Type	**Class name for the specific WAM implementation.** **Note** WAM Plug-in Type determines optional/required fields.
Agent Name	**This value must match the value used when the third-party WAM Web Agent was configured.**

Copyright © 2025 Ping Identity Corporation

Field	Description
Agent Secret	This value must match the value used when the third-party WAM Web Agent was configured.
Agent Config Location	Required for OAM, this value must contain the full path to an XML network-configuration file generated by the access-management system.
Failover	The default is false, indicating load balancing is enabled and user-session states and configuration data are shared among multiple WAM servers. Select true to enable failover, indicating that when one server fails, the next server is used.
Protected Resource	(Required) All files in the root directory (/*) is the default. Specify a different path to the resources in the protected realm, if necessary.
User Identifier	(Required) Defines which attribute that is parsed from the WAM session token is the user identifier for use in the assertion.
Session Token LOGGEDOFF Value	(Required) Value representing a logged-out session token.
Repad Token String	Enable this to pad the incoming session token string for Base64 encoding (if required).

8. Click Next.
9. (Optional) On the Token Attributes screen, select any or all attributes whose value you want to mask in the PingFederate log file.

   For more information about this screen, see the PingFederate *Administrator's Manual*. More information is available on the Help page.
10. Click Next.
11. On the Summary screen, verify that the information is correct and click Done.
12. On the Manage Token Processor Instances screen, click Save.

Configuring the SP token generator

About this task

If you are using PingFederate as a Service Provider (SP), configure the Token Generator using the following steps:

> **Important**
>
> You must first create a third-party WAM Web Agent within your WAM tool. Several properties used to configure the agent are then used on the Instance Configuration screen. Refer to your WAM documentation for details on agent configuration.

Copyright © 2025 Ping Identity Corporation

Steps

1. **Log on to the PingFederate administrative console and click Token Generators under Application Integration Settings in the SP Configuration section of the Main Menu.**

   If you do not see Token Generators on the Main Menu, enable WS-Trust under Server Settings on the Roles & Protocols screen by selecting WS-Trust for the SP role.

   > **Note**
   >
   > To enable token exchange, you may be prompted to provide SAML 1.x and SAML 2.0 federation identifiers for the STS on the Federation Info screen. Refer to the Federation Info screen's **Help** page for more information.

2. **On the Manage Token Generator Instances screen, click Create New Instance.**

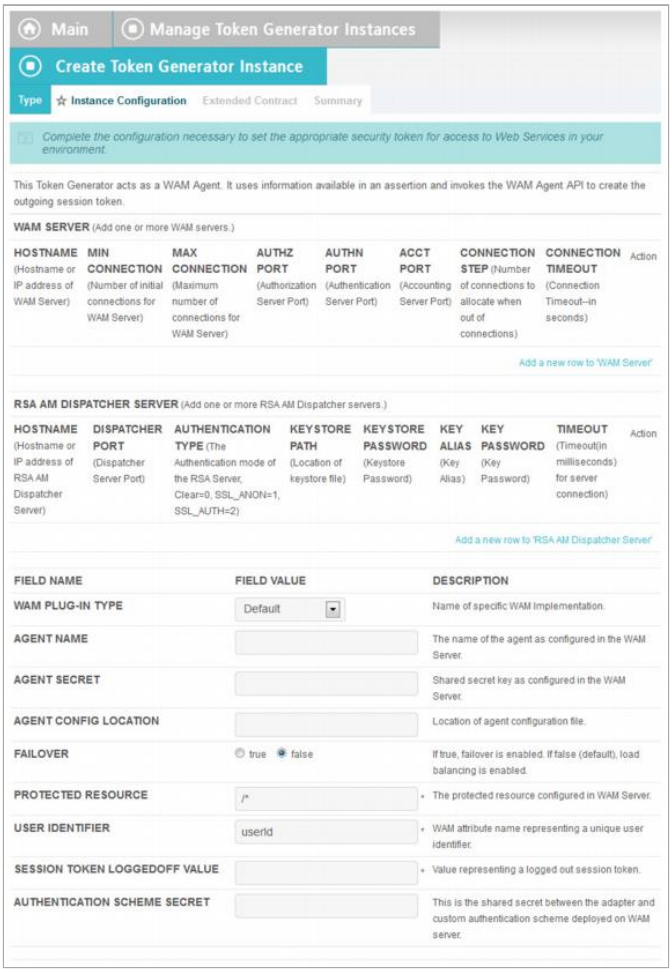
3. **On the Type screen, enter an Instance Name and Instance Id.**

   The Name is any you choose for identifying this instance. The ID is used internally and may not contain spaces or non-alphanumeric characters.

4. **Select WAM Token Generator 2.0 as the Type and click Next.**

   > **Note**
   >
   > If you are configuring the adapter for a custom plug-in (not bundled with this kit), then continue to step 5. If you are configuring the RSA Dispatcher server, then continue with step 6. If you are configuring OAM, continue at step 7.

Copyright © 2025 Ping Identity Corporation



5. (Only for custom plug-ins for WAM servers other than OAM or RSA) On the Instance Configuration screen, click Add a new row to 'WAM Server' and provide the following information into the table:
    1. Enter the Hostname or the IP address where the WAM server is running.
    2. Specify the remaining WAM server values that are required for your configuration.
    3. Click Update in the Action column.
    4. Repeat this step as needed, for additional WAM plug-ins.

       Skip the next step.
6. (Only for the RSA bundled plug-in) On the Instance Configuration screen, click Add a new row to 'RSA AM Dispatcher Server' and provide the following information in the table:

> **Note**
>
> You must specify at least one RSA AM Dispatcher Server

    1. Enter the Hostname or the IP address and the (optional) Dispatcher Port where the RSA AM Dispatcher server is running.

Copyright © 2025 Ping Identity Corporation

> **Note**
>
> You must specify the authentication method that is used by the dispatcher server. If you have specified multiple dispatcher servers, each server can have individual authentication methods.

2. Specify the Authentication Type used by the RSA Dispatcher Server.
    - Clear – clear text, no encryption
    - Anon – anonymous SSL, SSL encryption only
    - Auth – mutually authenticated SSL, SSL encryption with certificate-based encryption
3. If the selected Authentication Type is Auth, you must specify the following RSA server values:
    - Keystore Path – String filename of the private Keystore file (PKCS12 only)
    - Keystore Password – password for the private Keystore
    - Key Alias – the alias to your private key in the Keystore
    - Key Password – private Key Password for Keystore
4. (Optional) Specify the Timeout value required for your configuration.
5. Click Update in the Action column.
6. Repeat this step as needed for additional RSA Servers.

7. Provide entries on the Instance Configuration screen, as described on the screen and in the table below.

> **Note**
>
> The selected WAM Plug-in Type may override optional/required fields. For example, if the selected WAM Plug-in Type is `OAM`, the Agent Config Location becomes a required field. Leaving this field blank generates an error message.

Field	Description
WAM Plug-in Type	Class name for the specific WAM implementation. **Note** WAM Plug Type determines optional/required fields.
Agent Name	This value must match the value used when the third-party WAM Web Agent was configured.
Agent Secret	This value must match the value used when the third-party WAM Web Agent was configured.
Agent Config Location	Required for OAM, this value must contain the full path to an XML network-configuration file generated by the access-management system.

Copyright © 2025 Ping Identity Corporation

Field	Description
Failover	The default is false, indicating load balancing is enabled and user-session states and configuration data are shared among multiple WAM servers. Select true to enable failover, indicating that when one server fails, the next server is used.
Protected Resource	(Required) All files in the root directory (/*) is the default. Specify a different path to the resources in the protected realm, if necessary.
User Identifier	(Required) Defines which attribute that is parsed from the WAM session token is the user identifier for use in the assertion.
Session Token LOGGEDOFF Value	(Required) Value representing a logged out session token.
Authentication Scheme Secret	(Required, except for RSA) The shared secret between the adapter and the custom authentication scheme deployed on the WAM server.
Encode Token (Advanced Field)	The default is false. Check this box to url encode token string (if required).

8. Click Next.
9. (Optional) On the Extended Contract screen, add attributes you expect to retrieve in addition to the SAML subject (user ID). For more information, see Extending an SP adapter contract in the PingFederate documentation.
10. Click Next.
11. On the Summary screen, verify that the information is correct and click Done.
12. On the Manage Token Generator Instances screen, click Save.

Creating a custom authentication scheme for OAM

About this task

The Token Generator uses a custom authentication scheme when creating a WAM session and validates authentication requests coming from PingFederate. This section describes how to deploy the OAM-compatible Java-based PingFederate Custom Authentication Scheme.

Steps

1. Import the `<token_translator_install_dir>/dist/PingCustomAuthPlugin.jar` file into the OAM.

   The `PingCustomAuthPlugin.jar` file is a custom authentication scheme that supports OAM.
2. Configure your Access Server to use the custom authentication plug-in by creating or modifying a custom authentication scheme.

   Learn more in the Oracle Support documentation.

Copyright © 2025 Ping Identity Corporation

> **Note**
>
> The secret you specify when creating the custom authentication scheme must match the secret stored in the PingFederate Token Generator.

Using the STS client SDK

Ping Identity provides a Java STS-Client SDK for enabling Web Service applications (Client or Provider) to interact with the PingFederate STS. The SDK is available for download on the PingFederate server add-ons page.

The SDK provides functionality for sending a security token to the PingFederate STS for exchange with a returned SAML token, which can then be used to access Web Services across domains. The following code examples show how to send a token and request the exchange. Refer to the SDK documentation for modifications that apply to your site.

Token processor sample code

The code snippet below demonstrates using the PingFederate Java STS Client SDK to send a WAM session token to the PingFederate STS.

```
// Example method for obtaining the WAM Session token.
// You will need to implement this for your environment.
String wamSessionToken = getWAMSessionToken();

// Configure STS Client (IdP side / SP Connection)
STSClientConfiguration stsConfig = new STSClientConfiguration();
stsConfig.setAppliesTo("http://sp.domain.com");
stsConfig.setStsEndpoint("https://idp.domain.com:9031/idp/sts.wst");
stsConfig.setInTokenType(TokenType.BINARY);

// Instantiate the STSClient
STSClient stsClient = new STSClient(stsConfig);

// Send an RST Issue request to {pingfed} STS
Element samlToken = stsClient.issueToken(wamSessionToken);
```

Token generator sample code

The code snippet below demonstrates using the PingFederate Java STS Client SDK to retrieve a WAM session token through the PingFederate STS.

Copyright © 2025 Ping Identity Corporation

```
// Configure STS Client (SP side / IdP Connection)
STSClientConfiguration stsConfig = new STSClientConfiguration();
stsConfig.setStsEndpoint("https://sp.domain.com:9031/sp/sts.wst");
stsConfig.setOutTokenType(TokenType.BINARY);

// Instantiate the STSClient
STSClient stsClient = new STSClient(stsConfig);

// Send an RST Issue request to {pingfed} STS
Element wamsessionToken = stsClient.issueToken(samlToken);
```

Release notes

Changelog

Web Access Management Token Translator 2.0 – August 2014 (Current Release)

- Updated Token Translator with the WAM Agent Plugin 2.0
- Added "encode Token" field

Web Access Management Token Translator 1.2 – August 2013

- Updated Token Translator with the WAM Agent Plugin 1.2

Web Access Management Token Translator 1.1 – March 2013

- WAM Token Translator now includes WAM plug-in compatible with RSA Access Manager

Web Access Management Token Translator 1.0.1 – December 2012

- Updated to address security issues found since previous release
- Added support for OpenToken Adapter 2.5.1

Web Access Management Token Translator 1.0 – November 2012

- Initial Release
- Redesigned former product-specific kits to provide consistent functionality across multiple WAM products

Known issues and limitations

Known Limitations

- Due to a limitation with PingFederate 8.1 and earlier versions, when configuring two SP connections with the same provisioner, the second connection built may be pre-populated with the channel from the first connection. To avoid conflicts, delete this pre-populated channel and create a unique channel for each connection.

Copyright © 2025 Ping Identity Corporation

Download manifest

The distribution `.zip` archive for the WAM Token Translator contains the following:

- `ReadMeFirst.pdf` – contains links to this online documentation
- /legal – contains this document:
  - Legal.pdf – copyright and license information
- `/dist` – contains libraries needed to run the Token Translator:
  - `pf-wam-token-translator-2.0.jar` – the WAM Token Translator JAR file
  - `opentoken-adapter-2.5.1.jar` – OpenToken Adapter JAR file
- `/dist/oam` – contains Oracle Access Manager libraries needed to run the adapter:
  - `pf-oam-plugin.jar` – Pre-built OAM-compatible WAM plug-in JAR file
  - `PingCustomAuthPlugin.jar` – a Java-based PingFederate Custom Authentication Scheme
- `/dist/rsa` – contains RSA libraries needed to run the adapter:
  - `pf-rsa-plugin.jar` – Pre-built RSA-compatible WAM plug-in JAR file
  - `axm-runtime-api-6.1.4` - RSA API library
  - `jsafeFIPS-6.1.jar` – RSA API library
  - `jsafeJCEFIPS-6.1.jar` – RSA API library
- `/sdk` – contains build scripts, documents, libraries, and sample code to build a WAM plug-in:
  - `README.txt` – contains documentation on how to build a WAM plug-in
  - `/docs` – contains documentation on how to build a WAM plug-in
  - `/lib` – contains libraries and supporting files needed to build WAM plug-in
  - `/samples` – contains sample code used to build a WAM plug-in

Copyright © 2025 Ping Identity Corporation

Webex Provisioner

The Webex Provisioner allows PingFederate to integrate with Webex for provisioning and single sign-on (SSO).

The distribution file includes a quick connection template to help you get started.

Provisioning connector

Allows PingFederate to manage users in Webex based on changes in an external user datastore.

You can enable the create, update, and delete capabilities independently.

SSO support

Allows you to create a connection between PingFederate and Webex to support browser-based, IdP- and SP-initiated SSO.

Intended audience

This document is intended for PingFederate administrators.

If you need help during the setup process, see the following resources:

- The following sections of the PingFederate documentation:
  - Identity provider SSO configuration
  - Datastores
  - SP connection management
  - Configuring outbound provisioning
  - Configuring outbound provisioning settings

System requirements

- PingFederate 8.x or later.
- To allow PingFederate to make outbound connections to Webex, you might need to whitelist the following endpoints in your firewall:
  - https://<*subdomain*>.webex.com

User management

The Webex Connector synchronizes users from your datastore to Webex. The following describes the behavior of each provisioning capability.

Copyright © 2025 Ping Identity Corporation

> **Tip**
>
> You can configure the following capabilities and specify which users to provision when you get to the Configuring provisioning and single sign-on part of the setup process.

Synchronizing existing users

PingFederate synchronizes users based on the `email` attribute in Webex. If a user already exists in your datastore and Webex, mapping this attribute correctly links the two records together.

For example:

- In Webex, Janet's `email` is `jsmith@example.com`.
- In your datastore, Janet's `mail` is `jsmith@example.com`.
- On the Attribute Mapping tab of your provisioning connection configuration, map the `email` attribute to `mail`.
- When the provisioning connector runs, the datastore user is provisioned with a `email` of `jsmith@example.com`. That matches Janet's existing `email` in Webex, so her information in the datastore is synchronized to her Webex account.

User provisioning

PingFederate provisions users when any of the following happens:

- A user is added to the datastore group or filter that is targeted by the provisioning connector.
- A user with `disabled` status is added to the datastore group or filter that is targeted by the provisioning connector, and theProvision disabled users provisioning option is enabled. This feature is not available in all provisioning connectors.

> **Tip**
>
> You can define which users PingFederate targets for provisioning on the **Source Location** tab of your provisioning connection configuration.

User updates

PingFederate updates users when a user attribute changes in your datastore.

> **Tip**
>
> You can define which attributes PingFederate monitors for changes on the **Attribute Mapping** tab of your provisioning connection configuration.

User deprovisioning

PingFederate deprovisions users when any of the following happens:

- A user is deleted from the user store.
- A user is disabled in the user store.

Copyright © 2025 Ping Identity Corporation

- A user is removed from the datastore group or filter that is targeted by the provisioning connector.

Setup

Upgrade an existing connector

About this task

If you're upgrading from a previous version of the Webex Connector, copy your current configuration to a new connection.

Steps

1. Note the attribute mapping configuration from your existing Webex provisioning connection:
    1. On the PingFederate administrator console, on the Identity Provider tab, in the SP Connections section, click the name of your existing provisioning connection.
    2. On the Outbound Provisioning tab, click Configure Provisioning.
    3. On the Manage Channels tab, click your existing channel configuration.
    4. On the Attribute Mapping tab, note your existing attribute mappings.
2. Disable and delete your existing provisioning connection:
    1. On the Identity Provider tab, in the SP Connections area, click Manage All.
    2. For your existing provisioning connection, disable the connection.
    3. Click Select Action, and then click Delete.
    4. Click Save.
3. Complete the steps in Deploy the provisioning connector.
4. Complete the steps in Configuring provisioning and single sign-on to configure a new SP connection.
5. Go to Outbound Provisioning > Manage Channels > Attribute Mapping and configure the attribute mappings based on the information you noted in step 1.

Deploy the provisioning connector

Steps

1. Download the integration `.zip` archive from the Ping Identity Integration Directory.
2. Stop PingFederate.
3. Delete any existing Webex Connector files in the PingFederate directory:
    1. In the `<pf_install>/pingfederate/server/default/deploy` directory, delete the following files:
        - `pf-webex-quickconnection-<version>.jar`

Copyright © 2025 Ping Identity Corporation

- `webex-api-4.8.1.jar`

4. Deploy the provisioning connector:
    1. Extract the Webex Connector distribution file, and then go to the `dist` folder.
    2. Copy the `pf-webex-quickconnection-<version>.jar` file to the `<pf_install>` `/pingfederate/server/default/deploy` directory.
5. Enable the PingFederate provisioning engine:
    1. In the `<pf_install>/pingfederate/bin` direcotry, open the `run.properties` file for editing.
    2. Change `pf.provisioner.mode` to `STANDALONE`. Save the file.

    > **Note**
    >
    > To configure the `FAILOVER` mode instead, see Deploying provisioning failover in the PingFederate *Server Clustering Guide*.

6. Start PingFederate.
7. If you operate PingFederate in a cluster, repeat steps 1-5 for each instance of PingFederate.

Configuring provisioning and single sign-on

About this task

> **Note**
>
> You can follow these steps to create a new service provider (SP) connection, or you can modify an existing connection.

Steps

1. In the PingFederate administrator console, configure the datastore that PingFederate will use as the source of user data.

    For instructions, see Datastores in the PingFederate documentation.

    > **Tip**
    >
    > When targeting users and groups for provisioning, exclude the user account that you will use to administer users in your connection to Webex. This prevents the PingFederate provisioning engine from interfering with the account that provisions users and groups.

2. Enable provisioning:
    1. Go to System > Protocol Settings > Roles & Protocols and select Enable Identity Provider IdP Role and Support the Following.
    2. Select Outbound Provisioning. Click Save.
3. Create an SP connection with the Webex quick connection template:
    1. Follow the steps in Downloading your Webex SAML metadata file.

Copyright © 2025 Ping Identity Corporation

2. On the PingFederate Identity Provider tab, in the SP Connections section, click Create new.
3. On the Connection Template tab, select Use a template for this connection.
4. In the Connection Template list, select Webex Connector.
5. Click Choose File, select the Webex metadata file that you downloaded, and then click Open. Click Next.

4. On the Connection Type tab, select Browser SSO Profiles and Outbound Provisioning.
5. In the Type list, select Webex Connector. Click Next.
6. On the Connection Options tab, click Next.
7. On the General Info tab, the basic connection information is populated by the metadata XML file. Click Next.
8. On the Browser SSO tab, configure single sign-on (SSO) settings. Click Next.

   Follow the steps in Configure IdP Browser SSO in the PingFederate documentation, with the following specifics:

   1. Go to Browser SSO > SAML Profiles and select IdP-initiated SSO and SP-initiated SSO.
   2. Optional: Go to Browser SSO > Assertion Creation > Attribute Contract and extend the contract. Webex supports the following formats:
      - Unspecified
      - Email address
      - X509 subject name
      - Entity identifier
      - Persistent identifier
   3. Optional: Add the special `SAML_AUTHN_CTX` attribute.

   This indicates to the SP the type of credentials used to authenticate to the identity provider (IdP) application.

   1. Go to Browser SSO > Assertion Creation > Authentication Source Mapping to configure your authentication source mappings:
      - If you added the special `SAML_AUTHN_CTX` attribute, on the Attribute Contract Fulfillment tab, map the attribute to a text value, such as `urn:oasis:names:tc:SAML:2.0:ac:classes:unspecified`.
   2. Go to Browser SSO > Protocol Settings > Allowable SAML Bindings and select Post and Redirect. Clear Artifact and SOAP.
   3. If you want to enable SP-initiated SSO, go to Browser SSO > Protocol Settings > Signature Policy and select Require authn requests to be signed when received via the POST or Redirect bindings.

9. On the Credentials tab, configure your credentials. Click Next.

   See Configure credentials in the PingFederate documentation.

   1. Click Configure Credentials.

Copyright © 2025 Ping Identity Corporation

2. Go to Credentials > Digital Signature Settings and in the Signing Certificate list, select a certificate to use to sign SAML assertions.

10. On the Outbound Provisioning tab, configure the provisioning target and channel. Click Next.

    See Configuring outbound provisioning in the PingFederate documentation.

    1. Click Configure Provisioning.
    2. On the Target tab, complete the fields as follows.

Field	Description
Webex ID	Your Webex administrator username.
Password	Your Webex administrator password.
Site name	The subdomain of the Site Brand Name(s) listed on your Webex administration Site Information tab, such as *example* in example.webex.com.
Site ID	Optional: The Site ID listed on your Webex administration Site Information tab.
Partner ID	Optional: The Partner ID listed on your Webex administration Site Information tab.

> **Note**
>
> PingFederate verifies the credentials when you activate the channel and SP connection.

    3. Customize the provisioning connector actions. Click Next.

       See Provisioning options.

    4. On the Manage Channels tab, create a channel. Click Done.

    See Managing channels in the PingFederate documentation.

11. On the Activation and Summary tab, above the Summary section, click the toggle to turn on the connection. Click Save.

Downloading your Webex SAML metadata file

About this task

You can use a Webex SAML metadata file to automatically configure some of your PingFederate connection settings.

To download SAML 2.0 Metadata for Webex:

Steps

1. Sign on to your Webex administrator site.

Copyright © 2025 Ping Identity Corporation

2. In the upper-right corner, click your account, and then click Webex Administration.
3. Go to Configuration > Common Site Settings > SSO Configuration and in the Federation Protocol list, select SAML 2.0.
4. Click Export, and then save the `saml-metadata.xml` file.
5. Leave the SSO Configuration tab empty for now.

Provisioning options

Option	Description
User Create	• Selected (default) — PingFederate creates users in Webex. • Cleared — PingFederate does not create users in Webex.
User Update	• Selected (default) – PingFederate updates existing users in Webex. • Cleared – PingFederate does not update existing users in Webex.
User Disable	• Selected (default) – PingFederate disables users in Webex. • Cleared – PingFederate does not disable users in Webex.

Note
If any of the above options are cleared, PingFederate logs a warning in the user workflow section of `provisioner.log` when the related action fails.

Supported attributes reference

The following table lists the attributes that can be mapped for user provisioning.

Important
Make sure you send all user attributes that are required by your Webex account configuration.

Attribute	Description
`Email`	The user's email address. Must be a valid email address.
`Webex ID`	A reference to the Webex user account.
`First Name`	The user's first name.
`Last Name`	The user's last name.

Copyright © 2025 Ping Identity Corporation

Attribute	Description
`Password`	The user's password. A user password is validated against the password security options configured in the Webex Site Administration tool. If any of the security rules are violated, an exception occurs.
`Address 1`	The first line of the user's street address.
`Address 2`	The second line of the user's street address.
`City`	The user's city.
`Company`	The user's company name.
`Country`	The user's country. Must be a valid Country name. See Elements in WebEx XML Schema Definitions for the User Service in the Webex Meeting XML API Reference Guide.
`Fax`	The user's fax number.
`Language`	The user's preferred language. Must be a valid language. See Elements in WebEx XML Schema Definitions for the User Service in the Webex Meeting XML API Reference Guide.
`Meeting Type`	The user's meeting type IDs.
`Mobile Phone`	The user's mobile phone number.
`Pager`	The user's pager number.
`Phone`	The user's phone number.
`Pin`	The user's PIN number. Secondary level of authentication for PCN and when host is using the phone and inviting additional attendees. Single number values and simple sequences, like 1111 or 1234, are not allowed.
`State`	The user's state.
`Timezone`	The user's time zone. Must be a valid timezone. See Elements in WebEx XML Schema Definitions for the User Service in the Webex Meeting XML API Reference Guide.
`Title`	The user's title.
`Zip Code`	The user's zip code (postal code).

Copyright © 2025 Ping Identity Corporation

Configuring single sign-on in Webex

Use the following minimum required settings to configure single sign-on (SSO) for Webex.

About this task

For more information on how to setup SSO for Webex, see Configure Single Sign-On for Cisco Webex Site on the Webex site.

> **Note**
>
> You can customize your Webex configuration beyond the basics described here. However, the SP connection created by the connector template does not support the Webex Account Creation/Update options. These SAML assertion-based provisioning options conflict with the connector's outbound provisioning methodology.

Steps

1. **Download your PingFederate SAML metadata file.**

   For more information, see Metadata export in the PingFederate documentation.

2. **Sign on to your Webex administrator site.**
3. **In the upper-right corner, click your account, and then click Webex Administration.**
4. **Go to Configuration > Common Site Settings > SSO Configuration and in the Federation Protocol list, select SAML 2.0.**
5. **Click Import SAML Metadata, select the PingFederate metadata file that you downloaded, and then click Open.**

   > **Note**
   >
   > If you receive a message asking whether you want to overwrite an existing certificate, click **Yes**.

6. **Configure the fields based on the following table.**

   > **Important**
   >
   > At a minimum, you must change the Webex default AuthnContextClassRef value, as specified in the table. This setting is not contained in the SAML metadata.

Copyright © 2025 Ping Identity Corporation

Field	Description
SSO Profile	If you want to enable both identity provider (IdP)- and service provider (SP)-initiated SSO, select SP Initiated. **Note** Use this if you only want pre-authenticated users to be able to access Webex directly through another site, such as a company portal. If you want to enable only IdP-initiated SSO, select IdP Initiated. **Note** Use this if you want the SP-initiated behavior and also want users to be able to authenticate by clicking a link in Webex.
Webex SAML Issuer (SP ID)	If you are configuring a second (or greater) Webex Site for SSO, change this ID to match the Connection ID defined for the corresponding PingFederate SP connection. The default value is http://www.webex.com.
Issuer for SAML (IdP ID)	The Entity ID for SAML 2.0 at your site, as shown on the PingFederate administrative console Federation Info tab.
Customer SSO Service Login URL	Your site's PingFederate SAML 2.0 endpoint in the format: `https://<pf_host>:<pf_port>/idp/SSO.saml2`
AuthnContextClassRef	◦ If you customized the SAML_AUTHN_CTX value in Configuring provisioning and single sign-on, enter your custom value in this field. ◦ Otherwise, enter the following: `urn:oasis:names:tc:SAML:2.0:ac:classes:Password`
Default Webex Target page URL	The fully-qualified URL for your Webex site. For example, https://subdomain.webex.com.

7. Optional: If you selected SP Initiated SSO, select AuthnRequest Signed.
8. Optional: In the Destination URL field, paste the value from the Customer SSO Service Login URL field.
9. Click Update.

Release notes

Copyright © 2025 Ping Identity Corporation

Changelog

Webex Provisioner 2.3 – June 2023 (current release)

Users are no longer required to provide the WebEx Site ID or Partner ID to configure outbound provisioning.

Webex Provisioner 2.1 – July 2019

- Fixed an issue that prevented users with special characters from being provisioned to Webex.
- Improved error handling and reporting for cases where users in the target application do not have an ID.
- Improved error logging security.
- Added provisioning options to control the "create", "update", and "disable" functions individually.

Webex Provisioner 2.0.2 – March 2017

- Fixed synchronization on update of users, that were previously created with "User Create Enabled" set to false in configurable options.

Webex Provisioner 2.0.1 – February 2017

- Fixed handling of timezones not listed in Webex's 'Timezone encoding list' .

Webex Provisioner 2.0 – May 2016

- Added support for additional user attributes.
- Added configuration options for CRUD capabilities.
- Added support for Webex API 10.0 SP3.
- Improved exception handling and reporting.

Webex Provisioner 1.0.1 – February 2014

- Updated Webex API to use XMLBeans 2.6.0 to be compatible with PingFederate 7.x.

Webex Provisioner 1.0 – May 2010

- Initial Release.
- Added Support for User Provisioning.

Known issues and limitations

The following are known issues and limitations of the Webex Connector integration.

Known issues

There are no known issues.

Copyright © 2025 Ping Identity Corporation

Known limitations

- Updates made to users that are disabled in WebEx will not be visible in WebEx until they are reenabled.
- Because of a limitation with PingFederate 8.1 and earlier versions, when configuring two service provider (SP) connections with the same provisioner, the second connection built might be pre-populated with the channel from the first connection. To avoid conflicts, delete this pre-populated channel and create a unique channel for each connection.
- User attributes cannot be cleared when set.
- When an LDAP user is deleted in a targeted group distinguished name (DN), the provisioning connector does not propagate the deletion until a new user is added to the group. This limitation is compounded when the User Create provisioning option is disabled.

  For solutions, see SaaS provisioner does not remove the user in the Knowledge Base.
- The Webex API does not allow a user to be created in a disabled state. Webex activates all users after creation.
- The connector cannot re-enable user meeting types that have been disabled through the Webex administration console. If the connector tries to update the user's meeting types in this scenario, it can cause all meeting types for that user to be disabled.

Download manifest

The distribution archive contains the following files.

- `ReadMeFirst.pdf` – contains a link to the Ping Identity Support Home.
- `/legal`:
  - `Legal.pdf` – copyright and license information.
- `/dist` – contains libraries needed for the provisioning connector:
  - `pf-webex-quickconnection-<version>.jar` – Webex Connector

Copyright © 2025 Ping Identity Corporation

WebSphere Integration Guide

This guide describes how to configure PingFederate and IBM WebSphere Application Server to enable single sign-on (SSO).

Because this integration uses a standard SAML configuration, you don't need to download additional software.

Features

- Browser-based SSO initiated by the identity provider (IdP) or service provider (SP).

> **Note**
>
> WebSphere does not support single logout.

Intended audience

This document is intended for PingFederate administrators.

Before you start, you should be familiar with the following:

- Managing IdP adapters in the PingFederate documentation
- The following sections of the WebSphere documentation:
  - SAML web single sign-on
  - The Trust association model section of Trust associations

System requirements

- Administrator access to WebSphere 8.5 or later

Setup

Enabling single sign-on in PingFederate

Before you can configure single sign-on in WebSphere, you need to set a SAML entity ID in PingFederate.

Enabling single sign-on in PingFederate 10.1 or later

Steps

1. On the PingFederate administrative console, go to System > Protocol Settings > Federation Info.
2. In the SAML 2.0 Entity ID field, enter a name for PingFederate to use to identify itself to SAML partners.
3. Click Save.

Copyright © 2025 Ping Identity Corporation

Enabling single sign-on in PingFederate 10.0 or earlier

Steps

1. On the PingFederate administrative console, go to System > Protocol Settings > Roles & Protocols.
2. Select the Enable Identity Provider IdP Role and Support the Following check box.
3. Select the SAML 2.0 check box. Click Next.
4. Go to the Federation Info tab.
5. In the SAML 2.0 Entity ID field, enter a name for PingFederate to use to identify itself to SAML partners.
6. Click Save.

Exporting SAML metadata from PingFederate

Export a metadata file that describes your PingFederate identity provider configuration.

About this task

For general information about these steps, see Metadata export in the PingFederate documentation.

Steps

1. In the PingFederate administrative console, go to the Metadata Export window..

   Choose from:

   - For PingFederate 10.1 or later: go to System > Protocol Metadata > Metadata Export.
   - For PingFederate 10.0 or earlier: go to System > Metadata Export.
2. If you see the Metadata Role tab, select I am the identity provider (IdP). Click Next.
3. On the Metadata Mode tab, select Select information to include in metadata manually. Click Next.
4. On the Protocol tab, click Next.
5. On the Attribute Contract tab, click Next.
6. On the Signing Key tab, select a signing certificate. Click Next.
7. Optional: On the Metadata Signing tab, select a certificate to sign the metadata XML file. Click Next.
8. On the XML Encryption Certificate tab, select the certificate that you want to use to encrypt the XML content. Click Next.
9. On the Export & Summary tab, click Export.
10. Save `metadata.xml`.
11. Click Done.

Copyright © 2025 Ping Identity Corporation

Configuring single sign-on in WebSphere

Configure a SAML trust association interceptor (TAI) on your WebSphere Application Server (WAS).

Steps

1. Complete the steps in Enabling your system to use the SAML web single sign-on (SSO) feature in the WebSphere documentation. Add custom properties to your TAI based on the table below.

Trust association interceptor custom properties for PingFederate

Property	Description
`sso_<id>.sp.acsUrl`	The assertion consumer service URL for the WebSphere SAML ACS servlet, such as `https://was_host:was_port/samlsps/applicationacs`.
`sso_<id>.sp.EntityID`	Enter an entity ID of your choosing for your WAS. This is included in the SAML metadata file that you export in the next step.
`sso_<id>.idp_<id>.EntityID`	The SAML 2.0 Entity ID that you entered in Enabling single sign-on in PingFederate.
`sso_<id>.idp_<id>.SingleSignOnUrl`	The PingFederate SSO URL, such as `https://pf_host:pf_port/idp/SSO.saml2`.
`sso_<id>.idp_<id>.certAlias`	Enter a name of your choosing to identity the PingFederate signing certificate. You will use this when you import the certificate to WebSphere.
`sso_<id>.sp.login.error.page`	Your WAS authentication error page. This property is also used in the optional Configuring service provider-initiated SSO steps.
`sso_<id>.sp.targetUrl`	The URL of the target application. To test your configuration, you can enter `https://was_host:was_port/snoop`.

> **Note**
>
> For detailed specifications for these properties, see SAML web single sign-on (SS) trust association interceptor (TAI) custom properties in the WebSphere documentation.

2. Complete the steps in Exporting SAML web service provider metadata using the wsadmin command-line utility in the WebSphere documentation. Save the metadata file to your PingFederate server. You will use it in Creating a single sign-on connection.

Copyright © 2025 Ping Identity Corporation

3. Complete the steps in Importing SAML identity provider (IdP) partner metadata using the wsadmin command-line utility in the WebSphere documentation. Select the metadata file that you saved in Exporting SAML metadata from PingFederate. Use the alias that you chose for the `sso_<id>.idp_<id>.certAlias` property.
4. Complete the steps in Configuring single sign-on (SSO) partners in the WebSphere documentation.

Creating a single sign-on connection

To allow PingFederate to handle single sign-on (SSO) authentication for WebSphere, create a service provider (SP) connection.

Steps

1. Sign on to the PingFederate administrator console.
2. On the Identity Provider tab, in the SP Connections area, create a new connection.
    1. Click Create new.
    2. If you see the Connection Template tab, select Do not use a template for this connection. Click Next.
3. On the Connection Type tab, select only Browser SSO Profiles. Click Next.
4. On the Connection Options tab, select only Browser SSO. Click Next.
5. On the Import Metadata tab, select File. Click Choose File, select the `sp-metadata.xml` file that you saved in Configuring single sign-on in WebSphere, and then click Open. Click Next.
6. On the General Info tab, the basic connection information is populated by the metadata XML file. Click Next.
7. On the Browser SSO tab, configure browser SSO.

    For a complete guide, see Configure IdP Browser SSO in the PingFederate documentation.

    1. On the Browser SSO > SAML Profiles tab, select only IdP-Initiated SSO.
8. On the Credentials tab, configure the connection credentials. Click Next.

    For a complete guide, see Configuring credentials in the PingFederate documentation.

    - On the Credentials > Digital Signature Settings tab, select the Include the certificate in the signature <KeyInfo> element check box.
9. On the Activation and Summary tab, above the Summary section, turn on the connection.
10. Note the SSO Application Endpoint URL. You can use this to enable SP-initiated single sign-on in Configuring single sign-on in WebSphere. Click Save.

Configuring service provider-initiated SSO

You can imitate a service provider (SP)-initiated single sign-on (SSO) experience by configuring your WebSphere Application Server (WAS) to redirect unauthenticated users to the PingFederate SSO page.

About this task

Copyright © 2025 Ping Identity Corporation

> **Note**
>
> This part of the configuration is optional. If you only want identity provider-initiated SSO, skip these steps.

By default, when an unauthenticated user tries to access a protected resource, the WebSphere Application Server (WAS) captures the requested resource URL in a browser cookie, then directs the user to a default error page.

You can override this behavior to redirect the user to the PingFederate SSO page instead. After the user authenticates, the WebSphere Application Server redirects them to the original resource URL stored in the cookie.

This creates a sign on experience that is similar to SP-initated SSO. For a true SAML-based solution, see Enabling SAML SP-Initiated web single sign-on (SSO) in the WebSphere documentation.

Steps

1. **In your WebSphere trusted association interceptor (TAI) configuration, add filter properties to allow the WAS to identify requests that should be authenticated by PingFederate. Follow the guide in the SAML TAI filter property section of SAML web single sign-on (SSO) trust association interceptor (TAI) custom properties in the WebSphere documentation.**
2. **In your TAI configuration, change the `sso_<id>.sp.login.error.page` property to the SSO Application Endpoint URL that you noted in Creating a single sign-on connection.**

Copyright © 2025 Ping Identity Corporation

Workspace ONE UEM Integration Kit

The Workspace ONE UEM Integration Kit allows PingFederate to integrate with Workspace ONE (formerly known as AirWatch).

Features

- Make policy decisions based on the security posture of employees' devices
- Mitigate the risks that come with allowing employees' mobile and desktop devices to access corporate resources

Components

Workspace ONE IdP Adapter

- Allows PingFederate to communicate with the the Workspace ONE API

Intended audience

This document is intended for PingFederate administrators.

If you need help during the setup process, see the following resources:

- The following sections of the PingFederate documentation:
  - Identity provider SSO configuration
  - Managing IdP adapters
  - Authentication policies
  - Policy contracts
  - SP connection management
- If you want to use Workspace ONE to manage the PingID app on devices, see Configure Workspace ONE UEM for PingID in the PingID documentation

System requirements

- PingFederate 9.0 or later
- The X.509 Certificate Integration Kit deployed in PingFederate
- An on-premise or cloud deployment of the Workspace ONE platform
- The Workspace ONE certificate authority (CA) root certificate installed in the PingFederate trusted store

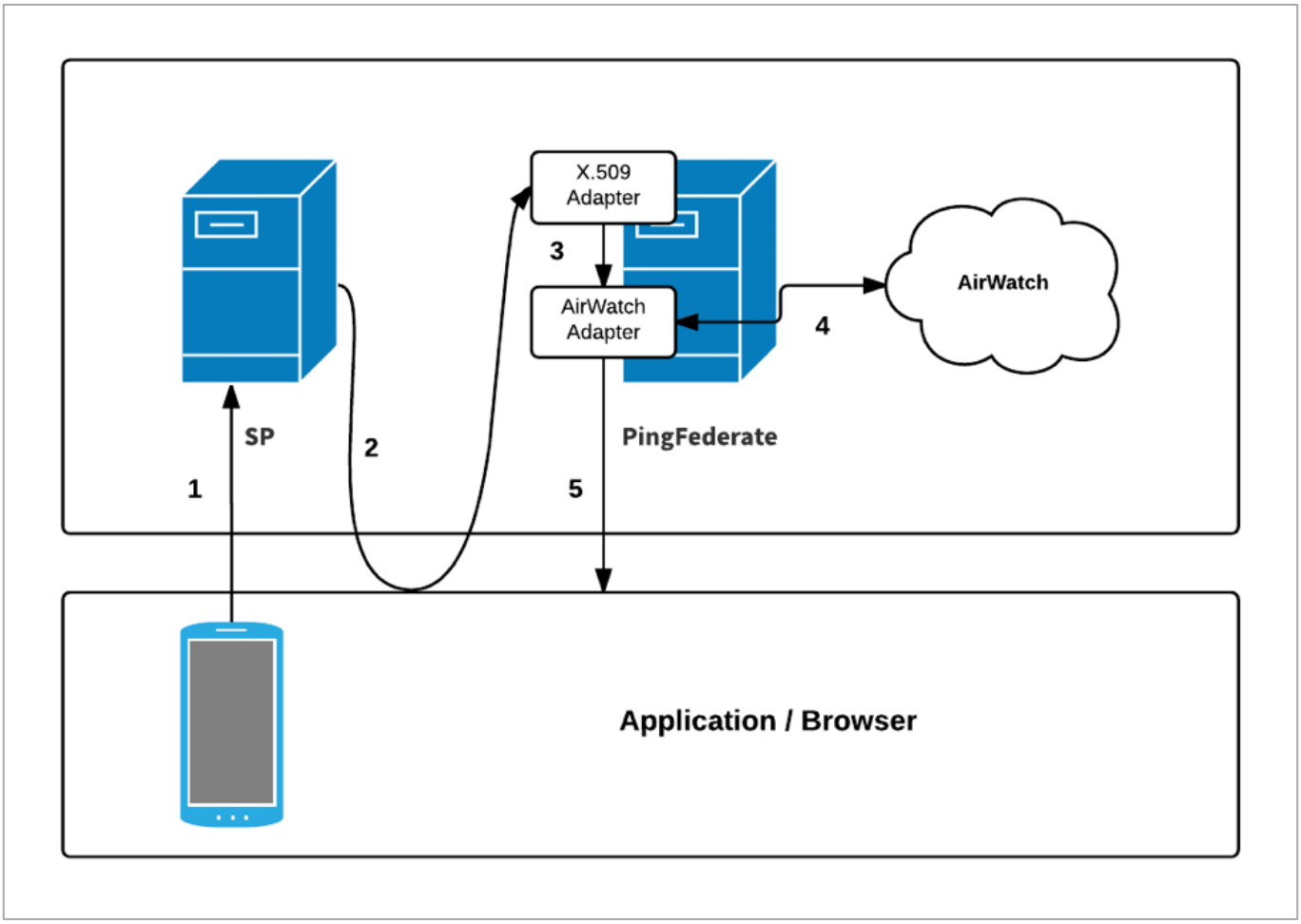
  For help, Manage trusted certificate authorities in the PingFederate documentation
- A device enrolled with Workspace ONE that you can use to test your configuration

Copyright © 2025 Ping Identity Corporation

- The following mobile single sign-on requirements:
  - A web controller, to allow the application to read the certificate from the device:
    - Android: Chrome 45 (or later) & Chrome Custom Tabs
    - Apple: iOS9 (or later) & Safari View Controller
  - AppAuth, for shared sessions between supported mobile applications

Overview of the SSO flow

The following figure shows a basic SSO scenario in which a PingFederate server authenticates users to an SP application using the Workspace ONE IdP Adapter.



Description

1. A user with an Workspace ONE enrolled device requests access to an SP resource. The request is redirected to PingFederate to perform X.509 authentication.
2. The browser requests the user's X.509 certificate. The PingFederate X.509 Certificate Adapter validates the certificate against a list of issuers. If no issuers are specified in the adapter setup, it uses the server's list of trusted CAs instead.

Copyright © 2025 Ping Identity Corporation

3. PingFederate validates the certificate, then passes the device ID from the certificate to the Workspace ONE IdP Adapter.
4. PingFederate contacts the the Workspace ONE API, provides the device ID to get information about the device's security posture.
5. The result of the authentication is returned, and if successful, the user is redirected to the requested resource.

Supported attributes reference

The Workspace ONE IdP Adapter retrieves the following attributes.

Attribute	Description
AcLineStatus	
AssetNumber	
ComplianceStatus	True if the status is compliant with MDM policies; false otherwise.
CompromisedStatus	True if the device is compromised; false otherwise.
DataEncryptionYN	True if data protection is enabled; false otherwise.
DeviceFriendlyName	The concatenated name used to identify the device/user combination.
EasId	
EnrollmentStatus	True if MDM value is Enrolled; false otherwise.
Imei	IMEI number of the device.
IsRemoteManagementEnabled	
IsSupervised	
LastComplianceCheckOn	The refresh date and timestamp of the last status reported.
LastCompromisedCheckOn	The refresh date and timestamp of the last status reported.
LastEnrolledOn	
LastSeen	Date and time the device last made successful contact with the MDM.
LocationGroupName	MDM location group configuration value.

Copyright © 2025 Ping Identity Corporation

Attribute	Description
MacAddress	The Wi-Fi MAC address.
Model	Model is automatically reported by the device during registration.
OperatingSystem	OS version.
Ownership	Values: C, E, or S (Corporate, Employee, or Shared).
PhoneNumber	Phone number entered during registration.
Platform	Platform specified during registration.
SerialNumber	Serial number.
UserEmailAddress	E-mail address of device user.
UserName	Name of device user.
VirtualMemory	

Setup

Deploying the integration files

To get started with the integration, deploy the Workspace ONE UEM Integration Kit files to your PingFederate directory.

About this task

> **Note**
> If you operate PingFederate in a cluster, the following steps refer to the console node.

Steps

1. Download the Workspace ONE UEM Integration Kit `.zip` archive from the Ping Identity Integration Directory.
2. Stop PingFederate.
3. If you're upgrading an existing deployment, delete the `pf-airwatch-adapter-<version>.jar` file from your `<pf_install>/pingfederate/server/default/deploy` directory.
4. Extract the `.zip` archive and merge the contents of the `dist` directory with your `<pf_install>/pingfederate` directory.
5. Start PingFederate.

Copyright © 2025 Ping Identity Corporation

6. If you operate PingFederate in a cluster, repeat steps 2-5 for each engine node.

Configuring a Workspace ONE IdP Adapter instance

Configure the Workspace ONE IdP Adapter to determine how PingFederate communicates with the Workspace ONE API.

Steps

1. In the PingFederate administrative console, create a new IdP adapter instance:
    1. For PingFederate 10.1 or later: go to Authentication > Integration > IdP Adapters. Click Create New Instance.
    2. For PingFederate 10.0 or earlier: go to Identity Provider > Adapters. Click Create New Instance.
2. On the Type tab, set the basic adapter instance attributes:
    1. In the Instance Name field, enter a name for the adapter instance.
    2. In the Instance ID field, enter a unique identifier for the adapter instance.From the Type list, select Workspace ONE IdP Adapter. Click Next.
3. On the IdP Adapter tab, configure the adapter instance. Click Next.
4. On the IdP Adapter tab, configure the adapter instance with information from your Workspace ONE configuration. Click Next.

> **Note**
>
> For Workspace ONE version 9.1.0.301, the API key is in **Groups & Settings > All Settings > System > Advanced > API > REST API**. We recommend that you create a new API key.

5. On the Adapter Attributes tab, set pseudonym and masking options as shown in Set pseudonym and masking options in the PingFederate documentation. Click Next.
6. On the Adapter Contract Mapping tab, configure the contract fulfillment details for the adapter as shown in Define the IdP adapter contract in the PingFederate documentation.
    1. Optional: Restrict device access based on criteria.
    2. Click Next.
7. On the Summary tab, check and save your configuration:
    1. For PingFederate 10.1 or later: click Save.
    2. For PingFederate 10.0 or earlier: click Done. On the Manage IdP Adapter Instances tab, click Save.

Restrict device access based on criteria

You can specify criteria to restrict which devices can access resources.

Steps

1. On the Issuance Criteria tab, in the Source list, select Adapter.

Copyright © 2025 Ping Identity Corporation

2. Optional: Restrict access based on device ownership:
    1. In the Attribute Name list, select Ownership.
    2. In the Condition list, select not equal to.
    3. In the Value field, type one of the following letters:
        - C (for corporate-owned devices)
        - S (for corporate shared devices)
        - E (for employee-owned devices)
    4. Click Add, and then click Done.
3. Optional: Restrict access based on device operating system:
    1. In the Attribute Name list, select OperatingSystem.
    2. In the Condition list, select not equal to.
    3. In the Value field, type one of the following letters:
        - Apple
        - Android
    4. Click Add, and then click Done.
4. Optional: Restrict access based on device MDM policy compliance:
    1. In the Attribute Name list, select ComplianceStatus.
    2. In the Condition list, select equal to (case insensitive).
    3. In the Value field, type Compliant.
    4. Click Add, and then click Done.
5. Optional: Restrict access for compromised devices:
    1. In the Attribute Name list, select CompromisedStatus.
    2. In the Condition list, select not equal to.
    3. In the Value field, type True.
    4. Click Add, and then click Done.
6. Click Done, and then click Next.

Create an X.509 Certificate Adapter instance

Pass the Workspace ONE device ID through an X.509 certificate to PingFederate.

Copyright © 2025 Ping Identity Corporation

Steps

1. In Workspace ONE, configure the X.509 certificate template to include the Workspace ONE device ID as an attribute in the SAN extension.

   For help, see the Workspace ONE UEM documentation.

2. In PingFederate, set up the X.509 Certificate Integration Kit, and create an X.509 Certificate Adapter instance.

   For complete documentation, see X.509 Certificate Integration Kit.

   1. In the adapter instance configuration, on the IdP Adapter tab, click Show Advanced Fields, then select Include subject alternative name (SAN).
   2. On the Extended Contract screen, add the SAN attribute from the X.509 certificate that contains the Workspace ONE device ID.

Create a policy contract

Create a policy contract that you will use in the authentication policy.

About this task

For help with these steps, see Policy contracts in the PingFederate documentation.

Steps

1. On the Identity Provider > Policy Contracts tab, click Create New Contract.
2. On the Contract Info tab, in the Contract Name field, type a unique name. Click Next.
3. On the Contract Attributes tab, click Next.
4. On the Summary tab, click Done.
5. On the Authentication Policy Contract tab, click Save.

Adding device postures to your authentication policy

Create an authentication policy to pass the Workspace ONE device ID from the X.509 Certificate Adapter instance to the Workspace ONE IdP Adapter instance.

About this task

These steps are designed to help you add to an existing authentication policy. For general information about configuring authentication policies, see PingFederate authentication API in the PingFederate documentation.

Steps

1. In the PingFederate administrative console, go to the Policies tab.
   1. For PingFederate 10.1 or later: go to Authentication > Policies > Policies.
   2. For PingFederate 10.0 or earlier: go to Identity Provider > Authentication Policies > Policies.

Copyright © 2025 Ping Identity Corporation

2. Select the IdP Authentication Policies check box.
3. Open an existing authentication policy, or click Add Policy.

   For help, see Defining authentication policies in the PingFederate documentation.
4. In the Policy area, from the Select list, select the X.509 Certificate Adapter instance that you created in Create an X.509 Certificate Adapter instance.
5. In the X.509 Fail section, configure the failure result.
6. In the X.509 Success section, select the Workspace ONE IdP Adapter instance that you created in Configuring a Workspace ONE IdP Adapter instance. Click Options.
7. On the Incoming User ID dialog, in the Source list, select the X.509 Certificate Adapter instance.
8. In the Attribute list, select the attribute that you added to the extended contract of the X.509 Certificate Adapter instance. Click Done.
9. In the Workspace ONE IdP Adapter Fail section, configure the failure result.
10. In the Workspace ONE IdP Adapter Success section, select the policy contract that you created in Create a policy contract.
11. Click Contract Mapping.
12. On the Contract Fulfillment tab, in the Source list, select the X.509 Certificate Adapter instance.
13. In the Value list, select the attribute that contains the Workspace ONE device ID.
14. Click Done. In the Policies window, click Save.

Configure the SP connection to use the policy contract

About this task

You can find more information in SP connection management in the PingFederate documentation.

Steps

1. In the PingFederate administrator console, create a new SP connection:
   1. For PingFederate 10.1 or later: go to Applications > Integration > SP Connections. Click Create Connection.
   2. For PingFederate 10.0 or earlier: go to Identity Provider > SP Connections. Click Create Connection.
2. On the Connection Template tab, select Do not use a template for this connection.
3. On the Connection Type tab, select Browser SSO Profiles. Click Next.
4. On the Connection Options tab, click Next.
5. On the Import Metadata tab, select None. Click Next.
6. On the General Info tab, enter the entity ID and choose a name for the connection. Click Next.
7. On the SAML Profiles tab, select IdP-Initiated SSO and any other profiles you need.

Copyright © 2025 Ping Identity Corporation

8. On the Assertion Lifetime tab, click Next.
9. On the Assertion Creation tab, configure the assertion creation. Click Next.

   Learn more in Configuring SSO token creation in the PingFederate documentation.

   1. On the Authentication Source Mapping tab, click Map New Authentication Policy.
   2. On the Authentication Policy Contract tab, in the Authentication Policy Contract list, select the policy contract that you created in Create a policy contract. Click Next.
   3. On the Attribute Contract Fulfillment tab, in the Source list, select Authentication Policy Contract.
   4. In the Value list, select the attribute that contains the Workspace ONE device ID. Click Next.
10. On the Assertion Creation tab, click Next.
11. On the Protocol Settings tab, configure the protocol settings. Click Next.

    Learn more in Configuring protocol settings in the PingFederate documentation.
12. On the Protocol Settings tab, click Next.
13. On the Summary tab, click Done.
14. On the Browser SSO tab, click Next.
15. On the Credentials tab, configure the connection credentials as shown in Configuring credentials in the PingFederate documentation. Click Next.
16. On the Activation and Summary tab, above the Summary section, click the toggle to turn on the connection. Click Save.

Test the integration

Test your configuration by adding a profile for your device and assigning it to a group.

Steps

1. Set up PingFederate to run the service provider (SP) application according to instructions in the Workspace ONE administrator documentation.
2. In the Workspace ONE administrative console, on the Devices > Profiles & Resources > Profiles page, click Add > Add Profile.
3. On the device type list, select the type that the profile applies to, and then complete the profile form.
4. In the Assigned Groups field, assign the profile to the group that includes your test device.
5. On the Credentials tab of the profile, click Configure. Select your credential source, verify the data is correct, and then click Save & Publish.
6. Install the profile on the device.

   1. On the Profiles & Resources page, on the profile that you created, click the Not Installed button.

Copyright © 2025 Ping Identity Corporation

2. Click Install Profile.
3. Turn on the test device.
4. On the device, open the agent app to force a "sync" between the device and the Workspace ONE Server.

7. On the test device, on the Devices > List View > Select your device > Certificates page, check that the certificate appears on the list of certificates installed on the device.
8. On the test device, in a browser, go to a protected resource.
9. When asked for X.509 authentication, select the certificate installed by the profile. If the certificate isn't available, manually add it to the browser certificate store and try again.

   The browser will be redirected to PingFederate for X.509 validation and the device should be redirected to the protected resource.

Release notes

Changelog

- The following is the change history for the Workspace ONE UEM Integration Kit.

Workspace ONE UEM Integration Kit 1.0.4 – November 2021

- Updated the product name from AirWatch Integration Kit to Workspace ONE UEM Integration Kit.
- Changed to a standardized `.zip` file structure to make automated deployments easier.

AirWatch Integration Kit 1.0.3 – September 2020

- Improved security by masking the API key in the administrator console.
- Fixed an issue that prevented the adapter from correctly returning a `FAIL` result to the PingFederate authentication policy when it received an error from the AirWatch API.

AirWatch Integration Kit 1.0.2 – March 2019

- Resolved an issue that prevented connections from being released for some AirWatch API responses.

AirWatch Integration Kit 1.0.1 – April 2018

- Updated adapter to use AirWatch's latest API version.

AirWatch Integration Kit 1.0 – April 2017

- Initial release.

Copyright © 2025 Ping Identity Corporation

Known issues and limitations

- The following are known issues or limitations for the Workspace ONE UEM Integration Kit.

Known issues

There are no known issues.

Known limitations

- Due to a limitation in how mobile devices handle X.509 certificates, clients using iOS must use Safari, and clients using Android must use Chrome.

Download manifest

- The following files are included in the Workspace ONE UEM Integration Kit `.zip` archive.
- `Legal.pdf` – copyright and license information
- `dist/pingfederate/server/default` – contains the integration files
    - `deploy` – contains the Java libraries
        - `pf-airwatch-adapter-<version>.jar` – The Workspace ONE IdP Adapter

Copyright © 2025 Ping Identity Corporation

Overview of X.509 integrations

PingFederate X.509 Certificate Integration Kit

The PingFederate X.509 Certificate Integration Kit provides an identity provider (IdP) adapter that allows a PingFederate IdP server to perform client X.509 certificate authentication for single sign-on (SSO) to Service Provider (SP) applications.

PingFederate X.509 Token Translator

The PingFederate X.509 Token Translator provides an identity provider (IdP) token processor for use with the PingFederate WS-Trust Security Token Service (STS).

X.509 Certificate Integration Kit

The PingFederate X.509 Certificate Integration Kit allows PingFederate to authenticate clients that use X.509 certificates to sign on to applications.

Components

X.509 Certificate IdP Adapter

- When PingFederate is acting as an identity provider (IdP), this adapter validates X.509 certificates against the certificate authorities (CA) in the PingFederate certificate store.

Intended audience

This document is intended for PingFederate administrators.

Before starting, read the Known issues and limitations.

System requirements

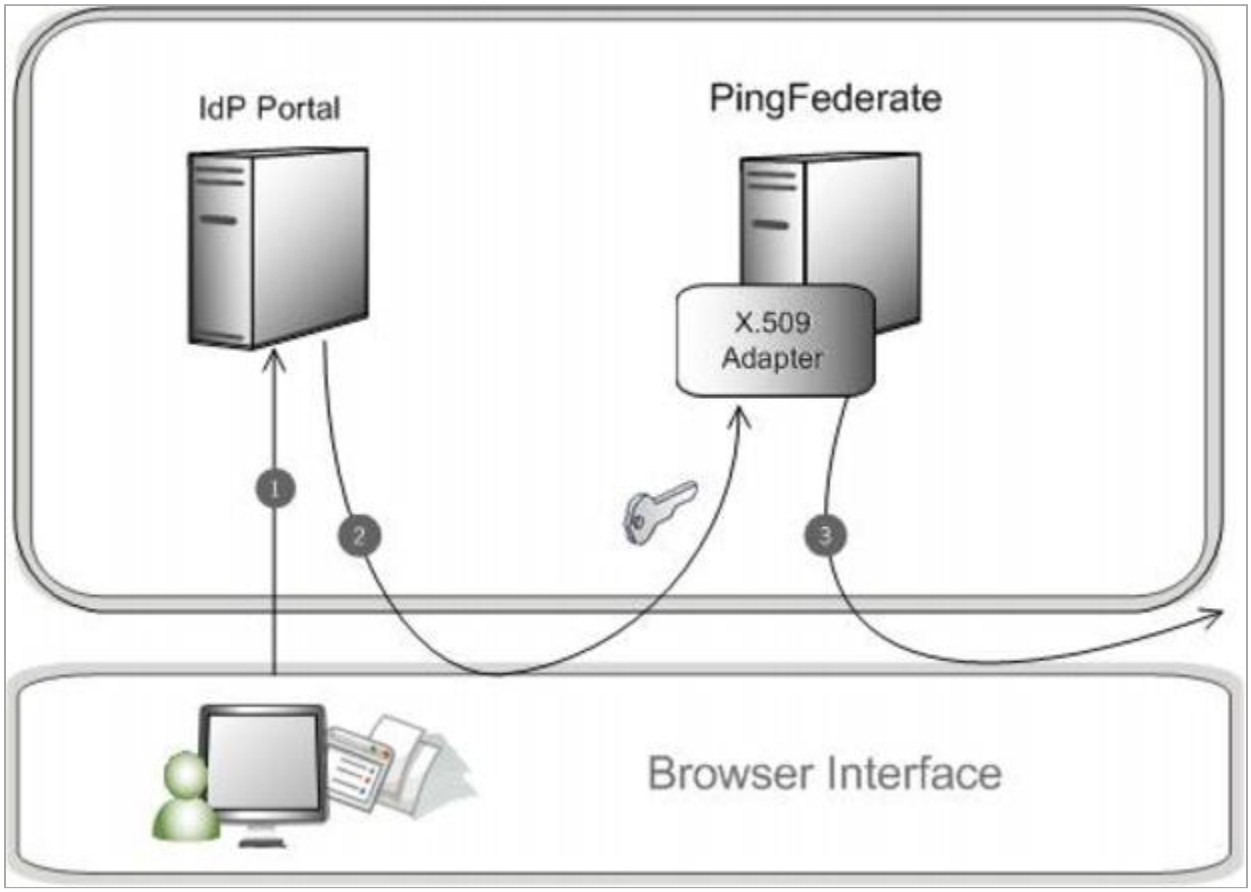
- PingFederate 10.3 or later.
- Client must use a compatible web browser. Learn more in Known issues and limitations for details.

Overview of the SSO flow

With the X.509 Certificate Integration Kit, PingFederate validates the X.509 certificate presented by a user or device during the sign-on flow.

The following figure shows a basic single sign-on (SSO) scenario where a PingFederate server authenticates users to a service provider (SP) application using the X.509 Certificate Adapter:

Copyright © 2025 Ping Identity Corporation



Description

1. A user initiates the sign-on process by requesting access to a protected resource.
2. The identity provider (IdP) portal redirects the request to PingFederate. The browser requests the user's client certificate. The PingFederate X.509 Certificate Adapter validates the certificate against a list of issuers. Issuers are specified in the adapter instance configuration. If no issuers are specified in the adapter instance configuration, the adapter checks against the server's list of trusted certificate authorities.
3. If the certificate is valid, PingFederate allows the sign on to proceed. Otherwise, the user is taken to an error page template.

Setup

Deploying the integration files

To get started with the integration, deploy the X.509 Certificate Integration Kit files to your PingFederate directory.

Steps

1. Download the X.509 Certificate Integration Kit `.zip` archive from the Ping Identity Integration Directory.
2. Stop PingFederate.

Copyright © 2025 Ping Identity Corporation

3. If you're upgrading an existing deployment, delete `x509-certificate-adapter-<version>.jar` from the `<pf_install>/pingfederate/server/default/deploy` directory.
4. Extract the `.zip` archive and merge the contents of the `dist` directory with your `<pf_install>/pingfederate` directory.
5. In `<pf_install>/pingfederate/bin/run.properties`, change the value of `pf.secondary.https.port` from `-1` to a valid port number.
6. Start PingFederate.
7. If you operate PingFederate in a cluster, repeat steps 2 - 6 for each engine node.

Configuring an adapter instance

Configure the X.509 Certificate IdP Adapter to determine how PingFederate handles X.509 certificates.

Steps

1. Sign on to the PingFederate administrative console.
2. On the Identity Provider > Manage IdP Adapter Instances page, click Create New Instance.
3. On the Type page, set the basic adapter instance attributes.
    1. In the Instance Name field, enter a name for the adapter instance.
    2. In the Instance ID field, enter a unique identifier for the adapter instance.
    3. In the Type list, select X.509 Certificate IdP Adapter. Click Next.
4. (Optional) On the IdP Adapter page, in the Constrain Acceptable Root Issuers section, specify the certificate authority (CA) that you want to use to validate end-user X.509 certificates.

    > **Note**
    >
    > Client certificates are always validated against all trusted CAs in PingFederate and the Java Virtual Machine (JVM). This section only restricts which issuers are used to validate end-user certificates.

    1. Click Add a new row to Constrain Acceptable Root Issuers.
    2. In the Issuer DN field, enter the subject distinguished name (DN) of an issuer listed on the Trusted CAs page in PingFederate.

        Learn more about Manage trusted certificate authorities in the PingFederate documentation.
    3. In the Action column, click Update.
    4. To add more acceptable issuers, repeat steps a - c.
5. On the IdP Adapter screen, configure the adapter instance by referring to X.509 Certificate IdP Adapter settings reference. Click Next.

Copyright © 2025 Ping Identity Corporation

6. On the Extended Contract screen, add any attributes that you want to include in the extended contract. Enter attributes in uppercase. Only attributes specified in RFC 2253 are allowed: `CN`, `L`, `ST`, `O`, `OU`, `C`, `STREET`, `DC`, and `UID`.

   > **Note**
   >
   > You can include subject DN components in this list.
   > If you selected **Parse Client Cert Subject and Issuer DNs** on the **IdP Adapter** page, you can also include the subject DN `email` component and issuer DN components.
   > For issuer DN components, prefix the attribute with `issuer_`, such as `issuer_CN`.

7. Complete the adapter configuration.
8. On the Summary page, verify that the configuration is correct. Click Done.
9. On the Manage IdP Adapter Instances page, click Save.
10. If you configured the Client Auth Hostname field, in `<pf_install>/pingfederate/server/default/data/config-store/session-cookie-config.xml`, add your domain with a preceding period to the `<c:item name="cookie-domain"></c:item>`, such as `<c:item name="cookie-domain">.example.com</c:item>`.

X.509 Certificate IdP Adapter settings reference

Field descriptions for the X.509 Certificate IdP Adapter configuration screen.

Standard fields

Field	Description
Client Auth Port	The port that PingFederate uses to validate client certificates. Enter the port number that you chose in Deploying the integration files. This field is blank by default.
Client Auth Hostname	The PingFederate hostname that is configured to use client-certificate authentication. This field is blank by default.
Parse Client Cert Subject and Issuer DNS	When enabled, the subject and issuer distinguished names (DN) in the client certificate are treated as separate attributes. This allows you to do the following: • Add subject or issuer DN attributes, such as CN or UID, to the adapter's extended contract. • Use the subject DN `email` attribute in the adapter's core contract. • Use Object-Graph Navigation Language (OGNL) expressions to extract other information from the X.509 certificate, as shown in Sample OGNL expressions. This checkbox is selected by default.

Copyright © 2025 Ping Identity Corporation

Field	Description
Match Issuer DN In Client Cert Subject And Issuer DNs	Determines how PingFederate validates the issuer distinguished name (DN) for the client certificate. When selected, the issuer DN is matched against the entries that are defined in the Constrain Acceptable Root Issuers section. When cleared, the issuer DN is matched against the default top-level certificate in the chain that is presented by the client. This checkbox is cleared by default.

Advanced fields

Field	Description
Return Success On SLO	When enabled, a success message is sent in response when the adapter receives a single logout (SLO) request. SLO is not supported by this adapter and the user session is not terminated. This feature only prevents other sites from experiencing an SLO failure. Learn more in Known issues and limitations. This checkbox is selected by default.
Authentication Context	The value used to populate the Authentication Context field in the SAML token that PingFederate sends after validating the X.509 certificate. ***Default*** Sets the value to `TLSClient`. ***Policy OID*** Sets the value to the identifier for the policy. ***Custom*** Sets the value based on the value you enter in the Custom Authentication Context field.
Custom Authentication Context	The value used to populate the Authentication Context field in the SAML token. Applies when Authentication Context is set to Custom. This field is blank by default.
Include Subject Alternative Name (SAN)	When enabled, the adapter includes the following decoded SAN attributes from the X.509 certificate and makes them available in the attribute contract: • `userPrincipalName` • `RFC822Name` • `fascn_sen` • `fascn_wo_sen` • `fascn_hex` This checkbox is cleared by default.

Copyright © 2025 Ping Identity Corporation

Field	Description
Skip Redirect	When enabled, the adapter does not redirect back from the Client Auth Port and Client Auth Hostname. This checkbox is cleared by default. **Note** Version 1.3.2 of the adapter introduced the ability to redirect back from the **Client Auth Port** and **Client Auth Hostname** as default behavior.

Sample OGNL expressions

The X.509 Certificate Integration Kit allows you to use Object Graph Navigation Library (OGNL) expressions to map attributes from X.509 certificates.

Client certificates are available as `java.security.cert.X509Certificate` objects. This allows you to map certificate attributes by using methods that are defined in the Java Platform, Standard Edition API Specification. You can find a full list of available methods in the J2SE online javadoc.

To get the first client certificate in the chain as an `X509Certificate` object, you must use the `ClientCertificateChain` keyword within the OGNL expression. For example:

```
#this.get("ClientCertificateChain").getObjectValue()
```

To get the entire client certificate chain as an iterable collection of `X509Certificate` objects, use the `getAllObjectValues()` method in the OGNL expression. For example:

```
#this.get("ClientCertificateChain").getAllObjectValues()
```

The following table lists example OGNL expressions you can use to map certificate attributes during the setup or deployment of the X.509 Certificate Adapter. Learn more about using OGNL for attribute mapping in Attribute mapping expressions in the PingFederate documentation.

OGNL Expression	Description
`#x509Cert = #this.get("ClientCertificateChain").getObjectValue() , #hexEncoded = new String (@org.apache.commons.codec.binary.Hex@encodeHex (#x509Cert.getSignature()))`	Returns a hex-encoded signature from the X.509 client certificate.

Copyright © 2025 Ping Identity Corporation

OGNL Expression	Description
`#x509Cert = #this.get("ClientCertificateChain").getObjectValue(), #hexEncoded = new String (@org.apache.commons.codec.binary.Hex@encodeHex (#x509Cert.getExtensionValue("2.16.840.1.113730.1.13")))`	Extracts the comment certificate extension (Object Identifier (OID) 2.16.840.1.113730.1.13) from the X.509 client certificate, which is then hex encoded. Other certificate extensions can be extracted by using the correct OID for the extension.

Proxy client authentication

To selectively perform client certification authentication on behalf of PingFederate, configure a front-end proxy or load balancer.

By configuring proxy client authentication, TLS is terminated at the proxy and the headers are passed back to PingFederate for validation by the X.509 Certificate IdP Adapter. This approach also allows you to have all traffic arrive on TCP port 443.

Learn more about configuring proxy client authentication for your product:

- Configure incoming proxy settings in the PingFederate documentation
- Creating header identity mappings and Defining engine listeners in the PingAccess documentation

Release notes

Changelog

The following is the change history for the X.509 Certificate Integration Kit. Updated April 19, 2024.

X.509 1.3.2 – April 2024

- Fixed an issue that caused the adapter to sometimes fail to redirect users who authenticated successfully back to the primary HTTPS port.

X.509 1.3.1 – September 2021

- Verified that the X.509 Certificate IdP Adapter is compliant with US Federal Information Processing Standards (FIPS). Learn more about usage in Integrating with Bouncy Castle FIPS provider in the PingFederate documentation.

X.509 1.3 – January 2020

- Extended the core contract to include the serial number of the client certificate.
- Added the option to use the client certificate for acceptable issuer validation.

Copyright © 2025 Ping Identity Corporation

X.509 1.2.1 – March 2017

- Updated third-party library.
- Adapter works with PingFederate authentication policies.

X.509 1.2 – November 2016

- Added ability to customize authentication context in SAML assertions.
- Added support for Subject Alternative Name (SAN) attributes.
- Added ability to redirect to a host for client-certificate authentication.

X.509 1.1 – February 2012

- Added support for retrieving additional certificate elements and mapping them to attributes.
- Added ability to access the certificate chain as J2SE `X509Certificate` objects through OGNL expressions.

X.509 1.0 – September 2008

- Initial release.

Known issues and limitations

The following are known issues or limitations for the X.509 Certificate Integration Kit.

Known issues

- If PingFederate is sitting behind a proxy and the X.509 certificate is sent encoded by the proxy, PingFederate can't decode it, resulting in a failure. To prevent this, ensure the proxy sends the certificate in RAW format as a header.

Known limitations

- The browser, browser version, and platform can affect the adapter's ability to obtain the X.509 certificate. If you experience issues using this adapter with a browser, contact Ping Identity support.
    - Users may be prompted to select the certificate even when only one certificate matches the configured Issuer CAs. Some browsers provide a setting that determines whether the user is prompted or the certificate is selected automatically.
    - The adapter has been tested with the following desktop browsers:
        - Firefox (tested with 89)
        - Chrome (tested with 91.0.4472.101)
        - Edge (tested with 91.0.864.54)
        - Safari (tested with 12.1.1 [14607.2.6.1.1])
        - Internet Explorer 11

Copyright © 2025 Ping Identity Corporation

    - Clients using iOS must use Safari. A limitation in iOS prevents Chrome and Firefox from working with this integration kit.

- Single logout (SLO) isn't supported because it isn't possible to force the browser to end the SSL session. The adapter can't force an authenticated user to select a new certificate or prompt the user to authenticate to a smart card again.
- Only attribute-type keywords specified in RFC2253 will be correctly parsed out of the subject distinguished name (DN): `CN`, `L`, `ST`, `O`, `OU`, `C`, `STREET`, `DC`, `UID`. The rest will be parsed as object identifiers (OIDs) and the corresponding name-value pairs are not human readable.
- Attribute-type keywords defined in the adapter contract will not work if they are mixed case, such as `Cn` or `sT`. Only all upper-case, such as `CN` or `ST`, or all lower-case, such as `cn` or `st`, will work.
- The adapter does not support the `isPassive` or `forceAuthn` portions of a Security Assertion Markup Language (SAML) authentication policy.

Download manifest

The following files are included in the X.509 Certificate Integration Kit `.zip` archive:

- `Legal.pdf`: Copyright and license information.
- `dist/pingfederate/server/default/deploy`: A directory that contains the Java libraries needed to run the adapter:
    - `X509-certificate-adapter-<version>.jar`: The X.509 Certificate IdP Adapter JAR file.

X.509 Token Translator

The PingFederate X.509 Token Translator provides an identity provider (IdP) token processor for use with the PingFederate WS-Trust Security Token Service (STS).

The token processor allows the STS to accept and validate an X.509 token from a Web Service Client (WSC) and then map user attributes into a SAML token for the WSC to send to a Web Service Provider (WSP).

> **Note**
>
> Ping Identity provides the Java Client SDK to allow web service applications (client or provider) to interact with the PingFederate STS. The SDK is available for download from the PingFederate server add-ons page.

The X.509 Token Processor uses the PingFederate security infrastructure for certificate validation and management. PingFederate validates the trust of all certificates. A certificate is trusted if the root certificate of the issuing Certificate Authority (CA) is imported into the PingFederate trusted certificate store or the CA is trusted by the Java Runtime Environment (JRE) in use.

Intended audience

This document is intended for PingFederate administrators.

Copyright © 2025 Ping Identity Corporation

You can get help during the setup process by consulting the following resources:

- The following sections of the PingFederate documentation:
  - WS-Trust STS configuration

System requirements

- PingFederate 6.0 or later

WS-Trust STS processing

The following steps describe a basic WS-Trust Security Token Service (STS) scenario where PingFederate validates an X.509 token and issues a SAML token:

1. A WSC sends a Request Security Token (RST) message containing an X.509 token to the PingFederate STS IdP endpoint.
2. The PingFederate X.509 Token Processor validates the X.509 token and, if valid, maps attributes from the X.509 token into a SAML token. PingFederate issues the SAML token based upon the SP connection configuration and embeds the token in a Request Security Token Response (RSTR), which is returned to the WSC.
3. The WSC binds the issued SAML token into a Web Service Security (WSSE) header and sends this through a SOAP request to the Web Service Provider (WSP).

Setup

Deploying the integration files

To get started with the integration, deploy the X.509 Token Translator files to your PingFederate directory.

About this task

> **Note**
>
> If you operate PingFederate in a cluster, the following steps refer to the console node.

Steps

1. Download the X.509 Token Translator `.zip` archive from the Ping Identity Integration Directory.
2. Stop PingFederate.
3. If you're upgrading an existing deployment, delete `pf-x509-token-translator-<version>.jar` from `<pf_install>/pingfederate/server/default/deploy`.
4. Extract the `.zip` archive and merge the contents of the `dist` directory with your `<pf_install>/pingfederate` directory.
5. Start PingFederate.
6. If you operate PingFederate in a cluster, repeat steps 2 - 5 for each engine node.

Copyright © 2025 Ping Identity Corporation

Configuring an X.509 Token Translator instance

Configure the X.509 Token Processor to determine how PingFederate validates and parses X.509 certificates.

Steps

1. Go to Authentication > Token Exchange > Token Processors.
2. On the Token Processor tab, click Create New Instance.
3. On the Type tab, set the basic token processor attributes:
    1. In the Instance Name field, enter a name for the token processor.
    2. In the Instance ID field, enter a unique identifier for the token processor.
    3. In the Type list, select X.509 Token Processor. Click Next.
4. Click Add a new row to 'Valid Certificate Issuer DNs' under Action, enter an applicable CA Issuer DN, and click Update.

> **Note**
>
> When this option is configured, only the CAs appearing under Valid DNs can be used to verify digital signatures on incoming tokens. Otherwise, all trusted CAs in the PingFederate or JRE trusted store can be used.

5. If you don't want the adapter to parse the certificate Subject DN to make its elements available for the extended contract, clear the Parse Subject DN check box. Click Next.
6. On the Extended Contract tab, add any Subject DN components and SAN settings:
    1. (Optional) Add any Subject DN components that you want to send individually in the SAML assertion in addition to the full DN.

    > **Note**
    >
    > The DN components are available separately only if the **Parse Subject DN** checkbox is selected, which is the default on the **Instance Configuration** tab. The attributes you enter must be in uppercase. Only attributes specified in RFC 2253 are allowed, such as `CN`, `L`, `ST`, `O`, `OU`, `C`, `STREET`, `DC`, and `UID`.

    2. (Optional) Select the Include Subject Alternative Name (SAN) checkbox to include the decoded SAN attributes from the certificate.

    When selected, this includes the following attributes from the X.509 certificate and makes them available in the attribute contract:

    - `userPrincipalName`
    - `RFC822Name`
    - `fascn_sen`
    - `fascn_wo_sen`
    - `deviceId`

Copyright © 2025 Ping Identity Corporation

3. Click Next.

7. On the Token Attributes tab, select any or all attributes whose values should be masked in PingFederate log files. Click Next.

   You can also select Mask all OGNL-expression generated log values. Learn more in Setting pseudonym and masking options in the PingFederate Administrator's Manual.

8. On the Summary tab, verify your configuration. Click Done.

9. On the Token Processor tab, click Save.

Using the Java Client SDK

Ping Identity provides the Java Client SDK to allow web service applications (client or provider) to interact with the PingFederate Security Token Service (STS).

The Java Client SDK provides functionality for sending a security token to the PingFederate STS for exchange with a returned SAML token, which can then be used to access web services across domains.

Download the Java Client SDK `.zip` archive from the Add-ons tab of the PingFederate downloads page. The `.zip` archive includes documentation in the `apidocs/index.html` file.

Java Client SDK sample code

The following code example shows how to send a token and request the exchange. Refer to the SDK documentation for modifications that apply to your site.

The code snippet below demonstrates using the PingFederate Java Client SDK to send an X.509 token to the PingFederate STS:

```
// Example method for obtaining the X.509 token
// You will need to implement this for your environment
X500PrivateCredential credential = getCredentialFromKeystore();

// Configure STS Client (IdP side / SP Connection)
STSClientConfiguration stsConfiguration = new STSClientConfiguration();
stsConfiguration.setAppliesTo("http://sp.domain.com");
stsConfiguration.setStsEndpoint("https://idp.domain.com:9031/idp/sts.wst");
stsConfiguration.setInTokenType(STSClientConfiguration.TokenType.X509);

// Instantiate the STSClient
STSClient stsClient = new STSClient(stsConfiguration);

// Send an RST Issue request to {pingfed} STS
Element samlToken = stsClient.issueToken(credential);
```

Release notes

Copyright © 2025 Ping Identity Corporation

Changelog

The following is the change history for the X.509 Token Translator:

X.509 1.1 – June 2021

- Added Include Subject Alternative Name (SAN) checkbox to the Extended Contract screen.

X.509 1.0 – April 2009

- Initial release.

Known issues and limitations

The following are known issues or limitations for the X.509 Token Translator.

Known issues

There are no known issues.

Known limitations

There are no known limitations.

Download manifest

The following files are included in the X.509 Token Translator `.zip` archive:

- `ReadMeFirst.pdf` : Contains links to this online documentation
- `Legal.pdf` : Copyright and license information
- `dist` : Contains the integration files
    - `deploy` : Contains the Java libraries
        - `pf-x509-token-translator-<version>.jar` : JAR file that contains the X.509 Token Processor

Copyright © 2025 Ping Identity Corporation

Zendesk Provisioner

The PingFederate provisioner for Zendesk enables an enterprise to provision its users to Zendesk.

This Zendesk provisioner includes a quick connection template to easily set up a single sign-on (SSO) connection requiring Zendesk provisioning.

Features

- User outbound provisioning
- Group outbound provisioning
- Browser-based service provider (SP) and identity provider (IdP)-initiated SSO

Intended audience

This document is intended for PingFederate administrators.

If you need help during the setup process, see the following resources:

- PingFederate documentation:
  - SP connection management
  - Identity provider SSO configuration
  - Managing digital signing certificates and decryption keys
  - Datastores
  - Configuring outbound provisioning
  - Configuring outbound provisioning settings

System requirements

- PingFederate 7.2.1 or later.
- Zendesk Plus and Enterprise accounts are required for SAML SSO integration. Some user attributes might require an Enterprise account.

User and group management

The Zendesk Provisioner synchronizes users and groups from your datastore to Zendesk. The following describes the behavior of each provisioning capability.

> **Tip**
>
> You can configure the following capabilities and specify which users to provision when you get to the Creating a connection part of the setup process.

Copyright © 2025 Ping Identity Corporation

Synchronizing existing users

PingFederate synchronizes users based on the `email` attribute in Zendesk. If a user already exists in your datastore and Zendesk, mapping this attribute correctly links the two records together.

For example:

- In Zendesk, Janet's `email` is `jsmith@example.com`.
- In your datastore, Janet's `mail` is `jsmith@example.com`.
- On the Attribute Mapping tab of your provisioning connection configuration, map the `email` attribute to `mail`.
- When the provisioning connector runs, the datastore user is provisioned with a `email` of `jsmith@example.com`. That matches Janet's existing `email` in Zendesk, so her information in the datastore is synchronized to her Zendesk account.

User provisioning

PingFederate provisions users when any of the following happens:

- A user is added to the datastore group or filter that is targeted by the provisioning connector.
- A user with `disabled` status is added to the datastore group or filter that is targeted by the provisioning connector, and theProvision disabled users provisioning option is enabled. This feature is not available in all provisioning connector versions.

You can define which users PingFederate targets for provisioning on the Source Location tab of your provisioning connection configuration.

User updates

PingFederate updates users when a user attribute changes in your datastore.

You can define which attributes PingFederate monitors for changes on the Attribute Mapping tab of your provisioning connection configuration.

User deprovisioning

PingFederate deprovisions users when any of the following happens:

- A user is deleted from the user store.
- A user is disabled in the user store.
- A user is removed from the datastore group or filter that is targeted by the provisioning connector.

The Remove User Action setting in the connection configuration determines whether the deprovisioning action disables or deletes the user.

Copyright © 2025 Ping Identity Corporation

Synchronizing existing groups

PingFederate synchronizes groups from the datastore to the target service based on the group name.

For example:

- In Zendesk, there is a group is named `Accounting`.
- In your datastore, there is a group with a `CN` of `Accounting`.
- When the provisioning connector runs, the two groups are synchronized.

Group provisioning

PingFederate provisions groups when a group is added to the datastore filter that is targeted by the provisioning connector.

You can define which groups PingFederate targets for provisioning and monitors for changes on the Source Location tab in your provisioning connection configuration.

Group name updates

PingFederate renames groups when they are renamed in the datastore.

Group membership updates

PingFederate updates group memberships when memberships change in the datastore, whether the change is in the group's properties or a user's properties.

Group memberships in the datastore overwrite the group memberships in Zendesk.

Group deletion

PingFederate deletes groups when any of the following happens:

- The group is deleted in the datastore.
- The group is removed from the datastore group or filter that is targeted by the provisioning connector.

> **Important**
>
> Group deletions are permanent and cannot be undone.

Setup

Upgrading an existing deployment

If you're upgrading from a previous version of the Zendesk Provisioner, note your existing service provider (SP) connection configuration and create a new connection.

Copyright © 2025 Ping Identity Corporation

Upgrading an existing deployment in PingFederate 10.1 or later

Steps

1. Back up your current PingFederate configuration as shown in Configuration archive in the PingFederate documentation.
2. In the PingFederate administrative console, go to Applications > Integration > SP Connections and select your connection.

   Note the attribute mappings for your existing SP connection.

   For help, see Mapping attributes in the PingFederate documentation.
3. Delete your existing SP connection.
    1. Go to Applications > Integration > SP Connections.For your existing connection, click Select action, and then click Delete. Click Confirm.
4. Complete the steps in Deploying the integration files.
5. Complete the steps in Creating a connection.
6. From Outbound Provisioning > Manage Channels > Channel, on the Attribute Mapping tab, configure the attribute mappings based on your notes.
7. Complete the steps in Configuring SAML SSO in Zendesk.

Upgrading an existing deployment in PingFederate 10.0 or earlier

Steps

1. Back up your current PingFederate configuration as shown in Configuration archive in the PingFederate documentation.
2. In the PingFederate administrative console, go to Identity Provider > SP Connections and select your connection.

   Note the attribute mappings for your existing SP connection.

   For help, see Mapping attributes in the PingFederate documentation.
3. Delete your existing SP connection.
    1. Go to Identity Provider > SP Connections > Manage All.For your existing connection, click Select action, and then click Delete. Click Save.
4. Complete the steps in Deploying the integration files.
5. Complete the steps in Creating a connection.
6. From Outbound Provisioning > Manage Channels > Channel, on the Attribute Mapping tab, configure the attribute mappings based on your notes.
7. Complete the steps in Configuring SAML SSO in Zendesk.

Copyright © 2025 Ping Identity Corporation

Obtain required information

Before you can configure this connector, you must complete the following steps.

> **Tip**
>
> Some of the following steps result in information that you'll need for subsequent tasks. You should copy this information to a secure location.

- Obtain your Zendesk administrator email
- Obtain your Zendesk subdomain
- Obtain your Zendesk API token
- Obtain your Zendesk SAML 2.0 metadata XML file
- Provision existing user accounts on Zendesk

Obtain your Zendesk administrator email

This connector requires a valid administrator account email to be used when configuring a service provider (SP) connection for outbound provisioning.

> **Note**
>
> You only require this information if you're configuring this connector for outbound provisioning.

Obtain your Zendesk subdomain

This connector requires the subdomain, such as <*YourSubDomain*>.zendesk.com, to access your Zendesk for single sign-on (SSO) and outbound provisioning.

Obtain your Zendesk SAML 2.0 metadata XML file

About this task

This connectors quick-connection template uses a metadata XML file to assist in configuring many settings in the SP connection. Before configuring your SP connection, you must first prepare the `zendesk-saml-metadata.xml` that comes packaged with the Zendesk Connector.

Steps

1. Open the `zendesk-saml-metadata.xml` file that came packaged with the Zendesk Connector in a text editor.
2. Replace all instances of `SUBDOMAIN` with your Zendesk subdomain value:

   Example:

   If your Zendesk subdomain is `pingidentity1234567890`, then `SUBDOMAIN.zendesk.com` would become `pingidentity1234567890.zendesk.com`.
3. Save your changes.

Copyright © 2025 Ping Identity Corporation

Provision existing user accounts on Zendesk

When configuring the connector, make sure that the value mapped to the `email` attribute matches the existing Zendesk user's email address exactly as it appears in Zendesk.

For example, on the Attribute Mapping tab, the user's `email` attribute on Zendesk is mapped to the user `mail` attribute in your LDAP. This will synchronize a user that already exists on Zendesk with an `email` address in Zendesk of `john.smith@mydomain.com`. In this case, the user's `mail` attribute in LDAP would also have to be `john.smith@mydomain.com`.

When the Zendesk connector provisions for the first time, this address is used to synchronize the user in your LDAP data store with the user in Zendesk.

Obtain your Zendesk API token

About this task

The Zendesk Connector's outbound provisioning functionality is built using Zendesk's REST API, which requires an authorized OAuth 2.0 Access Token.

> **Note**
>
> You only require this information if you are configuring this connector for outbound provisioning.

Steps

1. Sign on to Zendesk with your administrator account.
2. Click the Gear icon and select API in the Channels category.
3. On the Settings tab, ensure the Token Access check box is selected.
4. Note the API token value.

> **Note**
>
> These steps are subject to change at the discretion of Zendesk. If these steps are different from the steps in this document, contact your Zendesk representative for updated steps for obtaining this API token.

Enabling provisioning and single sign-on in PingFederate

To use PingFederate for provisioning and single sign-on, configure an external datastore and set a SAML entity ID.

About this task

Your external datastore acts as the source of data for provisioning. PingFederate also uses an internal datastore to store the state of synchronization between the source datastore and the target datastore.

For more information, see Datastores and Configuring outbound provisioning settings in the PingFederate documentation.

Copyright © 2025 Ping Identity Corporation

Enabling provisioning and single sign-on in PingFederate 10.1 or later

Steps

1. In the PingFederate administrative console, configure the datastore for PingFederate to use as the source of user data.

   For help, see Datastores in the PingFederate documentation.

2. Go to System > Server > Protocol Settings > Federation Info.
3. In the SAML 2.0 Entity ID field, enter a name for PingFederate to use when identifying itself to SAML partners.
4. On the Outbound Provisioning tab, in the Provisioner Data Store list, select the internal database that will store the synchronization state. Click Save.

   For more information, see Configuring outbound provisioning settings in the PingFederate documentation.

Enabling provisioning and single sign-on in PingFederate 10.0 or earlier

Steps

1. In the PingFederate administrative console, configure the datastore for PingFederate to use as the source of user data.

   For help, see Datastores in the PingFederate documentation.

2. Go to System > Protocol Settings > Roles & Protocols.
3. Select the Enable Identity Provider IdP Role and Support the Following check box.
4. Select the SAML 2.0 and Outbound Provisioning check boxes. Click Next.
5. Click the Federation Info tab.
6. In the SAML 2.0 Entity ID field, enter a name for PingFederate to use to identify itself to SAML partners.
7. On the Outbound Provisioning tab, in the Provisioner Data Store list, select the internal database that will store the synchronization state. Click Save.

   For more information, see Configuring outbound provisioning settings in the PingFederate documentation.

Deploying the integration files

To get started with the integration, deploy the Zendesk Provisioner files to your PingFederate directory.

Steps

1. Download the Zendesk Provisioner `.zip` archive from the Ping Identity Integration Directory.
2. Stop PingFederate.
3. If you're upgrading an existing deployment, delete the `pf-zendesk-quickconnection-<version>.jar` file from your `<pf_install>/pingfederate/server/default/deploy` directory.

Copyright © 2025 Ping Identity Corporation

4. Extract the `.zip` archive and merge the contents of the `dist` directory with your `<pf_install>/pingfederate/server/default/deploy` directory.

   Enable the PingFederate provisioning engine:

   1. Open your `<pf_install>/pingfederate/bin/run.properties` file.Change `pf.provisioner.mode` to `STANDALONE`. Save the file.

      > **Note**
      >
      > To configure the `FAILOVER` mode instead, see Deploying provisioning failover in the PingFederate Server Clustering Guide.

5. Start PingFederate.
6. If you operate PingFederate in a cluster, repeat steps 2-4 and step 6 for each engine node.

Creating a connection

To allow PingFederate to act as an identity provider and manage users in Zendesk, create a service provider (SP) connection.

Steps

1. In the PingFederate administrator console, create a new SP connection:
   1. For PingFederate 10.1 or later: go to Applications > Integration > SP Connections. Click Create Connection.
   2. For PingFederate 10.0 or earlier: go to Identity Provider > SP Connections. Click Create Connection.
2. Configure the basic connection details with the Zendesk quick connection template.
   1. On the Connection Template tab, select Use a template for this connection.
   2. In the Connection Template list, select Zendesk Provisioner.
   3. In the Metadata File row, upload the zendesk-saml-metadata.xml file that you saved in Obtain your Zendesk SAML 2.0 metadata XML file. Click Next.
   4. On the Connection Type tab, select Browser SSO Profiles and Outbound Provisioning. Click Next.
   5. On the Connection Options tab, click Next.
   6. On the General Info tab, in the Connection Name field, enter a name of your choosing. Click Next.
3. On the Browser SSO tab, configure your browser SSO settings as shown in Configuring IdP Browser SSO in the PingFederate documentation.
   1. On the Browser SSO > SAML Profiles tab, select only IdP-Initiated SSO and SP-Initiated SSO.
   2. On the Browser SSO > Assertion Creation > Attribute Contract tab, set the following name format.

Attribute Contract	Subject Name Format
SAML_SUBJECT	`[.codeph]` urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified````

Copyright © 2025 Ping Identity Corporation

3. On the Browser SSO > Protocol Settings > Signature Policy tab, select the Always Sign Assertion check box.

4. On the Credentials tab, configure the connection credentials as shown in Configuring credentials in the PingFederate documentation. Click Next.

5. On the Credentials > Digital Signature Settings tab, select the Include the certificate in the signature <KEYINFO> element check box.

6. On the Outbound Provisioning tab, configure provisioning as shown in Configuring outbound provisioning in the PingFederate documentation:

    1. On the Target tab, enter the Zendesk Administator email, Sub-domain and API Token you obtained in Obtain required information.

    2. On the Manage Channels > Attribute Mapping tab, complete the attribute mappings as shown in Managing channels in the PingFederate documentation.

       Provide a source of data or provide a default value for the `role` attribute. This attribute only accepts specific values, which are explained in the Supported attributes reference.

7. On the Activation and Summary tab, above the Summary section, click the toggle to turn on the connection. Click Save.

Supported attributes reference

The following table consists of the attributes that can be mapped on a user during provisioning.

Attribute	Description
`email`	The primary email address of the user. **Note** An update to the email attribute will make the new email as primary and leave the old email as secondary.
`external_id`	A unique id you can set on a user.
`name`	The name of the user
`alias`	An Agent's alias that is displayed visible by end-user type users.
`custom_role_id`	The if of the custom role for the user. The id of a custom role can be obtained from Zendesk after creating it.
`details`	Any details you wish to store about the user.
`locale_id`	The id of the user's locale. The id of a locale can be obtained from Zendesk.
`moderator`	Designates whether the user has forum moderation capabilities. Acceptable options include: "true" or "false".

Copyright © 2025 Ping Identity Corporation

Attribute	Description
`notes`	A general note field for the user.
`only_private_comments`	Designates whether the user can only create private comments. Acceptable options include: “true” or “false”.
`organization_id`	The id of the organization this user is associated with. The id of an organization can be obtained from Zendesk after creating it. Example: An organization called “ `Employee` ” may have an id that looks like “ `187178237` ”.
`phone`	The primary phone number of a user.
`role`	The role of the user in Zendesk. Acceptable options include: “end-user”, “agent” and “admin”.
`signature`	The user’s signature. Only available for agents and admins.
`tags`	A multi value attribute representing all the user’s tags. Tags will only be present if your account has user tagging enabled.
`ticket_restriction`	Specifies which tickets the user has access to. Possible values are: “organization”, “groups”, “assigned”, “requested”, null.
`time_zone`	The user’s time zone.
`verified`	Designates if the user’s identity has been verified. Acceptable options include: “true” or “false”.

Configuring SAML SSO in Zendesk

The following section describes the steps for configuring SP and IdP-initiated SSO to Zendesk.

About this task

> **Note**
>
> This section requires two pieces of information that can be found within PingFederate. The first is the SAML 2.0 Entity ID, which can be found under on the **Server Settings** page and the second is the exported certificate used to sign the SAML assertion (configured in step 19 of Configure a Connection).

Steps

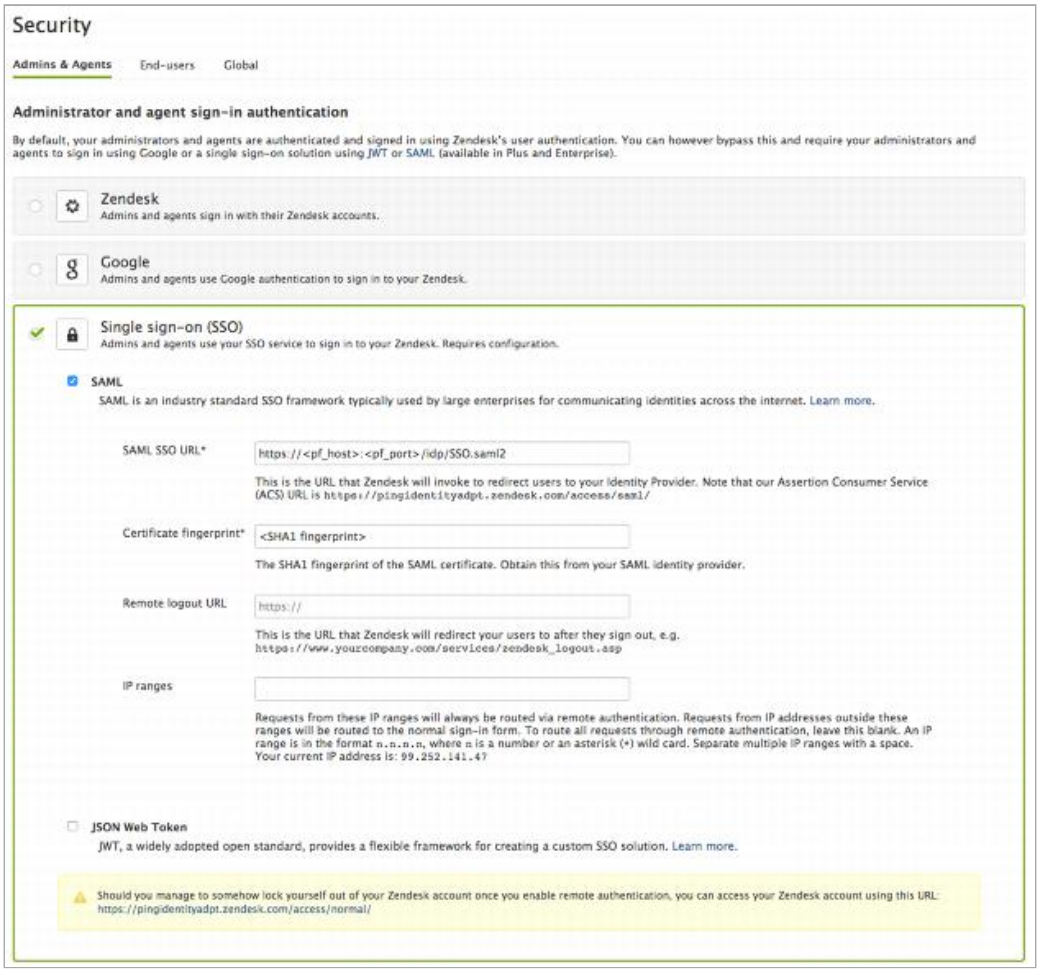
1. Navigate to `https://SUBDOMAIN.zendesk.com/agent/admin/` and sign in with your Admin credentials.
2. Navigate to Security under the Settings section to view single sign-on (SSO) configuration options.
3. If configuring SSO for Admins & Agents, select the Admins & Agents tab to view the Single sign-on (SSO) options.
4. If configuring SSO for End-users, select the End-users tab to view the Single sign-on (SSO) options.

Copyright © 2025 Ping Identity Corporation

5. Select Single sign-on (SSO) and select SAML to configure the options.

    1. Enter the PingFederate SAML endpoint into the SAML SSO URL field. `https://<pf_host>:<pf_port>/idp/SSO.saml2`

    2. Copy and paste the SHA1 fingerprint of the signing certificate into the Certificate fingerprint field.



6. Click Save to complete Zendesk SSO Setup.

Release notes

Changelog

Zendesk Connector 2.1 – June 2017 (current release)

- Added Support for updating user emails

Zendesk Connector 2.0 – October 2015

- Added Support for Group Provisioning
- Added Support for provisioning additional user attributes

Copyright © 2025 Ping Identity Corporation

Zendesk Connector 1.0 – October 2014

- Initial Release
- Added Support for User Provisioning

Known issues and limitations

Known Limitations

- Because of a limitation with PingFederate 8.1 and earlier versions, when configuring two service provider (SP) connections with the same provisioner, the second connection built can be pre-populated with the channel from the first connection. To avoid conflicts, delete this pre-populated channel and create a unique channel for each connection.
- Single logout (SLO) is not supported.
- User attributes cannot be cleared once set, they can only be updated.
- Emails that contain plus (+) symbols are not supported and may cause issues with provisioning.
- Syncing with existing groups is not supported. The connector is not able to manage groups, or add users to groups that it has not created.
- There is the potential to have a user in Zendesk managed by two users in Active Directory. This occurs when a new user is created with the secondary email of another user.

Download manifest

The distribution `.zip` archive for the Connector contains the following:

- `ReadMeFirst.pdf` – contains links to this online documentation.
- `zendesk-saml-metadata.xml` – The metadata used for Browser SSO
- `legal`:
  - `Legal.pdf` – copyright and license information.
- `dist` – contains libraries needed for the Connector:
  - `pf-zendesk-quickconnection-2.1.jar` – PingFederate Zendesk Connector

Copyright © 2025 Ping Identity Corporation

Zoom Provisioner

The Zoom Provisioner allows PingFederate to integrate with Zoom for user provisioning and single sign-on (SSO).

Features

- Manages users in Zoom based on changes in an external datastore attached to PingFederate.
  - Creates, updates, and disables/deletes users.
  - Allows you to enable the create, update, and disable/delete capabilities independently.
  - Allows you to choose to disable or delete users when deprovisioning.
- Browser-based single sign-on (SSO) and single logout (SLO) initiated by the service provider (SP) or identity provider (IdP).
- Pre-populates some connection settings with the included quick connection template and SAML metadata file.

Intended audience

This document is intended for PingFederate administrators. Before you start, familiarize yourself with the following:

- The following sections of the PingFederate documentation:
  - Identity provider SSO configuration
  - Datastores
  - Managing IdP adapters
  - Managing digital signing certificates and decryption keys
  - SP connection management
  - Configuring outbound provisioning
  - Configuring outbound provisioning settings
- The following sections of the Zoom documentation:
  - Guidelines for Vanity URL Requests
  - Getting Started with SSO
  - Accessing SAML Response Logs
  - SCIM2 API
  - SCIM2 API User attributes

System requirements

- PingFederate 9.0 or later.

Copyright © 2025 Ping Identity Corporation

- A Zoom system administrator account with SSO enabled.
- To allow PingFederate to make outbound connections to the Zoom API, you might need to allow list the following domain in your firewall:
    - https://api.zoom.us/scim2

User management

The Zoom Provisioner links users from the datastore to Zoom. The following describes the behavior of each provisioning capability.

Synchronizing existing users

By default, the provisioning connector synchronizes users from the datastore to the target service by matching the `mail` attribute in the datastore to the `Username` attribute in the target service. You can change the default mapping in Creating a provisioning connection.

For example:

- In the target service, Janet's `Username` is `jsmith@example.com`.
- In your datastore, Janet's `mail` is `jsmith@example.com`.
- On the Attribute Mapping tab of your PingFederate channel configuration, you map the `Username` attribute to `mail`.
- When the provisioning connector runs, the datastore user is provisioned with a `Username` of `jsmith`. That matches Janet's existing `Username` in the target service, so her information in the datastore is synchronized to her target service account.

User provisioning

User provisioning is triggered when a user is added to the datastore group or filter targeted by the provisioning connector.

The target is determined by the Source Location tab in the provisioning connection configuration.

User updates

User updates are triggered when a change occurs to a user attribute mapped in the provisioning connector configuration.

User deprovisioning

User deprovisioning is triggered by any of the following:

- A user is deleted from the user store.
- A user is disabled in the user store.
- A user is removed from the data store filter targeted by the provisioning connector.

Copyright © 2025 Ping Identity Corporation

The Remove User Action setting in the connection configuration determines whether the deprovisioning action disables or deletes the user.

Setup

Upgrading an existing deployment

If you're upgrading from a previous version of the Zoom Provisioner, note your existing service provider (SP) connection configuration and create a new connection.

Upgrading an existing deployment in PingFederate 10.1 or later
Steps

1. Back up your current PingFederate configuration as shown in Configuration archive in the PingFederate documentation.
2. In the PingFederate administrative console, go to Applications > Integration > SP Connections and select your connection.
3. Note the attribute mappings for your existing SP connection.

   You can find help in Mapping attributes in the PingFederate documentation.
4. Delete your existing SP connection:
    1. Go to Applications > Integration > SP Connections.
    2. For your existing connection, click Select action, and then click Delete. Click Confirm.
5. Complete the steps in Deploying the integration files.
6. Complete the steps in Creating a provisioning connection.
    1. Go to Outbound Provisioning > Manage Channels > Channel, and on the Attribute Mapping tab, configure the attribute mappings based on your notes.
7. (Optional) Complete the steps in Creating a single sign-on connection.

Upgrading an existing deployment in PingFederate 10.0 or earlier
Steps

1. Back up your current PingFederate configuration as shown in Configuration archive in the PingFederate documentation.
2. In the PingFederate administrative console, go to Identity Provider > SP Connections and select your connection.
3. Note the attribute mappings for your existing SP connection.

   You can find help in Mapping attributes in the PingFederate documentation.
4. Delete your existing SP connection:
    1. Go to Identity Provider > SP Connections > Manage All.

Copyright © 2025 Ping Identity Corporation

2. For your existing connection, click Select action, and then click Delete. Click Save.

5. Complete the steps in Deploying the integration files.
6. Complete the steps in Creating a provisioning connection.
    1. Go to Outbound Provisioning > Manage Channels > Channel, and on the Attribute Mapping tab, configure the attribute mappings based on your notes.
7. (Optional) Complete the steps in Creating a single sign-on connection.

Deploying the integration files

To get started with the integration, deploy the Zoom Provisioner files to your PingFederate directory.

Steps

1. Download the Zoom Provisioner `.zip` archive from the Add-ons tab of the PingFederate downloads page or the Ping Identity Integration Directory.
2. Stop PingFederate.
3. If you're upgrading an existing deployment, delete the `pf-zoom-quickconnection-<version>.jar` file from your `<pf_install>/pingfederate/server/default/deploy` directory.
4. Extract the `.zip` archive and merge the contents of the `dist` directory with your `<pf_install>/pingfederate/server/default/deploy` directory.
5. Enable the PingFederate provisioning engine:
    1. Open your `<pf_install>/pingfederate/bin/run.properties` file.
    2. Change `pf.provisioner.mode` to `STANDALONE`. Save the file.

    > **Note**
    >
    > To configure the `FAILOVER` mode instead, learn more in Deploying provisioning failover in the PingFederate server clustering guide.

6. Start PingFederate.
7. If you operate PingFederate in a cluster, repeat steps 2 - 4 and step 6 for each engine node.

Getting a vanity URL for Zoom

Before you can integrate with Zoom, you need to apply for a custom Zoom URL for your organization and add it to your connection metadata file.

About this task

> **Note**
>
> Zoom typically takes 1 - 2 business days to process vanity URL requests.

Copyright © 2025 Ping Identity Corporation

Steps

1. Sign on to the Zoom Account Profile page as an administrator.
2. In the Vanity URL section, click Apply.
3. On the Apply for Vanity URL dialog, enter a URL following the Guidelines for Vanity URL Requests in the Zoom documentation. Click Apply.
4. After your request is approved, copy the vanity URL to the `.xml` metadata file used to configure the connection in PingFederate:
    1. In a text editor, open the `zoom-saml-metadata.xml` file included in the integration `.zip` file.
    2. In the `.xml` file, replace every instance of *https://your_vanity_url.zoom.us* with the vanity URL that you requested.
    3. Save the changes.

    You will use this file in Creating a single sign-on connection.

Creating a Zoom app

To integrate PingFederate with Zoom, add it as a server-to-server OAuth app.

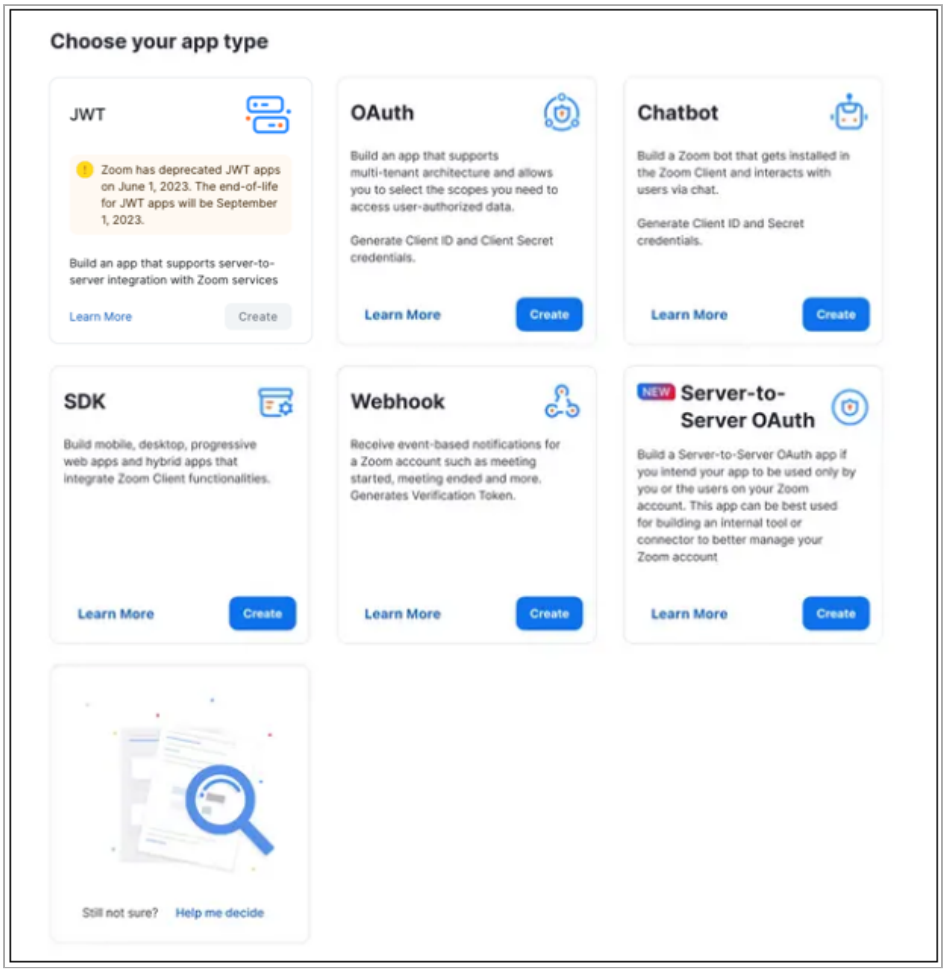
Steps

1. Go to the Zoom App Marketplace.
2. In the top right corner of the screen, click the Develop list and select Build App.

    Result:

    The Choose your app type page displays.

Copyright © 2025 Ping Identity Corporation



3. On the Server-to-Server OAuth app type tile, click Create.
4. In the App Name field, enter the name of the app. Click Create.

   Result:

   The account ID, client ID, and client secret for your new application display on the App credentials tab. Remember where this information is stored, as you'll need it to create a provisioning connection.
5. On the Information tab, add information about your app.

   For example, create a meaningful short description.
6. Enter the company name.

   > **Important**
   >
   > This information is required to activate the app.
7. In the Developer Contact Information section, enter your name and email address in the appropriate fields. Click Continue.
8. On the Feature tab, click the toggle to enable the features you want, then click Continue.

   For example, add events, team chats, and multi-platform support.

Copyright © 2025 Ping Identity Corporation

9. On the Scopes tab, click Add Scopes and select the following scopes:

   User scopes

   - user:master: Allows the app to view and manage sub account user information.
   - user:read:admin: Allows the app to view information for all users in a Zoom account. For example, profile information, user settings, user permissions, and the user's scheduling privileges.
   - user:write:admin: Allows the app to view and manage user profile information, such as settings and permissions.

   Account scopes

   - account:master: Allows the app to view and manage sub accounts.
   - account:read:admin: Allows an app to view account and sub account information. This includes account settings, account lock settings, managed domains, and the account's trusted domains.
   - account:write:admin: Allows an app to manage sub accounts on behalf of a master account. This includes creating or disassociating a sub account from a master account, updating an account's owner or account settings, and updating lock settings.

   SCIM2 scope

   - scim2: Allows an app to provide support for provisioning support.

10. After you're finished, click Done.
11. Go to the Activation tab and click Activate your app.

Enabling provisioning and single sign-on in PingFederate

To use PingFederate for provisioning and single sign-on, configure an external datastore and set a SAML entity ID.

About this task

Your external datastore acts as the source of data for provisioning. PingFederate also uses an internal datastore to store the state of synchronization between the source datastore and the target datastore.

Learn more in Datastores and Configuring outbound provisioning settings in the PingFederate documentation.

Enabling provisioning and single sign-on in PingFederate 10.1 or later

Steps

1. In the PingFederate administrative console, configure the datastore for PingFederate to use as the source of user data.

   Learn more in Datastores in the PingFederate documentation.

2. Go to System > Server > Protocol Settings > Federation Info.
3. In the SAML 2.0 Entity ID field, enter a name for PingFederate to use when identifying itself to SAML partners.

Copyright © 2025 Ping Identity Corporation

4. On the Outbound Provisioning tab, in the Provisioner Data Store list, select the internal database that will store the synchronization state. Click Save.

   Learn more in Configuring outbound provisioning settings in the PingFederate documentation.

Enabling provisioning and single sign-on in PingFederate 10.0 or earlier

Steps

1. In the PingFederate administrative console, configure the datastore for PingFederate to use as the source of user data.

   You can find help in Datastores in the PingFederate documentation.

2. Go to System > Protocol Settings > Roles & Protocols.
3. Select the Enable Identity Provider IdP Role and Support the Following checkbox.
4. Select the SAML 2.0 and Outbound Provisioning checkboxes. Click Next.
5. Click the Federation Info tab.
6. In the SAML 2.0 Entity ID field, enter a name for PingFederate to use to identify itself to SAML partners.
7. On the Outbound Provisioning tab, in the Provisioner Data Store list, select the internal database that will store the synchronization state. Click Save.

   Learn more in Configuring outbound provisioning settings in the PingFederate documentation.

Creating a provisioning connection

To allow PingFederate to manage users in Zoom, create a service provider (SP) connection.

Steps

1. In the PingFederate administrator console, create a new SP connection:

   Choose from:

   - For PingFederate 10.1 or later, go to Applications > Integration > SP Connections. Click Create Connection.
   - For PingFederate 10.0 or earlier, go to Identity Provider > SP Connections. Click Create Connection.

2. Configure the basic connection details with the Zoom quick connection template:

   1. On the Connection Template tab, select Use a template for this connection.
   2. In the Connection Template list, select Zoom Provisioner.
   3. On the Metadata File row, select the `zoom-saml-metadata.xml` file you modified in Getting a vanity URL for Zoom. Click Next.
   4. On the Connection Type tab select only Outbound Provisioning. Click Next.
   5. On the Connection Options tab, click Next.

Copyright © 2025 Ping Identity Corporation

6. On the General Info tab, in the Connection Name field, enter a name for the connection. Click Next.

3. On the Outbound Provisioning tab, configure the provisioning target and channel as shown in Configuring outbound provisioning in the PingFederate documentation.

    1. Click Configure Provisioning.

    2. On the Target tab, enter the account ID, client ID, and client secret noted in step 4.

        > **Note**
        >
        > PingFederate verifies the configurations when you activate the channel and SP connection.

    3. Under Provisioning Options, customize the provisioning connector actions as shown in Provisioning options reference. Click Next.

    4. On the Manage Channels tab, create a channel as shown in Managing channels in the PingFederate documentation. Click Done.

        > **Note**
        >
        > You can find more information about the attributes available in your channel configuration in Supported attributes reference.

    5. On the Outbound Provisioning tab, click Next.

4. On the Activation and Summary tab, above the Summary section, turn on the connection. Click Save.

Provisioning options reference

The following table lists the main provisioning capabilities available in the Zoom connection configuration.

Field Name	Description
User Create	*Selected (default)* PingFederate creates users in Zoom. *Cleared* PingFederate doesn't create users in Zoom.
User Update	*Selected (default)* PingFederate updates existing users in Zoom. PingFederate can also re-enable disabled users in Zoom. *Cleared* PingFederate doesn't update existing users in Zoom.

Copyright © 2025 Ping Identity Corporation

Field Name	Description
User Disable / Delete	*Selected (default)* PingFederate removes users in Zoom according to the Remove User Action setting. *Cleared* PingFederate doesn't disable or delete users in Zoom.
Note If any of the previous options are cleared, PingFederate logs a warning in the user workflow section of `provisioner.log` when the related action fails.	
Remove User Action	This option applies when User Disable / Delete is selected, and either: • A previously-provisioned user no longer meets the condition set on the Source Location tab • A user has been disabled or deleted from the datastore. *Disable (default)* PingFederate disables the user in Zoom. *Delete* PingFederate deletes the user from Zoom.

Supported attributes reference

The following table lists the attributes that can be mapped for user provisioning to Zoom.

Attribute	Description
`Username`	The user's unique identifier in Zoom. Must be in email address format. **Note** This attribute is required.
`Email`	The user's email address.
`Department`	The user's department or work group. For example, `Sales`.

Copyright © 2025 Ping Identity Corporation

Attribute	Description
`Family Name`	The user's family name. For example, `Jensen`, given the full name Ms. Barbara Jane Jensen, III.
`Given Name`	The user's given name. For example, `Barbara`, given the full name Ms. Barbara Jane Jensen, III.
`Locale`	Indicates the user's default location to localize such items as currency, date time format, or numerical representations.
`Organization`	The user's organization.
`Phone`	The user's phone number, formatted as `+1 2015550123`.
`Roles`	Roles held by the user. For example, `Student` or `Engineer`.
`Title`	The user's title. For example, `Vice President`.
`User Type`	The account type. This can be `Basic`, `Licensed`, or `On-Prem`.

Configuring single sign-on

To allow PingFederate to act as an identity provider for Zoom, enable single sign-on (SSO) in PingFederate and Zoom, then create a connection.

About this task

> **Note**
>
> Single sign-on integration is an optional part of this integration. If you only want to use the Zoom Provisioner for provisioning, skip these steps.

Steps

1. Complete the steps in Enabling provisioning and single sign-on in PingFederate.
2. Complete the steps in Configuring single sign-on in Zoom.
3. Complete the steps in Add the Zoom metadata URL in PingFederate.
4. Complete the steps in Creating a single sign-on connection.

Configuring single sign-on in Zoom

To allow PingFederate to manage authentication, enable single sign-on (SSO) in Zoom.

Copyright © 2025 Ping Identity Corporation

About this task

You can find more information in Getting Started with SSO in the Zoom documentation.

Steps

1. Sign on to the Zoom Single sign-on page as an administrator.
2. Click Enable Single Sign-On.
3. In the Sign-in page URL field, enter `https://pf_host:pf_port/idp/SSO.saml2`.
4. (Optional) In the Sign-out page URL field, enter `https://pf_host:pf_port/idp/SLO.saml2`.
5. In the Identity Provider Certificate field, enter the contents of your PingFederate signing certificate, without the `Begin Certificate` and `End Certificate` lines.

   To get your certificate, refer to the Exporting a certificate section in Managing digital signing certificates and decryption keys in the PingFederate documentation.
6. Note the Service Provider (SP) Entity ID for use in Creating a single sign-on connection.
7. In the Issuer (IDP Entity ID) field, enter the SAML 2.0 Entity ID that you created in Enabling provisioning and single sign-on in PingFederate.
8. In the Binding list, select a method.
9. In the Signature Hash Algorithm section, click SHA-256.
10. In the Security section, select Sign SAML request and Save SAML response logs on user sign-in.
11. In the Provision User list, select Prior to Sign-In. Click Save Changes.

Add the Zoom metadata URL in PingFederate

To provide PingFederate with information about Zoom as a SAML partner, register the Zoom metadata URL.

Steps

1. On the PingFederate administrative console, on the Security > Partner Metadata URLs tab, click Add New URL.
2. On the URL tab, in the Name field, enter a name.
3. In the URL field, enter `https://your_vanity_url.zoom.us/saml/metadata/sp`. Click Load Metadata, then Next.
4. On the Certificate Summary tab, click Next.
5. On the Summary tab, click Done.
6. On the Partner Metadata URLs tab, click Save.

Creating a single sign-on connection

To allow PingFederate to handle authentication for Zoom, create a service provider (SP) connection.

Copyright © 2025 Ping Identity Corporation

About this task

> **Note**
>
> You can follow these steps to create a new SP connection, or you can modify your provisioning connection.

Steps

1. In the PingFederate administrator console, create a new SP connection:

   Choose from:

   - For PingFederate 10.1 or later, go to Applications > Integration > SP Connections. Click Create Connection.
   - For PingFederate 10.0 or earlier, go to Identity Provider > SP Connections. Click Create Connection.

2. Configure the basic connection details with the Zoom quick connection template:
   1. On the Connection Template tab, select Use a template for this connection.
   2. In the Connection Template list, select Zoom Provisioner.
   3. On the Metadata File row, select the `zoom-saml-metadata.xml` file you modified in Getting a vanity URL for Zoom. Click Next.
   4. On the Connection Type tab select Browser SSO Profiles. Click Next.
   5. On the Connection Options tab, select only Browser SSO. Click Next.
   6. If you see the Metadata URL tab, clear the Enable automatic reloading checkbox. Click Next.
   7. On the General Info tab, in the Connection Name field, enter a name for your connection. Click Next.
3. On the Browser SSO tab, configure browser SSO.

   You can find a complete guide in Configure IdP Browser SSO in the PingFederate documentation.
   1. On Browser SSO > SAML Profiles, select all four checkboxes.
   2. On Browser SSO > Protocol Settings > SLO Service URLs, in the Binding list, select the method you chose for Binding in Configuring single sign-on in Zoom.
      1. In the Endpoint URL and Response URL fields, enter `/saml/SingleLogout`. Click Add.
   3. On Browser SSO > Protocol Settings > Allowable SAML Bindings, select the method that you chose for Binding in Configuring single sign-on in Zoom.
4. On the Credentials tab, configure the connection credentials. In the Signing Algorithm list, select RSA SHA256. Click Next.

   You can find a complete guide in Configure credentials in the PingFederate documentation.
5. On the Activation and Summary tab, above the Summary section, turn on the connection. Click Save.

Release notes

Copyright © 2025 Ping Identity Corporation

Changelog

The following is the change history for the Zoom Provisioner.

Zoom Provisioner 1.3 - June 2023

Added support for server-to-server OAuth apps, because JWT apps are being deprecated and won't be supported after September 1, 2023.

Zoom Provisioner 1.2 - December 2022

Added the ability to restore a user's license when the user is re-enabled. Updating the license is no longer required.

Zoom Provisioner 1.0 - March 2020

- Initial release
- Added support for user provisioning
- Added support for Zoom attributes
- Added support for API key and secret authentication
- Added configuration options for create, update, and disable/delete capabilities
- Added configuration options for deprovisioning actions

Known issues and limitations

The following are known issues or limitations with the Zoom Provisioner.

Known issues

There aren't any known issues.

Known limitations

- This integration doesn't support group provisioning.
- When an LDAP user is deleted in a targeted group distinguished name (DN), the provisioning connector doesn't propagate the deletion until a new user is added to the group. This limitation is compounded when the User Create provisioning option is disabled. You can find solutions in SaaS provisioner does not remove the user in the Ping Identity Knowledge Base.
- Because of PingFederate limitations, user attributes cannot be cleared once set.
- Zoom only allows a single value for the `Roles` attribute.
- Zoom doesn't allow users with the admin role to be disabled or deleted. Change the user's role first.
- Because of a limitation in Zoom, if a user's attributes change at the same time they're enabled or disabled, only the `disabled` status is updated in Zoom. The attributes are updated the next time a change is made to that user.

Copyright © 2025 Ping Identity Corporation

Download manifest

The following files are included in the Zoom Provisioner `.zip` archive:

- `legal/Legal.pdf` : Copyright and license information.
- `dist` : Contains the integration files.
    - `pf-zoom-quickconnection-<version>.jar` : The Zoom Provisioner quick connection template.
- `zoom-saml-metadata.xml` : A modifiable metadata file used to create the connection.

Copyright © 2025 Ping Identity Corporation

Overview of Zscaler integrations

Zscaler Internet Access Connector

The Zscaler Internet Access Connector allows PingFederate to integrate with Zscaler Internet Access for user provisioning, group provisioning, and single sign-on (SSO).

Zscaler Private Access Connector

The Zscaler Private Access Connector allows PingFederate to integrate with Zscaler Private Access for user provisioning, group provisioning, and single sign-on (SSO).

Zscaler Internet Access Provisioner

The Zscaler Internet Access Provisioner allows PingFederate to integrate with Zscaler Internet Access for user and group provisioning and single sign-on (SSO).

Features

- Manages users and groups in Zscaler Internet Access based on changes in an external data store that is attached to PingFederate.
    - Creates, updates, and deletes users
    - Allows you to enable the create, update, and delete capabilities independently
    - Creates groups and updates group memberships
- Supports browser-based SSO initiated by the service provider (SP) or identity provider (IdP)
- Pre-populates some connection settings with the included quick connection template

Intended audience

This document is intended for PingFederate administrators.

If you need help during the setup process, see the following resources:

- The following sections of the Zscaler Internet Access documentation:
    - About SAML
    - Configuring SAML
    - About SCIM
    - Configuring SCIM
- The following sections of the PingFederate documentation:
    - Datastores

Copyright © 2025 Ping Identity Corporation

- Configuring outbound provisioning
- Configuring outbound provisioning settings
- Identity provider SSO configuration
- Managing digital signing certificates and decryption keys
- SP connection management

System requirements

- PingFederate 9.0 or later.
- A Zscaler Internet Access administrator account.
- To allow PingFederate to make outbound HTTPS connections, you might need to allow the following host names in your firewall:
  - https://scim.<*your_Zscaler_domain*>.net

User and group management

The Zscaler Internet Access Provisioner synchronizes users and groups from your datastore to Zscaler Internet Access. The following describes the behavior of each provisioning capability.

> **Tip**
>
> You can configure the following capabilities and specify which users to provision when you get to the Creating a provisioning connection part of the setup process.

Synchronizing existing users

PingFederate synchronizes users based on the `Username` attribute in Zscaler Internet Access. If a user already exists in your datastore and Zscaler Internet Access, mapping this attribute correctly links the two records together.

For example:

- In Zscaler Internet Access, Janet's `Username` is `jsmith@example.com`.
- In your datastore, Janet's `mail` is `jsmith@example.com`.
- On the Attribute Mapping tab of your provisioning connection configuration, map the `Username` attribute to `mail`.
- When the provisioning connector runs, the datastore user is provisioned with a `Username` of `jsmith@example.com`. That matches Janet's existing `Username` in Zscaler Internet Access, so her information in the datastore is synchronized to her Zscaler Internet Access account.

User provisioning

PingFederate provisions users when any of the following happens:

- A user is added to the datastore group or filter that is targeted by the provisioning connector.

Copyright © 2025 Ping Identity Corporation

- A user with `disabled` status is added to the datastore group or filter that is targeted by the provisioning connector, and the Provision disabled users provisioning option is enabled. This feature is not available in all provisioning connector versions.

You can define which users PingFederate targets for provisioning on the Source Location tab of your provisioning connection configuration.

User updates

PingFederate updates users when a user attribute changes in your datastore.

You can define which attributes PingFederate monitors for changes on the Attribute Mapping tab of your provisioning connection configuration.

User deprovisioning

PingFederate deprovisions users when any of the following happens:

- A user is deleted from the user store.
- A user is disabled in the user store.
- A user is removed from the datastore group or filter that is targeted by the provisioning connector.

The Remove User Action setting in the connection configuration determines whether the deprovisioning action disables or deletes the user.

Synchronizing existing groups

PingFederate synchronizes groups from the datastore to the target service based on the group name.

For example:

- In Zscaler Internet Access, there is a group is named `Accounting`.
- In your datastore, there is a group with a `CN` of `Accounting`.
- When the provisioning connector runs, the two groups are synchronized.

Group provisioning

PingFederate provisions groups when a group is added to the datastore filter that is targeted by the provisioning connector.

You can define which groups PingFederate targets for provisioning and monitors for changes on the Source Location tab in your provisioning connection configuration.

Group name updates

PingFederate renames groups when they are renamed in the datastore.

Group membership updates

PingFederate updates group memberships when memberships change in the datastore, whether the change is in the group's properties or a user's properties.

Group memberships in the datastore overwrite the group memberships in Zscaler Internet Access.

Copyright © 2025 Ping Identity Corporation

Group deletion

PingFederate deletes groups when any of the following happens:

- The group is deleted in the datastore.
- The group is removed from the datastore group or filter that is targeted by the provisioning connector.

> **Important**
>
> Group deletions are permanent and cannot be undone.

Setup

Upgrading an existing deployment

If you're upgrading from a previous version of the Zscaler Internet Access Provisioner, note your existing service provider (SP) connection configuration and create a new connection.

Upgrading an existing deployment in PingFederate 10.1 or later

Steps

1. Back up your current PingFederate configuration as shown in Configuration archive in the PingFederate documentation.
2. In the PingFederate administrative console, go to Applications > Integration > SP Connections and select your connection.
3. Note the attribute mappings for your existing SP connection.

   Learn more in Mapping attributes in the PingFederate documentation.
4. Delete your existing SP connection.
   1. Go to Applications > Integration > SP Connections.
   2. For your existing connection, click Select action and then click Delete. Click Confirm.
5. Complete the steps in Deploying the integration files.
6. Complete the steps in Creating a provisioning connection.
   - From Outbound Provisioning > Manage Channels > Channel, on the Attribute Mapping tab, configure the attribute mappings based on your notes.
7. Complete the steps in Creating a single sign-on connection.

Copyright © 2025 Ping Identity Corporation

Upgrading an existing deployment in PingFederate 10.0 or earlier

Steps

1. Back up your current PingFederate configuration as shown in Configuration archive in the PingFederate documentation.
2. In the PingFederate administrative console, go to Identity Provider > SP Connections and select your connection.
3. Note the attribute mappings for your existing SP connection.

   Learn more in Mapping attributes in the PingFederate documentation.
4. Delete your existing SP connection.
    1. Go to Identity Provider > SP Connections > Manage All.
    2. For your existing connection, click Select action and then click Delete. Click Save.
5. Complete the steps in Deploying the integration files.
6. Complete the steps in Creating a provisioning connection.
    - From Outbound Provisioning > Manage Channels > Channel, on the Attribute Mapping tab, configure the attribute mappings based on your notes.
7. Complete the steps in Creating a single sign-on connection.

Deploying the integration files

To get started with the integration, deploy the Zscaler Internet Access Provisioner files to your PingFederate directory.

Steps

1. Download the Zscaler Internet Access Provisioner `.zip` archive from the Ping Identity Integration Directory.
2. Stop PingFederate.
3. If you're upgrading an existing deployment, delete the `pf-zscaler-zia-quickconnection-<version>.jar` file from your `<pf_install>/pingfederate/server/default/deploy` directory.
4. Extract the `.zip` archive and merge the contents of the `dist` directory with your `<pf_install>/pingfederate/server/default/deploy` directory.
5. Enable the PingFederate provisioning engine:
    1. Open your `<pf_install>/pingfederate/bin/run.properties` file.
    2. Change `pf.provisioner.mode` to `STANDALONE`.
    3. Save the file.

Copyright © 2025 Ping Identity Corporation

> **Note**
>
> To configure the `FAILOVER` mode instead, see Deploying provisioning failover in the PingFederate Server Clustering Guide.

6. Start PingFederate.
7. If you operate PingFederate in a cluster, repeat steps 2-4 and step 6 for each engine node.

Enabling provisioning and single sign-on in PingFederate

To use PingFederate for provisioning and single sign-on, configure an external datastore and set a SAML entity ID.

About this task

Your external datastore acts as the source of data for provisioning. PingFederate also uses an internal datastore to store the state of synchronization between the source datastore and the target datastore.

For more information, see Datastores and Configuring outbound provisioning settings in the PingFederate documentation.

Enabling provisioning and single sign-on in PingFederate 10.1 or later

Steps

1. In the PingFederate administrative console, configure the datastore for PingFederate to use as the source of user data.

   For help, see Datastores in the PingFederate documentation.
2. Go to System > Server > Protocol Settings > Federation Info.
3. In the SAML 2.0 Entity ID field, enter a name for PingFederate to use when identifying itself to SAML partners.
4. On the Outbound Provisioning tab, in the Provisioner Data Store list, select the internal database that will store the synchronization state. Click Save.

   For more information, see Configuring outbound provisioning settings in the PingFederate documentation.

Enabling provisioning and single sign-on in PingFederate 10.0 or earlier

Steps

1. In the PingFederate administrative console, configure the datastore for PingFederate to use as the source of user data.

   For help, see Datastores in the PingFederate documentation.
2. Go to System > Protocol Settings > Roles & Protocols.
3. Select the Enable Identity Provider IdP Role and Support the Following check box.
4. Select the SAML 2.0 and Outbound Provisioning check boxes. Click Next.

Copyright © 2025 Ping Identity Corporation

5. Click the Federation Info tab.
6. In the SAML 2.0 Entity ID field, enter a name for PingFederate to use to identify itself to SAML partners.
7. On the Outbound Provisioning tab, in the Provisioner Data Store list, select the internal database that will store the synchronization state. Click Save.

   For more information, see Configuring outbound provisioning settings in the PingFederate documentation.

Getting SAML details from Zscaler

To simplify the PingFederate configuration process, download your Zscaler Internet Access SAML metadata XML file.

About this task

For more information about configuring SAML, see About SAML and Configuring SAML in the Zscaler Internet Access documentation.

Steps

1. Sign on to Zscaler Internet Access as an administrator.
2. Go to Administration > Authentication > Authentication Settings.
3. Click Configure SAML.
4. In the SP's Metadata section, click Download.
5. Save `zscaler-metadata.xml`.

Configuring provisioning

To enable user provisioning, configure System for Cross-domain Identity Management (SCIM) in Zscaler Internet Access, and then create a connection in PingFederate.

Steps

1. Complete the steps in Getting a base URL and bearer token from Zscaler.
2. Complete the steps in Creating a provisioning connection.

Getting a base URL and bearer token from Zscaler

To enable provisioning to Zscaler Internet Access, enable the System for Cross-domain Identity Management (SCIM) feature and note your configuration information.

About this task

Learn more in About SCIM and Configuring SCIM in the Zscaler Internet Access documentation.

Steps

1. Sign on to Zscaler Internet Access as an administrator.

Copyright © 2025 Ping Identity Corporation

2. Go to Administration > Settings > Company Profile and note your Company ID.

   You'll use this as your organization ID in step 5.

3. Go to Administration > Authentication > Authentication Settings.
4. On the Authentication Profile tab, in the Authentication Type section, select SAML. Click Configure SAML.
5. Configure SAML and enable provisioning:
   1. In the Configure SAML window, select Enable SCIM-Based Provisioning.
   2. In the Base URL field, enter `https://scim.<Zscaler_cloud_name>.net/<organization_ID>/scim`.

      You can find the cloud name in the URL that you use to sign on to the Zscaler Internet Access service. For example, if you sign on to admin.zscalerbeta.net, then your cloud name is zscalerbeta.net.

   3. Note the Bearer Token. Click Save.

      You'll use the Bearer Token value in Creating a provisioning connection.

6. Click Save.
7. Activate the change as shown in Saving and Activating Changes in the Admin Portal in the Zscaler Internet Access documentation.

Creating a provisioning connection

To allow PingFederate to manage users in Zscaler Internet Access, create a service provider (SP) connection.

Steps

1. In the PingFederate administrator console, create a new SP connection:
   - For PingFederate 10.1 or later: go to Applications > Integration > SP Connections. Click Create Connection.
   - For PingFederate 10.0 or earlier: go to Identity Provider > SP Connections. Click Create Connection.
2. Configure the basic connection details with the Zscaler Internet Access quick connection template:
   1. On the Connection Template tab, select Use a template for this connection.
   2. In the Connection Template list, select Zscaler ZIA Provisioner.
   3. In the Metadata File row, upload the `zscaler-metadata.xml` file that you saved in Getting SAML details from Zscaler. Click Next.
   4. On the Connection Type tab, select only Outbound Provisioning. Click Next.
   5. On the General Info tab, in the Connection Name field, enter a name for the connection. Click Next.
3. On the Outbound Provisioning tab, configure provisioning, as shown in Configuring outbound provisioning in the PingFederate documentation, with the following details:
   1. On the Target page, enter the Base URL and Bearer Token values that you noted in Getting a base URL and bearer token from Zscaler.

Copyright © 2025 Ping Identity Corporation

> **Note**
>
> PingFederate verifies the access token when you activate the channel and SP connection.

2. Optional: In the Provisioning Options section, customize the provisioning connector actions as shown in Provisioning options reference. Click Next.
3. On the Manage Channels page, create a channel as shown in Managing channels in the PingFederate documentation. Click Done.

> **Note**
>
> Learn more about the attributes available for your channel configuration in Supported attributes reference.

4. On the Outbound Provisioning tab, click Next.

4. On the Activation and Summary tab, above the Summary section, click the toggle to turn on the connection. Click Save.

Provisioning options reference

The following are setting descriptions for the Zscaler Internet Access provisioning capabilities.

Field Name	Description
User Create	*Selected (default)* PingFederate creates users in Zscaler Internet Access. *Cleared* PingFederate does not create users in Zscaler Internet Access.
User Update	*Selected (default)* PingFederate updates existing users in Zscaler Internet Access. *Cleared* PingFederate does not update existing users in Zscaler Internet Access.
User Delete	*Selected (default)* PingFederate removes users from Zscaler Internet Access. *Cleared* PingFederate does not remove users from Zscaler Internet Access.

Copyright © 2025 Ping Identity Corporation

> **Note**
>
> If any of the above options are cleared, PingFederate logs a warning in the user workflow section of the `provisioner.log` file when the related action fails.

Supported attributes reference

The following attributes can be mapped for user provisioning to Zscaler Internet Access.

For more information on available attributes, see the Attribute Mapping section of About SCIM in the Zscaler Internet Access documentation.

Attribute	Description
Username	The user's unique identifier in Zscaler Internet Access. The expected format is `user@domain`. This attribute is required.
Display Name	The user's display name. This attribute is required.
Department	The user's department or work group, such as "Sales". If no department is specified, Zscaler sets a default value of `Service Admin`.
Email	The user's email address.
External ID	A string identifier assigned to a person by the provisioning client.
First Name	The user's given name. For example, "Barbara" in "Ms. Barabara Jane Jensen, III".
Last Name	The user's family name. For example, "Jensen" in "Ms. Barabara Jane Jensen, III".

Configuring single sign-on

To allow PingFederate to act as an identity provider (IdP) for Zscaler Internet Access, configure single sign-on (SSO), exchange signing certificates, and create a connection.

About this task

> **Note**
>
> SSO is an optional part of this integration. If you only want user provisioning, skip these steps.

Steps

1. Complete the steps in Creating a single sign-on connection.

Copyright © 2025 Ping Identity Corporation

2. Ensure that your traffic is being forwarded to Zscaler Internet Access.

   Learn more in Choosing Traffic Forwarding Methods in the Zscaler Internet Access documentation.

3. Complete the steps in Registering PingFederate as an identity provider in Zscaler.

Creating a single sign-on connection

To allow PingFederate to handle single sign-on (SSO) to Zscaler Internet Access, create a service provider (SP) connection.

About this task

> **Note**
>
> You can follow these steps to create a new connection, or you can modify your provisioning connection.

Steps

1. In the PingFederate administrator console, create a new SP connection:
   - For PingFederate 10.1 or later: go to Applications > Integration > SP Connections. Click Create Connection.
   - For PingFederate 10.0 or earlier: go to Identity Provider > SP Connections. Click Create Connection.
2. Configure the basic connection details with the Zscaler Internet Access quick connection template:
   1. On the Connection Template tab, select Use a template for this connection.
   2. In the Connection Template list, select Zscaler ZIA Provisioner.
   3. In the Metadata File row, upload the `zscaler-metadata.xml` file that you saved in Getting SAML details from Zscaler. Click Next.
   4. On the Connection Type tab, select Browser SSO Profiles. Click Next.
   5. On the General Info tab, in the Connection Name field, enter a name for the connection. Click Next.
3. On the Browser SSO tab, configure SSO as shown in Configuring IdP Browser SSO in the PingFederate documentation, with the following details:
   1. On the Browser SSO > SAML Profiles tab, select only IdP-Initiated SSO and SP-Initiated SSO.

      > **Important**
      >
      > It is recommended to leverage SP-initiated SSO because IdP-initiated SSO is not commonly used.
      > Learn more in IdP-Initiated SAML in the Zscaler Internet Access documentation and Setting Assertion Consumer Service URLs (SAML) in the PingFederate documentation.
      > If you want to use both IdP-initiated SSO and SP-initiated SSO, both endpoints are accessible using the `ACSIdx` parameter.
      > Learn more in IdP endpoints in the PingFederate documentation.

   2. On the Browser SSO > Protocol Settings > Allowable SAML Bindings tab, select only POST.
   3. On the Browser SSO > Protocol Settings > Signature Policy tab, select Always sign assertion.

Copyright © 2025 Ping Identity Corporation

4. On the Credentials tab, configure the connection credentials as shown in Configuring credentials in the PingFederate documentation. Click Next.
5. On the Activation and Summary tab, above the Summary section, click the toggle to turn on the connection. Click Save.

Registering PingFederate as an identity provider in Zscaler

Export your PingFederate signing certificate and use it to configure SAML in Zscaler Internet Access.

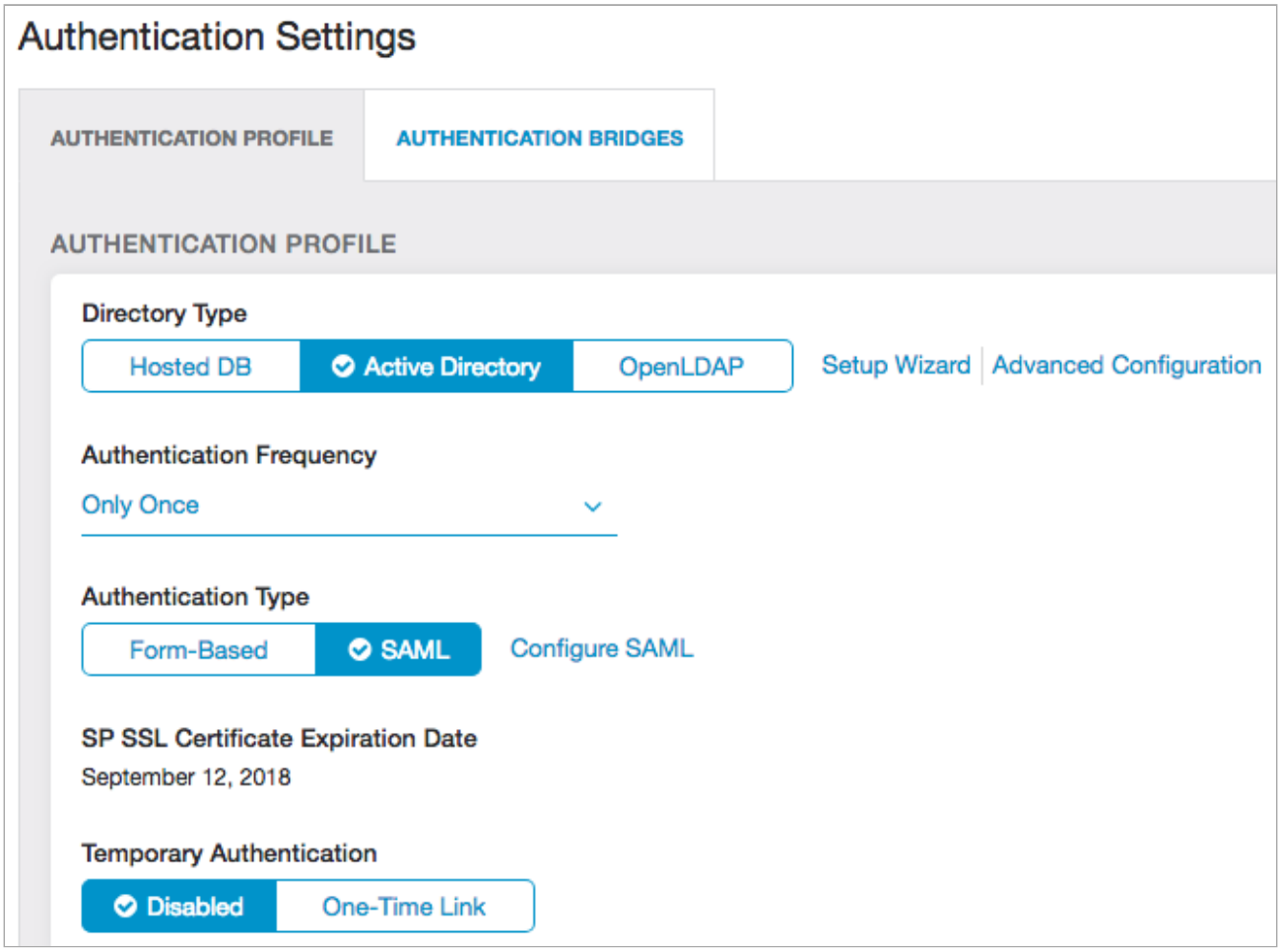
About this task

For more information on setting up single sign-on (SSO), see Configuring SAML in the Zscaler Internet Access documentation.

Steps

1. In PingFederate, export your signing certificate:
    1. Go to Security > Signing & Decryption Keys & Certificates.
    2. For the certificate that you want to use, in the Action column, click Export.
    3. On the Export Certificate tab, click Next.
    4. On the Export & Summary tab, click Export.
    5. Open the `.crt` file in a text editor and copy the contents.
    6. Rename the file extension to `.pem`.
2. In Zscaler Internet Access, configure SAML:
    1. Go to Administration > Authentication > Authentication Settings.
    2. On the Authentication Profile tab, in the Authentication Type section, select SAML. Click Configure SAML.

Copyright © 2025 Ping Identity Corporation



3. In the SAML Portal URL field, enter your PingFederate SSO endpoint. For example:

```
https://<pf_host>:<pf_port>/idp/SSO.saml2
```

4. In the Login Name Attribute field, enter the LDAP attribute that maps to the login name that users enter when they authentication with Zscaler Internet Access, such as `NameID`.
5. In the Public SSL Certificate section, click Upload. Click Choose File, select the `.pem` file that you exported from PingFederate, and then click Upload.
6. Click Save to exit the window.
7. Click Save and activate the change as shown in Saving and Activating Changes in the Admin Portal in the Zscaler Internet Access documentation.

Release notes

Changelog

The following is the change history for the Zscaler Internet Access Provisioner.

Copyright © 2025 Ping Identity Corporation

Zscaler Internet Access Provisioner 1.1.1 – October 2023

- Fixed an issue that caused null values or missing members in `PATCH` operations for group updates.
- Improved efficiency of `PATCH` operations for group updates.
- Improved volume of log entries for group create and update operations to reduce log density.

Zscaler Internet Access Provisioner 1.1 – February 2020

- Renamed the integration to "Zscaler Internet Access Provisioner" to match official branding.
- Added the ability to update the `username` attribute in Zscaler Internet Access.
- Improved error handling and reporting when encountering a user that does not have an ID.
- Improved provisioner efficiency by adding support for Patch operations for group updates.
- Fixed an issue that prevented synchronization of groups with certain special characters in the name.
- Fixed an issue that caused groups to be serialized incorrectly.

Zscaler Internet Access Provisioner 1.0.2 – August 2019

- Fixed an issue that prevented synchronization of groups with certain special characters in the name.

Zscaler Internet Access Provisioner 1.0.1 – July 2018

- Fixed compatibility issue when other System for Cross-domain Identity Management (SCIM)-based connectors exist along side Zscaler provisioning connections

Zscaler Internet Access Provisioner 1.0 – June 2018

- Initial release.
- Added support for user and group life cycle management (including creates, updates, and deletes).
- Added support for adding users to groups.
- Added configuration options for workflow capabilities (for example, the ability to disable updates).

Known issues and limitations

The following are known issues or limitations for the Zscaler Internet Access Provisioner.

Known issues

There are no known issues.

Copyright © 2025 Ping Identity Corporation

Known limitations

- When an LDAP user is deleted in a targeted group distinguished name (DN), the provisioning connector does not propagate the deletion until a new user is added to the group. This limitation is compounded when the User Create provisioning option is disabled. Learn more about solutions in SaaS provisioner does not remove the user in the Knowledge Base.
- Because of PingFederate limitations, user attributes cannot be cleared once set.
- Because of PingFederate limitations, the group name is the only supported group attribute and must be the Common Name (CN) of the LDAP group being provisioned. Additional group attributes are not supported in PingFederate.
- Performance:
    - Zscaler Internet Access System for Cross-domain Identity Management (SCIM) servers have a rate limit of 5 requests per second. In order to avoid retries, reduce the number of threads in the PingFederate channel configuration. For more information, see Specifying channel information in the PingFederate documentation and SCIM API Examples in the Zscaler Internet Access documentation.
    - When provisioning users to a group, all users must be set to active. If an inactive user is present, the active users and group will be provisioned but no users will be added to the group.
    - Rate limiting can prevent the connector from provisioning users to groups. If rate limiting forces a retry, and a user is created on the retry, the user's group memberships are applied the next time the relevant groups are updated.

Download manifest

The following files are included in the Zscaler Internet Access Provisioner `.zip` archive:

- `legal` – a directory that contains the legal document.
    - `Legal.pdf` – copyright and license information.
- `dist` – contains the integration files.
    - `pf-zscaler-zia-quickconnection-<version>.jar` – The Zscaler Internet Access Provisioner quick connection template.

Zscaler Private Access Provisioner

The Zscaler Private Access Provisioner allows PingFederate to integrate with Zscaler Private Access for user and group provisioning and single sign-on (SSO).

Features

- Manages users and groups in Zscaler Private Access based on changes in an external data store that is attached to PingFederate.
    - Creates, updates, and deletes users.

Copyright © 2025 Ping Identity Corporation

- Allows you to enable the create, update, and delete capabilities independently.
- Create groups and update group memberships.

- Browser-based single sign-on (SSO) initiated by the service provider (SP) or identity provider (IdP).
- Pre-populates some connection settings with the included quick connection template.

Intended audience

This document is intended for PingFederate administrators working with the Zscaler Private Access Provisioner.

> **Note**
>
> If you use Zscaler Internet Access, see the Zscaler Internet Access Provisioner documentation.

Before you start, you should be familiar with the following:

- The following sections of the Zscaler documentation:
  - About IdP Configuration
  - Configuring an IdP for Single Sign-On
  - About SAML Attributes
  - About SCIM
  - Enabling SCIM for Identity Management
- The following sections of the PingFederate documentation:
  - Identity provider SSO configuration
  - Datastores
  - Managing IdP adapters
  - Managing digital signing certificates and decryption keys
  - SP connection management
  - Configuring outbound provisioning
  - Configuring outbound provisioning settings

System requirements

- PingFederate 9.0 or later.
- A Zscaler Private Access administrator account.
- To allow PingFederate to make outbound connections to the Zscaler API, you may need to whitelist the following domain in your firewall.
  - https://scim.*your_Zscaler_domain*.net

Copyright © 2025 Ping Identity Corporation

User and group management

The Zscaler Private Access Provisioner links users and groups from the data store to Zscaler. The behavior of each provisioning capability is described below.

Synchronizing existing users

By default, the provisioning connector synchronizes users from the data store to Zscaler by matching the `mail` attribute in the data store to the `Username` attribute in Zscaler. You can change the default mapping in Creating a provisioning connection.

For example:

- In Zscaler, Janet's `Username` is `jsmith@example.com`.
- In your data store, Janet's `mail` is `jsmith@example.com`.
- On the Attribute Mapping screen of your PingFederate channel configuration, you map the `Username` attribute to `mail`.
- When the provisioning connector runs, the data store user is provisioned with a `Username` of `jsmith@example.com`. That matches Janet's existing `Username` in Zscaler, so her information in the data store is synchronized to her Zscaler account.

User provisioning

Triggered when a user is added to the data store group or filter that is targeted by the provisioning connector.

The target is determined by the Source Location screen in the provisioning connector configuration.

User updates

Triggered when a change occurs to a user attribute that is mapped in the provisioning connector configuration.

User deprovisioning

Triggered by any of the following:

- A user is deleted from the user store.
- A user is disabled in the user store.
- A user is removed from the data store group or filter that is targeted by the provisioning connector.

The Remove User Action setting in the connection configuration determines whether the deprovisioning action disables or deletes the user.

Synchronizing existing groups

The provisioning connector synchronizes groups from the data store to Zscaler based on the group name.

For example:

- In Zscaler Private Access, there is a group is named `Accounting`.

Copyright © 2025 Ping Identity Corporation

- In your data store, there is a group with a `CN` of `Accounting`.
- When the provisioning connector runs, the two groups are synchronized.

Group provisioning

Triggered when a group is added to the data store filter that is targeted by the provisioning connector.

The target is determined by the Source Location screen in the provisioning connector configuration.

Group name updates

Renaming the group in the data store will trigger PingFederate to rename the group in Zscaler.

Group membership updates

Changing group memberships through the group's properties or a user's properties will trigger PingFederate to update the group membership in Zscaler.

Group memberships in the data store overwrite the group memberships in Zscaler.

Group deletion

Triggered by any of the following:

- Deleting the group in the data store will trigger PingFederate to delete the group in Zscaler Private Access. Group deletions are permanent and cannot be undone.
- The group is removed from the data store group or filter that is targeted by the provisioning connector.

Setup

Upgrading an existing deployment

If you're upgrading from a previous version of the Zscaler Private Access Provisioner, note your existing service provider (SP) connection configuration and create a new connection.

Upgrading an existing deployment in PingFederate 10.1 or later

Steps

1. Back up your current PingFederate configuration as shown in Configuration archive in the PingFederate documentation.
2. In the PingFederate administrative console, go to Applications > Integration > SP Connections and select your connection.
3. Note the attribute mappings for your existing SP connection.

   For help, see Mapping attributes in the PingFederate documentation.

Copyright © 2025 Ping Identity Corporation

4. Delete your existing SP connection.
    1. Go to Applications > Integration > SP Connections.
    2. For your existing connection, click Select action, and then click Delete. Click Confirm.
5. Complete the steps in Deploying the integration files.
6. Complete the steps in Creating a provisioning connection.
    - From Outbound Provisioning > Manage Channels > Channel, on the Attribute Mapping tab, configure the attribute mappings based on your notes.
7. Complete the steps in Creating a single sign-on connection.

Upgrading an existing deployment in PingFederate 10.0 or earlier

Steps

1. Back up your current PingFederate configuration as shown in Configuration archive in the PingFederate documentation.
2. In the PingFederate administrative console, go to Identity Provider > SP Connections and select your connection.
3. Note the attribute mappings for your existing SP connection.

   For help, see Mapping attributes in the PingFederate documentation.
4. Delete your existing SP connection.
    1. Go to Identity Provider > SP Connections > Manage All.
    2. For your existing connection, click Select action, and then click Delete. Click Save.
5. Complete the steps in Deploying the integration files.
6. Complete the steps in Creating a provisioning connection.
    - From Outbound Provisioning > Manage Channels > Channel, on the Attribute Mapping tab, configure the attribute mappings based on your notes.
7. Complete the steps in Creating a single sign-on connection.

Deploying the integration files

To get started with the integration, deploy the Zscaler Private Access Provisioner files to your PingFederate directory.

Steps

1. Download the Zscaler Private Access Provisioner `.zip` archive from the Ping Identity Integration Directory.
2. Stop PingFederate.
3. If you're upgrading an existing deployment, delete the `pf-zscaler-zpa-quickconnection-<version>.jar` file from your `<pf_install>/pingfederate/server/default/deploy` directory.

Copyright © 2025 Ping Identity Corporation

4. Extract the `.zip` archive and merge the contents of the `dist` directory with your `<pf_install>/pingfederate/server/default/deploy` directory.
5. Enable the PingFederate provisioning engine:
    1. Open your `<pf_install>/pingfederate/bin/run.properties` file.
    2. Change `pf.provisioner.mode` to `STANDALONE`.
    3. Save the file.

    > **Note**
    >
    > To configure the `FAILOVER` mode instead, see Deploying provisioning failover in the PingFederate Server Clustering Guide.

6. Start PingFederate.
7. If you operate PingFederate in a cluster, repeat steps 2-4 and step 6 for each engine node.

Enabling single sign-on in PingFederate

Before you can configure single sign-on in Zscaler Private Access, you need to set a SAML entity ID in PingFederate.

Enabling single sign-on in PingFederate 10.1 or later

Steps

1. On the PingFederate administrative console, go to System > Protocol Settings > Federation Info.
2. In the SAML 2.0 Entity ID field, enter a name for PingFederate to use to identify itself to SAML partners.
3. Click Save.

Enabling single sign-on in PingFederate 10.0 or earlier

Steps

1. On the PingFederate administrative console, go to System > Protocol Settings > Roles & Protocols.
2. Select the Enable Identity Provider IdP Role and Support the Following check box.
3. Select the SAML 2.0 check box. Click Next.
4. Go to the Federation Info tab.
5. In the SAML 2.0 Entity ID field, enter a name for PingFederate to use to identify itself to SAML partners.
6. Click Save.

Exporting SAML metadata from PingFederate

Export a metadata file that describes your PingFederate identity provider configuration.

Copyright © 2025 Ping Identity Corporation

About this task

For general information about these steps, see Metadata export in the PingFederate documentation.

Steps

1. In the PingFederate administrative console, go to the Metadata Export window..

   Choose from:

   - For PingFederate 10.1 or later: go to System > Protocol Metadata > Metadata Export.
   - For PingFederate 10.0 or earlier: go to System > Metadata Export.

2. If you see the Metadata Role tab, select I am the identity provider (IdP). Click Next.
3. On the Metadata Mode tab, select Select information to include in metadata manually. Click Next.
4. On the Protocol tab, click Next.
5. On the Attribute Contract tab, click Next.
6. On the Signing Key tab, select a signing certificate. Click Next.
7. Optional: On the Metadata Signing tab, select a certificate to sign the metadata XML file. Click Next.
8. On the XML Encryption Certificate tab, select the certificate that you want to use to encrypt the XML content. Click Next.
9. On the Export & Summary tab, click Export.
10. Save `metadata.xml`.
11. Click Done.

Enabling provisioning and single sign-on in Zscaler

Register PingFederate as an identity provider in Zscaler and download the SAML metadata information.

About this task

For more information about configuring Zscaler, see Configuring an IdP for Single Sign-On and Enabling SCIM for Identity Management in the Zscaler Private Access documentation.

Steps

1. Sign on to Zscaler Private Access as an administrator.
2. On the Administration > Authentication > Settings page, click Add IdP Configuration.
3. On the Add IdP Configuration modal, on the IdP Information tab, complete the basic information. Click Next.

> **Note**
>
> If you cannot select an authentication domain, contact Zscaler. For more information, see Configuring Authentication Settings in the Zscaler Private Access documentation.

Copyright © 2025 Ping Identity Corporation

4. On the SP Metadata tab, click Download Metadata. Save the file as `sp_metadata.xml`.
5. Click Download Certificate. You will use this in Creating a single sign-on connection. Click Next.
6. On the Create IdP tab, complete the information from PingFederate.
    1. For the IdP Metadata File, upload the `metadata.xml` file that you exported in Exporting SAML metadata from PingFederate.
    2. For the IdP Certificate, upload your PingFederate signing certificate. For instructions, see Managing digital signing certificates and decryption keys in the PingFederate documentation.
    3. In the Single Sign-On URL field, enter your PingFederate single sign-on endpoint based on the following.

        ```
        https://pf_host:pf_port/idp/SSO.saml2
        ```

    4. In the IdP Entity ID field, enter the SAML 2.0 Entity ID that you created in Enabling single sign-on in PingFederate.
7. In the SCIM section, configure SCIM provisioning. Click Save.
    1. For SCIM Sync, click Enable.
    2. Note the SCIM Service Provider Endpoint and Bearer Token. You will use these in Creating a provisioning connection.

Creating a provisioning connection

To allow PingFederate to manage users in Zscaler Private Access, create a service provider (SP) connection.

About this task

> **Note**
>
> You can follow these steps to create a new SP connection, or you can modify your provisioning connection.

Steps

1. In the PingFederate administrator console, configure the data store that PingFederate will use as the source of user data. For instructions, see Datastores in the PingFederate documentation.
    - When targeting users and groups for provisioning, exclude the user account that you will use to administer users in your connection to Zscaler Private Access. This prevents the PingFederate provisioning engine from interfering with the account that provisions users and groups.
2. Enable provisioning.
    1. On the System > Protocol Settings > Roles & Protocols screen, select Enable Identity Provider IdP Role and Support the Following.
    2. Select Outbound Provisioning. Click Save.

Copyright © 2025 Ping Identity Corporation

3. On the Identity Provider screen, in the SP Connections area, open an existing connection or create a new one as follows:
    1. Click Create new.
    2. On the Connection Template screen, select Use a template for this connection.
    3. In the Connection Template list, select Zscaler ZPA Connector.
    4. Click Choose File, select the `sp_metadata.xml` file that you downloaded in Enabling provisioning and single sign-on in Zscaler, and then click Open. Click Next.
4. On the Connection Type screen, select Outbound Provisioning and clear any unwanted types. Click Next.
5. On the General Info screen, the basic connection information is populated by the metadata XML file. Click Next.
6. On the Outbound Provisioning screen, configure the provisioning target and channel as shown in Configuring outbound provisioning in the PingFederate documentation.
    1. Click Configure Provisioning.
    2. On the Target screen, in the Base URL field, enter the SCIM Service Provider Endpoint that you noted in Enabling provisioning and single sign-on in Zscaler.
    3. On the Target screen, enter the Bearer Token that you noted in Enabling provisioning and single sign-on in Zscaler.

        > **Note**
        >
        > PingFederate verifies the access token when you activate the channel and SP connection.

    4. Under Provisioning Options, customize the provisioning connector actions as shown in Provisioning options reference. Click Next.
    5. On the Manage Channels screen, create a channel as shown in Managing channels in the PingFederate documentation. Click Done.

        > **Note**
        >
        > For more information about the attributes available in your channel configuration, see Supported attributes reference.

    6. On the Outbound Provisioning screen, click Next.
7. On the Activation and Summary screen, above the Summary section, turn on the connection. Click Save.

Provisioning options reference

Lists the main provisioning capabilities available in the Zscaler connection configuration.

Copyright © 2025 Ping Identity Corporation

Field Name	Description
User Create	Selected (default) – PingFederate creates users in Zscaler. Cleared – PingFederate does not create users in Zscaler.
User Update	Selected (default) – PingFederate updates existing users in Zscaler. Cleared – PingFederate does not update existing users in Zscaler.
User Disable / User Delete	Selected (default) – PingFederate disables or deletes users in Zscaler according to the Remove User Action setting. Cleared – PingFederate does not disable or delete users in Zscaler.

> **Note**
> If any of the above options are cleared, PingFederate logs a warning in the user workflow section of `provisioner.log` when the related action fails.

Remove User Action	This option applies when User Disable / User Delete is selected and either of the following is true: • a previously provisioned user no longer meets the condition set on the Source Location tab. • a user has been disabled or deleted from the data store. Disable (default) – PingFederate disables the user in Zscaler Private Access. Delete – PingFederate deletes the user from Zscaler Private Access.

Supported attributes reference

Lists the attributes that can be mapped for user provisioning to Zscaler.

Attribute	Description
Username	The user's unique identifier in Zscaler. The expected format is `user@domain`. This attribute is required.
Email	The user's email address.
Display Name	The user's display name. For example, "Barb Jensen".
Nickname	The user's nickname. For example, "Barb" instead of "Barbara".
First Name	The user's first name. For example, "Barbara" in "Ms. Barabara Jane Jensen, III".
Last Name	The user's last name. For example, "Jensen" in "Ms. Barabara Jane Jensen, III".
Formatted Name	The user's complete name, including all middle names, titles, and suffises. For example, "Ms. Barbara Jane Jenson, III".

Copyright © 2025 Ping Identity Corporation

Attribute	Description
Department	The user's department or work group, such as "Sales". If no department is specified, Zscaler sets a default value of `Service Admin`.
Division	The user's division in the organization.
Organization	The user's organization.
Cost Center	The user's cost center location.
User type	The type of user in the organization. For example, "Employee", "Contractor", "Intern", or "Temp".
External ID	A string identifier assigned to a person by the provisioning client.

Creating a single sign-on connection

To allow PingFederate to handle single sign-on (SSO) to Zscaler Private Access, create a service provider (SP) connection.

About this task

> **Note**
>
> You can follow these steps to create a new SP connection, or you can modify your provisioning connection.

Steps

1. In the PingFederate administrator console, configure the data store that PingFederate will use as the source of user data. For instructions, see Datastores in the PingFederate documentation.
2. On the Identity Provider tab, in the SP Connections area, open an existing connection or create a new one as follows:
    1. Click Create new.
    2. On the Connection Template tab, select Use a template for this connection.
    3. In the Connection Template list, select Zscaler ZPA Connector.
    4. Click Choose File, select the `sp_metadata.xml` file that you downloaded in Enabling provisioning and single sign-on in Zscaler, and then click Open. Click Next.
3. On the Connection Type tab, select Browser SSO Profiles and clear any unwanted types. Click Next.
4. On the General Info tab, the basic connection information is populated by the metadata XML file. Click Next.
5. On the Browser SSO tab, configure browser SSO.

    For a complete guide, see Configuring IdP Browser SSO in the PingFederate documentation.
    1. On the Browser SSO > SAML Profiles tab, select only IdP-Initiated SSO and SP-Initiated SSO.
    2. On the Browser SSO > Protocol Settings > Allowable SAML Bindings tab, select only POST.

Copyright © 2025 Ping Identity Corporation

3. On the Browser SSO > Protocol Settings > Signature Policy tab, select Always sign assertion.

6. On the Credentials tab, configure the connection credentials. Click Next.

   For a complete guide, see Configuring credentials in the PingFederate documentation.

   - On the Credentials > Signature Verification Settings > Signature Verification Certificate tab, click Manage Certificates and import the certificate that you downloaded in Enabling provisioning and single sign-on in Zscaler.

7. On the Activation and Summary tab, above the Summary section, turn on the connection. Click Save.

Release notes

Changelog

Lists the changes made in each version of the Zscaler Private Access Connector.

Zscaler Private Access Provisioner 1.0.1 – October 2023

- Fixed an issue that caused null values or missing members in `PATCH` operations for group updates.
- Improved efficiency of `PATCH` operations for group updates.
- Improved volume of log entries for group create and update operations to reduce log density.

Zscaler Private Access Provisioner 1.0 – April 2020

- Initial release.
- Added support for user and group provisioning.
- Added support for ZPA SCIM attributes.
- Added configuration options for create, update, and disable/delete capabilities.
- Added configuration options for deprovisioning actions

Known issues and limitations

The following are known issues or limitations with the Zscaler Private Access Provisioner.

Known issues

There are no known issues.

Copyright © 2025 Ping Identity Corporation

Known limitations

- When an LDAP user is deleted in a targeted group distinguished name (DN), the provisioning connector does not propagate the deletion until a new user is added to the group. This limitation is compounded when the User Create provisioning option is disabled. For solutions, see SaaS provisioner does not remove the user in the Knowledge Base.
- Due to PingFederate limitations, user attributes cannot be cleared once set.
- Due to PingFederate limitations, the Group name is the only supported group attribute and must be the Common Name (CN) of the LDAP Group being provisioned. Additional group attributes are not supported in PingFederate.
- Performance
    - Zscaler SCIM servers have a rate limit of 5 requests per second. In order to avoid retries, reduce the number of threads in the PingFederate channel configuration. For more information, see Specifying channel information in the PingFederate documentation and SCIM API Examples in the Zscaler documentation.
    - When provisioning users to a group, all users must be set to active. If an inactive user is present, the active users and group will be provisioned but no users will be added to the group.
    - Rate limiting can prevent the connector from provisioning users to groups. If rate limiting forces a retry, and a user is created on the retry, the user's group memberships are applied the next time the relevant groups are updated.

Download manifest

The following files are included in the Zscaler Private Access Provisioner `.zip` archive:

- `legal` – a directory that contains the legal document.
    - `Legal.pdf` – copyright and license information.
- `dist` – contains the integration files.
    - `pf-zscaler-zpa-quickconnection-<version>.jar` – The Zscaler Private Access Provisioner quick connection template.

Copyright © 2025 Ping Identity Corporation