

Authentication nodes

December 3, 2025



ADVANCED IDENTITY CLOUD

Copyright

All product technical documentation is Copyright © 2010-2025 Ping Identity Corporation
Ping Identity Corporation
1001 17th Street, Suite 100
Denver, CO 80202
U.S.A.

Visit <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

Basic nodes

Data Store Decision node 9

Kerberos node 11

LDAP Decision node 13

Identity Store Decision node 18

Password Collector node 21

Username Collector node. 22

Zero Page Login Collector node

Failure node

Success node

Multi-factor nodes

Combined MFA Registration node

Device Binding node.

Device Binding Storage node.

Device Signing Verifier node

Enable Device Management node

Get Authenticator App node

HOTP Generator node.

MFA Registration Options node

OATH Device Storage node

OATH Registration node

OATH Token Verifier node

OTP Collector Decision node

OTP Email Sender node

OTP SMS Sender node.

Opt-out Multi-Factor Authentication node

Push Registration node

Push Result Verifier node.

Push Sender node

Push Wait node.

Recovery Code Collector Decision node.

Recovery Code Display node

TypingDNA Decision node

TypingDNA Recorder node

TypingDNA Reset Profile node

TypingDNA Short Phrase Collector node

WebAuthn Authentication node

WebAuthn Device Storage node

WebAuthn Registration node.

Risk management nodes

Account Active Decision node

Account Lockout node.

Auth Level Decision node

CAPTCHA node

reCAPTCHA Enterprise node

Legacy CAPTCHA node

Modify Auth Level node.

PingOne Protect Evaluation node

PingOne Protect Initialization node

PingOne Protect Result node.

Behavioral nodes

Increment Login Count node.

Login Count Decision node

Contextual nodes

Certificate Collector node.

Certificate User Extractor node

Certificate Validation node

Cookie Presence Decision node

Device Geofencing node

Device Location Match node

Device Match node

Device Profile Collector node.

Device Profile Save node

Device Tampering Verification node.

Persistent Cookie Decision node.

Set Custom Cookie node

Set Persistent Cookie node

Federation nodes

OIDC ID Token Validator node

SAML2 Authentication node

Social Provider Handler node

Legacy Social Provider Handler node

Write Federation Information node

Identity management nodes

Accept Terms and Conditions node

Attribute Collector node

Attribute Present Decision node.

Attribute Value Decision node

Consent Collector node

Create Object node

Display Username node.

Identify Existing User node

KBA Decision node

KBA Definition node

KBA Verification node

Pass-through Authentication node

Patch Object node

PingOne Create User node

PingOne Delete User node

PingOne Identity Match node

PingOne Verify Completion Decision node

PingOne Verify Evaluation node

Platform Password node

Platform Username node

Profile Completeness Decision node

Query Filter Decision node

Required Attributes Present node

Select Identity Provider node

Terms and Conditions Decision node

Time Since Decision node

Utility nodes

Agent Data Store Decision node

Amster Jwt Decision node

Anonymous Session Upgrade node

Anonymous User Mapping node

Backchannel Initialize node

Backchannel Status node

Choice Collector node

Configuration Provider node

Email Suspend node

Email Template node

Failure URL node

Flow Control node

Get Session Data node

Inner Tree Evaluator node

Message node

Meter node

Page node

Polling Wait node

Query Parameter node

Register Logout Webhook node

Remove Session Properties node

Request Header node

Retry Limit Decision node

Scripted Decision node

Set Error Details node

Set Failure Details node	
Set Session Properties node	
Set State node	
Set Success Details node	
State Metadata node	
Success URL node	
Timer Start node	
Timer Stop node	
Update Journey Timeout node	
Uncategorized nodes	
Identity Assertion node	
SpyCloud Auth Node.	
Thing nodes	
Authenticate Thing node	
Register Thing node	
Marketplace nodes	
PingOne integration	
PingOne Service	
PingOne node (deprecated).	
PingOne Authorize node.	
PingOne Credentials nodes	
PingOne Credentials Delete Wallet node	
PingOne Credentials Find Wallets node	
PingOne Credentials Issue node	
PingOne Credentials Pair Wallet node	
PingOne Credentials Revoke node	
PingOne Credentials Update node	
PingOne Credentials Verification node	
PingOne DaVinci API node.	
PingOne Verify Authentication node (deprecated).	
PingOne Verify Proofing node (deprecated).	
PingOne Verify service (deprecated).	
BioCatch Session node	
BioCatch Session Collector node.	
BioCatch Session Profiler node.	
Duo node (deprecated)	
Duo Universal Prompt node	
Fingerprint nodes	
Setup	
Fingerprint Profiler node	
Fingerprint Response node	
Gateway Communication node	
HTTP Client node.	
IdentityX Auth Request Decision node	

IdentityX Auth Request Initiator node	
IdentityX Check Enrollment Status node	
IdentityX Mobile Auth Request node	
IdentityX Mobile Auth Request Validate node	
IdentityX Sponsor User node	
iProov Authentication node	
Jumio identity verification	
Jumio initiate node	
Jumio Decision node	
Microsoft Intune node.	
LexisNexis One-Time Passcode (OTP)	
LexisNexis OTP Sender node	
LexisNexis OTP Collector node	
LexisNexis OTP Decision node	
OneSpan	
Set up OneSpan.	
OneSpan nodes.	
OneSpan Auth Activate Device node	
OneSpan Auth Add Device node	
OneSpan Auth Check Activation node	
OneSpan Auth Check Session Status node	
OneSpan Auth VDP User Register node	
OneSpan Auth Assign Authenticator node.	
OneSpan Auth Generate VOTP node	
OneSpan Get User Authenticator node	
OneSpan Identity Verification node	
OneSpan Sample journeys	
Onfido Check node	
Onfido Registration node	
RSA SecurID node	
Secret Double Octopus (SDO) nodes	
SpyCloud Auth Node.	
ThreatMetrix Authentication nodes	
ThreatMetrix Profiler node	
ThreatMetrix Session Query node	
ThreatMetrix Review Status node	
ThreatMetrix Reason Code node	
ThreatMetrix Update Review node.	
Twilio Identifier node	
Twilio Verify Collector Decision node	
Twilio Verify Lookup node	
Twilio Verify Sender node.	
TypingDNA	
TypingDNA Decision node.	

TypingDNA Recorder node

TypingDNA Reset Profile node

TypingDNA Short Phrase Collector node

Basic nodes

Data Store Decision node

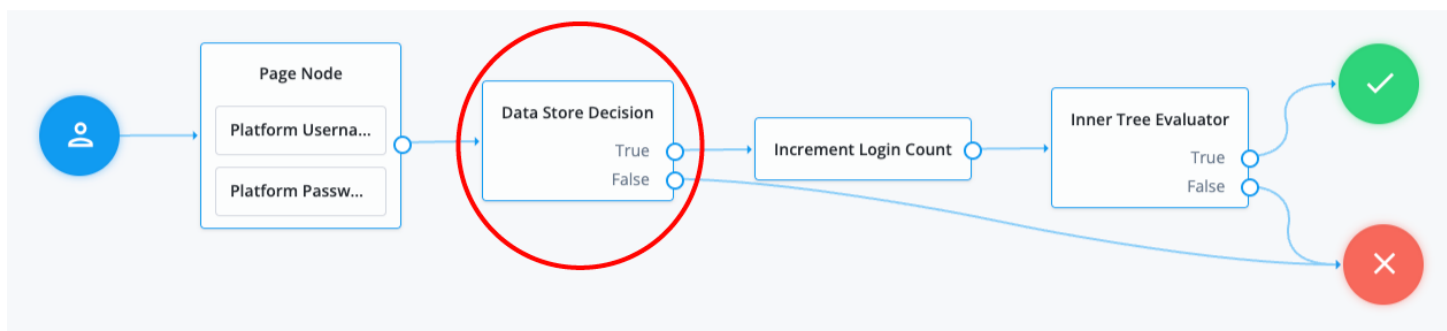
The **Data Store Decision** node checks that the credentials provided during authentication match the ones stored in the configured data store for the realm.

The following are alternate nodes that you can use in your journeys depending on your specific use cases:

- The [LDAP Decision node](#) supports LDAP Behera Password Policies with separate outcomes for accounts that are locked and passwords that have expired.
- The [Identity Store Decision node](#) is an enhanced node with additional outcomes. Use this node if your authentication journey needs more functionality than a simple **True** or **False** outcome.

Examples

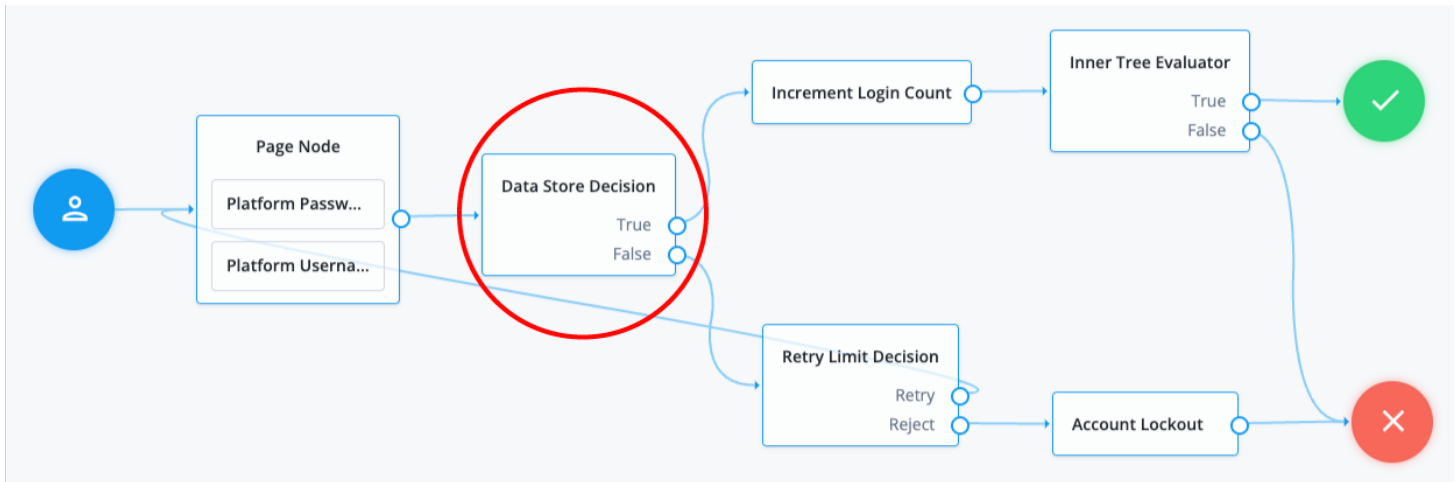
Example 1: Simple username and password collector nodes with Data Store Decision node



This example illustrates a simple login process. The journey involves a **Page node** that contains two embedded nodes: **Platform Username node** and **Platform Password node**. To enhance user experience, the Page node lets users input their username and password on a single page, instead of splitting them across two different pages.

The Data Store Decision node has two outcomes: **True** or **False**. When the outcome is **True**, it triggers a **Login Count Decision node**. The **Increment Login Count node** then moves to an **Inner Tree Evaluator node**, which performs additional login processes. The **False** outcome connects directly to a failure node, indicating a failed state where the username and/or password provided by the user did not match the information stored in the data store.

Example 2: Grant the user several attempts to enter their credentials correctly



In the following example, when an authentication attempt fails at the Data Store Decision node, you can direct it to a [Retry Limit Decision node](#). The [Retry Limit Decision node](#) determines the number of retries allowed and either retries the login attempt or rejects it. If the journey rejects the login attempt after reaching the configured limit, for example three attempts, the operation results in an account lockout.

Availability

Product	Available?
PingOne Advanced Identity Cloud	Yes
PingAM (self-managed)	Yes
Ping Identity Platform (self-managed)	Yes

Inputs

This node requires the `realm`, `username`, and `password` properties in the incoming node state.

Implement a [Platform Username node](#) and a [Platform Password node](#) earlier in the journey.

Alternatively, implement a [Zero Page Login Collector node](#).

Dependencies

The Data Store Decision node is a basic node used in many authentication application types, such as basic, push, OAuth 2.0, and social provider authentication applications.

Configuration

This node has no configurable properties.

Outputs

This node copies shared and transient state into the outgoing node state.

Outcomes

Returns a boolean outcome:

True

The credentials match those found in the data store.

False

The credentials don't match those found in the data store.

Errors

The following Data Store Decision node warnings and errors can appear in the logs:

Warnings

- "invalid password error"
- "invalid username error"

Errors

- "Exception in data store decision node"

Kerberos node

Not supported in Advanced Identity Cloud

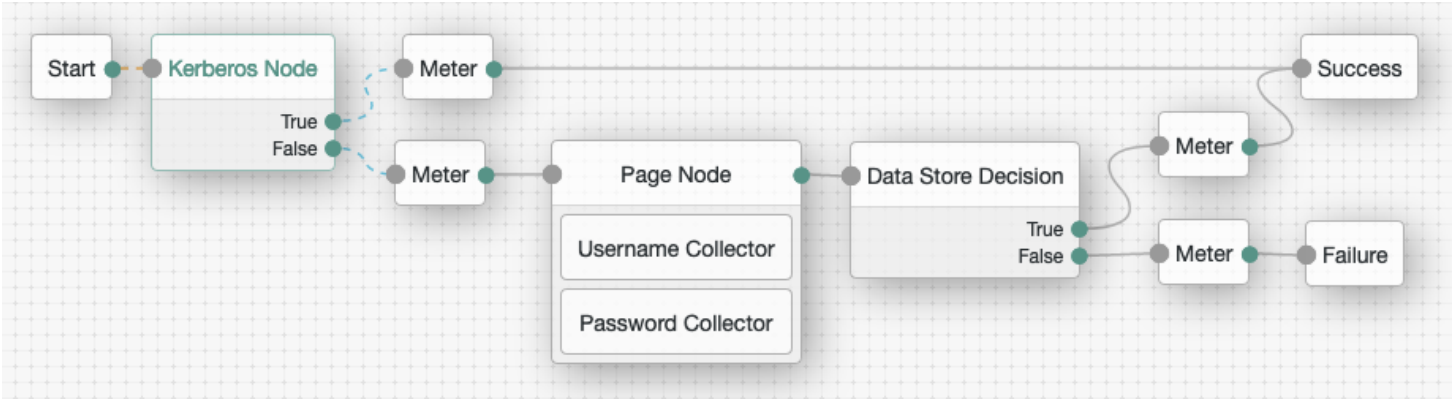
Enables desktop single sign-on such that a user who has already authenticated with a Kerberos Key Distribution Center can authenticate to AM without having to provide the login information again.

To achieve this, the user presents a Kerberos token to AM through the Simple and Protected GSS-API Negotiation Mechanism (SPNEGO) protocol.

End users may need to set up Integrated Windows Authentication in Microsoft Edge to benefit from single sign-on when logged on to a Windows desktop.

Example

This flow attempts to authenticate the user with Windows Desktop SSO. If unsuccessful, AM requests the username and password for login. Meter nodes are used to track metrics for the various paths through the flow:



Availability

Product	Available?
PingOne Advanced Identity Cloud	No
PingAM (self-managed)	Yes
Ping Identity Platform (self-managed)	Yes

Outcomes

- True
- False

Evaluation continues along the **True** path if Windows Desktop SSO is successful; otherwise, evaluation continues along the **False** path.

Configuration

Property	Usage
Service Principal	Specifies the Kerberos principal for authentication in the format <code>HTTP/AM-DOMAIN@AD-DOMAIN</code> , where <i>AM-DOMAIN</i> corresponds to the host and domain names of the AM instance, and <i>AD-DOMAIN</i> is the domain name of the Kerberos realm (the FQDN of the Active Directory domain). <i>AD-DOMAIN</i> can differ from the domain name for AM. In multi-instance AM deployments, configure <i>AM-DOMAIN</i> as the FQDN or IP address of the load balancer in front of the AM instances. For example, <code>HTTP/AM-LB.example.com@KERBEROSREALM.INTERNAL.COM</code> .

Property	Usage
Key Tab File Path	Specifies the full, absolute path of the keytab file for the specified Service Principal.  Tip You generate the keytab file using the Windows <code>ktpass</code> utility. For example: <pre>C:\> ktpass -out fileName.keytab -princ HTTP/ am.example.com@AD_DOMAIN.COM -pass +rdnPass -maxPass 256 -mapuser amKerberos@frdpcloud.com -crypto AES256-SHA1 -ptype KRB5_NT_PRINCIPAL -kvno 0</pre>
Kerberos Realm	Specifies the name of the Kerberos (Active Directory) realm used for authentication. Must be specified in ALL CAPS.
Kerberos Server Name	Specifies the fully qualified domain name, or IP address of the Kerberos (Active Directory) server.
Trusted Kerberos realms	Specifies a list of trusted Kerberos realms for user Kerberos tickets. If realms are configured, then Kerberos tickets are only accepted if the realm part of the user principal name of the user's Kerberos ticket matches a realm from the list. Each trusted Kerberos realm must be specified in all caps.
Return Principal with Domain Name	When enabled, AM returns the fully qualified name of the authenticated user rather than just the username.
Lookup User In Realm	Validates the user against the configured data stores. If the user from the Kerberos token is not found, evaluation continues along the <code>False</code> path. This search uses the <code>Alias Search Attribute Name</code> from the core realm attributes. Find more information about this property in Core authentication attributes > User profile.
Is Initiator	When enabled (<code>true</code>), specifies that the node is using <i>initiator</i> credentials, which is the default. When disabled (<code>false</code>), specifies that the node is using <i>acceptor</i> credentials.

LDAP Decision node

The **LDAP Decision** node verifies that the provided username and password exist in the specified LDAP user data store. The node also checks whether the associated user account has expired or is locked out.

Availability

Product	Available?
PingOne Advanced Identity Cloud	Yes
PingAM (self-managed)	Yes
Ping Identity Platform (self-managed)	Yes

Inputs

The node reads the `username` and `password` fields from the node state.

Implement a [Platform Username node](#) and a [Platform Password node](#) earlier in the journey.

Alternatively, implement a [Zero Page Login Collector node](#).

Dependencies

None.

Configuration

Property	Usage
Primary LDAP Server <i>(required)</i>	Specify one or more primary directory servers. Specify each directory server in the following format: <code>host:port</code> . For example, <code>directory_services.example.com:389</code> .
Secondary LDAP Server	Specify one or more secondary directory servers. Specify each directory server in the following format: <code>host:port</code> . The journey uses the secondary servers when none of the primary servers are available. For example, <code>directory_services_backup.example.com:389</code> .
DN to Start User Search <i>(required)</i>	Specify the DN from which to start the user search. More specific DNs, such as <code>ou=sales,dc=example,dc=com</code> , result in better search performance. If multiple entries with the same attribute values exist in the directory server, make sure this property is specific enough to return only one entry.
Bind User DN, Bind User Password	The credentials used to connect to the LDAP user data store.
Attribute Used to Retrieve User Profile <i>(required)</i>	The attribute used to retrieve a user profile from the directory server. The user search will have already happened, as specified by the Attributes Used to Search for a User to be Authenticated and User Search Filter properties.

Property	Usage
Attributes Used to Search for a User to be Authenticated <i>(required)</i>	The attributes the node uses to match the credentials provided by the user to an entry in the directory server. For example, a value of <code>uid</code> forms the search filter <code>uid=user</code>. If you specify multiple values, such as <code>uid</code> and <code>cn</code>, the node forms a complex search filter <code>((uid=user)(cn=user))</code>. Multiple attribute values let the user authenticate with any one of the values. For example, if you set both <code>uid</code> and <code>mail</code>, then Barbara Jensen can authenticate with either <code>bjensen</code> or <code>bjensen@example.com</code>. Note If you're using account lockout and you set multiple attribute values here, you must add those attributes to the Alias Search Attribute Name property in the User profile. Find more information about this property in Core authentication attributes > User profile.
User Search Filter	A filter to append to user searches. For example, if your search attribute is <code>mail</code> and you set User Search Filter to <code>(objectClass=inetOrgPerson)</code>, the node uses <code>(&(mail=address)(objectClass=inetOrgPerson))</code> as the resulting search filter. In this example, <code>address</code> is the mail address provided by the user.
Search Scope	The extent of the search for users in the directory server: OBJECT: The search extends only to the entry specified by the DN to Start User Search. ONELEVEL: The search extends to the entries that are direct children of the DN to Start User Search. SUBTREE: The search extends to the DN to Start User Search and every entry under it. Default: <code>SUBTREE</code>
LDAP Connection Mode	Specifies whether to use SSL or StartTLS to connect to the directory server. The node must be able to trust the certificates used. Possible values: <code>LDAP</code>, <code>LDAPS</code>, and <code>StartTLS</code> Default: <code>LDAP</code>

Property	Usage
mTLS Enabled	Enables mTLS (mutual TLS) between Advanced Identity Cloud and the directory server. This setting applies to <i>all</i> configured LDAP servers; that is, Advanced Identity Cloud uses mTLS to authenticate to all LDAP servers configured for this node. When mTLS is enabled, Advanced Identity Cloud ignores the values for Bind User DN and Bind User Password. If you enable this property, you must: • Set the LDAP Connection Mode to LDAPS • Provide an mTLS Secret Label Identifier Default: Disabled
mTLS Secret Label Identifier	Identifier used to create a secret label for mapping to the mTLS certificate in the secret store. Advanced Identity Cloud uses this identifier to create a specific secret label for this node. The secret label takes the form <code>am.authentication.nodes.ldap.decision.mtls.identifier.cert</code>, where <code>identifier</code> is the value of mTLS Secret Label Identifier. The identifier can only contain alphanumeric characters (<code>a-z</code>, <code>A-Z</code>, <code>0-9</code>) and periods (<code>.</code>). It can't start or end with a period. All LDAP servers configured for this node share the same secret label. For more security, you should rotate certificates periodically. When you rotate a certificate, update the corresponding mapping in the realm secret store configuration to reflect this label. When you rotate a certificate, Advanced Identity Cloud closes any existing connections using the old certificate. A new connection is selected from the connection pool and no server restart is required.
Return User DN to DataStore	When enabled, the node returns the DN rather than the User ID. From the DN value, Advanced Identity Cloud uses the RDN to search for the user profile. For example, if a returned DN value is <code>uid=demo,ou=people,dc=example,dc=com</code>, Advanced Identity Cloud uses <code>uid=demo</code> to search the directory server. Default: Enabled
User Creation Attributes	This list lets you map (external) attribute names from the LDAP directory server to (internal) attribute names used by Advanced Identity Cloud.
Minimum Password Length	The minimum acceptable password length. Default: <code>8</code>
LDAP Behera Password Policy Support	When enabled, support interoperability with servers that implement the Internet-Draft, Password Policy for LDAP Directories. Default: Enabled
Trust All Server Certificates	When enabled, the server blindly trusts server certificates, including self-signed test certificates. Default: Disabled

Property	Usage
LDAP Connection Heartbeat Interval	Specifies how often Advanced Identity Cloud should send a heartbeat request to the directory server to ensure that the connection doesn't remain idle. Some network administrators configure firewalls and load balancers to drop connections that are idle for too long. Set the units for the interval in the LDAP Connection Heartbeat Time Unit property.  Note Setting this property to 0 does <i>not</i> disable the heartbeat (keepalive) or load balancer availability checks. Default: 10
LDAP Connection Heartbeat Time Unit	The time unit for the LDAP Connection Heartbeat Interval. Default: seconds
LDAP Operations Timeout	The timeout, in seconds, that Advanced Identity Cloud should wait for a response from the directory server. Default: 0 (means no timeout)
Use mixed case for password change messages	Specifies whether the server returns password change messages in mixed (sentence) case or transforms them to uppercase. By default, the server transforms password reset and password change messages to uppercase. Enable this setting to return messages in sentence case. Default: Disabled
LDAP Affinity Level	Level of affinity used to balance requests across LDAP servers. Affinity-based load balancing means that each request for the same user entry goes to the same DS server. The DS server used for a specific operation is determined by the DN of the identity involved. List the directory server instances that form part of the affinity deployment in the Primary LDAP Server and Secondary LDAP Server properties. Options are: • NONE – no affinity • BIND – affinity for BIND requests only • ALL – affinity for all requests Default: NONE

Outcomes

True

The provided credentials match those found in the LDAP user data store.

False

The provided credentials don't match those found in the LDAP user data store.

Locked

The profile associated with the provided credentials is locked.

Cancelled

The user must change their password. When the journey prompts the user to change their password, the user cancels the password change.

Expired

The profile is found, but the password has expired.

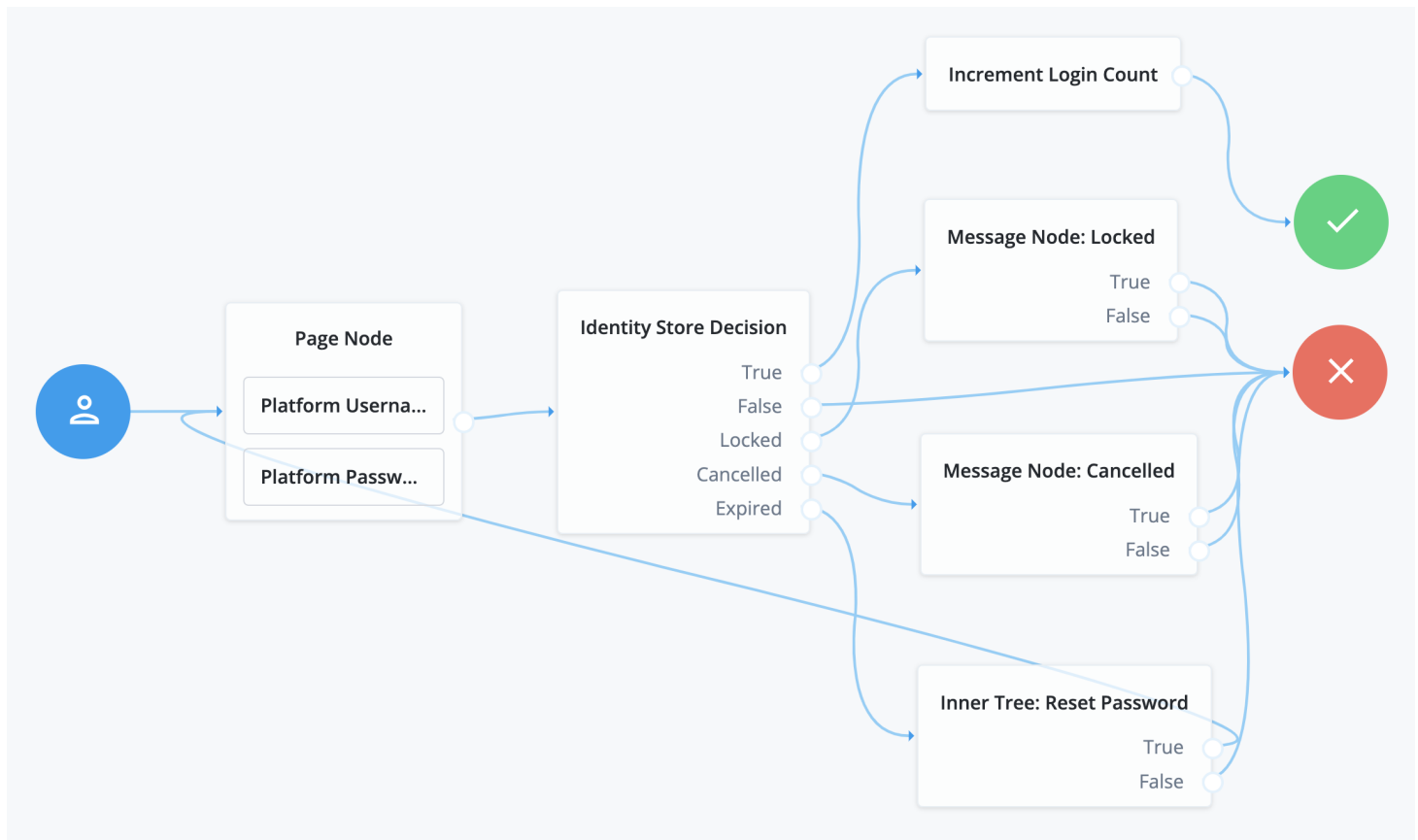
Identity Store Decision node

The **Identity Store Decision** node attempts to match the provided username and password with the credentials stored in the identity store.

If the credentials exist, the node checks the following:

- Is the profile locked?
- Has the provided password expired?
- Has the user cancelled a password reset?

Example



This example illustrates a simple login process.

- A **Page node** with the embedded nodes (**Platform Username node** and **Platform Password node**) prompts the user for their credentials.
- The Identity Store Decision node assesses the credentials:
 - If it finds the credentials in the data store and the credentials are valid, the journey follows the **True** outcome. An **Increment Login Count node** increments the login count and the user is authenticated.
 - If the credentials don't exist in the data store, the journey follows the **False** outcome and authentication fails.
 - If the credentials exist in the data store but the account is locked, the journey follows the **Locked** outcome. A **Message node** displays a custom lockout message and authentication fails.
 - If the credentials exist in the data store but the user must change their password, the node prompts the user to change their password. If the user cancels this change request, the journey follows the **Cancelled** outcome. A **Message node** displays a custom message and authentication fails.
 - If the credentials exist in the data store but the password has expired, the node follows the **Expired** outcome. The user is routed to an inner tree journey that contains the password reset logic and then routes the user to the start of the journey to authenticate again.

Alternative nodes

- The [Data Store Decision node](#) is a simpler node with only two outcomes, `True` and `False`. Use this node if the flow only requires these outcomes.

Availability

Product	Available?
PingOne Advanced Identity Cloud	Yes
PingAM (self-managed)	No
Ping Identity Platform (self-managed)	No

Inputs

The node reads the `username` and `password` fields from the node state.

The journey can provide these credentials in a number of ways, for example, with a combination of the [Platform Username node](#) and [Platform Password node](#), or by using the [Zero Page Login Collector node](#).

Dependencies

None

Configuration

Property	Usage
Minimum Password Length	For password change requests, the node rejects passwords that are shorter than this value. If you set this value to <code>0</code> , the node doesn't check the password length. Default: <code>8</code>
Username as Universal Identifier	If you enable this property, the <code>username</code> property is set to the value of the <code>uuid</code> . For example, <code>"username": "c636b756-ba6b-481d-ab4a-ab8c064cb24b"</code> . If this property is false, the value of the <code>username</code> property remains unchanged. For example, <code>"username": "bjensen"</code> . Default: false
Use mixed case for password change messages	Return password change messages in mixed (sentence) case. By default password reset and password change messages are transformed to upper case. Enable this option to return messages in sentence case. Default: Disabled

Outputs

This node copies shared and transient state into the outgoing node state.

Outcomes

True

The credentials match those found in the identity store.

False

The credentials don't match those found in the identity store.

Locked

The profile associated with the provided credentials is locked.

Cancelled

The user cancelled a password change request. The [example](#) provides a detailed explanation of this outcome.

Expired

The credentials match those found in the identity store, but the password has expired.

Password Collector node

Not supported in Advanced Identity Cloud

Prompts the user to enter their password.

The captured password is transient, persisting only until the authentication flow reaches the next node requiring user interaction.

Availability

Product	Available?
PingOne Advanced Identity Cloud	No
PingAM (self-managed)	Yes
Ping Identity Platform (self-managed)	No

Outcomes

Single outcome path.

Evaluation continues after capturing the password.

Configuration

This node has no configurable properties.

Username Collector node

Not supported in Advanced Identity Cloud

Prompts the user to enter their username.

Availability

Product	Available?
PingOne Advanced Identity Cloud	No
PingAM (self-managed)	Yes
Ping Identity Platform (self-managed)	No

Outcomes

Single outcome path.

Evaluation continues after capturing the username.

Configuration

This node has no configurable properties.