Getting Started

Use this guide to get a quick, hands-on look at Autonomous Identity. Download the image, install, and play around with the software.

ForgeRock® Autonomous Identity is an entitlements analytics system that lets you fully manage your company's access to your data.

An entitlement refers to the rights or privileges assigned to a user or thing for access to specific resources. A company can have millions of entitlements without a clear picture of what they are, what they do, and who they are assigned to. Autonomous Identity solves this problem by using advanced artificial intelligence (AI) and automation technology to determine the full entitlements landscape for your company. The system also detects potential risks arising from incorrect or over-provisioned entitlements that lead to policy violations. Autonomous Identity eliminates the manual re-certification of entitlements and provides a centralized, transparent, and contextual view of all access points within your company.

> IMPORTANT
>
> This guide is for developers, technical consultants, and ForgeRock partners who are familiar with Autonomous Identity. The deployment example in this guide is for evaluation purposes only. For production deployments, see the Install Guide.

Quick Start

Features Learn about the Autonomous Identity features.	**Architecture in Brief** Learn about the Autonomous Identity architecture.	**How It Works** Learn how Autonomous Identity works with a simple example.
Next Steps Learn where to go from here.	**FAQ** Read some FAQs on Autonomous Identity.	**Glossary** Look up Autonomous Identity terms in the glossary.

For installation instructions, see the Autonomous Identity Installation Guide.

For a description of the Autonomous Identity UI console, see the Autonomous Identity Users Guide.

Features

Autonomous Identity 2021.3.1 provides the following features:

- **Broad Support for Major Identity Governance and Administration (IGA) Providers**. Autonomous Identity supports a wide variety of Identity as a Service (IDaaS) and Identity Management (IDM) data including but not limited to comma-separated values (CSV), Lightweight Directory Access Protocol (LDAP), human resources (HR), database, and IGA solutions.
- **Highly-Scalable Architecture**. Autonomous Identity deploys using a microservices architecture, either on-prem, cloud, or hybrid-cloud environments. Autonomous Identity's architecture supports scalable reads and writes for efficient processing.
- **Powerful UI dashboard.** Autonomous Identity displays your company's entitlements graphically on its UI console. You can immediately investigate those entitlement outliers as possible security risks. The UI also lets you quickly identify those entitlements that are good candidates for automated low-risk approvals or re-certifications. Users can also view a trend-line indicating how well they are managing their entitlements. The UI also provides an application-centric view and a single-page rules view for a different look at your entitlements.
- **Powerful Analytics Engine**. Autonomous Identity's analytics engine is capable of processing millions of access points within a short period of time. Autonomous Identity lets you configure the machine learning process and prune less productive rules. Customers can run analyses, predictions, and recommendations frequently to improve the machine learning process.
- **UI-Driven Schema Extension**. Autonomous Identity lets administrators discover and extend the schema, and set up attribute mappings using the UI.
- **UI-Driven Data Ingestion and Mappings**. Autonomous Identity provides improved data ingestion tools to define multiple csv input files needed for analysis and their attribute mappings to the schema using the UI.
- **Broad Database Support**. Autonomous Identity supports both Apache Cassandra and MongoDB databases. Both are highly distributed databases with wide usage throughout the industry.
- **Improved Search Support**. Autonomous Identity now incorporates Open Distro for Elasticsearch, a distributed, open-source search engine based on Lucene, to improve database search results and performance.

Architecture in Brief

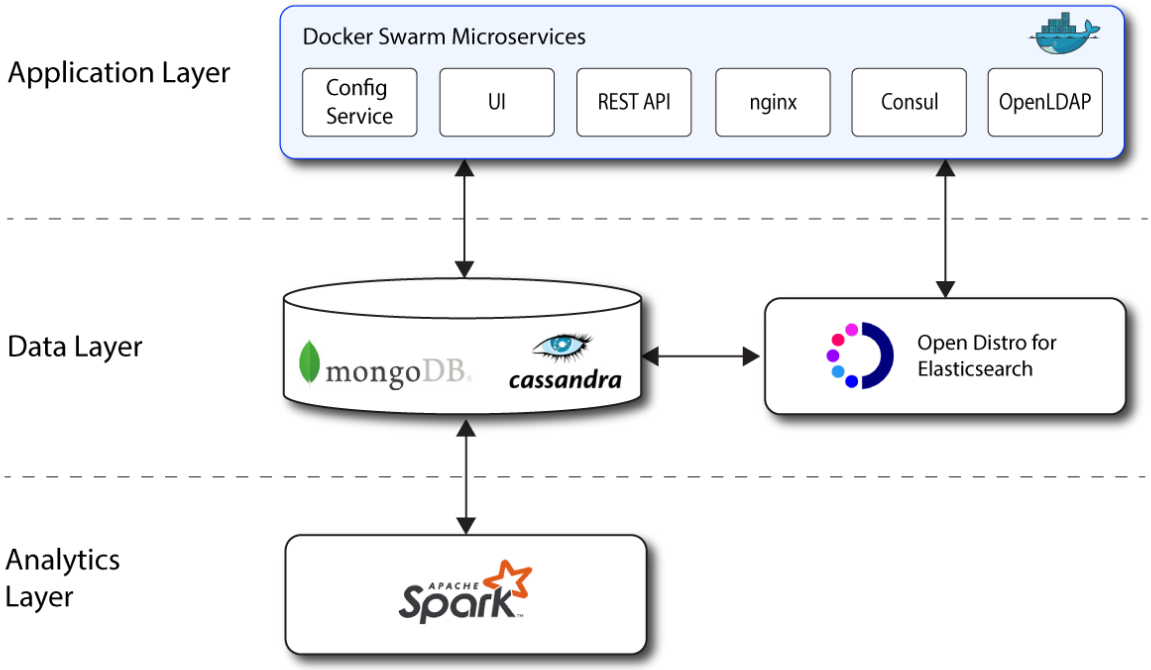
Autonomous Identity’s flexible architecture can deploy in any number of ways: single-node or multi-node configurations across on-prem, cloud, hybrid, or multi-cloud environments. The Autonomous Identity architecture has a simple three-layer conceptual model:

- **Application Layer**. Autonomous Identity implements a flexible Docker Swarm microservices architecture, where multiple applications run together in containers. The microservices component provides flexible configuration and end-user interaction to the deployment. The microservices components are the following:
  - **Autonomous Identity UI**. Autonomous Identity supports a dynamic UI that displays the entitlements, confidence scores, and recommendations.
  - **Autonomous Identity API**. Autonomous Identity provides an API that can access endpoints using REST. This allows easy scripting and programming for your system.
  - **Self-Service Tool**. The self-service tool lets users reset their Autonomous Identity passwords.
  - **Backend Repository**. The backend repository stores Autonomous Identity user information. To interface with the backend repository, you can use the `phpldapadmin` tool to enter and manage users.
  - **Configuration Service**. Autonomous Identity supports a configuration service that allows you to set parameters for your system and processes.
  - **Nginx**. Nginx is a popular HTTP server and reverse proxy for routing HTTPS traffic.
  - **Hashicorp Consul**. Consul is a third-party system for service discovery and configuration.
  - **Apache Livy**. Autonomous Identity supports Apache Livy to provide a RESTful interface to Apache Spark.
  - **Java API Service**. Autonomous Identity supports the Java API Service for RESTful interface to the Cassandra or MongoDB database.
- **Data Layer**. Autonomous Identity supports Apache Cassandra NoSQL and MongoDB databases to serve predictions, confidence scores, and prediction data to the end user. Apache Cassandra is a distributed and linearly scalable database with no single point of failure. MongoDB is a schema-free, distributed database that uses JSON-like documents as data objects. Java API Service (JAS) provides a REST interface to the databases.

  Autonomous Identity also implements Open Distro for Elasticsearch and Kibana to improve search performance for its entitlement data. Elastic Persistent Search supports scalable writes and reads.

- **Analytics and Administration Layer**. Autonomous Identity uses a multi-source Apache Spark analytics engine to generate the predictions and confidence scores. Apache Spark is a distributed, cluster-computing framework for AI machine learning for large datasets. Autonomous Identity runs the analytics jobs directly from the Spark master over Apache Livy REST interface.

Figure 1: A Simple Conceptual Image of the Autonomous Identity Architecture



How Autonomous Identity Works

Autonomous Identity is an AI-based analytics engine that discovers, analyzes, and generates a complete profile of your company’s entitlements.

Autonomous Identity looks at each entitlement and its relationship to the assigned user within the company. These relationships are modelled and assigned a single confidence score (from 0 to 100%) indicating the strength of correlation between the model and the assigned entitlement. The results are displayed on the UI console.

NOTE

For a definition of Autonomous Identity terms, see the Glossary.

Figure 2: A simple conceptual diagram of Autonomous Identity

image::how-autoid-works-2.png["Autonomous Identity ingests entitlement, HR, and application data to calculate confidence scores of the assigned entitlements."]

Let’s Run a Simple Example

Let’s run a simple example to see how Autonomous Identity models the entitlements and calculates confidence scores. Each company can decide the level and scope of the analysis. However, in most cases, the more data you analyze, the better the entitlement models.

Before you can ingest and process the data. You must run three pre-analytics tasks to prepare your data and machine learning runs.

Set Entity Definitions

The process begins by adding any new entity definitions to the schema. Autonomous Identity provides a UI interface that lets you add any new entity definitions or attributes to the schema.

Set Data Sources and Mappings

The next step is to define your data source files using the UI. Data can come from application services, HR databases, and other sources and must be in comma-separated values (.csv) formatted files.

You must also define mappings that map your data attributes to those in the schema. This is a critical step to ensure key data elements are properly included in your Autonomous Identity deployment.

Set Machine Learning Thresholds

The next step is to adjust any machine learning thresholds from the default values. In general, most deployments use the default values. You should only edit the values if you fully understand the machine learning process.

Data Ingestion

After you have set your entity definitions, defined your data sources and mappings, and set your thresholds, you can ingest the data into the system. There are four basic types of data required: applications, assignments, entitlements, and identities. Examples are provided below. Note that actual production data will have additional columns of information.

- **Applications**. Applications data include those attributes that define your applications.

  The following table shows a simple example of application data:

  Table 1: Applications

APP_ID	APP_NAME	APP_OWNER_ID
adp-1	ADP	irene.9
expensify-1	Expensify	janice.10

- **Assignments**. The second type of data is an assignments file that maps each user to their assigned entitlement. The data can come from your IAM/IGA system. If a user has multiple entitlements, each row represents a single assigned mapping to each entitlement.

  The following table shows a simple example of assignments data:

  Table 2: Assignments

ENT_ID	USR_ID	HIGH_RISK
Payroll Report	alice.1	High
Expenses	alice.1	High
Payroll Report	bob.2	High
Expenses	bob.2	High
Payroll Report	chris.3	High
Expenses	chris.3	High
Payroll Report	diane.4	High
Payroll Report	ellen.5	High
Payroll Report	fred.6	High
Expenses	gary.7	High
Payroll Report	harry.8	High
Expenses	irene.9	High
Payroll Report	janice.10	High
Expenses	karen.11	High

- **Entitlements**. In another file, you have entitlement data. The file may include data, such as full entitlement name, application name, entitlement or role owner, or any other information that helps with the machine learning.

The following table shows a simple example of entitlement data:

Table 3: Entitlements

ENT_ID	ENT_OWNER	APP_ID
Payroll Report	alice.1	adp-1
Expenses	bob.2	expensify-1

- **Identities**. The fourth type of data file is identities, which stores user profile information from your HR data including attributes that contain a minimally required set of information required for analytics processing.

  The following table shows a simple example of identities data:

  Table 4: Identities

USR_ID	DEPARTMENT	CITY
alice.1	San Jose	Finance
bob.2	Finance	San Jose
chris.3	Finance	San Jose
daine.4	HR	San Jose
ellen.5	HR	Austin
fred.6	HR	Austin
gary.7	HR	Austin
harry.8	HR	Austin
irene.9	IT	Dublin
janice.10	Finance	Dublin
karen.11	Finance	San Jose

Training

Next, Autonomous Identity runs a two-stage machine-learing process to generate the association rules and confidence scores. An association rule is an IF-THEN rule that expresses patterns between random data variables in a large transaction set. For example, [San Jose, Finance] → [Payroll Report] indicates that if a company's finance office is located in San Jose and a person works in that office and department, it is likely that they get access to the Payroll Report.

During stage one of the training, Autonomous Identity analyzes the user attribute data using machine learning algorithms to pattern-mine and create itemsets of rules. The frequency of occurrence is counted for each itemset. Only rules that appear three times or more are considered. Itemsets less than three are ignored.

NOTE

In a typical deployment, Autonomous Identity can create a million or more association rules for a company’s dataset.

The following table shows the results of the initial training process:

Table 4: Itemsets of User Data

Itemset	Freq
[San Jose]	6
[Finance]	5
[HR]=]	5
[San Jose, Finance]	4
[Austin]	3
[Austin, HR]	3

During this training run, the analytics engine creates a unique row in a table for each user and their assigned entitlements. In the example below, alice.1, bob.2, and chris.3 have multiple rows, one for each assigned entitlement. Again, only frequency sets (freqUnion) of three or more are considered.

The following table shows the results of the second training process:

Table 5: Training Stage 2

USER ID	CITY	DEPT	ENT
alice.1	San Jose	Finance	Payroll Report
alice.1	San Jose	Finance	Expenses
bob.2	San Jose	Finance	Payroll Report
bob.2	San Jose	Finance	Expenses

USER ID	CITY	DEPT	ENT
chris.3	San Jose	Finance	Payroll Report
chris.3	San Jose	Finance	Expenses
diane.4	San Jose	HR	Payroll Report
ellen.5	San Jose	HR	Payroll Report
fred.6	Austin	HR	Payroll Report
gary.7	Austin	HR	Expenses
harry.8	Austin	HR	Payroll Report
irene.9	Dublin	IT	Expenses
janice.10	Dublin	Finance	Payroll Report
karen.11	San Jose	Finance	Expenses

Autonomous Identity applies the association rules to the entitlement mappings and calculates the risk confidence scores by dividing the freqUnion by frequency numbers (FreqUnion/Freq). The FREQ column show the number of occurrences of a rule from Table 5, for example, the rule [San Jose] appears 9 times. The FreqUnion is the union of a rule with an entitlement, for example, the union of the rule [San Jose] with the entitlement, Expenses, appears 4 times in Table 5. The confidence score indicates the scale from 0 to 100% to indicate the strength of each correlation. A confidence score of 100% indicates that the assigned entitlement is highly correlated to the user's job function. Only rules that appear three times or more are considered.

The following table shows the applied association rules:

Table 6: Applied Association Rules to Entitlements

RULE	ENT	FREQ	FreqUnion	Confidence
[San Jose]	Payroll Report	9	5	56%
[San Jose]	Expenses	9	4	44%
[Finance]	Payroll Report	8	4	50%
[Finance]	Expenses	8	4	50%
[HR]	Payroll Report	5	4	80%

RULE	ENT	FREQ	FreqUnion	Confidence
[San Jose,Finance]	Payroll Report	7	3	43%
[San Jose,Finance]	Expenses	7	4	57%

Next, Autonomous Identity must re-adjust the frequency numbers from the previous table as the occurrences of a rule are inflated due to multiple appearances of a user’s entitlements (that is, one entitlement per row) as seen in Table 5. The re-adjustment provides a more accurate confidence score for each association rule.

The following are the results of the readjusted confidence scores:

Table 7: Applied Association Rules to Entitlements

RULE	ENT	FREQ (Corrected)	FreqUnion	Confidence
[San Jose]	Payroll Report	6	5	83%
[San Jose]	Expenses	6	4	67%
[Finance]	Payroll Report	5	4	80%
[Finance]	Expenses	5	4	80%
[HR]	Payroll Report	5	4	80%
[San Jose,Finance]	Payroll Report	4	3	75%
[San Jose,Finance]	Expenses	4	4	100%

As-is Predictions

After the training process has determined the association rules for each entitlement, the analytics engine runs through an *as-is predictions* process, where user accesses are mapped to each entitlement using the association rules.

In the example, the user `alice.1` has the following mapped entitlements, which are called *justifications* for their entitlement accesses.

The following table shows the results of the as-is-predictions process:

Table 8: Initial As-is Predictions for a User

USER ID	ENT	RULE	Confidence	FreqUnion
alice.1	Payroll Report		83%	5
alice.1	Payroll Report		80%	4
alice.1	Payroll Report		75%	3
alice.1	Expenses		67%	4
alice.1	Expenses		80%	4
alice.1	Expenses		100%	4

The as-is predictions filter the justifications from the previous step using confidence score properties that are set in the configuration file. The maximum confidence score is set by the `maxConf` property. The minimum confidence score is set by the `maxConf` minus the `pred_conf_window` property, which is set to 5% in the configuration file by default. Thus, for this example, the maximum confidence and minimum confidence score filters for each entitlement is as follows:

The following table shows the result of the as-is-predictions:

Table 9: As-is Predictions Filter for a User

ENT	maxConf	Min
Payroll Report	83%	78%
Expenses	100%	95%

Applying the filters in Table 9 to the mapped entitlements in Table 8, we get the following filtered assigned entitlements while discarding the rest. These filters are applied to all users in your analysis.

The following table shows the results for alice.1:

Table 10: Filtered As-is Predictions for a User

USER ID	ENT	RULE	Confidence	FreqUnion
alice.1	Payroll Report		83%	5
alice.1	Payroll Report		80%	4
alice.1	Expenses		100%	4

Finally, the highest `freqUnion` is used to find the users with a specific rule and entitlement access. All rules with the lower `freqUnion` values are filtered out to favor rules that apply to the largest number of employees within a company. This ensures that the most generalized rules are used for the analysis.

The following table shows the final as-is predictions for a user:

Table 11: Final As-is Predictions for a User

USER ID	ENT	RULE	Confidence	FreqUnion
alice.1	Payroll Report		83%	5
alice.1	Expenses		100%	4

Recommendations

The analytics process goes through a recommendations predictions process that takes the entitlement rules and identifies any users who should have access to the entitlement but do not. The analytics engine looks at each user's confidence score associated with the entitlement and if the confidence score exceeds a pre-configured hreshold value, the recommendation are made for the user.

The process begins by assigning the entitlements to all users and removing already existing accesses. Autonomous Identity assigns the rules and confidence scores for these new assignments.

The following table shows the recommendations assigned to users who do not have a particular entitlement:

Table 12: Recommendations

USER ID	ENT	RULE	Confidence
diane.4	Expenses		67%
ellen.5	Expenses		67%
fred.6	Expenses		67%
gary.7	Payroll Report		80%
harry.8	Expenses		20%
irene.9	Payroll Report	no rule found	0%
janice.10	Expenses		80%
janice.11	Payroll Report		83%

The analytics engine determines the rules and confidence scores that meet a threshold property, `conf_thresh`, which is set to 80% in the configuration file by default.

The following example shows the final recommendations:

Table 13: Final Recommendations

USER ID	ENT	RULE	Confidence
gary.7	Payroll Report		80%
janice.10	Expenses		80%
janice.11	Payroll Report		83%

The results will be uploaded to the Cassandra database as a recommended new entitlement and appears on the UI console on the Recommendations screen.

Output to the UI Console

The final step of the process is for Autonomous Identity to display the confidence scores graphically on the UI as a distribution from low, medium, to high scores. The console lets you immediately identify the low confidence scores that could pose a potential security risk as well as the high confidence scores that can be automatically approved or certified. Autonomous Identity displays the attributes that justified each confidence score as well as other data to help you manage your entitlements.

You can run the analytics weekly or monthly to ensure near realtime assessment of your entitlements. This ensures that some entitlements can immediately be flagged if it goes stale and is no longer necessary.

Next Steps

To see how Autonomous Identity works, you can do the following:

- Install a single-node deployment.

  You can run an Autonomous Identity single-node deployment. See Install a Single Node Deployment.

- View the UI Console

  You can view the various page views of the Autonomous Identity UI. See the Autonomous Identity Users Guide.

- Manage Users

You can add, remove, and change any passwords to Autonomous Identity. See Creating Users.

- Using the Autonomous Identity API

  Autonomous Identity provides API for developers who want to create scripts or applications to access the endpoints. See the Autonomous Identity API Guide.

- Browse the Documentation

  The following documentation is available for this release:

  - Autonomous Identity Release Notes
  - Autonomous Identity Installation Guide
  - Autonomous Identity Deployment Planning Guide
  - Autonomous Identity Administration Guide
  - Autonomous Identity Users Guide
  - Autonomous Identity API Guide

Frequently Asked Questions

How is Autonomous Identity different to peer group analytics?

Peer group analysis compares a user to their peer group to identify whether they have any accesses that may be anomalous, relative to that peer group. Autonomous Identity differs from peer group comparisons in that it compares users in the same department, with the same job title, or with every single person in the company. This granular approach provides a more comprehensive and global view of your entitlements. Autonomous Identity can therefore identify many potential patterns that may be missed with peer group analysis.

Can we weight an attribute within the features file to be more important and influence the resulting confidence scores?

No. Autonomous Identity is entirely data-driven. All user attributes are equally weighted. This means that it is possible to have association rules that do not lead to an entitlement assignment, such as [Expenses, Finance] → San Jose. In these cases, the rules are discarded. In general, weighted association rules would negatively impact the analytics results and thus are not implemented.

Glossary

anomaly report

A report that identifies potential anomalous assignments.

applications

A data source file type that includes application attributes.

as-is predictions

A process where confidence scores are assigned to the entitlements that users have.

assignment

A relationship between the user and an entitlement.

assignments

A data source file type that includes labels for the user and an entitlement.

association rule

The result of an Autonomous Identity machine learning process. It describes the rule for any given entitlement along with fequency, frequency union, and confidence score.

auto-certify

An action that an entitlement owner can do to approve a justification. Auto-certify indicates that anyone who has the justification is automatically approved for the entitlement.

auto-request

An action that an entitlement owner can do to approve a justification. Auto-request indicates that anyone who matches these justification attributes but may not already have access should automatically get provisioned for this entitlement.

confidence score

A score from a scale from 0 to 100% that indicates the strength of correlation between an assigned entitlement and a user's data profile. The score is derived from the ratio of frequency union over frequency.

data ingestion

A pre-analytics process that pushes the seven .csv files into the Cassandra database. This allows the entire training process to be performed from the database.

data sparsity

A reference to data that has null values. Autonomous Identity requires dense, high quality data with very few null values in the user attributes to get accurate analysis scores.

data validation

A pre-analytics process that tests the data to ensure that the content is correct and complete prior to the training process. This is run automatically.

driving factor

A single user attribute and its value that exceeds a confidence threshold level (e.g., 75%).

entitlement

An entitlement is a unit of `privilege`, whether fine-grained or course-grained. A user or device with an entitlement gets access rights to specified resources.

entitlements

A data source file type that includes the entitlement name, application-to-entitlement, roleowner, entitlement attributes.

frequency

The total occurrence of a rule within a user population.

frequency union

The total occurrence of a rule with a user population that has a specific entitlement.

identities

A data source file type that includes job and department descriptions, HR name, and any identifying features of a user.

insight report

A report that provides metrics on the rules and predictions generated in the analytics run.

recommendation

A process run after the as-is predictions that assigns confidence scores to all entitlements and recommends entitlements that users do not currently have. If the confidence score meets a threshold, set by the `conf_thresh` property in the configuration file, the entitlement will be recommended to the user in the UI console.

resource

An external system, database, directory server, or other source of identity data to be managed and audited by an identity management system.

REST

Representational State Transfer. A software architecture style for exposing resources, using the technologies and protocols of the World Wide Web. REST describes how distributed data objects, or resources, can be defined and addressed.

rule

A collection of driving factors (i.e., justifications), where each driving factor is related to each other through an AND relationship.

stemming

A process that occurs after training that removes similar association rules that exist in a parent-child relationship. If the child meets three criteria, then it will be removed by the system. The criteria are: 1) the child must match the parent; 2) the child (e.g., [San

Jose, Finance]) is a superset of the parent rule. (e.g., [Finance]); 3) the child and parent’s confidence scores are within a +/- range of each other. The range is set in the configuration file.

training

A multi-step process that generates the association rules with confidence scores for each entitlement. First, Autonomous Identity models the frequent itemsets that appear in the user attributes for each user. Next, Autonomous Identity merges the user attributes with the entitlements that were assigned to the user. It then applies association rules to model the sets of user attributes that result in an entitlement access and calculates confidence scores, based on their frequency of appearances in the dataset.

Was this helpful? 👍 👎