Deployment planning

Use this chapter to plan your Autonomous Identity deployment.

IMPORTANT

This chapter is for deployers, technical consultants, and administrators who are familiar with Autonomous Identity and are responsible for architecting a production deployment.

Architecture in brief

Learn about the Autonomous Identity architecture.

Security controls

Learn about the Autonomous Identity security controls.

Topology planning

Review topology sizing considerations.

Deployment checklist

Use the checklist.

For installation instructions, refer to the Autonomous Identity installation guide.

For component versions, refer to the Autonomous Identity Release notes.

Architecture in brief

Autonomous Identity has a powerful and flexible architecture that lets you deploy Autonomous Identity 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 as follows:

- **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.
    - **Backend Repository**. The backend repository stores Autonomous Identity user information.
    - **Nginx**. Nginx is a popular HTTP server and reverse proxy for routing HTTPS traffic.
    - **Apache Livy**. Autonomous Identity supports Apache Livy to provide a RESTful interface to Apache Spark.
    - **Java API Service**. Autonomous Identity supports a private Java API Service (JAS) for a 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 RESTful interface to the databases.

  Autonomous Identity also implements Opensearch and Opensearch Dashboards to improve search performance for its entitlement data. Opensearch supports scalable writes and reads. Opensearch Dashboards provides a useful visualization tool for your Opensearch backend.
- **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 main over Apache Livy REST interface.

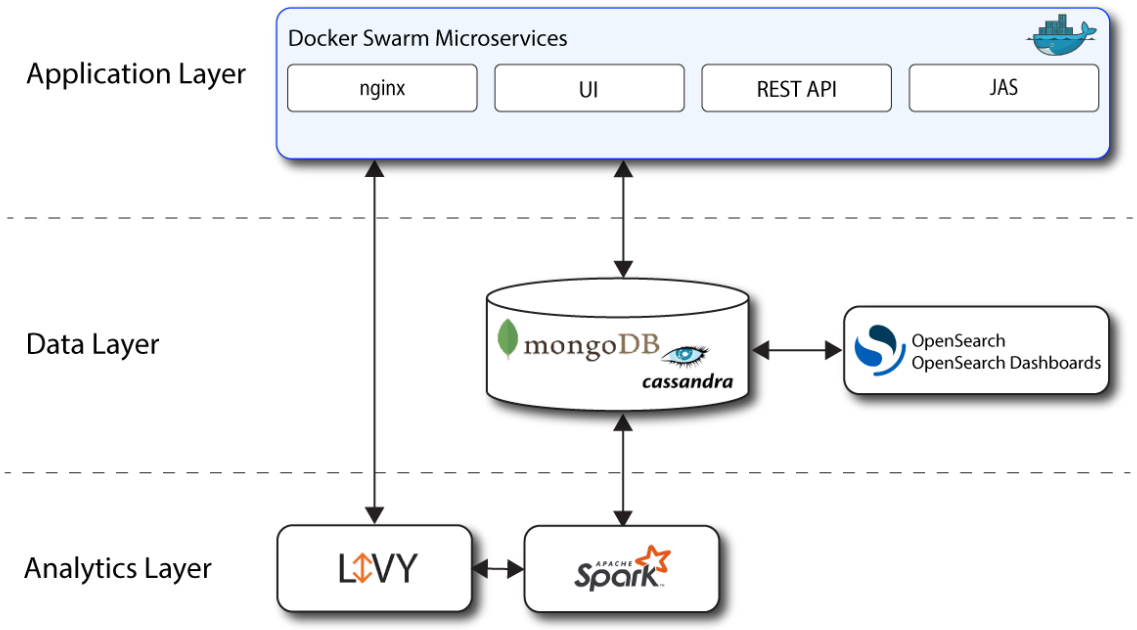

Figure 1. A Simple Conceptual Image of the Autonomous Identity Architecture

Security controls overview

Autonomous Identity uses a number of security protocols as summarized below.

Security Controls Summary

Security	Description
Encryption Protocol	TLSv1.2
Encryption: External Data in Transit	All data in transit from Autonomous Identity to the outside world is encrypted. SSL certificates must be configured with the load balancer. By default, Autonomous Identity configures self-signed certificates used by Nginx. Customers can also use their own certificates during deployment.

Encryption: Internal Data in Transit	Within the Autonomous Identity secure server network, most data in transit between the Autonomous Identity services is encrypted, but not all. The exception is any non-encrypted communication between Autonomous Identity servers. You can protect this communication via network firewalls. It is also recommended to disable access on network and firewall ports for services like Spark and Livy that are meant for internal access only. The rest of the services are SSL/TLS-protected including all Nginx-protected services, MongoDB, Cassandra, and Opensearch nodes.
Encryption: Data at Rest	MongoDB is not encrypted natively in Autonomous Identity, but can be encrypted via third-party disk encryption or using the MongoDB enterprise version. If encryption at rest is required, please confirm with the MongoDB vendors how this is handled in existing MongoDB clusters. Likewise, Cassandra is not natively encrypted, but can be supported through its enterprise versions.
Authentication	Autonomous Identity uses various authentication methods within its systems, such as the following: • **Local Authentication**. User credentials (user/groups) are stored in Opensearch. Users can log in with a username and password. This is mostly used for development or QA scenarios. • **OpenID Connect**. Autonomous Identity can use Single Sign-On (SSO) by integrating SSO providers like Azure AD and ForgeRock® Access Management (AM). The API service and Java API Service (JAS) are protected by authentication handlers that support token-based access. JAS also supports certificate-based authentication, which is only used by internal services that require elevated access.

Topology planning

Based on existing production deployments, we have determined a suggested number of servers and settings based on the numbers of identities, entitlements, assignments, and applications. These suggested number of servers and settings are general guidelines for your particular deployment requirements. Each deployment is unique, and requires review prior to implementation.

For a description of possible production deployments, refer to Deployment Architecture in the Autonomous Identity Installation Guide.

Data sizing

ForgeRock has determined general categories of dataset sizes based on a company's total number of identities, entitlements, assignments, and applications.

A key determining factor for sizing is the number of applications. If a company has identities, entitlements, and assignments in the Medium range, but if applications are close to 150, then the deployment could be sized for large datasets.

Data Set Ranges

	Small	Medium	Large	Extra Large
Total Identities	<10K	10K-50K	50K-100K	100K-1M
Total Entitlements	<10K	10K-50K	50K-100K	100K+
Total Assignments	<1M	1M-6M	6M-15M	15M+
Total Applications	<50	50-100	100-150	150+

Suggested number of servers

Based on dataset sizing, the following chart shows the number of servers for each deployment. These numbers were derived from existing customer deployments and internal testing setups.

IMPORTANT

These numbers are not hard-and-fast rules, but are only presented as starting points for deployment planning purposes. Each deployment is unique and requires proper review prior to implementation.

Suggested Number of Servers

	Small	Medium	Large	Extra Large

Deployer	1[1]	1	1	1
Docker	1	2 (manager; worker)	2 (manager; worker)	Custom[2]
Database	1	2 (2 seeds)	3 (3 seeds)	Custom[2]
Analytics	1	3 (master; 2 workers)	5 (master; 4 workers)	Custom[2]
Opensearch	1	2 (master; worker)	3 (master; 2 workers)	Custom[2]
Opensearch Dashboards	1	1	1	1

[1] This figure assumes that you have a separate deployer machine from the target machine for single-node deployments. You can also run the deployer on the target machine for a single-node deployment. For multi-node deployments, we recommend running the deployer on a dedicated low-spec box.
[2] For extra-large deployments, server requirements will need to be specifically determined.

Suggested analytics settings

Analytics settings require proper sizing for optimal machine-learning performance.

The following chart shows the analytics settings that are for each deployment size. The numbers were derived from customer deployments and internal testing setups.

IMPORTANT

These numbers are not hard-and-fast rules, but are only presented as starting points for deployment planning purposes. Each deployment is unique and requires proper review prior to implementation.

Suggested Analytics Settings

	Small	Medium	Large	Extra Large
Driver Memory (GB)	2	10	50	Custom[1]
Driver Cores	3	3	12	Custom[1]
Executor Memory (GB)	3	3-6	12	Custom[1]

Executor Cores	6	6	6	Custom[1]
Elastic Heap Size[2]	2	4-8	8	Custom[1]

[1] For extra-large deployments, server requirements will need to be specifically customized.
[2] Set in the `vars.yml` file.

Production technical recommendations

Autonomous Identity 2022.11.7 has the following technical specifications for production deployments:

Production Technical Specifications

	Deployer	**Database**	**Database**	**Analytics**	**Opensearch**
Installed Components	Docker	Cassandra	MongoDB	Spark (Spark Master)/Apache Livy	Opensearch
OS	CentOS	CentOS	CentOS	CentOS	CentOS
Number of Servers	Refer to Suggested number of servers	Refer to Suggested number of servers	Refer to Suggested number of servers	Refer to Suggested number of servers	Refer to Suggested number of servers
RAM (GB)	4-32	32	32	64-128	64
CPUs	2-4	8	8	16	16
Non-OS Disk Space (GB)[1]	32	1000	1000	1000	1000

NFS Shared Mount	N/A	N/A	N/A	1 TB NFS mount shared across all Docker Swarm nodes (if more than 1 node is provisioned) at location separate from the non-OS disk space requirement. For example, `/data` or `shared`.	N/A
Networking	nginx: 443 Docker Manager: 2377 (TCP) Docker Swarm: 7946, 4789 (UDP) 7946, 2049 (TCP)	Client Protocol Port: 9042 Cassandra Nodes: 7000	Client Protocol Port: 27017 MongoDB Nodes: 30994	Spark Master: 7077 Spark Workers: Randomly assigned ports	Opensearch: 9300 Opensearch (REST): 9200 Opensearch Dashboards: 5601
Licensing	N/A using Docker CE free version	N/A	N/A	N/A	N/A

Software Version	Docker: 20.10.17	Cassandra: 4.0.6	MongoDB: 4.4	Spark: 3.3.2 Apache Livy: 0.8.0-incubating	Opensearch /Opensearch Dashboards 1.3.13
Component Reference	Refer to below.[2]	Refer to below.[3]	Refer to below.[4]	Refer to below.[5]	Refer to below.[6]

[1] At root directory "/"
[2] https://docs.docker.com/ee/ucp/admin/install/system-requirements/
[3] https://docs.datastax.com/en/dse-planning/doc/planning/planningHardware.html
[4] http://cassandra.apache.org/doc/latest/operating/hardware.html
[4] http://www.mongodb.com
[5] https://spark.apache.org/docs/latest/security.html#configuring-ports-for-network-security
[6] https://Opensearch.org/

Deployment checklist

Use the following checklist to ensure key considerations are covered for your 2022.11.7 deployment:

Deployment Checklist

Check	Requirement	Details
Access		
☐	Remote Access	The Autonomous Identity Team is a global team. To support the needs of client teams, remote access to all servers is required for deployment and support of product.
☐	Service Account	The service account must have the ability to run passwordless sudo commands. The deployer will not without this ability.
☐	File Transfer Process	The Autonomous Identity Team require access to a file transfer process, which lets specified packages be transferred from the vendor to the client infrastructure.
Service Account		

☐	Service Account Group	The service account group must be the same as the service account name. For example, if the service account name is `srv-autoid`, that user must be in the group `srv-autoid`.
☐	Autonomous Identity Team Access	Autonomous Identity team members must be able to switch to this user after logging in to the servers.
☐	SSH Ability	The service account must be able to passwordless SSH between all Autonomous Identity servers; preferred method is RSA SSH key authentication.
☐	Default Shell	The default shell of the service account must be Bash.
☐	Directory Ownership	Ownership of the following directories must be given to the Service Account. • /data or applicable name of the shared mount (Docker and Spark servers) • /opt/autoid (all servers) • /tmp (R, W, E required + NOEXEC flag must not be present)
☐	Docker Commands	The service account must have permissions to run Docker commands. Note that Docker should NOT need to be installed as a prerequisite; this will be installed by deployment team.
Networking/Internet		
☐	Access to the Internet	If available, the front-end servers downloads the required Docker images from the official Autonomous Identity image repository.

☐	SSL Certificates	If SSL is being implemented, SSL certificates are required for the UI, Cassandra or MongoDB nodes, and Spark nodes. These certificates can be generated using one of the following four options: • Self-signed certificates for all 3 components • Valid certificate for the UI and self-signed certificates for Cassandra, MongoDB, and Spark nodes (self-signed certs only used in server-server traffic) • Valid and separate certificates for the UI, Cassandra, MongoDB, and Spark • *.domainname.com certificate (wildcard)
☐	Ports Open (Internal)	All internal ports specified in the Networking section of the Environment Specifications need to be opened for the specified servers.
☐	Ports Open (external browser)	The following ports must be accessible from a web browser within the client network: • 443 (Front-end) For a list of Autonomous Identity ports, refer to Autonomous Identity Ports.
Required Packages		
☐	Dependencies	The following packages must be installed on specified servers as prerequisites: • Analytics Servers: ◦ OpenJDK version "11.0.16" ◦ Python 3.10.9 with symlinks to Python 3 (sudo ln -s /usr/bin/python3.10 /usr/bin/python3)
Other		
☐	Infrastructure Support POC	A point-of-contact (POC) with sufficient access to the infrastructure is required. The POC can support in case of infrastructure blockers arise (e.g., proxy, account access, or port issues).

☐	SELinux	SELinux must be disabled on the Docker boxes. The package "container-selinux" must be present (this can be done as part of the root scripts described in the "Root Access" category).