

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, see the [Autonomous Identity Installation Guide](#).

For component versions, see the [Autonomous Identity Release Notes](#).

Architecture in Brief

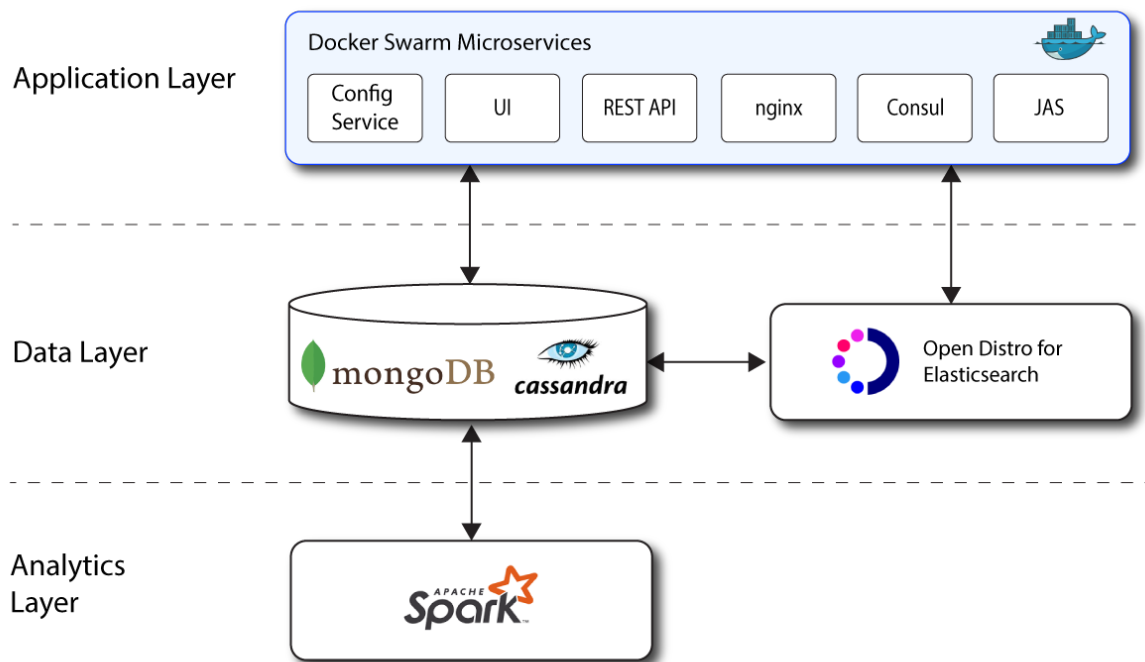
Autonomous Identity's flexible architecture can deploy in any number of ways: single-node or multi-node configurations across on-prem, cloud, hybrid, or multi-cloud environments. The Autonomous Identity architecture has a simple three-layer conceptual model:

- **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.
 - **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



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. Autonomous Identity configures self-signed certificates used by Nginx. Customers can also use their own certificates during deployment.

<p>Encryption: Internal Data in Transit</p>	<p>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.</p> <p>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, Mongo, Cassandra, and Elasticsearch nodes.</p>
<p>Encryption: Data at Rest</p>	<p>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.</p> <p>Likewise, Cassandra is not natively encrypted, but can be supported through its enterprise versions.</p>
<p>Authentication</p>	<p>Autonomous Identity uses various authentication methods within its systems, such as the following:</p> <ul style="list-style-type: none"> • LDAP Authentication. User credentials (user/groups) are stored in LDAP (OpenLDAP). 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). <p>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.</p>

Based on existing production deployments, we have determined a suggested number of servers based on total entitlement assignments. These suggested number of servers are guidelines for your particular deployment requirements. Unique client requirements may require customization, which may differ from the listed number of servers.

For a description of possible production deployments, see [Deployment Architecture](#) in the [Autonomous Identity Installation Guide](#).

Suggested Number of Servers

Data Set Ranges

	Small	Medium	Large	Custom
Total Assignments	<1M	1-5M	5-15M	15M+
Suggested # of Servers				
Application	Discuss with Autonomous Identity Team (dependent on HA requirements)[1]			
Database	2	2	3	Custom[2]
Analytics	1	2	3	Custom[2]
Deployer[3]	1	1	1	1

[1] Docker Swarm is designed to be highly available and may require 5 or more nodes. For a production deployment, the specific requirements can be discussed with the Autonomous Identity Team. Docker Swarm requirements are not correlated to data set size, but to front-end user requirements (for example, the number of users and frequency of use).

[2] For environments with more than 15 million assignments, server requirements will need to be specifically customized.

Production Technical Specifications

Autonomous Identity 2021.8.0 has the following technical specifications for production deployments:

Production Technical Specifications

	Application	Database	Database	Analytics	Deployer

Installed Components	Docker Swarm	Cassandra	MongoDB	Spark (Spark Master)/Apache Livy	Docker
OS	CentOS	CentOS	CentOS	CentOS	CentOS
# of Servers	See Suggested Number of Servers	See Suggested Number of Servers	See Suggested Number of Servers	See Suggested Number of Servers	See Suggested Number of Servers
RAM (GB)	32	32	32	64	32
CPUs	8	8	8	16	8
Non-OS Disk Space (GB)[1]	1000	1000	1000	1000	1000
NFS Shared Mount	Application layer services require access to the shared mount for analytics.	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	No specific port requirements
Licensing	N/A using Docker CE free version	N/A	N/A	N/A	N/A
Software Version	Docker: 19.03.8	Cassandra: 3.11.2	MongoDB: 4.4	Spark: 3.0.1 Apache Livy: 0.8.0-incubating	Docker: 19.03.8
Component Reference	See below. [2]	See below. [3]	See below. [4]	See below. [5]	See 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>

[5] <https://spark.apache.org/docs/latest/security.html#configuring-ports-for-network-security>

[6] <https://docs.ansible.com/ansible/latest/index.html>

Deployment Checklist

Use the following checklist to ensure key considerations are covered for your 2021.8.0 deployment:

Deployment Checklist

Check	Requirement	Details
-------	-------------	---------

Access		
<input type="checkbox"/>	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.
<input type="checkbox"/>	Service Account	The service account must have the ability to run passwordless sudo commands. The deployer will not without this ability.
<input type="checkbox"/>	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		
<input type="checkbox"/>	Service Account Group	The service account group must be the same as the service account name. For example, if the service account name is <code>srv-autoid</code> , that user must be in the group <code>srv-autoid</code> .
<input type="checkbox"/>	Autonomous Identity Team Access	Autonomous Identity team members must be able to switch to this user after logging in to the servers.
<input type="checkbox"/>	Passwordless Sudo	Root access via passwordless sudo is required to run required package installations (YUM), perform Docker installation, Docker Swarm-based installation applicable boxes, and potential troubleshooting. Please discuss with delivery team if this requirement is a concern. If so, submit a specified contact to run admin tasks.
<input type="checkbox"/>	SSH Ability	The service account must be able to passwordless SSH between all Autonomous Identity servers; preferred method is RSA SSH key authentication.
<input type="checkbox"/>	Default Shell	The default shell of the service account must be Bash.

<input type="checkbox"/>	Directory Ownership	Ownership of the following directories must be given to the Service Account. <ul style="list-style-type: none"> • /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)
<input type="checkbox"/>	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		
<input type="checkbox"/>	Access to the Internet	If available, the front-end servers downloads the required Docker images from the official Autonomous Identity image repository.
<input type="checkbox"/>	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: <ul style="list-style-type: none"> • 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)
<input type="checkbox"/>	Ports Open (Internal)	All internal ports specified in the Networking section of the Environment Specifications need to be opened for the specified servers.

<input type="checkbox"/>	Ports Open (external browser)	<p>The following ports must be accessible from a web browser within the client network:</p> <ul style="list-style-type: none"> • 443 (Front-end) • 8080 (Spark) • 8081 (Spark) <p>For a list of Autonomous Identity ports, see Autonomous Identity Ports.</p>
Required Packages		
<input type="checkbox"/>	Dependencies	<p>The following packages must be installed on specified servers as prerequisites:</p> <ul style="list-style-type: none"> • Cassandra Servers: java-1.8.0-openjdk-devel.x86_64 • MongoDB: see Deployment Prerequisites. • Analytics Servers: java-1.8.0-openjdk-devel.x86_64
Other		
<input type="checkbox"/>	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).
<input type="checkbox"/>	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).

<input type="checkbox"/>	Components Not Pre- installed	The following software must NOT be pre-installed on the box: <ul style="list-style-type: none">• Docker• Cassandra• MongoDB• Spark• Apache Livy• Ansible If any do come pre-installed, discuss the details with the Delivery Team ahead of time.
--------------------------	-------------------------------------	---

Was this helpful? 