Installing and Upgrading Jama Connect 9.6.6

Installing Jama Connect (KOTS)

Jama Connect is a Linux-based application that uses containerd to manage containers and depends on Replicated KOTS software to "orchestrate" deploying applications. The process of installing Jama Connect includes installing and configuring the software. These tasks deliver the components necessary to run Jama Connect.

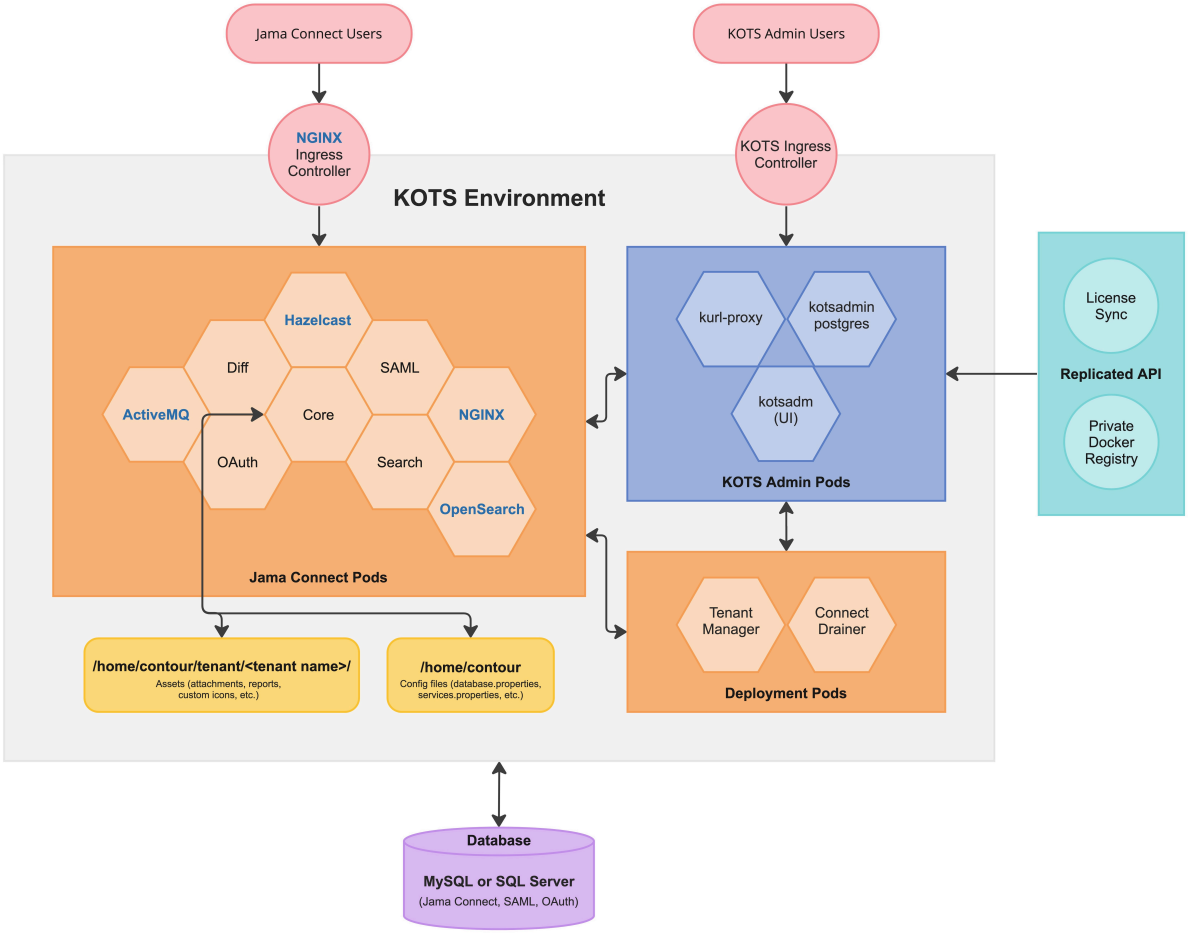
Components and what they do

Replicated KOTS — A container-based platform for easily deploying cloud native applications inside customers' environments, providing greater security and control. The KOTS Admin Console is the interface for installing and administering the Jama Connect application. See https://www.replicated.com/ for details.

Containerd — A container runtime that assists in the deployment, management, and operation of containers that support Jama Connect KOTS. See https://containerd.io/ for details.

Jama Connect license — The license included in your Welcome email. You save the license to your application server, then begin installing Jama Connect.

Jama Connect architecture



- Users access Jama Connect by browsing to the instance URL (https://jamainstanceurl.customer.com/).
- Administrators browse to the KOTS Admin Console using the same instance URL, but on port 8800 (https://jamainstanceurl.customer.com:8800/).

- Jama Connect and the license are updated via API calls for internet-enabled environments. Our airgap option removes the need for remote API calls.
- Content that is added to your Jama Connect instance is stored in the database.
- Uploaded artifacts, such as attachments and report templates, are stored in a Persistent Volume created by a Persistent Volume Claim (PVC) called **tenantfs**.

For more information about KOTS, see https://www.replicated.com/blog/announcing-kots/

Installation workflow (KOTS)

Whether your environment is internet-enabled or airgapped, the installation process consists of three stages: planning, preparation, and installation.

Review the system and server requirements for your environment, then follow the instructions for each stage.

1 Plan

- Review Release Notes
- System requirements
- Application server requirements and resource sizing
- Database server requirements and resource sizing

2 Prepare

- Prepare application and database servers
- Install and configure database
- Configure memory settings for Elasticsearch

3 Install

- Install KOTS software
- Provision your Jama Connect dataset
- Create a Replicated Snapshot

For this component...	Follow these instructions
MySQL	Install and configure MySQL [8]
Microsoft SQL	Install and configure Microsoft SQL Server [9]
Internet	Install KOTS software (internet) [13]
Airgap	Install KOTS software (airgap) [16]
Local Elasticsearch	Included by default
Remote Elasticsearch	Configure dedicated Elasticsearch nodes [28]

Planning your installation (KOTS)

Before you install the Admin Console and Jama Connect, make sure you have the following according to your type of installation.

All installations	• The license file sent from Jama Software (included in the Welcome email)
	• An application server with the necessary preparation [4] and sizing requirements [4]
	• A database server with the necessary preparation [7]
	• Supported [2] 64-bit Linux distribution with a kernel of:
	  • 4.x or greater (recommended)
	  • 3.10 (minimum)

Airgap installations	• URL to the airgap-safe portal (included in the Welcome email) for downloading the Jama Connect application file • A unique password (included in the Welcome email) to access the airgap-safe portal • PDF of this installation guide for the version of Jama Connect you are installing **IMPORTANT** If you lose the URL and password, contact Support to generate new ones.
Optional	• TLS certificate and private key to secure the Admin Console and Jama Connect application

System requirements and supported software (KOTS)

Make sure that your environment conforms to all requirements and recommendations before installing Jama Connect software.

After reviewing the information on this page, see Things to do before installation [6].

Application server

Use the information in this table for the server that runs the Jama Connect application. For details on sizing your application server to your environment, see Resource sizing for application server [4].

Component	
Minimum • 8 CPU • 32 GB RAM • 200 GB storage per node • Every node has the same storage space	**Recommended** • 16 CPU • 64 GB RAM • 200 GB storage • Every node has the same storage space
Operating system • *Recommended* — Ubuntu 20.04 or Ubuntu 22.04 • Red Hat 8.6 or 8.8 — Supported only when the RHEL Container Tools are not installed.	
Software installed with Jama Connect • KOTS • Containerd	
Musts • Dedicated server — Is running only Jama Connect • Accessible by admin with permissions • Uses only supported software and environments	

Database server

Use the information in this table for the server that runs your database. For details on sizing your database server to your environment, see Resource sizing for database server [6].

Component	
Minimum • 4–8 CPU • 16–24 GB RAM	**Recommended** • 8 CPU • 24 GB RAM • Dedicated volumes for data
Database software • MySQL 8 (recommended) • Microsoft SQL Server 2019 & 2022	
Operating system • *Recommended* — Ubuntu 20.04 or Ubuntu 22.04 • Red Hat 8.6 or 8.8	
Musts • Database is hosted on a server separate from the Jama Connect application. • Database server can host other databases, but no other applications. • Accessible by admin with permissions. • Uses only supported software and environments. • Databases must be able to accept a minimum of 300 concurrent connections.	
Not supported • Azure database • MariaDB • Custom configurations of Jama Connect databases (for example, query optimization and additional indexes that aren't shipped with Jama Connect)	

Supported software

Make sure your environment uses only supported software.

Component	
Browsers • Edge Chromium • Firefox* • Google Chrome* • Safari* *Versions released over the past 12 months are supported.	**Important** Browser zoom is supported only at 100%. Use of browser extensions/add-ons or enabling Compatibility View is not supported while using Jama Connect. **Tip** To prevent session issues, use the application in a single browser window.
Word processor and spreadsheet programs • Office 365 for Mac • Office 365 for Windows	Office 365 is used for exports and reports.

Application server requirements (KOTS)

To install and run Jama Connect successfully, your application server must meet these requirements.

Requirement	Notes
A dedicated application server	Jama Connect is the only application running on the application server. External services can affect stability of the application, for example by consuming memory resources.
Sufficient storage, CPU, and memory for optimal performance	To estimate the size of and required resources for your application server, see Resource sizing for application server [4].
Accessible by an admin with permissions	An admin must have proper permissions to maintain the application, perform upgrades, and access the server for regular maintenance.
Uses compatible software and environments	Verify that you're using supported software and environments [2] compatible with the most recent self-hosted release.

Resource sizing for application server (KOTS)

For optimal performance, estimate your application server needs before you install Jama Connect.

Requirements

- Each node must have a minimum volume of 200 GB. Increase this size based on the size of the assets that you plan to save in Jama Connect. We recommend that every node has the same storage space.
- KOTS must be up and running before you configure the application settings in the KOTS Admin Console.

IMPORTANT

To avoid performance issues, use the recommended requirements for horizontal scaling, rather than minimum requirements.

Use the following tables to help determine resources for the primary node of your application server.

Table 1. Minimum size (AWS instance sizing = m5.2xlarge)

CPU	RAM	CPU + memory settings	CPU + memory setting with horizontal scaling jamacores
8	32 GB	N/A	*jamacore application settings:* • Maximum CPU: 1000m • Maximum memory: 2 G • Maximum memory per container: 3 G • Number of ingress nodes 2

Table 2. Recommended size (AWS instance size = m5.4xlarge)

CPU	RAM	CPU + memory settings	CPU + memory setting with horizontal scaling jama-cores
16	64 GB	*Supports:* • 1,250 users with a ramp-up time of 30 sec-onds	*Supports:* • 1,250 users with a ramp-up time of 10 seconds • 2,500 users with a ramp-up time of 30 seconds
		jamacore application settings: • Maximum CPU: 12000m • Maximum memory: 48 G • Maximum memory for container: 60 G	*jamacore application settings:* • Maximum CPU: 3000m • Maximum memory: 12 G • Maximum memory for container: 15 G • Number of ingress nodes: 2
		Elasticsearch settings: • Maximum CPU: 8000m • Maximum memory: 8 G • Maximum memory for container: 10 G	*Elasticsearch settings:* • Maximum CPU: 8000m • Maximum memory: 8 G • Maximum memory for container: 10 G
		Diff Service settings: • Maximum memory: 2 G	*Diff Service settings:* • Maximum memory: 2 G

Use the following table to help determine resources for the secondary node of your application server.

Table 3. Secondary nodes dedicated to Elasticsearch: Recommended size (AWS instance size = m5.2xlarge)

CPU	RAM	CPU + memory settings
8	32 GB	*Supports:* • 2,500 users with a ramp-up time of 10 seconds *Elasticsearch settings:* • Maximum CPU: 8000m • Maximum memory: 8 G • Maximum memory for container: 10 G

TIP

Once you're up and running, you can monitor usage and adjust settings as needed.

Database server requirements (KOTS)

The database must be hosted on a server separate from the Jama Connect application. This server can host other databases, but we don’t support running other applications on the same server as the database.

Supported databases

- MySQL 8 (recommended)
- Microsoft SQL Server 2019 & 2022

What is not supported

- Azure database
- MariaDB
- Custom configurations of Jama Connect databases. Customizations such as query optimization and additional indexes that aren't shipped with Jama Connect aren't supported.

Resource sizing for database server (KOTS)

For optimal performance, estimate your database server needs before you install Jama Connect.

Use the information in this table to determine resources needed for your database server.

Database server	Small	Medium	Large	Enterprise
Active items in system	$\leq 600,000$	≤ 2 million	2–4 million	4 million+
Active projects	≤ 100	≤ 500	$\leq 1,000$	1,000+
Concurrent users	≤ 50	≤ 500	$\leq 1,000$	1,000+
CPU	4	8	16	Contact Support
Total systems of RAM	16 GB	32 GB	64 GB	Contact Support

If your usage approaches the Enterprise threshold, contact Support for customized recommendations and advanced, multi-server setup.

TIP

Once you're up and running, you can monitor usage and adjust settings as needed.

Important considerations

- Total system RAM for your database server can vary if you're using memory intensive workflows such as reuse, exporting, moving items, integrations, and batch updates. Database sizing is based on your usage patterns and platform. You must have a minimum of 4–8 cores and 16–24 GB of memory. Consult with your database admin when determining database size.
- The memory allocation allows for minimum headroom. If you need to run additional software for monitoring and analysis, consider the system requirements for that software. Configure dynamic memory settings as needed in the Admin Console.

Things to do before installation (KOTS)

Whether your environment is internet-enabled or airgap, make sure that your application server and database server are ready before installing Jama Connect.

- Review the Jama Connect Release Notes.
- Prepare your application server [7].
- Prepare your database server [7].
- Install and configure your database (MySQL [8] or SQL Server [9]).
- Configure custom memory settings for Elasticsearch [12].

Prepare your application server (KOTS)

Make sure your application server meets all requirements. See System requirements and supported software [2].

For users and admins to properly access Jama Connect, specific ports must be accessible to inbound traffic. Work with your network admin to make sure your network is configured properly.

1. **Inbound rules and ports for nodes** — Make sure the ports in the following table are accessible to inbound traffic and the inbound rules are configured for each server in the KOTS cluster.

Protocol	Port range	Source*	Inbound rule applies to node...	Description
HTTPS	443	Anywhere	All	Jama Connect port for SSL/TLS communication (HTTPS), which is used to access Jama Connect. It can be disabled or the port number can be reconfigured.
HTTP	80	Anywhere	All	Jama Connect port for clear text communication (HTTP), which is used to access Jama Connect. It can be disabled or the port number can be reconfigured.
TCP	8800	Anywhere	All	Allows admins to access the KOTS Admin Console, which is used to configure, install, and upgrade Jama Connect.
SSH	22	Anywhere	All	Allows admins to make remote connections to the nodes using SSH.
TCP	6443	Anywhere Any node	Primary	Allows admins and KOTS nodes to access the Kubernetes API server.**
TCP	2379–2380	Any node	Primary	Allows the KOTS nodes to access the etcd server client API.**
TCP	10250	Any node	All	Allows the KOTS nodes to access the Kubelet API server.**
UDP	8472	Any node	All	Allows KOTS (Flannel) to create a virtual network that connects the services running inside the cluster.**

* *Anywhere* means anyone or anything that must consume the resources in the environment.

** Can be disabled in single node clusters.

2. **User IDs** — Verify that the following User IDs are available and unused on the application server.
   - **User ID 91** — Used by Tomcat to read and write to directories inside jamacore pods.
   - **User IDs 480–499** — Used by the various services.
3. **Time sync setting** — To ensure accurate time on the application server, set up a cron job to sync the time on a routine schedule (for example, every day or hour). Use this command to set up the cron job:

```
ntpdate pool.ntp.org
```

Preparing your database server (KOTS)

The following information is needed when connecting the application server to the database server.

Information	Requirements
Type/vendor	Database must be one of the following: • MySQL 8 (recommended) — Install and configure MySQL [8] • Microsoft SQL Server 2019 & 2022 — Install and configure Microsoft SQL Server [9]
Database hostname	Example: *jama.companydb.com*

Information	Requirements
Listening ports	The application server must be allowed to communicate remotely with the database server over the listening ports. Default ports are: • MySQL = 3306 • Microsoft SQL Server = 1433
Database schema name	The database owner must be able to create one: • A new database schema • Tables inside an existing database schema of the given name The database name must follow these rules: • Start with a letter (a–z) • Contain any number of characters: a–z, 0–9 or an underscore ("_") • Letters must be lowercase
Username	*jamauser*
Password	
Connections	The database must be able to accept a minimum of 300 concurrent connections.
SAML schema username	*samluser*
OAuth database username	*oauthuser*

The username and password for SAML and OAuth must match what's entered in the Microsoft SQL Server upgrade preparation script. See Install and configure Microsoft SQL Server for more details.

Install and configure MySQL (KOTS)

MySQL is the recommended database server. Follow these steps to install and configure it.

Important considerations

- You must have full database admin permissions to the server hosting the MySQL database.
- For the Jama Connect installation to succeed, you must first create two additional database schemas.
- If you need to upgrade MySQL, see "Install and configure MySQL (upgrading traditional to KOTS)" in the *Jama Connect User Guide.*

Recommended settings and sample

The following recommended settings require 8 GB of memory allocated to MySQL Server for a typical installation and 16 GB for an enterprise installation.

These settings can be added to your my.cnf file (Linux) or my.ini file (Windows).

Property	Typical installation	Enterprise installation
max_allowed_packet	1 GB	1 GB
tmp_table_size	2 GB	2 GB
max_heap_table_size	2 GB	2 GB
table_open_cache	512	512
innodb_buffer_pool_size	2 GB	12 GB
innodb_log_file_size	256 MB	256 GB
innodb_log_buffer_size	12 MB	12 MB
innodb_thread_concurrency	16	16
max_connections	151	351
wait_timeout	259200	259200

Here is a sample text config file at an enterprise level. You must add the following values for your environment:

```
bind-address=0.0.0.0
key_buffer_size=16M
max_allowed_packet=1G
thread_stack=192K
thread_cache_size=8
tmp_table_size=2G
max_heap_table_size=2G
table_open_cache=512
innodb_buffer_pool_size=12G
innodb_log_file_size=256M
innodb_log_buffer_size=12M
innodb_thread_concurrency=16
max_connections=351
wait_timeout=259200
```

To install and configure MySQL:

1. Make sure that the InnoDB engine is enabled.
2. Download and install a supported version of MySQL [2].
3. On the MySQL database server, create an empty Jama Connect schema / database that uses UTF8:

   ```
   CREATE DATABASE jama character set utf8mb4;
   ```
4. On the MySQL database server, create two additional database schemas and a user ("jamauser") with the ability to access, create, and update tables within the database:

   ```
   CREATE DATABASE saml;
   CREATE DATABASE oauth;
   CREATE USER 'jamauser'@'%' IDENTIFIED BY 'password';
   CREATE USER 'oauthuser'@'%' IDENTIFIED BY 'password';
   CREATE USER 'samluser'@'%' IDENTIFIED BY 'password';
   GRANT ALL PRIVILEGES ON jama.* TO 'jamauser'@'%';
   GRANT ALL PRIVILEGES ON oauth.* TO 'oauthuser'@'%';
   GRANT ALL PRIVILEGES ON saml.* TO 'samluser'@'%';
   ```
5. Create a database schema for Quartz to support horizontal scaling in KOTS:

   ```
   CREATE DATABASE quartz;
   CREATE USER 'quartzuser'@'%' IDENTIFIED BY 'password';
   GRANT ALL PRIVILEGES ON quartz.* TO 'quartzuser'@'%'
   ```
6. Restart the database server.

Install and configure Microsoft SQL Server (KOTS)

If you are using Microsoft SQL Server for your database, follow these steps to install and configure it.

Important considerations

- You must have full database admin permissions to the server hosting the SQL Server database.
- If you need to upgrade the Microsoft SQL Server, see "Install and configure Microsoft SQL Server (upgrading traditional to KOTS)" in the *Jama Connect User Guide*.

Before installing Jama Connect 9.6.x

- Install Microsoft SQL Server 2019 or 2022 for the database server.
- Create an empty Jama Connect database and two additional database schemas for the installation to succeed.

- Jama Connect requires that the MSSQL COMPATIBILITY_LEVEL value is 130 or greater.
  To confirm the current value:

  ```
  SELECT compatibility_level FROM sys.databases WHERE name = <DATABASENAME>;
  ```

  To modify the value:

  ```
  ALTER DATABASE <DATABASENAME> SET COMPATIBILITY_LEVEL = 130;
  ```

  For more information, see https://learn.microsoft.com/en-us/sql/t-sql/statements/alter-database-transact-sql-compatibility-level?view=sql-server-ver16

Organizations using Microsoft SQL Server must enter database users in Replicated. Without these entries, the installation will fail.

The new schema must be created for a successful installation. Otherwise, the system continues to attempt to connect to the databases and produces log failures. After you create the database schemas, you must restart Jama Connect.

For more information, go to Supported software, environments, and system requirements and select your version of Jama Connect.

Follow these steps for a first-time installation of Jama Connect:

1. Connect to the SQL Server using a SQL management application (such as SQL Server Management Studio).
2. Replace the following values in the installation script: **<JAMA_LOGIN_Psswd>, <SAML_LOGIN_Psswd> & <OAUTH_LOGIN_Psswd>**.
3. Copy and store the passwords you create here. You will need them later to configure the Admin Console settings.
4. In a new query window, run this SQL query script:

```
-- Fresh Install Preparation SCRIPT
/*
Jama Connect Preparation Commands for a fresh install. It is required to
run
 these command / script on the Microsoft SQL Server BEFORE running the
Jama
Connect 8.62.x install
for ON-PREM installation using Microsoft SQL Server 2016 - 2019
DATE: 05/10/2021
NOTES:
This script assumes this is a new Installation of JAMA Connect. DO NOT
RUN
THIS SCRIPT ON AN EXISTING JAMA INSTALLATION.
The script will create a new empty JAMA database, add 2 new schemas
(empty) to
the Jama Database, 2 new DB Logins and Database users to support the
Multi-
Auth functionality released in Jama Connect 8.62.0.

INSTRUCTIONS:
This script must be run prior to Jama installation or installation may
fail to
complete.
Modify the <JamaUser_LOGIN_Psswd>, <SAML_LOGIN_Psswd> &
<OAUTH_LOGIN_Psswd>
values in the script below before Execution.
```

```
Passwords must be enclosed in single quotes.
*/

USE master;
CREATE LOGIN jamauser with password = 'password';
CREATE LOGIN samluser with password = 'password';
CREATE LOGIN oauthuser with password = 'password';
GO

USE master;
CREATE DATABASE jama;
GO
ALTER DATABASE jama SET READ_COMMITTED_SNAPSHOT ON WITH ROLLBACK IMMEDIAT
E
GO
ALTER DATABASE jama CONVERT TO CHARACTER SET latin1 [COLLATE =
'latin1_general_CI_AI'];
GO

USE jama;
EXEC ('CREATE SCHEMA oauth');
EXEC ('CREATE SCHEMA saml');
GO

USE jama;
CREATE USER jamauser for LOGIN jamauser;
CREATE USER samluser for LOGIN samluser with DEFAULT_SCHEMA=saml;
CREATE USER oauthuser for LOGIN oauthuser with DEFAULT_SCHEMA=oauth;
GO

EXEC sp_addrolemember N'db_owner', jamauser;
EXEC sp_addrolemember N'db_owner', samluser;
EXEC sp_addrolemember N'db_owner', oauthuser;
GO
```

5. Create a database schema for Quartz to support horizontal scaling in KOTS:

```
USE master;
CREATE LOGIN quartzuser with password = 'password';
GO

USE jama;
EXEC ('CREATE SCHEMA quartz');
GO

USE jama;
CREATE USER quartzuser for LOGIN quartzuser with
DEFAULT_SCHEMA=quartz;
GO

EXEC sp_addrolemember N'db_owner', quartzuser;
GO
```

6. Confirm that these actions were successful:
   - **Script completed** — Check the Query Execution results for errors.
   - **Users created** — Run the following SQL script in a new query window.

```
USE jama
SELECT * from master.sys.sql_logins
SELECT * from Jama.sys.sysusers
```

The results include **jamauser**, **samluser**, and **oauthuser** in the "Name" column of the result panes.

- **Users granted the DB_owner role** — Run the following SQL script in a new query window.

```
USE jama
SELECT DP1.name AS DatabaseRoleName,
isnull (DP2.name, 'No members') AS DatabaseUserName
FROM sys.database_role_members AS DRM
RIGHT OUTER JOIN sys.database_principals AS DP1
ON DRM.role_principal_id = DP1.principal_id
LEFT OUTER JOIN sys.database_principals AS DP2
ON DRM.member_principal_id = DP2.principal_id
WHERE DP1.type = 'R'
ORDER BY DP1.name;
```

The results show that db_owner role is granted to **jamauser**, **samluser**, and **oauthuser**.

7. Keep the database from locking users' accounts while they are logging in or working in Jama Connect (you must have db_owner permissions):

```
ALTER DATABASE jama SET READ_COMMITTED_SNAPSHOT ON WITH
ROLLBACK IMMEDIATE;
```

8. Make sure the flag was successfully enabled:

```
SELECT is_read_committed_snapshot_on FROM sys.databases WHERE
name='jama';
```

If the returned value is 1, the flag is on.

Configure custom memory settings for Elasticsearch (KOTS)

To prepare for installing Jama Connect, you must first update the system that hosts the application. The update consists of configuring memory settings for Elasticsearch.

Requirements

- The memory settings must be configured on each server in the KOTS cluster for Elasticsearch to run on these servers. If you use the remote Elasticsearch setting, the memory settings can be applied only to servers that are dedicated to Elasticsearch.
- You must have admin permissions to configure the memory settings for Elasticsearch.

To configure memory settings:

1. As an admin, open the /etc/sysctl.conf file, add the following line to the file, then save the file.

```
vm.max_map_count=262144
```

2. Reload the sysctl.conf file:

```
sudo sysctl -p
```

3. To confirm, type this command:

```
sudo sysctl -a | grep max_map_count
```

The system responds with:

```
vm.max_map_count=262144
```

Installing the software (KOTS)

KOTS is an open-source application for Kubernetes clusters that streamlines the process to remotely install, manage, and update Jama Connect, all from the KOTS Admin Console or command-line interface (CLI).

IMPORTANT
KOTS and Jama Connect must be installed on a new cluster that is created during installation and dedicated to KOTS.

Whether your organization is internet-enabled or requires an airgap installation, follow these instructions to download, install, and configure the software you need for your Jama Connect instance.

The software includes:

- KOTS Admin Console (Replicated)
- Jama Connect

Jama Software sends a Welcome email that includes your Jama Connect license file.

The installation process consists of these tasks:

- Install KOTS and Jama Connect (internet [13] or airgap [16])
- Provision your Jama Connect dataset [20]
- Create a Replicated Snapshot [21]

Depending on your environment, the process can also include these tasks:

- Configure KOTS to save tenant assets in Amazon EFS [30]
- Enable horizontal scaling [25]
- Configure dedicated Elasticsearch nodes [28]
- Configure Federated Authentication for KOTS Admin Console [35]

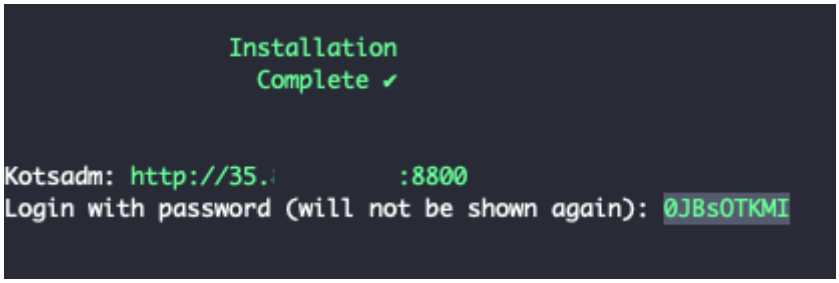
Install Jama Connect and KOTS (internet)

The installation script and the installation wizard guide you through the process of installing the KOTS-required software and Jama Connect, then configuring the KOTS Admin Console.

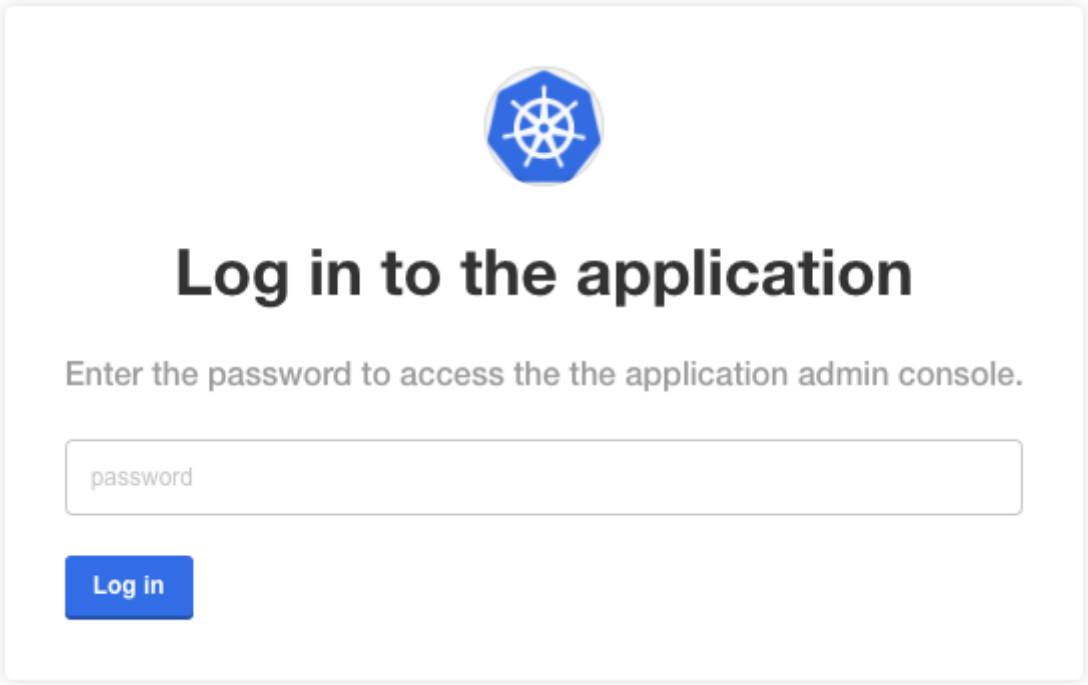
The license file is included in the Welcome email you received from Jama Software.

1. Open the Welcome email from Jama Connect, then save the attached license file on your local system.
2. Run the command on the application server provisioned for Jama Connect:

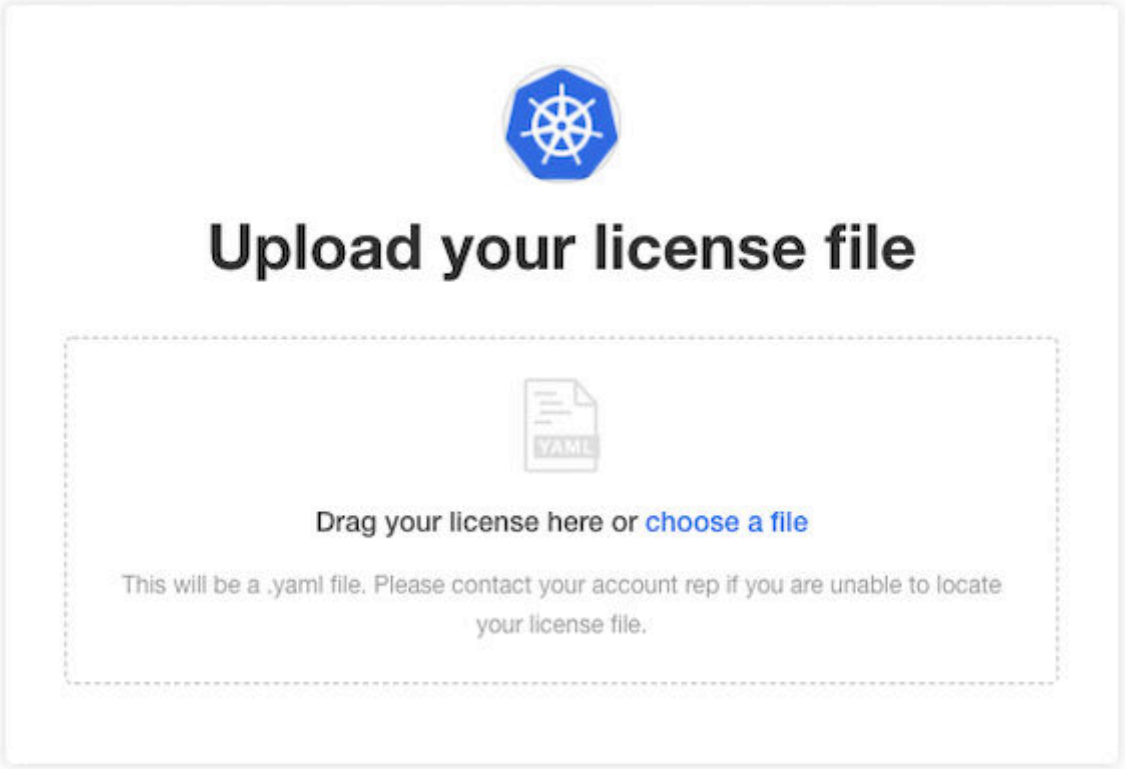
   ```
   curl -sSL https://kurl.sh/jama-k8s-standardkots | sudo bash
   ```
3. After the command runs, save the KOTS admin URL, password, and other configuration options for future reference. This is the only time these credentials appear, so make sure you save them.



4. In a supported browser, enter the URL for **Kotsadm**, which was generated when you installed KOTS.
5. Log in to the KOTS Admin Console using the password you just saved.



6. Select the appropriate option:
   - **Have key/certificate** — Click **Choose file** under Private key and Certificate, navigate to the files and select them, then click **Upload & Continue**.
   - **No key/certificate** — Select **Self-Signed Cert**.
7. Upload the license file that you saved on your local system.



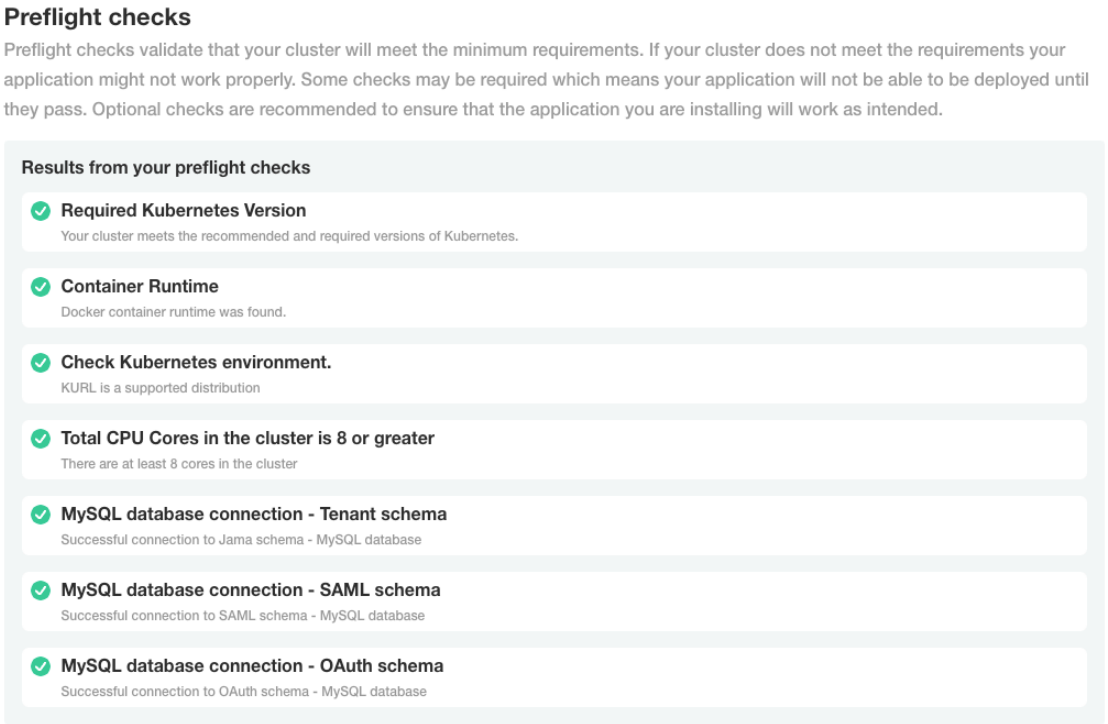
8. Configure the settings for each group, as needed. Scroll down to see each group of settings.

- **Database Settings** — Select your database type (**MySQL** or **Microsoft SQL Server**), then use the information from Preparing your database server [7] to complete the settings.
- **Host Name** — Enter the base URL for Jama Connect. Ensure this domain name is routable on your network.
- **TLS Key Pair Source** — (Optional) If you have a custom key and certificate for the host name, select **Custom TLS Configuration**. In the TLS Configuration section, upload the key and certificate.
- **Assets Size** — Enter the estimated size of the assets that you are planning to store in Jama Connect.
- **Elasticsearch Settings > Volume Size** — Enter the amount of disk space that each Elasticsearch node is allowed to use.
- **Tenant Manager Settings** — Enable this setting for optimal performance. Disable this setting if background operations are required before you provision the tenant (for example, when reusing traditional Replicated or using remote Elasticsearch).
  The *Tenant Manager* provisions, restores, upgrades, and sets licenses during application startup.

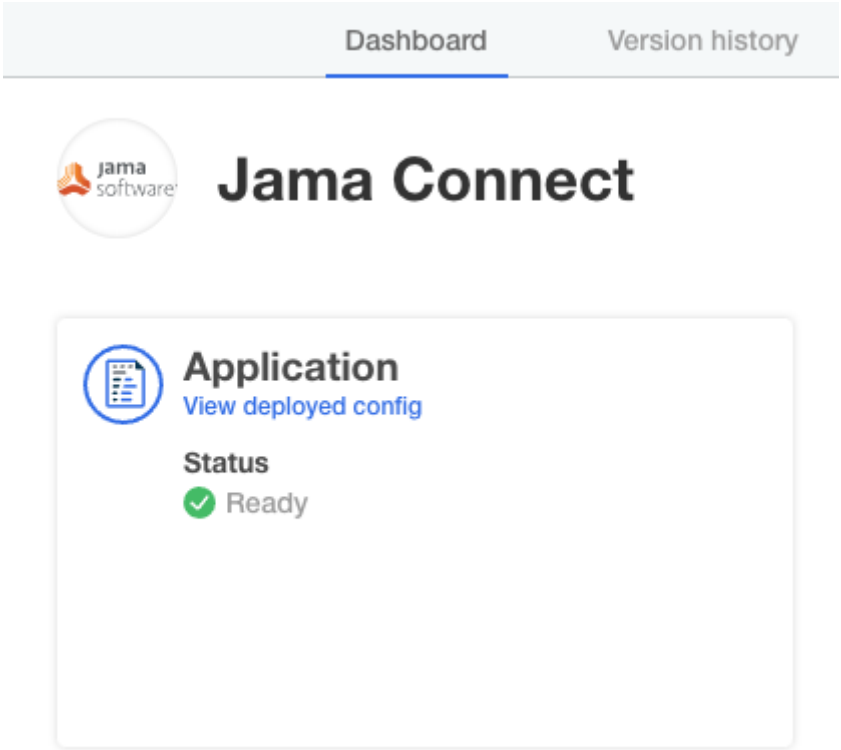
9. (Optional) From the Config tab in the KOTS Admin Console, follow the steps to configure KOTS to save tenant assets in the Amazon EFS [30].

> **NOTE**
> To use Ubuntu 22.04, you must update the memory or Elasticsearch fails. From the KOTS Admin Console, adjust the memory settings so that Maximum Memory is 6G and Maximum Memory for Container is 8G.

10. Scroll to the bottom of the page and click **Continue**.
    The system performs the preflight checks.



11. From the Preflight checks screen, click **Continue** to open the KOTS Admin Console.
    The process can take up to an hour. When the system is available, the status changes to **Ready**.



12. Log in to Jama Connect as root using the hostname configured for Jama Connect.
13. **Important:** Once Jama Connect is installed, use these instructions to provision a Jama Connect dataset [20].

> **IMPORTANT**
> You must provision a Jama Connect dataset [20] before you allow your users access to Jama Connect. If you need the link to the dataset, contact your Customer Success Manager.

Install Jama Connect and KOTS (airgap)

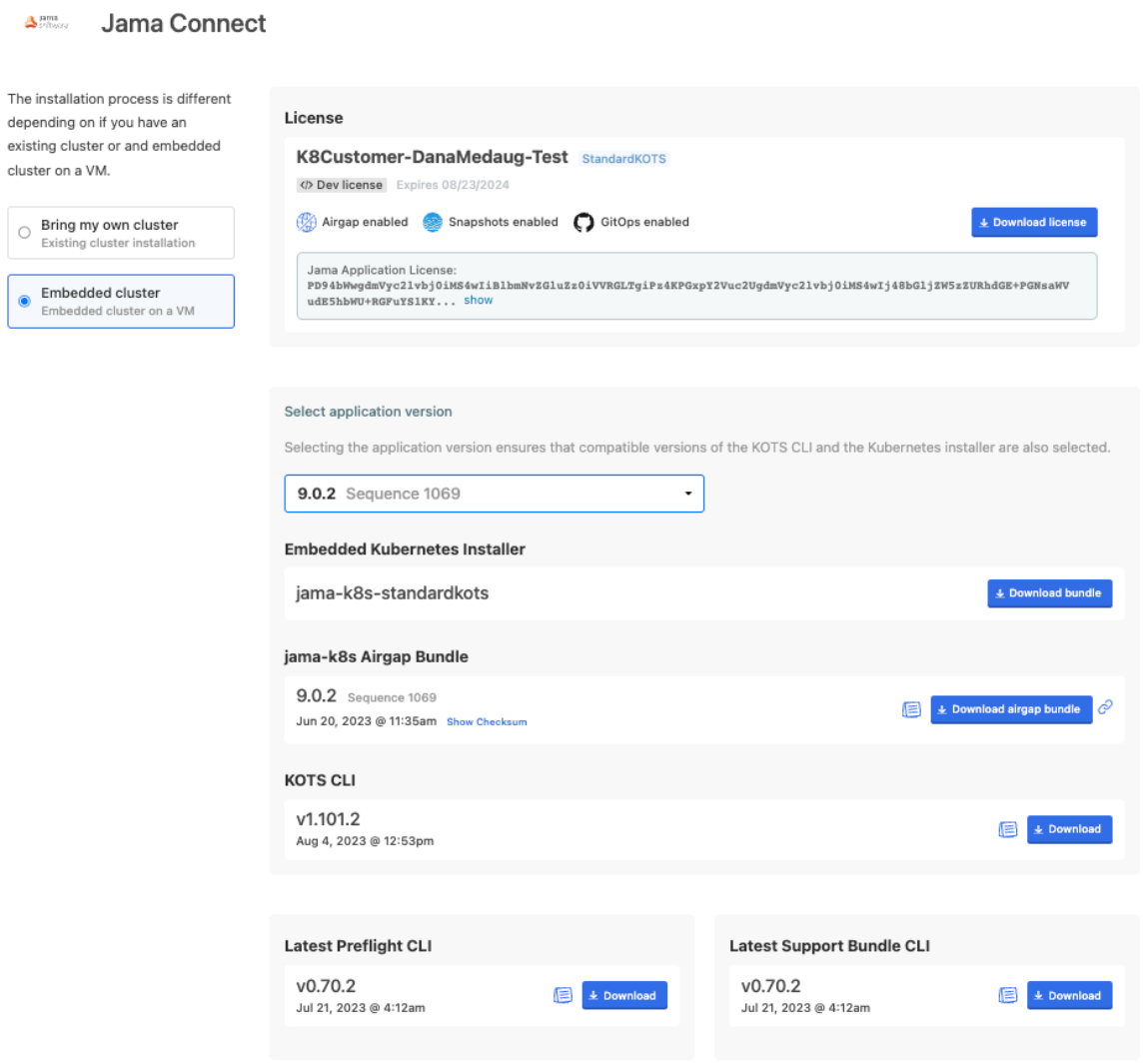
The installation script and the installation wizard guide you through the process of installing the KOTS-required software and Jama Connect, then configuring the KOTS Admin Console.

The following is included in the Welcome email you received from Jama Software:

- License file
- URL to the airgap-safe portal for downloading the Jama Connect application file
- A unique password to access the airgap-safe portal

To install Jama Connect and KOTS:

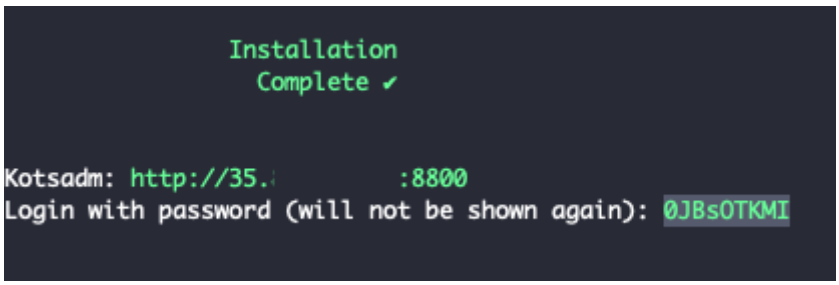
1. Open the Welcome email from Jama Connect and save the attached license file to your local system.
2. Log in to the airgap portal, select **Embedded Cluster**, then download the **jama-k8s Airgap Bundle** and **Embedded Kubernetes Installer** files to your local system.



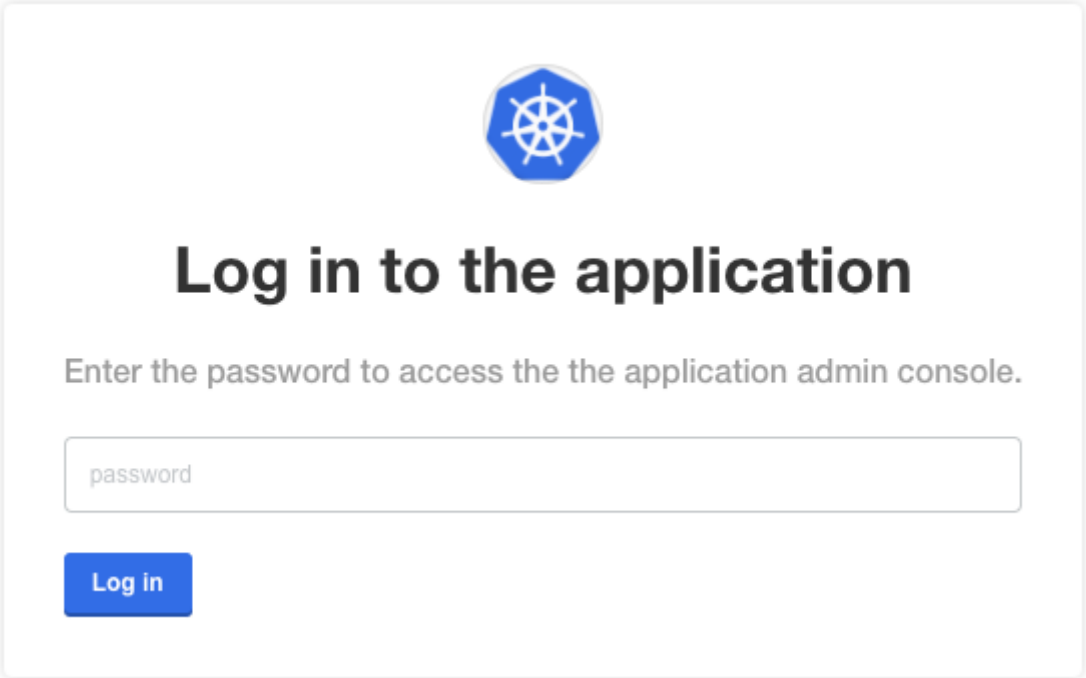
3. Move the downloaded Embedded Kubernetes Installer to your application server.
4. Replace <installer-name> with the name of the downloaded installer in the following commands, then run these commands to extract the installer and run it:

```
tar -zxvf <installer-name>.tar.gz
cat install.sh | sudo bash -s airgap
```

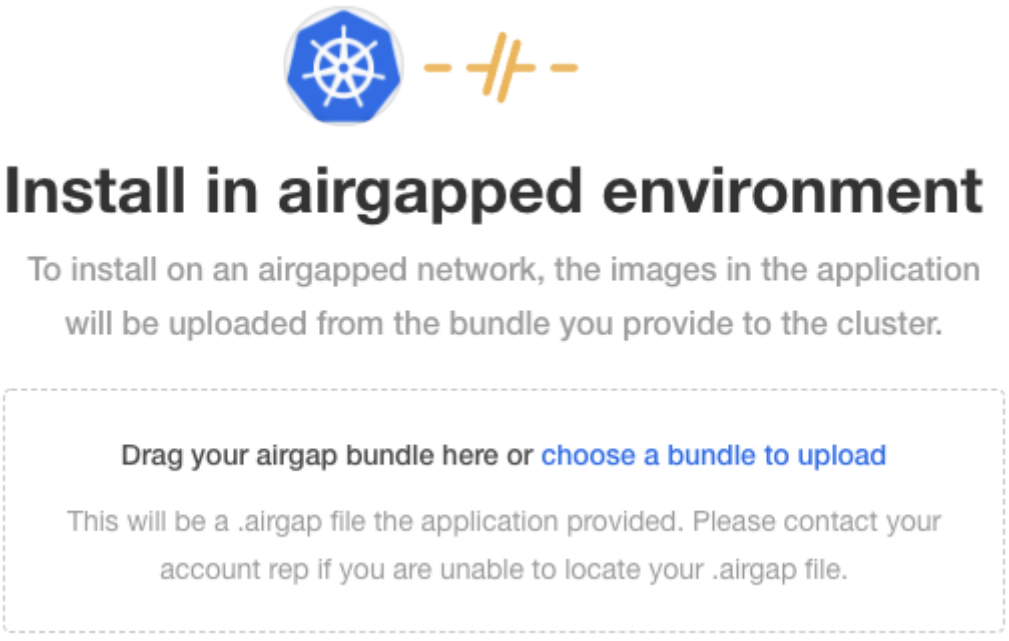
5. After the command runs (which might take several minutes), save the KOTS admin URL, password, and other configuration options for future reference. This is the only time these credentials appear, so make sure you save them.
6. In a supported browser, enter the URL for **Kotsadm**, which was generated when you installed KOTS.



7. Log in to the KOTS Admin Console using the password you just saved.



8. Select the appropriate option:
   - **Have key/certificate** — Select **Choose file** under Private key and Certificate, navigate to the files and select them, then click **Upload & Continue**.
   - **No key/certificate** — Select **Use Self-Signed Cert**.
9. Upload the license file saved on your local system.
10. Upload your jama-k8s airgap bundle, then click **Continue**.



The Config tab in the KOTS Admin Console opens, where you can configure Jama Connect.

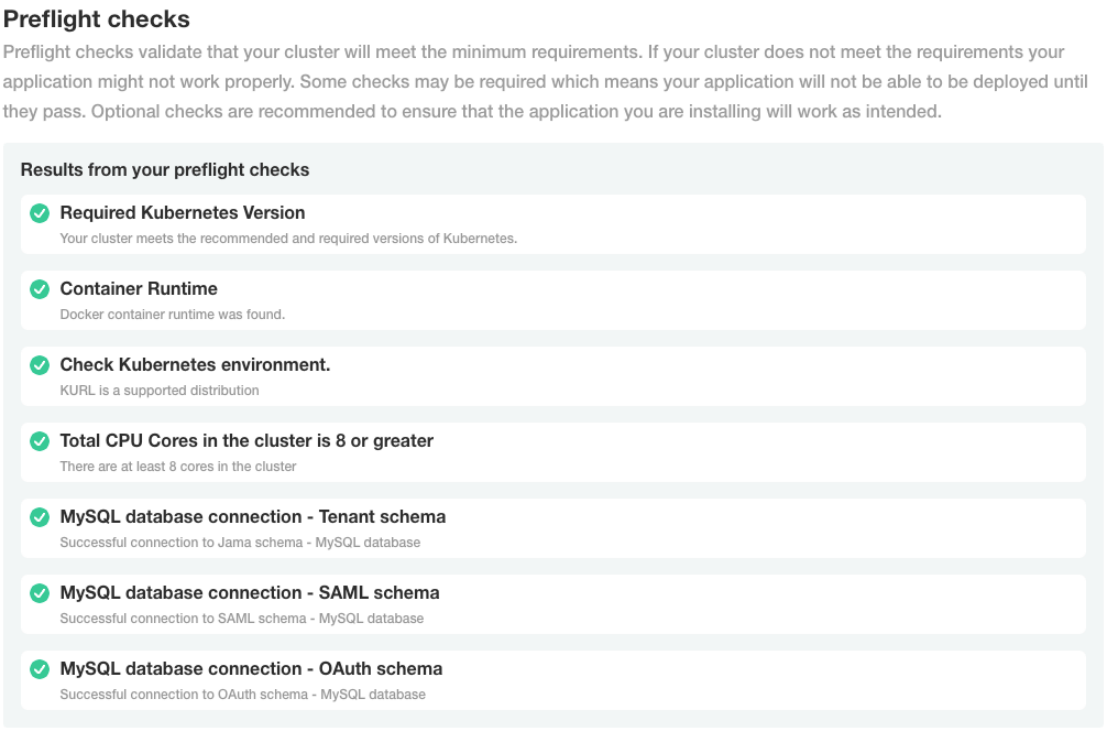
11. Configure the settings for each group, as needed. Scroll down to see each group of settings.
    - **Database Settings** — Select your database type, then use information from Preparing your database server [7] to complete the settings.
    - **Host Name** — Enter the host name for the cluster.

- **TLS Key Pair Source** — (Optional) If you have a custom key and certificate for the host name, select **Custom TLS Configuration**. In the TLS Configuration section, upload the key and certificate.
- **Assets Size** — Enter the estimated size of the assets that you are planning to store in Jama Connect.
- **Elasticsearch Settings > Volume Size** — Enter the amount of disk space that each Elasticsearch node is allowed to use.
- **Tenant Manager Settings** — Enable this setting for optimal performance. Disable this setting if background operations are required before you provision the tenant (for example, when reusing traditional Replicated or using remote Elasticsearch).
  The *Tenant Manager* provisions, restores, upgrades, and sets licenses during application startup.

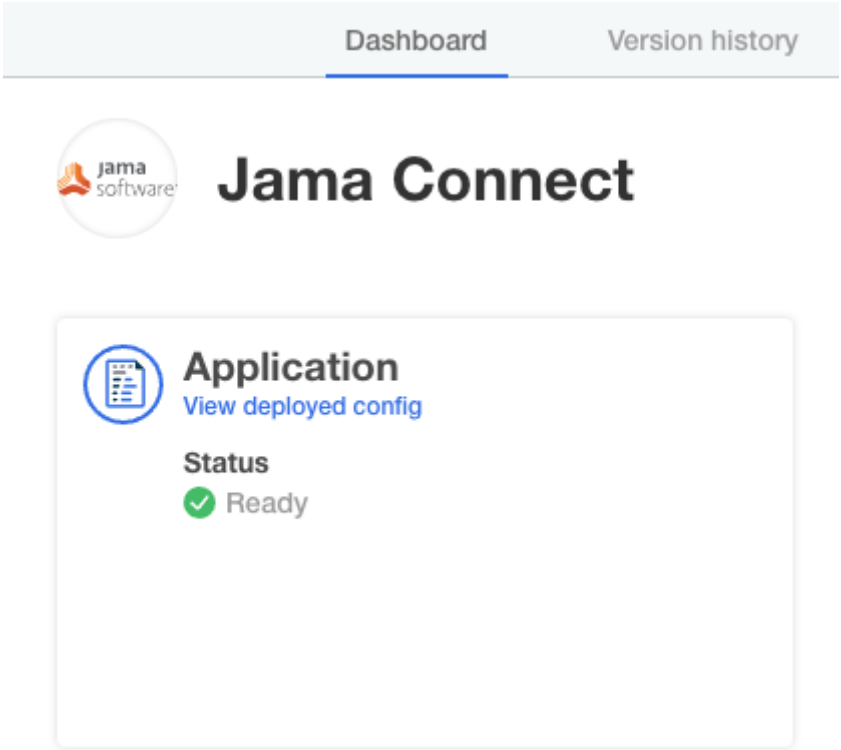
12. (Optional) From the Config tab in the KOTS Admin Console, follow the steps to configure KOTS to save tenant assets in the Amazon EFS [30].

> **NOTE**
> To use Ubuntu 22.04, you must update the memory or Elasticsearch fails. From the KOTS Admin Console, adjust the memory settings so that Maximum Memory is 6G and Maximum Memory for Container is 8G.

13. Scroll to the bottom of the page and click **Save config**.
    The system performs the preflight checks.



14. From the Preflight checks screen, click **Continue** to open the KOTS Admin Console.
15. In the Application section of the dashboard, wait until the status changes to **Ready**.



16. Log in to Jama Connect as root using the hostname configured for Jama Connect.
17. **Important:** Once Jama Connect is installed, use these instructions to provision a Jama Connect dataset [20].

> **IMPORTANT**
> You must provision a Jama Connect dataset [20] before you allow your users access to Jama Connect. If you need the link to the dataset, contact your Customer Success Manager.

Provision your Jama Connect dataset (KOTS)

Although optional, we strongly recommend that you provision an industry dataset. It ensures that your organization has a sample framework as you begin to use Jama Connect.

If you don't provision an industry dataset in your installation, you don't see sample data or an industry framework when you log in and begin using Jama Connect. Otherwise, your use of Jama Connect isn't impacted.

Your purchase confirmation email includes the .jama license file and a link to the industry dataset. If you don't have this link, contact your Customer Success Manager.

Requirements

- Jama Connect must be installed before you provision your dataset. Otherwise, the provisioning will fail.

To provision your dataset:

1. Using the link that was included in your purchase confirmation email, download the .jama license file for the industry dataset.

2. Copy the .jama file to a host system with a node within the KOTS cluster.
3. On the host system, copy the .jama file to the /data/restore directory:

```
kubectl cp -c core <path to .jama archive> default/core-0:/data/restore/
```

4. List the files stored in the /data/restore mount point, along with their permissions:

```
kubectl exec --tty -c core pods/core-0 -- ls -la /data/restore
```

5. Configure the permissions for the file to be read by all users:

```
kubectl exec --tty -c core pods/core-0 -- chmod 644 /data/restore/
<filename>.jama
```

6. Delete the tenant properties file:

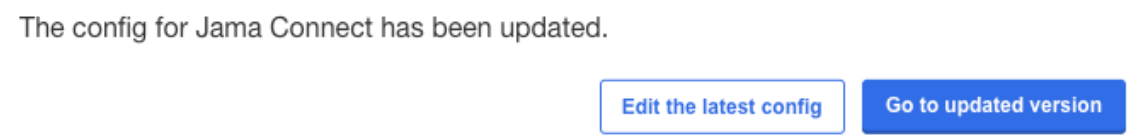
```
kubectl exec --tty -c core pods/core-0 -- rm /home/contour/
tenant_properties/tenant.properties
```

7. Remove the resources:

```
kubectl delete sts/core
kubectl delete job/tenant-manager
kubectl delete pod/hazelcast-0
```

8. Drop the current database and create a new database, SQL Server [9] or MySQL [8], with the same name. If you decide to create a database with a new name, update the database settings in the config tab of the KOTS Admin Console.
9. From the KOTS Admin Console in the Restore Jama Backup section, enter the path to the backup file, then click **Save**.

10. Select **Go to updated version**, then click **Deploy**.



In the Application section of the dashboard, the status changes to **Ready**. The provisioning of your dataset is complete.

Create a Replicated Snapshot (KOTS)

Taking a full snapshot creates a backup of the KOTS Admin Console and application data. It can be used for full Disaster Recovery by restoring over the same instance or in a new cluster. Tenant assets are included in the snapshot. Elasticsearch data is included by default.

A Replicated Snapshot can be taken while Jama Connect is running without interruption.

Requirements

- Replicated Snapshots must be enabled for your Replicated customer license.
- KOTS Admin Console 1.79 and later.
- Replicated Snapshots don't include your database. You must use a proprietary backup/restore system for your type of database, MySQL or SQL Server.

Important considerations

- When restoring from a snapshot in a new cluster, you must reinstall KOTS.
- *Recommended* — Include Elasticsearch data in snapshots to avoid having to reindex your data after performing a restore. However, if your snapshot is not recent, we recommend reindexing your data.
- Replicated Snapshots don't support IAM authentication against EFS. Saving Replicated Snapshots in EFS requires that you use the default file system policy to allow all nodes in the cluster to mount the EFS.

To create a Replicated snapshot:

1. Capture the KOTS installer [23].
2. (Recommended) Include Elasticsearch data in snapshots: From the KOTS Admin Console under the Elasticsearch Settings section, select **Include Elasticsearch in Replicated Snapshots**.
3. *Airgap only* — Capture the IP address of the private registry, which is the IP address value in the Cluster-IP column:

   ```
   kubectl get service/registry -n kurl
   ```
4. Configure the storage destination:
   a. In the KOTS Admin Console, select **Snapshots > Settings & Schedule**.
   b. From the Destination drop-down menu, select a storage destination for your snapshots.
      - **For AWS S3** — The IAM role assigned to the underlaying servers or the user associated with the credentials (access and secret key) must have the Policy template attached.
        Use the following template to create a policy, replacing the **<arn-S3>** parameter with ARN of the S3 bucket. For example: arn:aws:s3:::jama-snapshots.

        ```
        {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Effect": "Allow",
                    "Action": [
                        "s3:PutObject",
                        "s3:GetObject",
                        "s3:AbortMultipartUpload",
                        "s3:DeleteObject",
                        "s3:ListMultipartUploadParts"
                    ],
                    "Resource": "<arn-s3>/*"
                },
                {
                    "Effect": "Allow",
                    "Action": "s3:ListBucket",
                    "Resource": "<arn-s3>"
                }
            ]
        }
        ```

- **For NFS** — If using EFS as an NFS server, the **Server** field = the **DNS name** of the EFS and the **Path** field = a directory inside the EFS, writable by the user:group 1001:1001.

c. Click **Update storage settings** to save your preferences.

5. Schedule Full Snapshots:
   a. In the KOTS Admin Console, select **Snapshots > Settings & Schedule**.
   b. Select **Enable automatic scheduled snapshots**, then click **Update schedule**.
6. Create a Full Snapshot (follow the steps provided by Replicated).

Capture KOTS Installer (KOTS)

When you restore a snapshot in a new cluster, the version of KOTS and its add-ons must match those of the original cluster. Capture each KOTS Installer that was used to create or update your clusters.

Why capture the kurl URL?

A hashed kurl URL (for example, https://kurl.sh/c601b1e) points to a website where you can get the installation script or Kubernetes airgap bundle. Both require you to install the same version of KOTS and add-ons. You must capture this kurl URL because the Replicated Channel URL that was used to install KOTS always pulls the latest KOTS installer that has been promoted. If you rerun the installer from the channel to enable an advanced option or you create a cluster to restore a snapshot, you might accidentally update the KOTS version and it's add-ons.

NOTE

Replicated Vendor maintains a history of every installer that has been promoted to a channel. If for any reason the kurl URL captured in this procedure doesn't work, it can be provided to Jama Software and we might be able to find a similar installer in our KOTS Installer History.

To capture the KOTS Installer:

1. Use the following installer resource information to create a .yaml file named **installer.yaml**:

```
cat <<EOT >> installer.yaml
apiVersion: cluster.kurl.sh/v1beta1
kind: Installer
metadata:
  name: latest
EOT
```

2. Get all installer resources in your cluster, and copy down the name of the installer that you used to download it:

```
kubectl get installers
```

3. Gather the installer details, replacing the **<installer-name>** parameter:

```
kubectl get installers <installer-name> -o yaml
```

4. From the results, copy the **spec** section and paste it at the end of the installer.yaml file that you created.
   The file looks similar to this example of a KOTS Installer:

```
apiVersion: cluster.kurl.sh/v1beta1
kind: Installer
metadata:
  name: latest
```

```
spec:
  certManager:
    version: 1.9.1
  containerd:
    version: 1.6.24
  contour:
    version: 1.25.2
  ekco:
    version: 0.28.3
  flannel:
    version: 0.22.3
  kotsadm:
    applicationSlug: jama-k8s/standardkots
    version: 1.103.3
  kubernetes:
    version: 1.27.6
  metricsServer:
    version: 0.6.4
  minio:
    version: 2023-09-30T07-02-29Z
  openebs:
    isLocalPVEnabled: true
    localPVStorageClassName: local
    version: 3.9.0
  prometheus:
    version: 0.68.0-51.0.0
  registry:
    version: 2.8.3
  velero:
    version: 1.11.1
```

5. Send the installer.yaml file to the create installer API from Replicated to receive a hashed URL:

```
curl -X POST -H "Content-Type: text/yaml" --data-binary
"@installer.yaml" https://kurl.sh/installer && echo ""
```

6. Save the kurl URL that is displayed. It looks similar to https://kurl.sh/c601b1e.

Restore KOTS Admin Console and Jama Connect from a Replicated Snapshot (KOTS)

When you set up a new application server for Jama Connect, you can restore the KOTS Admin Console settings that you saved in a Replicated Snapshot.

Snapshots include the registry images and data for Jama Connect.

Requirements

If restoring to a new cluster, it must match these specifications and settings of the cluster where the backup was taken:

- Number of nodes
- Inbound and outbound traffic rules
- Virtual memory settings for Elasticsearch
- Connectivity to external services and resources (for example, AWS EFS, AWS S3)

To restore from a snapshot:

1. Configure servers for a new cluster:

a. After the servers for the cluster are provisioned, install KOTS on one node using the captured KOTS Installer [23]. You must pass the same flags to the installation script that were passed to the original cluster.
   - **Restoring an online cluster** — Run the appropriate installation script that was generated from the captured KOTS installer.
   - **Restoring an airgap cluster** — Download the appropriate KOTS installer bundle, replacing **<ip>** with the IP address of the private registry from the original cluster:

     ```
     cat install.sh | sudo bash -s airgap kurl-registry-ip=<ip>
     ```

b. When the installer has finished, run the command displayed on the screen so the other servers join the cluster. If required, label the nodes dedicated for Elasticsearch [28].
c. Install any add-ons that were manually installed in the cluster. For example, the EFS Driver.

2. Configure the storage destination: From the KOTS CLI, point the cluster to the storage destination where the Replicated Snapshots were saved.

AWS S3	See configure-aws-s3.
Azure	See configure-azure.
GCP	See configure-gcp.
S3-Other	See configure-other-s3.
NFS	See configure-nfs. If the cluster uses EFS or NFS, also see Configuring an NFS Storage Destination. **Note:** If using EFS as an NFS server, **Server** field = **DNS name** of the EFS and **Path** field = a directory inside the EFS, writable by the user:group 1001:1001.

3. Locate the snapshot and restore it: From the KOTS CLI, run a backup ls and full restore.

   ```
   backup ls
   ```

   This can take a few minutes. If the snapshots don't appear, rerun this command.
4. If you changed the host name for Jama Connect:
   a. Update the Host Name field in the KOTS Admin Console and deploy the change.
   b. From your browser, log in to Jama Connect and change your URL.
5. View scheduled jobs to verify that the list isn't empty.
6. If the Elasticsearch data wasn't included in the snapshot or if the snapshot isn't recent, reindex all search items.
7. Verify that you can locate your assets.

Enable horizontal scaling (KOTS)

To avoid performance issues, you can enable horizontal scaling and add more instances of Jama Connect. For each KOTS node, configure each Jama Connect instance to use more CPU and memory.

Requirements

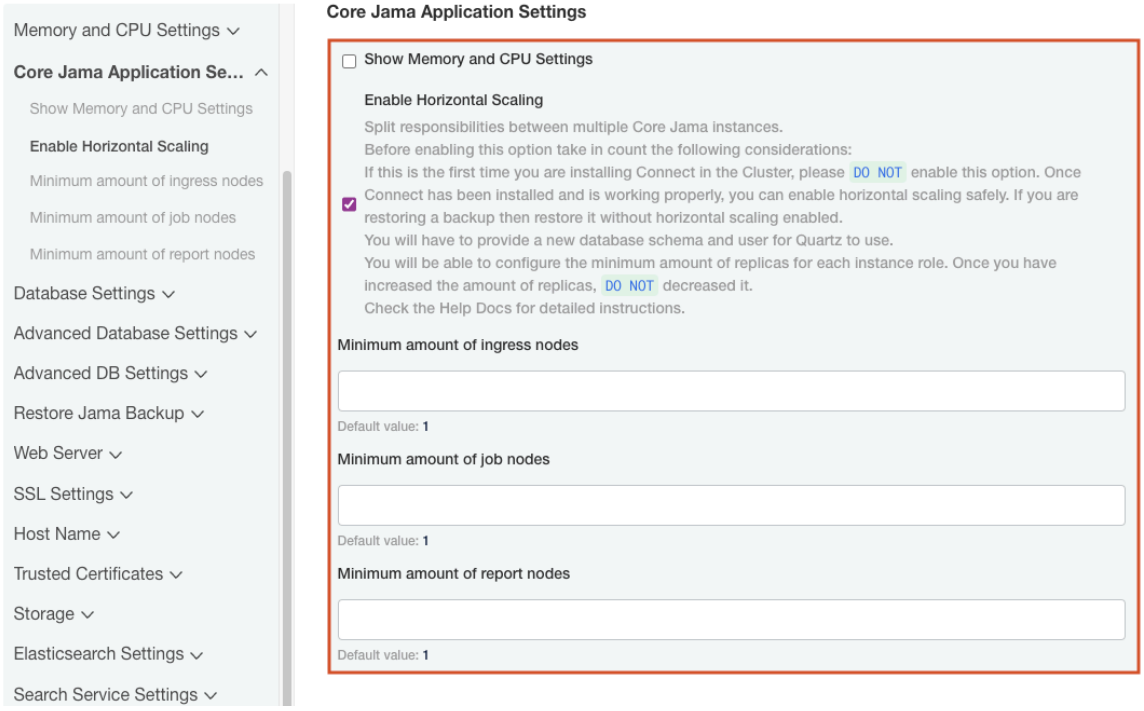
- Jama Connect must already be installed and running before enabling this option.
- If restoring your environment from a backup, restore it without horizontal scaling enabled.
- To use horizontal scaling, you must provide a new database schema and user.
- Once you increase the number of replicas for each instance role, don't decrease the number.

IMPORTANT

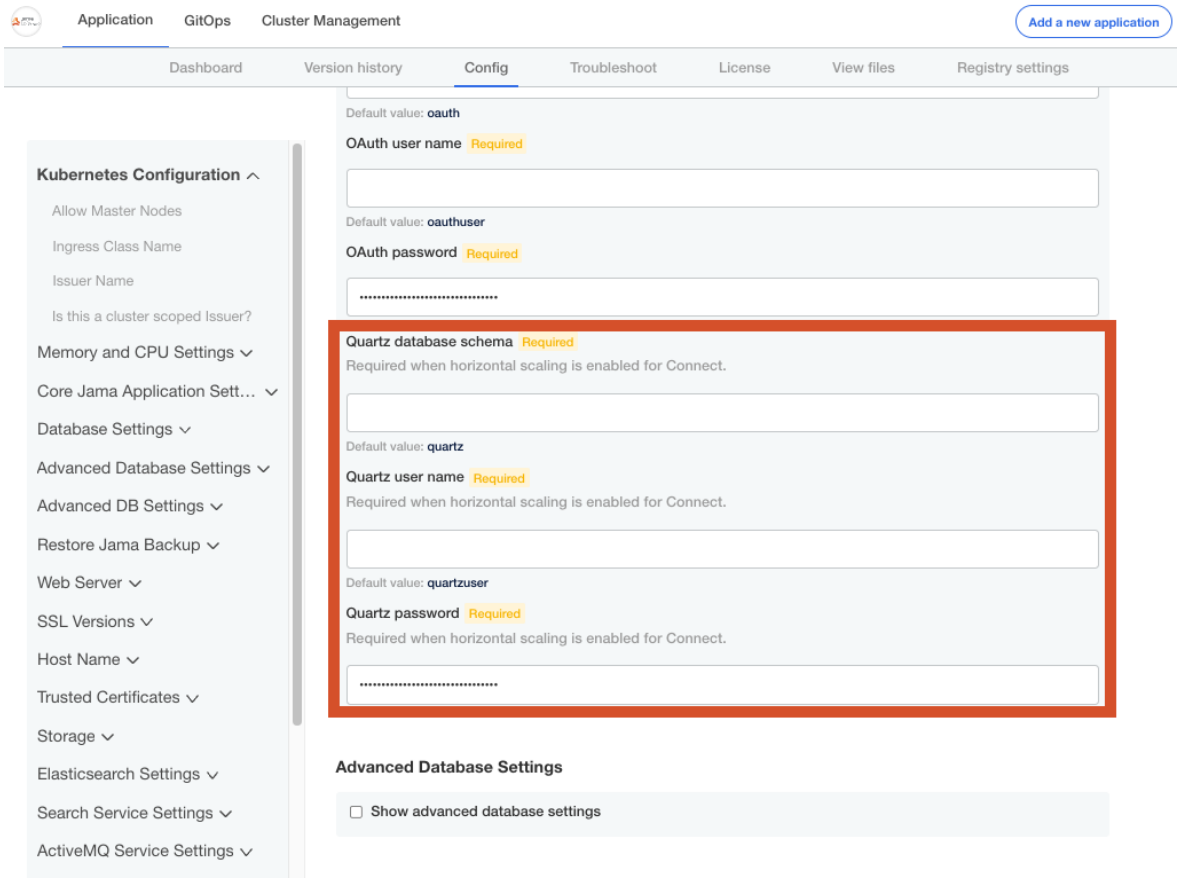
To avoid performance issues, use the recommended requirements [4] for horizontal scaling, rather than minimum requirements.

To enable horizontal scaling:

1. On the KOTS Admin Console, go to the **Config** tab.
2. Scroll to the **Core Jama Application Settings** section, and select **Enable Horizontal Scaling**. Extra fields are displayed for setting horizontal scaling.



3. Specify the number of nodes that you want per role (default is 1). For recommended values, see Application server requirements [4].
   - Minimum number of ingress nodes
   - Minimum number of job nodes
   - Minimum number of report nodes
4. Adjust the maximum memory and CPU for each node. For recommended values, see Application server requirements [4].
5. Scroll down to the Database Settings section and specify the **Quartz database schema** information.



You can use the following scripts as a base to create the schema for Quartz in your database. They were created assuming that you already set up your database [7].
In the scripts, change the schema name, username, or user password to match what you specified in the KOTS Admin Console.

MySQL:	`CREATE DATABASE quartz;`
	`CREATE USER 'quartzuser'@'%' IDENTIFIED BY 'password';`
	`GRANT ALL PRIVILEGES ON quartz.* TO 'quartzuser'@'%';`

Microsoft SQL:	`USE master;`
	`CREATE LOGIN quartzuser with password = 'password';`
	`GO`
	
	`USE jama;`
	`EXEC ('CREATE SCHEMA quartz');`
	`GO`
	
	`USE jama;`
	`CREATE USER quartzuser for LOGIN quartzuser with DEFAULT_SCHEMA=quartz;`
	`GO`
	
	`EXEC sp_addrolemember N'db_owner', quartzuser;`
	`GO`

6. Click **Save config**.
7. Deploy the new version: Select the **Version history** tab and click **Deploy** in the row of your newly configured version.
8. Verify the status of your application: Select the **Dashboard** tab and make sure the status is **Ready**.
9. (Optional) Verify that the new pods are ready:

```
kubectl get pods -o wide
```

Configure dedicated Elasticsearch nodes (KOTS)

Your primary KOTS server (node) is referred to as a KOTS stack. To run Elasticsearch, you must add one or more secondary nodes where Elasticsearch will run, and configure the nodes to run Elasticsearch.

Important considerations

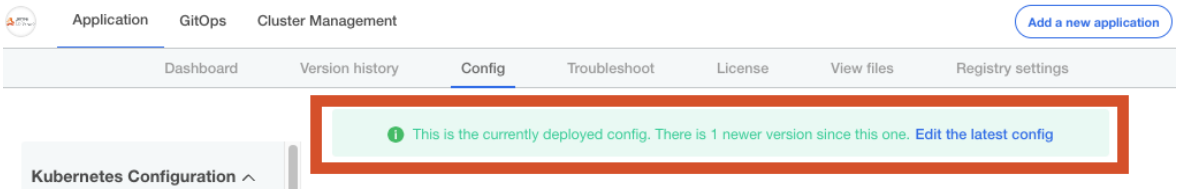
- You must have a functioning primary KOTS server and a secondary server where Replicated KOTS has not yet been installed. For secondary node specifications, see Application server requirements [4].
- This task is appropriate for a new node and an existing node.
- Contact Support to enable remote Elasticsearch for your Replicated license.

To configure your nodes:

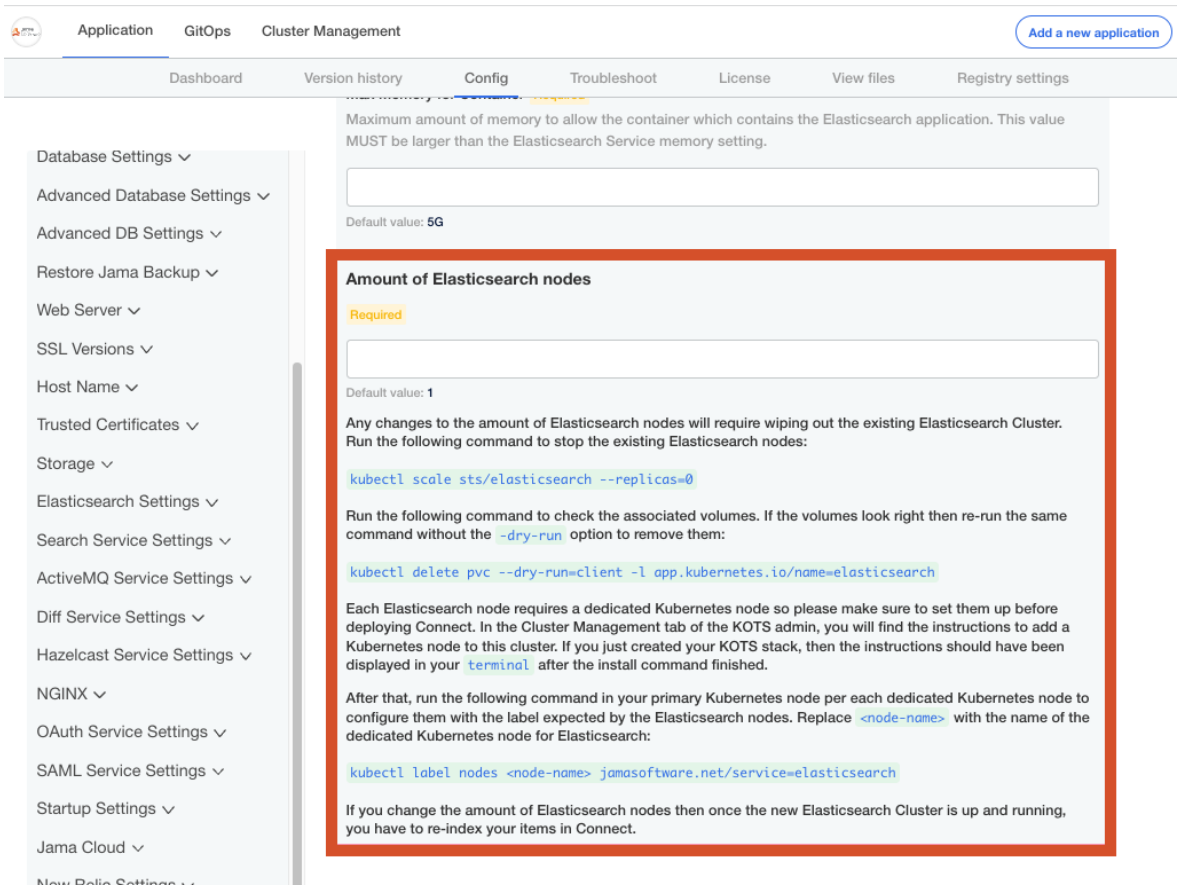
1. Make sure communication is established between primary (KOTS stack) and secondary (where Elasticsearch will run) KOTS nodes. For more information, see Prepare your application server [7].
2. On the secondary node, configure the memory settings for Elasticsearch [12].

```
echo "vm.max_map_count=262144" | sudo tee -a /etc/sysctl.conf
sudo sysctl -p
```

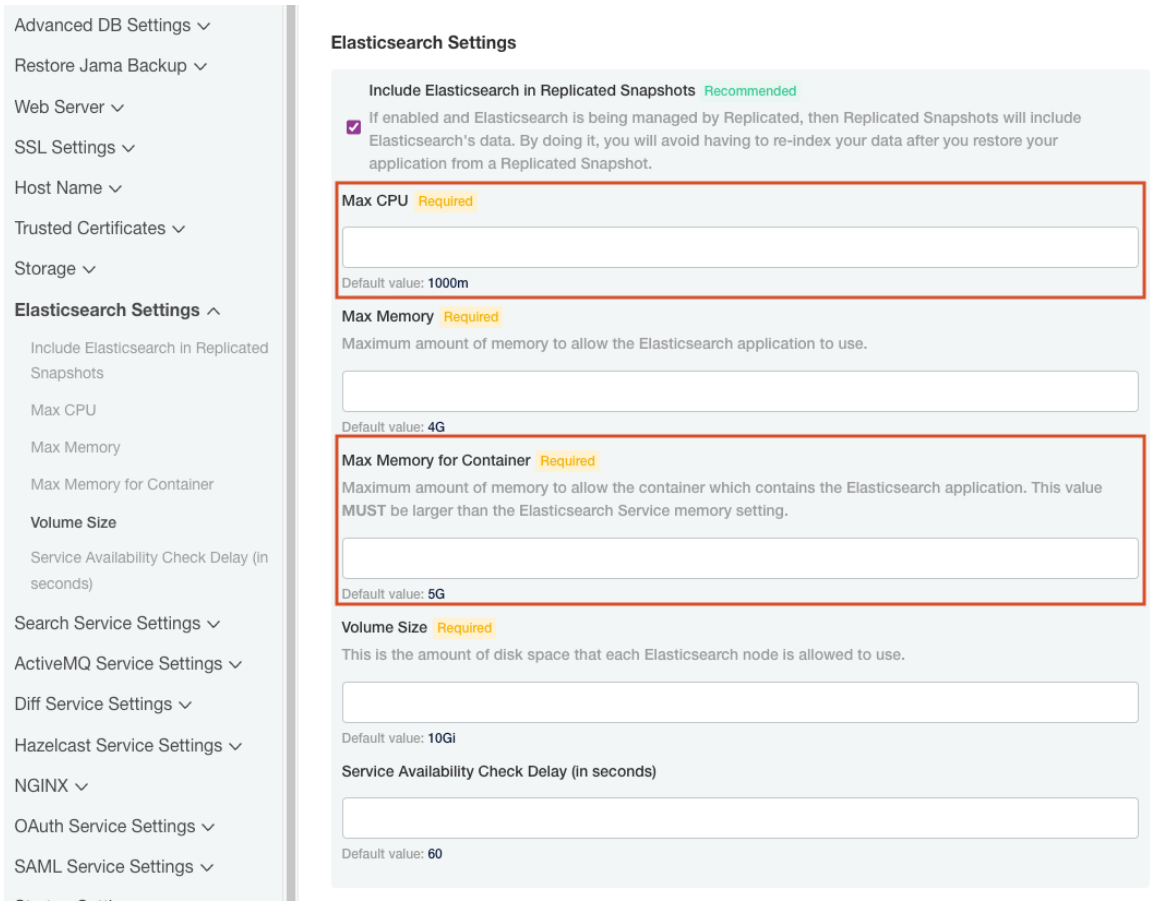
3. On the Admin Console of the primary node below Version history, click **Check for updates** to synch the changes made to your license.
4. Select the **Config** tab and, if you see the following message, click **Edit the latest config**.



5. Scroll down to the **Elasticsearch Settings** section and follow the steps shown to add a dedicated Elasticsearch node.



6. Set the number of Elasticsearch nodes to match the number of dedicated KOTS nodes that you configured.
7. Adjust the maximum memory and CPU that each Elasticsearch node can use based on the specifications of each dedicated KOTS node set up for Elasticsearch. For more information, see Application server requirements [4] and Resource sizing for application server [4].



8. Click **Save config**.
9. Deploy the changes.
10. When the Elasticsearch cluster is up and running, reindex all items.

Configure KOTS to save tenant assets in Amazon EFS

When you configure KOTS to save tenant assets in Amazon EFS, the tenant assets are saved if a cluster fails. EFS provides automatic backups of the tenant assets and EFS is automatically scaled as you add and remove assets.

> **IMPORTANT**
> Complete this task before Jama Connect is deployed. Otherwise, if you want to move your assets to EFS, you must first back up tenant assets to a TAR in KOTS [34].

Requirements

- A KOTS cluster must be up and running.
- You must be able to create and modify these AWS resources: IAM roles, IAM policies, security groups, EC2 instances, and EFS file systems.
- The cluster must have internet access to download the EFS driver and associated containerd images.
- Ports 9909 and 9809 must be available for the EFS driver to function successfully.
- Create a new EFS dedicated to your KOTS stack because each Persistent Volume requires an EFS point, and access points are limited. Currently, each EFS can have a maximum of 120 access points. A dedicated EFS allows you to future-proof the cluster. The steps are provided below.

NOTE
Make sure you copy the Amazon Resource Names (ARNs) for the IAM role that is assigned to the EC2 instances included in the KOTS cluster.

To save tenant assets:

1. Create a new security group for the EFS that allows inbound access for the TCP protocol on the NFS port (2049) from all EC2 instances that are included in the KOTS cluster.
   a. Select a security group that is assigned to the EC2 instances as the source.
   b. Confirm that the EC2 instances included in the KOTS cluster have a security group that allows outbound access on the NFS port to the security group created in the previous step.
2. Create the Amazon EFS file system:
   a. From the Amazon EFS Management Console, select **Create file system**.
   b. In the Create file system page, click **Customize**.
   c. On the File systems setting page, configure the following, then click **Next**:
      - **Name** — Enter a name that allows you to easily identify the EFS.
      - **Availability and Durability** — Regional.
      - **Automatic backups** — Enable automatic backups during off hours to avoid backup inconsistencies.
      - **Performance mode** — General Purpose.
      - **Throughput mode** — Bursting.
      - **Encryption** — Enable encryption of data at rest.
   d. On the Network access page, configure the following, then click **Next**:
      - **Virtual Private Cloud (VPC)** — Enter the name of the VPC where the KOTS cluster is running.
      - **Mount targets** — Verify that a mount target is created per Availability Zone, then assign the security group you created earlier.
   e. Using the template below, generate a file system policy for the EFS, replacing the **<arn-cluster-role>** parameter with the ARN of the cluster role. Then, attach the policy to the EFS.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Principal": {
        "AWS": "*"
      },
      "Action": "*",
      "Condition": {
        "Bool": {
          "aws:SecureTransport": "false"
        }
      }
    },
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "<arn-cluster-role>"
      },
      "Action": [
```

```
        "elasticfilesystem:ClientRootAccess",
        "elasticfilesystem:ClientWrite",
        "elasticfilesystem:ClientMount"
      ]
    }
  ]
}
```

f. On the Review and create page that opens, review the file system configuration groups, then select **Create** to create your file system and return to the File systems page.

3. Edit the cluster role:

a. Generate the ARN of the newly created EFS, replacing **<region>**, **<account-id>**, and **<file-system-id>** parameters:

```
arn:aws:elasticfilesystem:<region>:<account-id>:file-system/<file-system-id>
```

b. Generate the ARN for the access points, replacing **<region>** and **<account-id>** parameters:

```
arn:aws:elasticfilesystem:<region>:<account-id>:access-point/*
```

IMPORTANT

The template must be used as is, with the policy targeting all access points.

c. Use the following template to create a new IAM policy, replacing the **<arn-efs>** with the ARN generated in step 3a and replacing the **<arn-access-points>** parameters with the ARN generated in step 3b. Then, attach the new policy to the cluster role.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeAvailabilityZones"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "elasticfilesystem:DescribeAccessPoints",
        "elasticfilesystem:DescribeFileSystems",
        "elasticfilesystem:DescribeMountTargets"
      ],
      "Resource": "<arn-efs>"
    },
    {
      "Effect": "Allow",
      "Action": "elasticfilesystem:CreateAccessPoint",
      "Resource": "<arn-efs>",
      "Condition": {
        "StringLike": {
          "aws:RequestTag/efs.csi.aws.com/cluster": "true"
        }
      }
```

```
    },
    {
      "Effect": "Allow",
      "Action": "elasticfilesystem:DeleteAccessPoint",
      "Resource": "<arn-access-points>",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/efs.csi.aws.com/cluster": "true"
        }
      }
    }
  ]
}
```

4. Install the AWS EFS driver version **1.3.8** in your cluster:

```
kubectl apply -k "github.com/kubernetes-sigs/aws-efs-csi-driver/deploy/
kubernetes/overlays/stable/?ref=tags/v1.3.8"
```

The following containerd images are downloaded to your EFS driver: amazon/aws-efs-csi-driver, public.ecr.aws/eks-distro/kubernetes-csi/node-driver-registrar, public.ecr.aws/eks-distro/kubernetes-csi/livenessprobe.

5. Verify that the driver was successfully installed:

```
kubectl get daemonset.apps/efs-csi-node csidriver/efs.csi.aws.com
deployments/efs-csi-controller -n kube-system
```

```
NAME                          DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODE SELECTOR            AGE
daemonset.apps/efs-csi-node   1         1         1       1            1           beta.kubernetes.io/os=linux   26m

NAME                                ATTACHREQUIRED   PODINFOONMOUNT   STORAGECAPACITY   TOKENREQUESTS   REQUIRESREPUBLISH
csidriver.storage.k8s.io/efs.csi.aws.com   false            false            false             <unset>         false

NAME                                READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/efs-csi-controller   1/2     2            1           26m
```

6. Enable the EFS Storage Class.
   a. Log in to the KOTS Admin Console, select the **Config** tab, then scroll to the **AWS Resources** section.
   b. Select **Enable EFS Storage Class**.
   c. In the AWS EFS Storage Class section under File System ID, enter the ID of the newly created EFS.
7. Remove existing PVC and assets.

> **IMPORTANT**
> If Jama Connect has been deployed and you want to move your assets to EFS, you must first back up tenant assets to a TAR in KOTS [34].

   a. From the primary node, delete the StatefulSets of the core pods:

```
kubectl delete sts/core sts/core-ingress sts/core-reports sts/core-
jobs
```

   b. Delete the PVC that contains the assets, so that a new PVC can be created that points to EFS:

```
kubectl delete pvc/tenantfs
```

8. Save assets in EFS.
   a. From the KOTS Admin Console, scroll to the Storage section, then in the **Assets Storage Class** field, enter the name assigned to the EFS Storage Class.
   b. Save your changes and deploy Jama Connect.

c. (Optional) Once the core pods are running, restore tenant assets from a TAR in KOTS [34].

9. Run this command:

```
kubectl get pvc/tenantfs
```

The output displays **storage class** as the name assigned to the EFS storage class.

Back up tenant assets to a TAR file in KOTS

If Jama Connect was deployed to KOTS and you want to move your assets to EFS, you must first back up tenant assets to a Tape Archive file (TAR) in KOTS.

IMPORTANT

You must have a core-0 pod running, unless you have horizontal scaling enabled for jamacore, then a core-ingress-0 pod is running.

To back up tenant assets:

1. Set an environment variable with your tenant name:

```
export TENANT_NAME=jama
```

2. Copy the assets from a core pod to an **assets** local directory in the KOTS node. To reduce the backup size, exclude the **tempreports**.

```
kubectl cp -c core default/core-0:/home/contour/tenant/${TENANT_NAME}/
attachments assets/attachments
kubectl cp -c core default/core-0:/home/contour/tenant/${TENANT_NAME}/
avatars assets/avatars
kubectl cp -c core default/core-0:/home/contour/tenant/${TENANT_NAME}/
diagrams assets/diagrams
kubectl cp -c core default/core-0:/home/contour/tenant/${TENANT_NAME}/
equations assets/equations
kubectl cp -c core default/core-0:/home/contour/tenant/${TENANT_NAME}/
reports assets/reports
kubectl cp -c core default/core-0:/home/contour/tenant/${TENANT_NAME}/
tempreports assets/tempreports
```

3. List the contents of the assets directory inside the core pod:

```
kubectl exec --tty -c core pods/core-0 -- ls -la /home/contour/tenant/$
{TENANT_NAME}/
```

4. Verify that the commands from step 2 included every folder and file displayed.
5. Create a TAR file from the local directory:

```
tar -zcvf assets.tar.gz assets/
```

6. Copy the TAR file from the node to a different system and review its content:

```
scp <user>@<ip-another-machine>:<destination-path> assets.tar.gz
```

You now have a backup file that includes all the assets.

Restore tenant assets from TAR in KOTS

Follow this process when you have an existing cluster and want to save your tenant assets on an external storage device.

Requirements

- In EFS, the tenant assets must be backed up in a TAR file [34] and restored once EFS has been configured.
- Make sure that you use our process to create the TAR file; the restore commands expect a TAR file with a certain structure.

To restore tenant assets:

1. Set an environment variable with your tenant name:

```
export TENANT_NAME=jama
```

2. Copy the TAR file from its current location to a master node:

```
scp assets.tar.gz <user>@<ip-master-node>:~/assets.tar.gz
```

3. Log in to the master node and extract the TAR file:

```
tar -xvzf assets.tar.gz
```

4. Copy the assets to a core pod:

```
cd assets
kubectl cp -c core . default/core-0:/home/contour/tenant/${TENANT_NAME}/
kubectl exec --tty -c core pods/core-0 -- chmod -R 755 /home/contour
kubectl exec --tty -c core pods/core-0 -- chown -R tomcat:tomcat /home/
contour
```

5. Verify that the assets were copied:

```
kubectl exec --tty -c core pods/core-0 -- ls -la /home/contour/tenant/$
{TENANT_NAME}/
kubectl exec --tty -c core pods/core-0 -- du -shc /home/contour/tenant/$
{TENANT_NAME}/
```

Configure Federated Authentication for KOTS Admin Console

By default, you can log in to the KOTS Admin Console with a shared password. To improve security, configure this feature so that KOTS admin authentication is managed by your Identity Provider.

Requirements

- You must have the KOTS software installed.
- Identity Service must be enabled by Jama Software Support for your Replicated license.
- You must have an Identity Provider that is compatible with OpenID.

Important considerations

- When you enable identity provider access to the KOTS Admin Console, shared password authentication is disabled. To reset authentication and reenable shared password authentication:

```
kubectl kots identity-service enable-shared-password --namespace default
```

To configure Federated Authentication:

1. Update the KOTS license if Support enabled Identity Service for your license.
2. Connect KOTS Admin Console to the Identity Provider.
   a. Log in to the KOTS Admin Console, then select the **Access** tab.
   b. In the Configure Identity Provider section:
      - Verify that the Admin Console URL matches the URL for your KOTS Admin Console.
      - **Connector name** — Enter a name that works best for your team.

- **Issuer** — Enter the Issuer or OpenID Configuration URL from your IdP application.
- **Client ID** and **Client secret** — Enter the Client ID and Client Secret from your IdP application.

c. Select the **Access** tab to expand the Advanced options menu, complete the following, then click **Save provider settings**:
   - **Scopes** — Enter the OpenID, profile, and email.
   - **Skip email verification** — Enable or disable this option based on your organization's needs and IdP support.
   - **Remaining fields** — Use the default values.

d. Click **Logout**.

You are redirected to a new login screen, where you can log in to Jama Connect. If a "Failed to attempt login" error appears, see Troubleshooting KOTS errors [36].

After installing Jama Connect (KOTS)

Whether your environment is internet-enabled or airgap, after you install Jama Connect you can continue to set up your Jama Connect environment.

Follow any post installation instructions that apply to your organization.

The setup tasks to configure your environment include:

- Add Organization Admin account
- Modify organization details
- Configure email/collaboration settings
- Configure user authentication
- Create XML backups (optional)
- Update the license for KOTS environments (optional)

If you have further questions about Jama Connect installation and setup, visit the Jama Support Community or contact Support.

Troubleshooting your installation (KOTS)

If you run into problems with your KOTS installation, here are some resources that might help.

- Connection errors [37]
- Federated Authentication errors [37]
- Backup and restore errors [38]
- Installation errors [36]
- Generate a support bundle [38]

Installation errors (KOTS)

If any errors occurred during installation, use this table to fix the issues.

Error message	Solution
This webpage is not available	Verify that the "Host Name" section of the settings was correctly entered to point to the application server.
Not private or *Not secure*	This might happen if you chose a self-signed certificate or uploaded an invalid certificate. Verify that you correctly entered the **Custom TLS configuration** in the **Host Name** window. If this happens only for other users and not the system administrator, and the Admin Console is using a self-signed certificate, you might have already told your web browser to "Proceed ... (unsafe)" or "Add exception," while other users haven't. Verify that you selected the setting you want for **Reuse admin console TLS configuration** in the **Host Name** window.

Error message	Solution
Problem: Cannot create database jama: Connections could not be acquired from the underlying database!	Most likely, something is wrong with your Admin Console database settings (for example, bind-address, DBO credentials), or the connection between the application server and the database server. Double-check your database settings in the Admin Console.

Connection errors (KOTS)

The KOTS installation process includes using the kubectl command line tool. If you see an error message that relates to kubectl, use the workaround tips for the issue.

Error message	Reason	Workaround
The connection to the server localhost:8080 was refused - did you specify the right host or port? error: error loading config file "/etc/kubernetes/admin.conf": open /etc/kubernetes/admin.conf: permission denied	kubectl might not be configured properly for the user and/or node where you tried to use it.	• **Wrong user** — Switch to the user that installed KOTS or to root (**sudo su -**), then rerun the **kubectl** command. • **Wrong node** — Switch to the server where KOTS was installed initially or to a primary node, then rerun the **kubectl** command. kubectl can be configured for other users and nodes but it requires some research. See Embedded Cluster: How to get kubectl working for other users.
Waited for 1.184446141s due to client-side throttling, not priority and fairness	When using kubectl with Ubuntu 18.04, you might see this warning message.	Log in to Jama Connect as the root user (**sudo su -**).
Application status is not accurate	This error usually occurs when the cluster is restarted or if a cluster was restored from a snapshot. The Application status in the KOTS Admin Console might differ from what you see in the UI when using kubectl.	Redeploy the latest license version from the Version history tab in the KOTS Admin Console.

Federated Authentication errors (KOTS)

The "Failed to attempt login" error can occur when you log in to the KOTS Admin Console from a browser for the first time. This error can occur if you didn't specify the host name and chose to upload custom certificates, or you specified the host name but it wasn't retained by the KOTS Admin Console.

To resolve this issue:

If provided, the KOTS Admin Console uses the custom certificate. If none was provided, a new self-signed certificate is generated with the host name you specified. The KOTS Admin Console retains the host name.

1. Review the KOTS Admin pod logs:
   a. Check the name for your KOTS Admin pod:

      ```
      kubectl get pods -o wide
      ```

   b. Check the logs for your KOTS Admin pod:

      ```
      kubectl logs -f pods/<kotsadm-pod-name>
      ```

      Review the logs and confirm that the following error appears:

      ```
      {
        "level": "error",
        "ts": "2022-08-25T18:36:03Z",
        "msg": "failed to get kotsadm oidc provider: failed
      to query provider \"https://<your-kots-admin-hostname>:8800/dex\":
      Get \"https://<your-kots-admin-hostname>:8800/dex/.well-known/openid-
      ```

```
configuration\": x509: certificate is valid for kotsadm,
kotsadm.default, kotsadm.default.svc, kotsadm.default.svc.cluster,
kotsadm.default.svc.cluster.local, not <your-kots-admin-hostname>"
}
```

2. Restore the ability to configure the TLS certificates:

```
kubectl -n default annotate secret kotsadm-tls acceptAnonymousUploads=1
--overwrite
```

3. Restart the kurl-proxy pod:

```
kubectl delete pod $(kubectl get pod | grep kurl-proxy | awk '{print
$1}')
```

4. Open the KOTS Admin Console with this link: http://<your-kots-admin-hostname>8800/tls
5. Choose one:
   - Select **Skip & continue** if you don't want to provide custom certificates.
   - Upload the files and select **Upload & continue** if you want to provide custom certificates.

Backup and restore errors (KOTS)

Replicated has documented the following scenarios. For more information, see Troubleshooting Backup and Restore.

Error message	Reason	Workaround
Error executing hook	When a cluster is restarted, some pods might be in a **Shut-down** state, meaning they were likely replaced by new pods.	Delete the pods that are in a **Shut-down** state: `kubectl delete pods/<pod-name>`
Connect is not reach-able after a restore even when pods are ready	If you restored a cluster on a new server with a different host name than the original, and updated the Host Name field in the KOTS Admin Console and deployed it, the httpproxy resource for nginx might not have been updated.	Delete the httproxy resource for nginx and redeploy it: `kubectl delete httpproxy/nginx`

Generate a support bundle (KOTS)

To troubleshoot and diagnose problems with application deployments, you can generate a support bundle to collect and analyze data from your environment.

Jama Support uploads the support bundle to the Replicated vendor portal to view and interpret the analysis, and can open a support request ticket if needed. Severity 1 issues are resolved three times faster when submitted with support bundles.

1. For internet environments, generate a support bundle from the CLI:
   a. Log in to the KOTS Admin Console, then select the **Troubleshoot** tab.
   b. Scroll down to the Analyze Jama Connect for support section, then click **If you'd prefer to get a command to manually generate a support bundle**.
   A cURL command appears.
   c. Copy the command.
   d. From the CLI, run the command to generate a support bundle.
2. For airgap environments, generate a support bundle from the CLI:
   a. Log in to the KOTS Admin Console, then select the **Troubleshoot** tab.
   b. Scroll down to the Analyze Jama Connect for support section, then click **If you'd prefer to get a command to manually generate a support bundle**.
   A cURL command appears.
   c. Remove the following code from the cURL command:

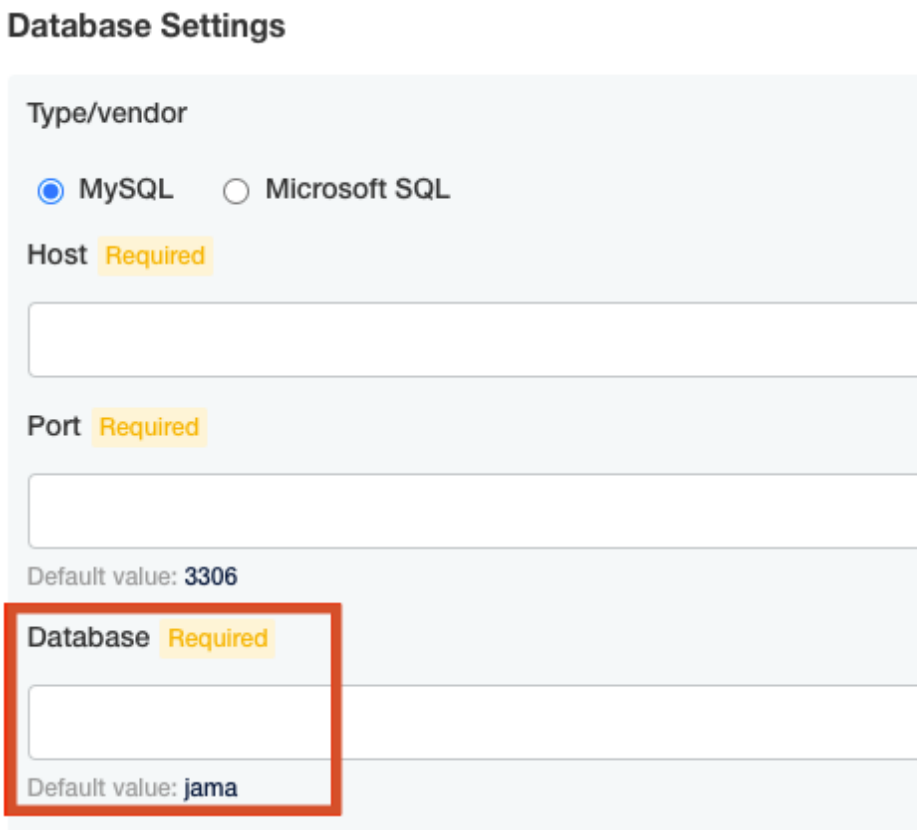
```
curl https://krew.sh/support-bundle | bash
```

   Your command looks like this:

```
kubectl support-bundle secret/default/
kotsadm-jama-k8s-supportbundle --redactors=configmap/default/
kotsadm-redact-spec/redact-spec,configmap/default/kotsadm-jama-k8s-
redact-spec/redact-spec
```

d. Copy the command.
e. From the CLI, run the command to generate a support bundle.

KOTS FAQ

Question	Answer
What is my tenant name?	Your tenant name is the text you entered as the database name from the Config tab in the KOTS Admin Console. 
How can I find the name of a node?	Run this command, then check the **Name** column: `kubectl get nodes -o wide`
How do I shut down my cluster?	Ideally, your cluster is always up and running. If all nodes require maintenance, shut down and perform maintenance on one node at a time. The KOTS installer deploys EKCO, which is a utility tool to perform maintenance operations on the cluster. Run this command to prepare the node for a reboot: `sudo /opt/ekco/shutdown.sh` When the process is finished, shut down the node.
Does Jama Connect support NFS?	If running Jama Connect in AWS, you can configure the application to save your tenant assets in EFS, or configure KOTS to save Replicated Snapshots to an NFS server.

Upgrading Jama Connect (KOTS)

Upgrading Jama Connect to 8.79.6, 9.0.4, or 9.6.x requires that you first update the Jama Connect KOTS platform. The updated KOTS platform optimizes how data is stored in Jama Connect and how KOTS resources communicate with one another.

IMPORTANT
Upgrading your current environment involves significant maintenance downtime and requires that you have a recovery plan in case you need to revert to the original environment. Instead, we recommend that you install a new Jama Connect environment (referred to as a *clean installation*), then copy elements of your current environment to the new environment.

Here are the supported upgrade scenarios:

- ***(Recommended)* Clean installation of Jama Connect KOTS** — This recommended scenario requires that you install a clean Jama Connect KOTS instance on a new application server, then copy data assets and the tenant.properties file from your current environment to the new environment. The new instance must point to a restored backup of your database.
- **In-place upgrade of Jama Connect KOTS** — This scenario requires upgrading your current environment in place, which involves significant maintenance downtime and requires that you have a recovery plan in case you need to revert to the original version. You must run a pre-upgrade script before running the Kubernetes (kURL) installer.

Recommended upgrade paths

Use this table to determine the best upgrade path for your organization.

If your Jama Connect instance is running this version...	Upgrade to one of these versions...
8.79.x	8.79.6 9.0.4 9.6.x
9.0.x	9.0.4 9.6.x

Perform a clean installation of Jama Connect

Whether your environment is internet-enabled or airgapped, we recommend that you install a new Jama Connect environment (referred to as a *clean installation*) to support new versions of the Jama Connect application.

The process includes using a new application server and a database instance that was restored from a backup of your current production instance. Once the new environment is up and running, you must copy elements of your current environment to the new environment (move from one KOTS environment to another KOTS environment).

To perform a clean installation:

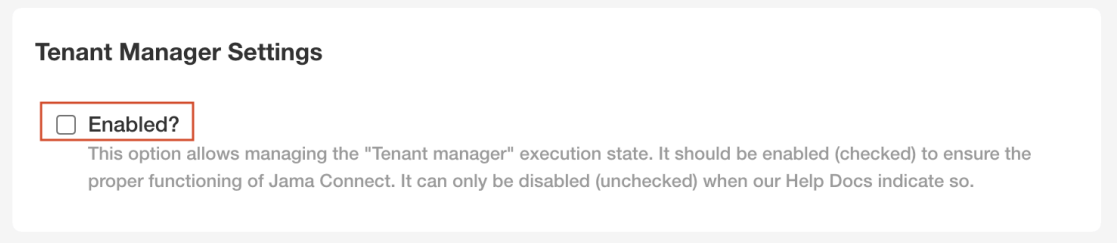
1. Install the KOTS software.

2. Provision your tenant in Jama Connect KOTS:
   a. From the KOTS Admin Console, select the **Config** tab.
   b. Configure the settings for each group, as needed. Scroll down to see each group of settings.

> **IMPORTANT**
> Use the settings from your current environment as a guide when configuring the new settings.
>
> Make sure that the current Host name, Database name, Username, and Password are configured correctly in the KOTS Admin Console. When you install KOTS in a new environment, you must point to the newly installed database host or the deployment fails.

   - **Database Settings** — Select your database type (**MySQL** or **Microsoft SQL Server**), then use the information from Preparing your database server to complete the settings.
   - **Host Name** — Enter the base URL for Jama Connect. Ensure this domain name is routable on your network.
   - **TLS Key Pair Source** — (Optional) If you have a custom key and certificate for the host name, select **Custom TLS Configuration**. In the TLS Configuration section, upload the key and certificate.
   - **Assets Size** — Enter the estimated size of the assets based on the current data assets size of your environment and its projected growth.
   - **Elasticsearch Settings > Volume Size** — Enter the amount of disk space that each Elasticsearch node is allowed to use.

   c. Scroll down to Tenant Manager Settings and deselect the **Enabled** checkbox to disable it. Disabling the tenant manager allows you to pause provisioning while copying data assets and tenant.properties from the existing KOTS environment to the new KOTS environment.



   d. Scroll to the bottom of the page and click **Save config**.
   The preflight checks run.
   e. From the Preflight checks screen, click **Deploy** to deploy the Jama Connect application and services.
   When the system is available, the status in the KOTS Admin Console changes to **Ready**.
   The deployment process can take at least an hour.
   f. From the application server CLI, verify that the Kubernetes pods were successfully created:

```
kubectl get pods
```

   The status of the pods change to ready and running.

```
root@ip-              kubectl get pods
NAME                                READY   STATUS      RESTARTS   AGE
activemq-0                          1/1     Running     0          79m
connect-drainer-s7ncs               0/1     Completed   0          79m
core-0                              1/1     Running     0          79m
diff-0                              1/1     Running     0          79m
elasticsearch-0                     1/1     Running     0          79m
hazelcast-0                         1/1     Running     0          79m
kotsadm-684954474d-dvjgl            1/1     Running     0          120m
kotsadm-rqlite-0                    1/1     Running     0          120m
kurl-proxy-kotsadm-68d64bf84c-gq29x 1/1     Running     0          120m
nginx-0                             1/1     Running     0          79m
oauth-0                             1/1     Running     0          79m
saml-0                              1/1     Running     0          79m
search-0                            1/1     Running     0          79m
root@ip-
```

g. When the pods are ready and running, copy the data assets and tenant.properties file from the original instance to an accessible location on the new application server (data assets are located in the core-0 pod under /home/contour/tenant/jama).
   - Create a TAR file of the data assets:

     ```
     tar -zcvf assets.tar.gz avatars/ attachments/ diagrams/ reports/
     equations/ tempreports/
     ```

     The tenant.properties file is located in the core pod under /home/contour.

h. On the application server, set the tenant name for the environment (the tenant name, usually jama, and can be found in the tenant.properties file that you preserved earlier):

   ```
   export TENANT_NAME=<tenant_name>
   ```

i. On the application server:
   i. Copy and extract the previously preserved data assets into the running core pod and change the ownership permissions:

      ```
      kubectl cp -c core /tmp/contour/assets.tar.gz default/core-0:/
      home/contour/tenant/${TENANT_NAME}/
      kubectl exec --tty -c core pods/core-0 -- tar -xvzf /
      home/contour/tenant/${TENANT_NAME}/assets.tar.gz -C /home/contour/
      tenant/${TENANT_NAME}/
      kubectl exec --tty -c core pods/core-0 -- chmod -R 755 /home/
      contour
      kubectl exec --tty -c core pods/core-0 -- chown -R tomcat:tomcat
      /home/contour
      ```

   ii. Copy the previously preserved tenant.properties file into the running core pod and change the ownership permissions:

      ```
      kubectl cp -c core tenant.properties default/core-0:/home/contour/
      tenant_properties/tenant.properties
      kubectl exec --tty -c core pods/core-0 -- chmod -R 755 /home/
      contour
      kubectl exec --tty -c core pods/core-0 -- chown -R tomcat:tomcat
      /home/contour
      kubectl exec --tty -c core pods/core-0 -- cat /home/contour/
      tenant_properties/tenant.properties
      ```

   iii. Delete the core stateful set to recreate the core pod:

      ```
      kubectl delete sts/core
      ```

j. From the KOTS Admin Console, select the **Config** tab, enable the Tenant Manager Settings that were previously disabled, then click **Save config**.
k. After the preflight checks run successfully, deploy the new version of Jama Connect:
   i. When the deployment is complete and all pods are running, log in to Jama Connect as root using the hostname configured for Jama Connect.
   ii. If upgrading with new servers:
      • Configure SAML authentication in the new Jama Connect KOTS environment.
      • Update the base URL before performing a full reindex.
l. Perform a full reindex in Jama Connect to complete the deployment.

Perform an in-place upgrade of Jama Connect

Upgrading Jama Connect to 8.79.6, 9.0.4, or 9.6.x requires that you first update the Jama Connect KOTS platform. The updated KOTS platform optimizes how data is stored in Jama Connect and how KOTS resources communicate with one another.

IMPORTANT
Upgrading your current environment involves significant maintenance downtime and requires that you have a recovery plan in case you need to revert to the original environment. Instead, we recommend that you install a new Jama Connect environment (referred to as a *clean installation*), then copy elements of your current environment to the new environment. See Perform a clean installation [1].

To perform an in-place upgrade, see the instructions for your environment:

- Update the Jama Connect platform (internet) [4]
- Update the Jama Connect platform (airgap) [8]

Update the Jama Connect KOTS platform (internet)

Before you can deploy Jama Connect 8.79.6, 9.0.4, and 9.6.x, you must first update the Jama Connect KOTS platform (referred to as an *in-place upgrade*).

This method requires that you first run a pre-upgrade script, then run the Kubernetes (kURL) installer. After you run the script and installer, new versions of Jama Connect can be deployed from the KOTS Admin Console.

Upgrade from this Jama Connect version...	To one of these versions...
8.79.x	8.79.6 9.0.4 9.6.x
9.0.x	9.0.4 9.6.x

Important considerations

- Make sure that the file system on your primary node/application server has enough free space to support a data migration of the assets associated with your Jama Connect instance. Measure the disk space occupied by the var/lib/longhorn directory (du /var/lib/longhorn) and confirm that the disk has twice that amount available.
- Expect downtime. Upgrade during off hours to minimize the impact.

- The Kubernetes (kURL) installer is interactive and prompts you to confirm some of the steps. Depending on the number of remote nodes in your environment, expect this part of the upgrade process to take at least two hours.
- If you have dedicated Elasticsearch nodes, you are prompted to run separate installer commands on the secondary nodes during the installation.
- Recommended — Run the install commands inside a terminal multiplexer session to keep the session active in the event that the connection is dropped or the terminal is closed.
- If your Jama Connect deployment fails with a HorizontalPodAutoscaler error, you must manually update the Kubernetes secret associated with the deployed Helm release and redeploy Jama Connect. See Troubleshoot PersistentVolumeAccessMode errors if Jama Connect deployment fails [13].

To update the Jama Connect KOTS platform:

1. Back up your Jama Connect instance.
2. Create a shell script using the following contents:

```
#!/bin/bash

# Function to log messages
log() {
    echo "[INFO] $1"
}

# Function to log error messages
logError() {
    echo "[ERROR] $1"
}

# Function to annotate the TenantFS PVC
annotate_pvc() {
    local tenantfs_sc=$(kubectl get pvc/tenantfs
-o=jsonpath='{.spec.storageClassName}')
    if [ "$tenantfs_sc" != "longhorn" ]; then
            log "The tenantfs pvc will not be annotated since its
storage class is $tenantfs_sc"
            return
    fi

    log "Annotating the TenantFS PVC to allow an access mode change
during migration..."
    kubectl annotate pvc tenantfs kurl.sh/pvcmigrate-
destinationaccessmode='ReadWriteOnce' --overwrite=true
    if [ $? -eq 0 ]; then
        log "Successfully annotated the TenantFS PVC."
    else
        logError "Failed to annotate the TenantFS PVC."
        exit 1
    fi
}

# Function to delete Oauth and Saml volumes to avoid issues during the
process
delete_unused_pvc() {
    kubectl delete sts/saml sts/oauth
    kubectl delete pvc/volume-saml-0 pvc/volume-oauth-0
}
```

```
# Function to update Longhorn volume replicas
update_replicas() {
    local namespace="longhorn-system"
    local default_replicas=3
    # Check the number of nodes in the cluster
    local node_count=$(kubectl get nodes --no-headers | wc -l)

    if [ "$node_count" -ge "$default_replicas" ]; then
        log "There are $node_count nodes in the cluster. Will not scale
down Longhorn volume replicas"
        return
    fi

    log "Fetching Longhorn volumes in the $namespace namespace..."
    local volumes=$(kubectl get volumes -n $namespace
-o=jsonpath='{range .items[*]}{.metadata.name}{" "}{end}')

    local replicas=$node_count
    log "Updating spec.numberOfReplicas to $replicas for each volume..."
    for volume in $volumes; do
        kubectl patch volume $volume -n $namespace --type='json'
-p="[{\"op\": \"replace\", \"path\": \"/spec/numberOfReplicas\",
\"value\": $replicas}]"
        if [ $? -eq 0 ]; then
            log "Successfully updated volume $volume."
        else
            logError "Failed to update volume $volume."
        fi
    done
}

# Function to remove stopped Longhorn replicas
remove_unscheduled_replicas() {
    log "Removing unscheduled Longhorn replicas..."
    kubectl get replicas -n longhorn-system -o=jsonpath='{range .items[?
(@.spec.nodeID=="")]}{.metadata.name}{"\n"}' | xargs kubectl delete
replicas -n longhorn-system || true
    log "All unscheduled Longhorn replicas have been removed."
}

# Function to remove pods in shutdown status to avoid upgrade issues
# if the cluster has been restarted and there are shutdown Longhorn pods
remove_shutdown_pods() {
    local namespace="longhorn-system"
    log "Removing Longhorn pods in shutdown status."
    kubectl get pods -n $namespace | grep Shutdown | awk '{print $1}' |
xargs kubectl delete pod -n $namespace || true
    log "All Longhorn pods in shutdown status have been removed."
}

delete_unused_pvc
annotate_pvc
update_replicas
```

```
remove_unscheduled_replicas
remove_shutdown_pods
```

3. Run the shell script created in step 2 as a user with adequate privileges:

```
bash preupgrade.sh
```

4. Run the kURL installer:

> **IMPORTANT**
> The kURL installer is interactive and prompts you to continue several times throughout the upgrade process. Kubernetes is upgraded incrementally in steps from version 1.23.17 to 1.27.6 and requires you to confirm several of the steps before proceeding to the next version.

   a. From the command line on the primary node/application server, enter the following command to initiate the installation:

```
curl -sSL https://kurl.sh/jama-k8s-standardkots | sudo bash -s
```

   b. Once the Jama Connect KOTS platform upgrade is complete, run the following command to manually delete the **projectcontour** namespace on all nodes before proceeding with the Jama Connect upgrade.

```
kubectl delete namespace projectcontour
```

   c. Prepare your instance for the new Jama Connect release. This command deletes targeted KOTS resources, which is required before deploying the new version of Jama Connect.

```
kubectl delete sts/activemq sts/core sts/diff sts/elasticsearch
sts/hazelcast sts/oauth sts/saml sts/search sts/nginx sts/core-
ingress sts/core-reports sts/core-jobs jobs/tenant-manager pvc/
volume-oauth-0 pvc/volume-saml-0
```

5. Upgrade Jama Connect [7].

Upgrade Jama Connect with KOTS (internet)

When a new version of KOTS is available, you can apply and deploy it from the KOTS Admin Console.

> **IMPORTANT**
> If you are upgrading Jama Connect 8.79.6 or 9.0.4 to 9.6.x, you must run this command on the application server CLI before deploying Jama Connect:
>
> ```
> kubectl delete sts/saml sts/oauth pvc/volume-oauth-0 pvc/volume-
> saml-0
> ```

To upgrade Jama Connect with KOTS:

1. From the KOTS Admin Console, select the **Version history** tab, then click **Check for update**.
2. When the preflight checks are complete, find your Jama Connect upgrade version, then click **Deploy**.

The new version is tagged as **Currently deployed version**.

Update the Jama Connect KOTS platform (airgap)

Before you can deploy Jama Connect 8.79.6, 9.0.4, and 9.6.x, you must first update the Jama Connect KOTS platform (referred to as an *in-place upgrade).*

This method requires that you first run a pre-upgrade script, then run the kURL installer. After you run the script and installer, new versions of Jama Connect can be deployed from the KOTS Admin Console.

See also: Updating Embedded Clusters.

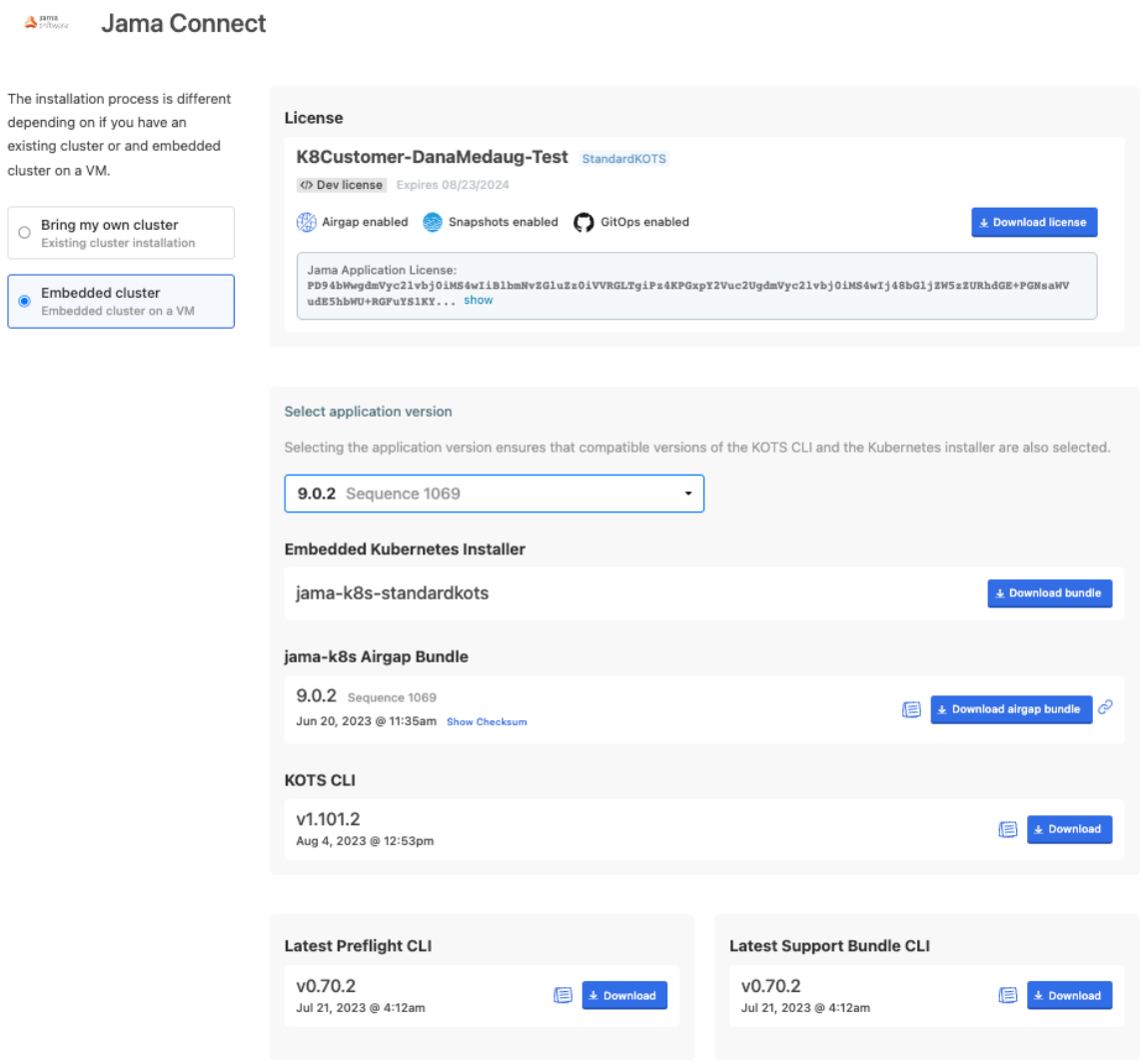
Upgrade from this Jama Connect version...	To one of these versions...
8.79.x	8.79.6 9.0.4 9.6.x
9.0.x	9.0.4 9.6.x

Important considerations

- Make sure that the file system on your primary node/application server has enough free space to support a data migration of the assets associated with your Jama Connect instance. Measure the disk space occupied by the var/lib/longhorn directory (du /var/lib/longhorn) and confirm that the disk has twice that amount available.
- Expect downtime. Upgrade during off hours to minimize the impact.
- The Kubernetes (kURL) installer is interactive and prompts you to confirm some of the steps. Depending on the number of remote nodes in your environment, expect this part of the upgrade process to take at least two hours.
- If you have dedicated Elasticsearch nodes, you are prompted to run separate installer commands on the secondary nodes during the installation.
- Recommended — Run the install commands inside a terminal multiplexer session to keep the session active in the event that the connection is dropped or the terminal is closed.
- If your Jama Connect deployment fails with a HorizontalPodAutoscaler error, you must manually update the Kubernetes secret associated with the deployed Helm release and redeploy Jama Connect. See Troubleshoot PersistentVolumeAccessMode errors if Jama Connect deployment fails [13].

To update the Jama Connect KOTS platform:

1. Log in to the airgap portal, select **Embedded Cluster**, then download the **Embedded Kubernetes Installer** files to your local system.



2. Create a shell script using the following contents:

```
#!/bin/bash

# Function to log messages
log() {
    echo "[INFO] $1"
}

# Function to log error messages
logError() {
    echo "[ERROR] $1"
}

# Function to annotate the TenantFS PVC
annotate_pvc() {
    local tenantfs_sc=$(kubectl get pvc/tenantfs
-o=jsonpath='{.spec.storageClassName}')
    if [ "$tenantfs_sc" != "longhorn" ]; then
            log "The tenantfs pvc will not be annotated since its
storage class is $tenantfs_sc"
            return
    fi
```

```
    log "Annotating the TenantFS PVC to allow an access mode change
during migration..."
    kubectl annotate pvc tenantfs kurl.sh/pvcmigrate-
destinationaccessmode='ReadWriteOnce' --overwrite=true
    if [ $? -eq 0 ]; then
        log "Successfully annotated the TenantFS PVC."
    else
        logError "Failed to annotate the TenantFS PVC."
        exit 1
    fi
}

# Function to delete Oauth and Saml volumes to avoid issues during the
process
delete_unused_pvc() {
    kubectl delete sts/saml sts/oauth
    kubectl delete pvc/volume-saml-0 pvc/volume-oauth-0
}

# Function to update Longhorn volume replicas
update_replicas() {
    local namespace="longhorn-system"
    local default_replicas=3
    # Check the number of nodes in the cluster
    local node_count=$(kubectl get nodes --no-headers | wc -l)

    if [ "$node_count" -ge "$default_replicas" ]; then
        log "There are $node_count nodes in the cluster. Will not scale
down Longhorn volume replicas"
        return
    fi

    log "Fetching Longhorn volumes in the $namespace namespace..."
    local volumes=$(kubectl get volumes -n $namespace
-o=jsonpath='{range .items[*]}{.metadata.name}{" "}{end}')

    local replicas=$node_count
    log "Updating spec.numberOfReplicas to $replicas for each volume..."
    for volume in $volumes; do
        kubectl patch volume $volume -n $namespace --type='json'
-p="[{\"op\": \"replace\", \"path\": \"/spec/numberOfReplicas\",
\"value\": $replicas}]"
        if [ $? -eq 0 ]; then
            log "Successfully updated volume $volume."
        else
            logError "Failed to update volume $volume."
        fi
    done
}

# Function to remove stopped Longhorn replicas
remove_unscheduled_replicas() {
    log "Removing unscheduled Longhorn replicas..."
    kubectl get replicas -n longhorn-system -o=jsonpath='{range .items[?
```

```
(@.spec.nodeID=="")]}{.metadata.name}{"\n"}' | xargs kubectl delete
replicas -n longhorn-system || true
    log "All unscheduled Longhorn replicas have been removed."
}

# Function to remove pods in shutdown status to avoid upgrade issues
# if the cluster has been restarted and there are shutdown Longhorn pods
remove_shutdown_pods() {
    local namespace="longhorn-system"
    log "Removing Longhorn pods in shutdown status."
    kubectl get pods -n $namespace | grep Shutdown | awk '{print $1}' |
xargs kubectl delete pod -n $namespace || true
    log "All Longhorn pods in shutdown status have been removed."
}

delete_unused_pvc
annotate_pvc
update_replicas
remove_unscheduled_replicas
remove_shutdown_pods
```

3. Run the shell script created in step 2 as a user with adequate privileges:

   ```
   bash preupgrade.sh
   ```

4. Extract (untar) the kURL installer:

   ```
   tar -xzvf jama-k8-standardkots.tar.gz
   ```

   The following contents are extracted: kurl directory, install.sh, join.sh, tasks.sh, and upgrade.sh scripts.

5. Run the kURL script to ensure all required images are available:

   ```
   cat tasks.sh | sudo bash -s load-images
   ```

6. Run the kURL installer:

   IMPORTANT
   The kURL installer is interactive and prompts you to continue several times throughout the upgrade process. Kubernetes is upgraded incrementally in steps from version 1.23.17 to 1.27.6 and requires you to confirm several of the steps before proceeding to the next version.

   a. From the command line on the primary node/application server, enter the following command to initiate the installation:

      ```
      cat install.sh | sudo bash -s airgap
      ```

   b. Once the Jama Connect KOTS platform upgrade is complete, run the following command to manually delete the **projectcontour** namespace on all nodes before proceeding with the Jama Connect upgrade.

      ```
      kubectl delete namespace projectcontour
      ```

   c. When the installation is complete, prepare your instance for the new Jama Connect release. This command deletes targeted KOTS resources, which is required before deploying the new version of Jama Connect.

      ```
      kubectl delete sts/activemq sts/core sts/diff sts/elasticsearch
      sts/hazelcast sts/oauth sts/saml sts/search sts/nginx sts/core-
      ```

```
ingress sts/core-reports sts/core-jobs jobs/tenant-manager pvc/
volume-oauth-0 pvc/volume-saml-0
```

7. Upgrade Jama Connect [12].

Upgrade Jama Connect with KOTS (airgap)

When a new version of KOTS is available, you can apply and deploy it from the KOTS Admin Console.

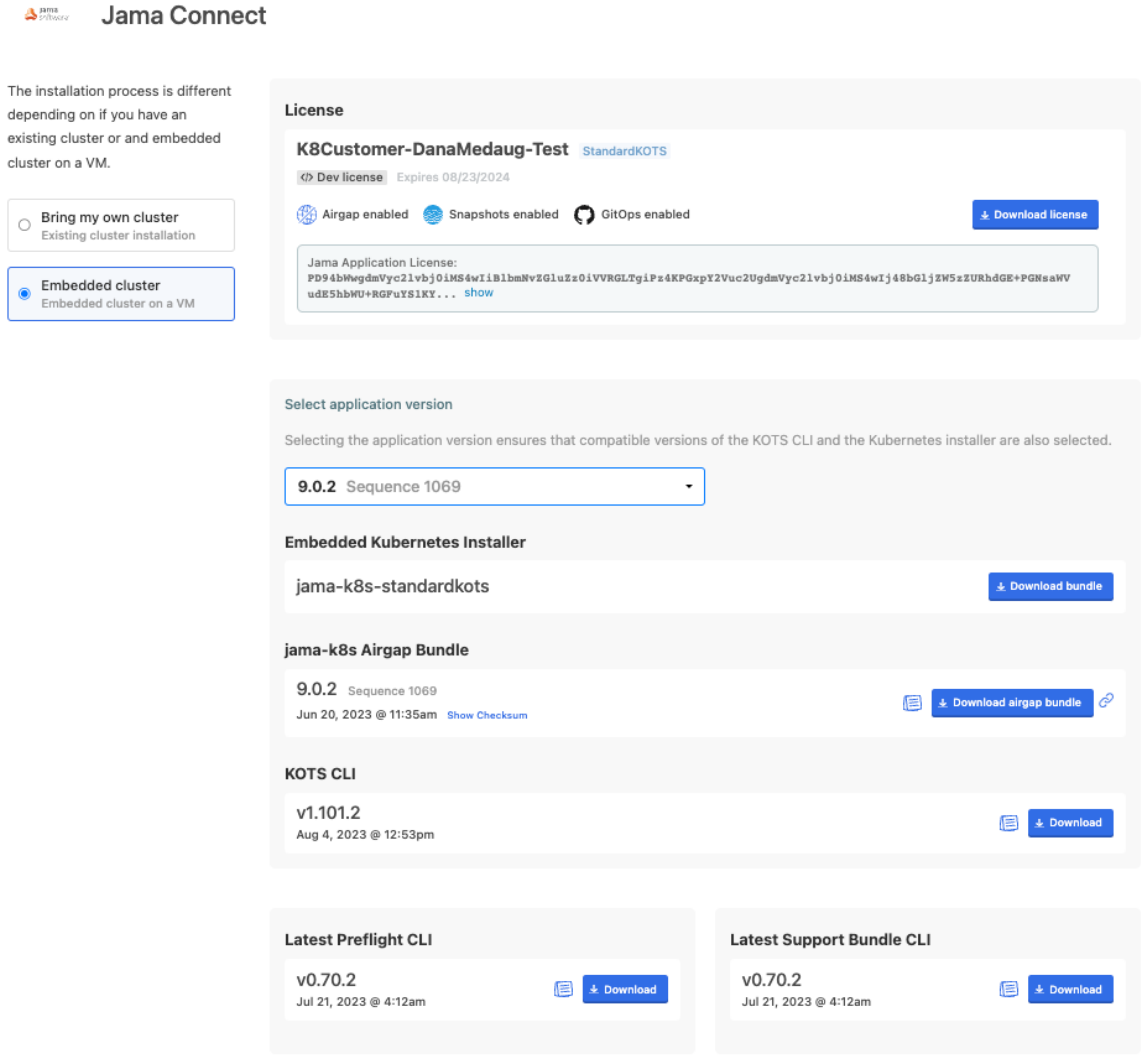
IMPORTANT

If you are upgrading Jama Connect 8.79.6 or 9.0.4 to 9.6.x, you must run this command on the application server CLI before deploying Jama Connect:

```
kubectl delete sts/saml sts/oauth pvc/volume-oauth-0 pvc/volume-
saml-0
```

To upgrade Jama Connect with KOTS:

1. From the air-gap safe portal, download the new **jama-k8s airgap** bundle for embedded clusters.



2. From the KOTS Admin Console, select the **Version history** tab:
   You must complete this step if the new airgap bundle hasn't been uploaded yet.

a. Click **Upload new version**.
b. Select the new airgap bundle.

A new version is created, and the system performs the preflight checks.

3. When the preflight checks are complete, click **Deploy**.

The new version is tagged as **Currently deployed version**.

Troubleshooting your upgrade (KOTS)

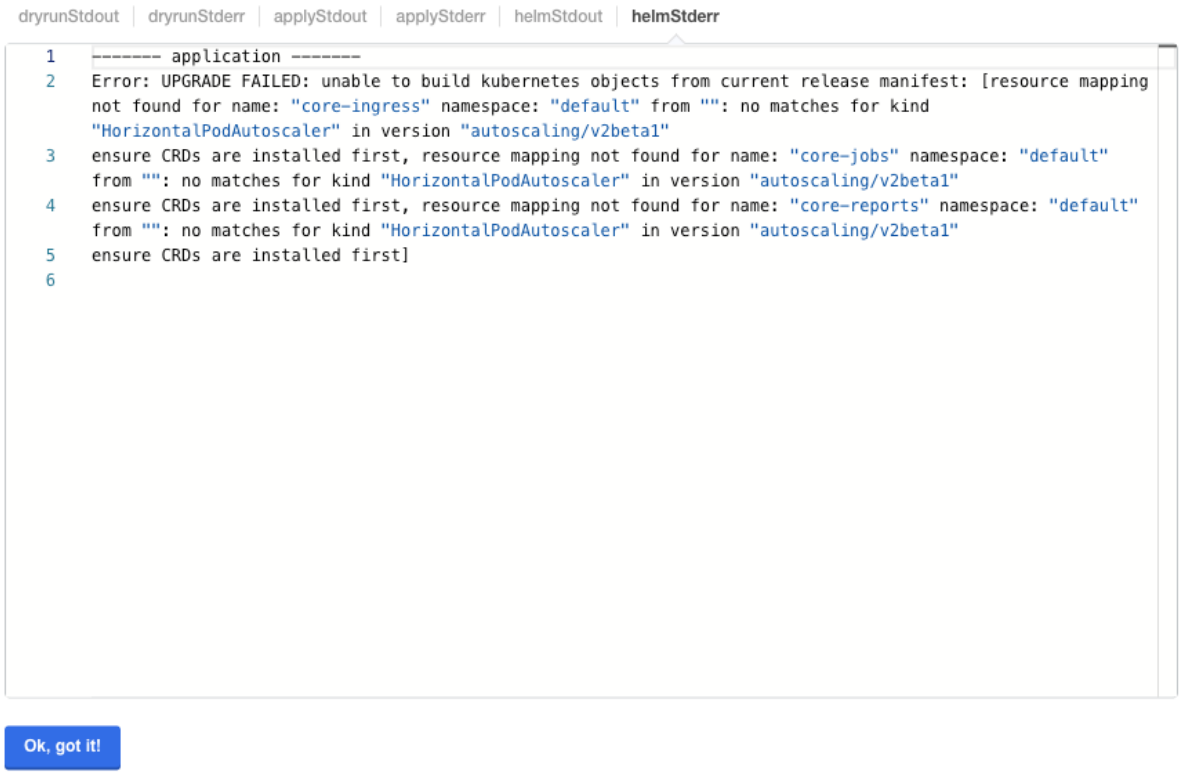
If you run into problems with your KOTS upgrade, here are some resources that might help.

- Troubleshoot HorizontalPodAutoscaler errors if Jama Connect deployment fails [13]
- Troubleshoot PersistentVolumeAccessMode errors if Jama Connect deployment fails [14]
- Troubleshoot kURL installer errors if node connectivity tests fail [15]

Troubleshoot HorizontalPodAutoscaler errors if Jama Connect deployment fails

In previous versions of Jama Connect with Kubernetes 1.27.6, the HorizontalPodAutoscaler resources for horizontal scaling were deprecated. If your Jama Connect deployment fails with the following error, you must manually update the Kubernetes secret associated with the deployed Helm release and redeploy Jama Connect.

IMPORTANT
This process applies only to environments with horizontal scaling enabled.



To modify the Kubernetes Helm release secret:

1. Retrieve the name of the secret associated with the latest deployed Helm release:

   ```
   kubectl get secret -l owner=helm,status=deployed,name=application | awk
   '{print $1}' | grep -v NAME
   ```
2. Use the secret to save the latest deployed release details to a file:

   ```
   kubectl get secret <secret-name> -o yaml > release.yaml
   ```
3. Create a backup of the file you created:

   ```
   cp release.yaml release.bak
   ```
4. Decode and generate output of the release object (JSON) found in the file you created:

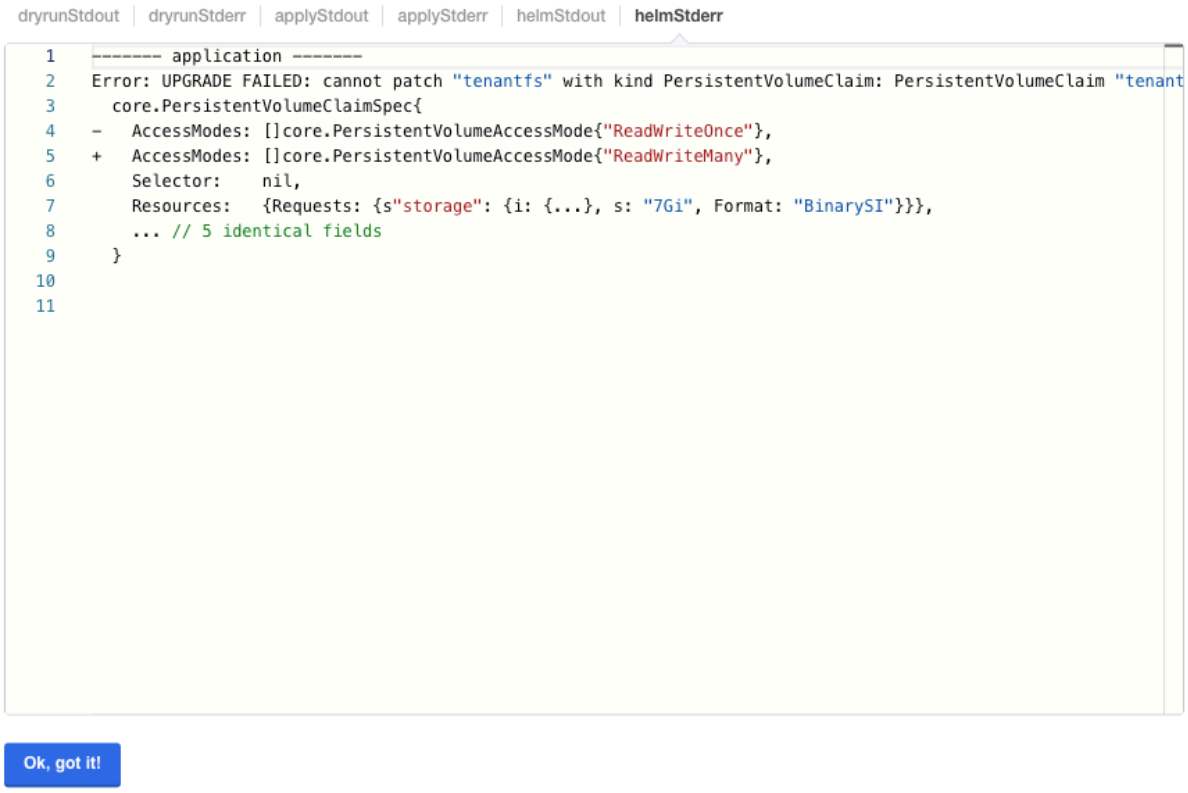
   ```
   cat release.yaml | grep -oP '(?<=release: ).*' | base64 -d | base64 -d |
   gzip -d > release.data.decoded
   ```
5. Using an editor tool, edit the release object data by changing all occurrences that reference the deprecated API version (autoscaling/v2beta1) with the new value (autoscaling/v2) found in the manifest field.
6. Encode the modified release object:

   ```
   cat release.data.decoded | gzip | base64 | base64
   ```
7. If the output contains line breaks, you must remove them before you can continue.
8. Using an editor tool, replace the JSON property value “data.release” in release.yaml with the newly encoded release object value you just created.
9. Apply the release file:

   ```
   kubectl apply -f release.yaml
   ```
10. Deploy Jama Connect.

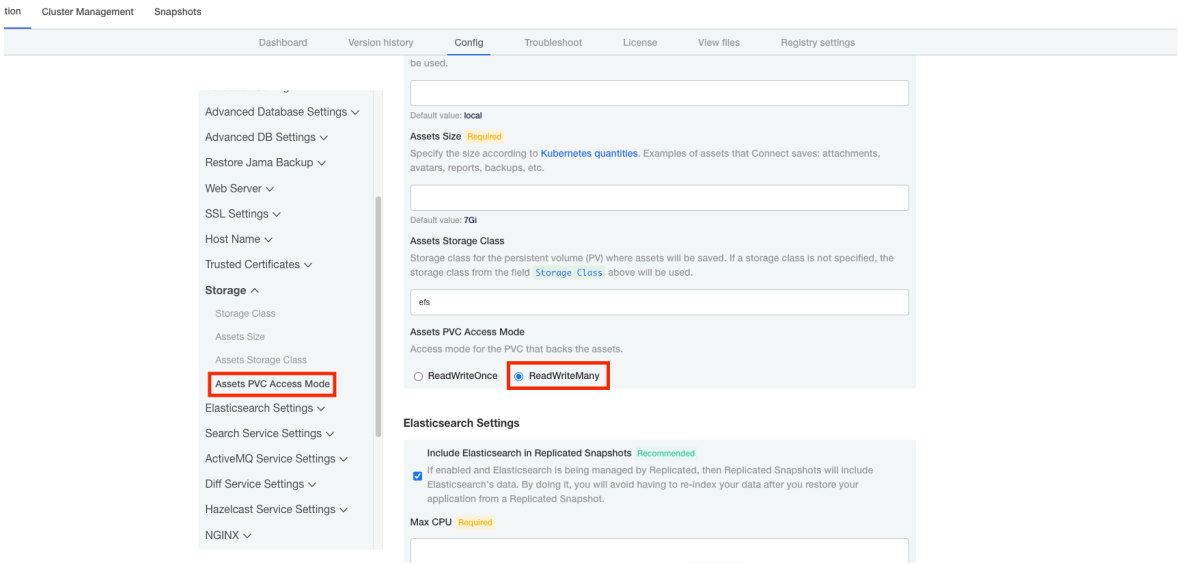
Troubleshoot PersistentVolumeAccessMode errors if Jama Connect deployment fails

If you configured Jama Connect to use a third-party storage class to save assets, you might get the following deployment error.



To modify Assets PVC Access Mode:

1. From the Config tab in the KOTS Admin Console, set the **Assets PVC Access Mode** to **ReadWriteMany**.



2. Deploy Jama Connect.

Troubleshoot kURL installer errors if node connectivity tests fail

When the kURL installer runs, internal tests confirm that all nodes can communicate with each other. If these tests fail, the Linux "fs.inotify.max_user_instances" host setting must be updated.

The Linux "fs.inotify.max_user_instances" is a host setting that defines user limits on the number of available inotify resources on the application server.

If the connectivity tests fail, these error messages are displayed:

```
✔ In cluster Preflights success
The migration from Weave to Flannel will require whole-cluster downtime.
Would you like to continue?
(Y/n) Y
Verifying if all nodes can communicate with each other through port 8472/UDP.
Testing intra nodes connectivity using port 8472/UDP.
Connection between all nodes will be attempted, this can take a while.
Deploying node connectivity listeners DaemonSet.
Listeners DaemonSet deployed successfully.
Testing connection from                                 (1/5)
Failed to connect from                                  %!s(MISSING)
Testing connection from                                 (2/5)
Failed to connect from                                  %!s(MISSING)
Testing connection from                                 (3/5)
Failed to connect from                                  %!s(MISSING)
Testing connection from                                 (4/5)
Failed to connect from                                  %!s(MISSING)
Testing connection from                                 (5/5)
Failed to connect from                                  %!s(MISSING)

Attempt to connect from                                          (UDP) failed.
Please verify if the active network policies are not blocking the connection.
Error: Failed to test nodes connectivity: node                 failed to connect from
Flannel requires UDP port 8472 for communication between nodes.
Please make sure this port is open prior to running this upgrade.
Not migrating from Weave to Flannel
```

For more information, see How to increase the inotify.max_user_watches and inotify.max_user instances sysctls on a Linux host.

To update the Linux host setting:

1. Check the current inotify user instance limit:

   ```
   cat /proc/sys/fs/inotify/max_user_instances
   ```
2. To update the limits temporarily (the value is set to 65536 in this example):

   ```
   sudo sysctl fs.inotify.max_user_instances=65536
   sudo sysctl -p
   ```
3. To apply the changes permanently, add **fs.inotify.max_user_instances=65536** to the file **/etc/sysctl.conf**.