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:

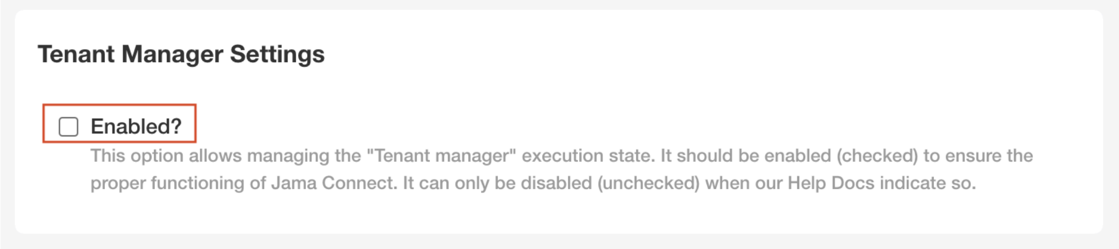
1. Install the KOTS software.

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.



Tenant Manager Settings

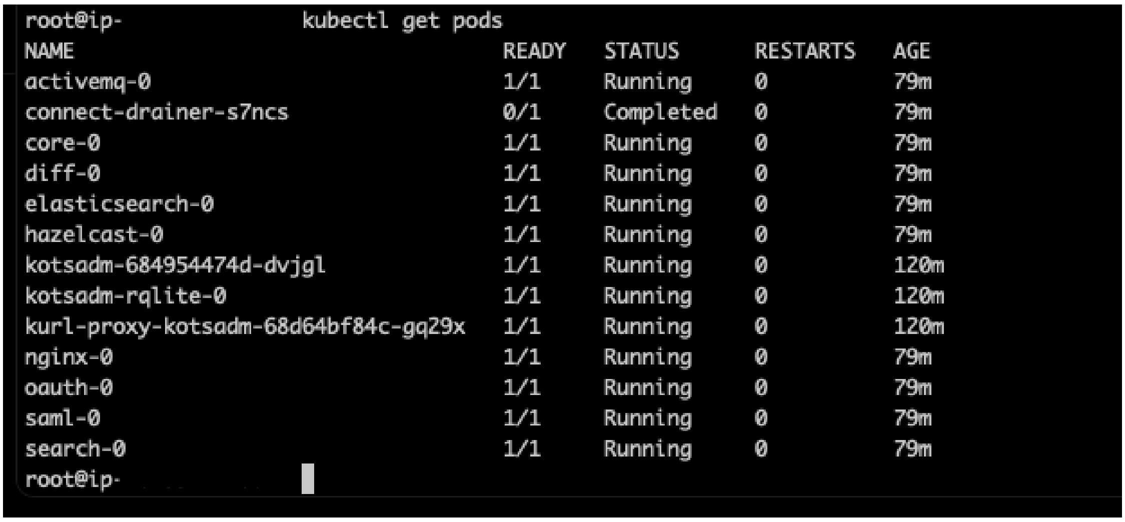
□ Enabled?

This option allows managing the "Tenant manager" execution state. It should be enabled (checked) to ensure the proper functioning of Jama Connect. It can only be disabled (unchecked) when our Help Docs indicate so.

   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) [7]

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. 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.

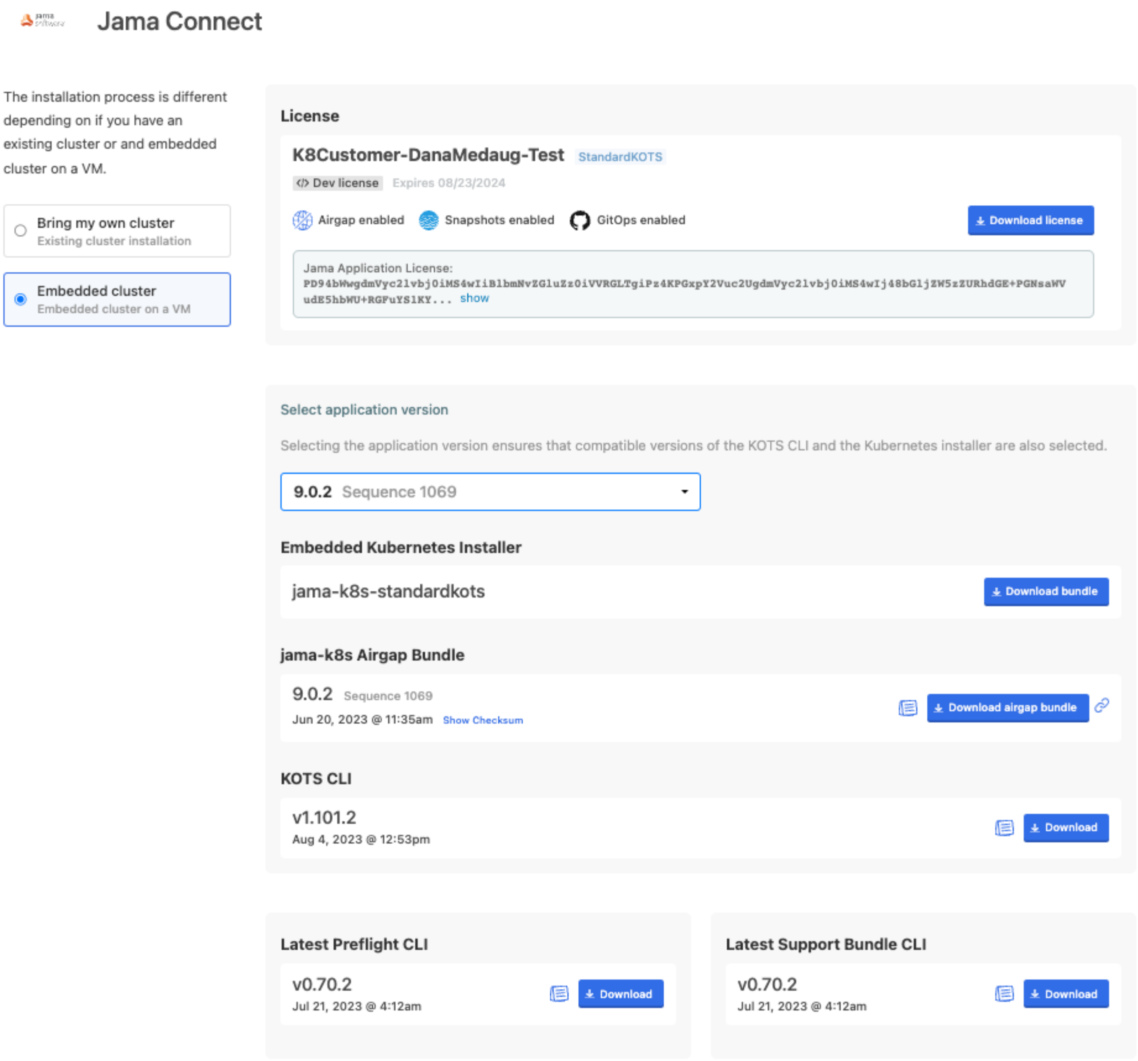
Upgrade from this Jama Connect version...	To one of these versions...
8.79.x	8.79.6 9.0.4 9.6.x
9.0.x	9.0.4 9.6.x

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.



Jama Connect

The installation process is different depending on if you have an existing cluster or and embedded cluster on a VM.

○ Bring my own cluster
Existing cluster installation

● Embedded cluster
Embedded cluster on a VM

License

K8Customer-DanaMedaug-Test StandardKOTS

Expires 08/23/2024

Airgap enabled · Snapshots enabled · GitOps enabled

Jama Application License:

PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZGluZz0iVVRGLTgiPz4KPGxpY2Vuc2UgdmVyc2lvbj0iMS4wIj48bGljZW5zZURhdGE+PGNsaWV
udE5hbWU+RGFuYS1KY... show

Select application version

Selecting the application version ensures that compatible versions of the KOTS CLI and the Kubernetes installer are also selected.

9.0.2 Sequence 1069

Embedded Kubernetes Installer

jama-k8s-standardkots

jama-k8s Airgap Bundle

9.0.2 Sequence 1069
Jun 20, 2023 @ 11:35am Show Checksum

KOTS CLI

v1.101.2
Aug 4, 2023 @ 12:53pm

Latest Preflight CLI

v0.70.2
Jul 21, 2023 @ 4:12am

Latest Support Bundle CLI

v0.70.2
Jul 21, 2023 @ 4:12am

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. 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].

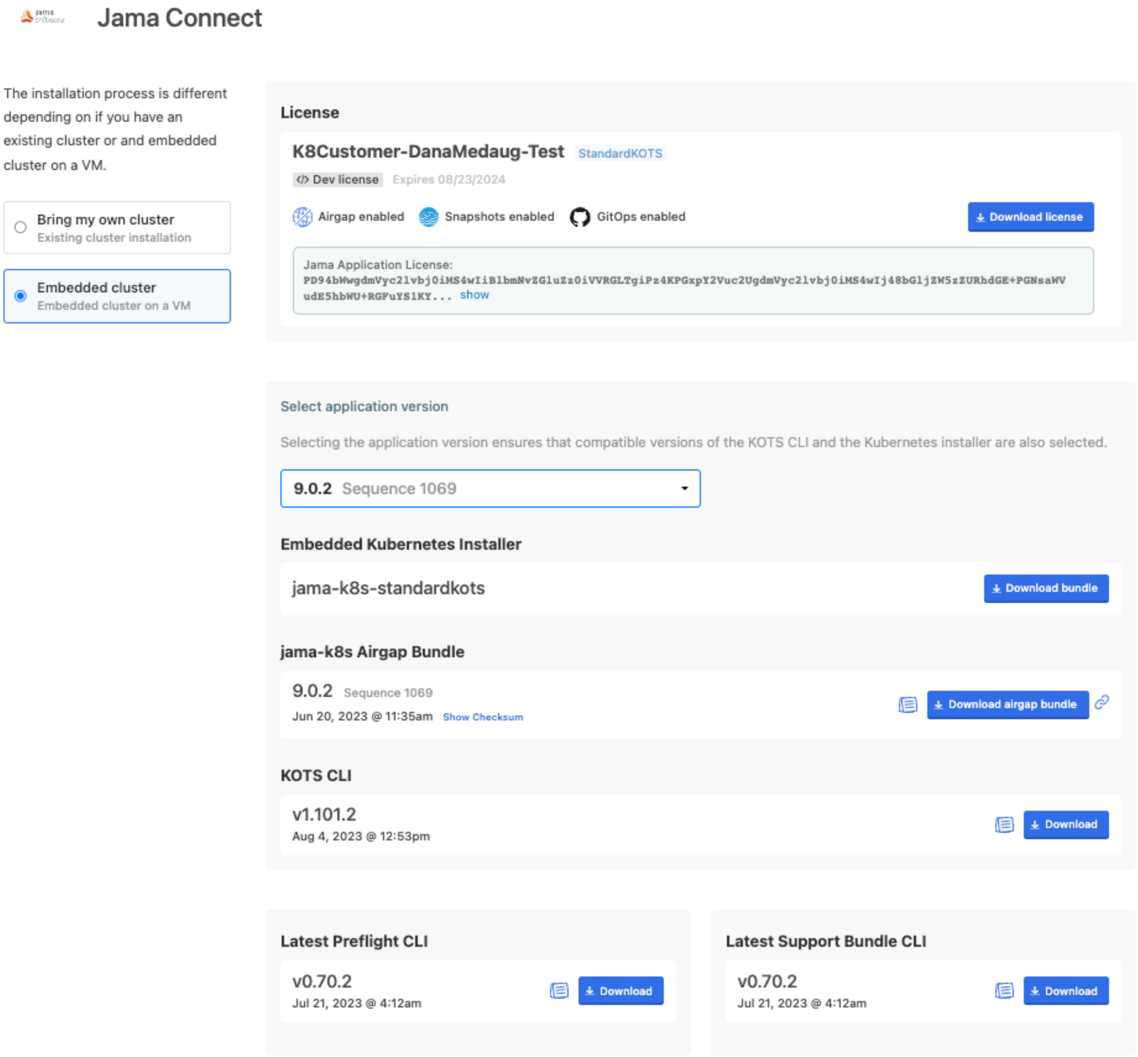
Upgrade Jama Connect with KOTS (airgap)

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.



Jama Connect

The installation process is different depending on if you have an existing cluster or and embedded cluster on a VM.

○ Bring my own cluster
Existing cluster installation

● Embedded cluster
Embedded cluster on a VM

License

K8Customer-DanaMedaug-Test StandardKOTS

</> Dev license Expires 08/23/2024

Airgap enabled Snapshots enabled GitOps enabled

Download license

Jama Application License:
PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZGluZz0iVVRGLTgiPz4KPGxpY2Vuc2UgdmVyc2lvbj0iMS4wIj48bGljZW5zZURhdGE+PGNsaWV udE5hbWU+RGFuYS1KY... show

Select application version

Selecting the application version ensures that compatible versions of the KOTS CLI and the Kubernetes installer are also selected.

9.0.2 Sequence 1069

Embedded Kubernetes Installer

jama-k8s-standardkots Download bundle

jama-k8s Airgap Bundle

9.0.2 Sequence 1069
Jun 20, 2023 @ 11:35am Show Checksum
Download airgap bundle

KOTS CLI

v1.101.2
Aug 4, 2023 @ 12:53pm
Download

Latest Preflight CLI

v0.70.2
Jul 21, 2023 @ 4:12am
Download

Latest Support Bundle CLI

v0.70.2
Jul 21, 2023 @ 4:12am
Download

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)

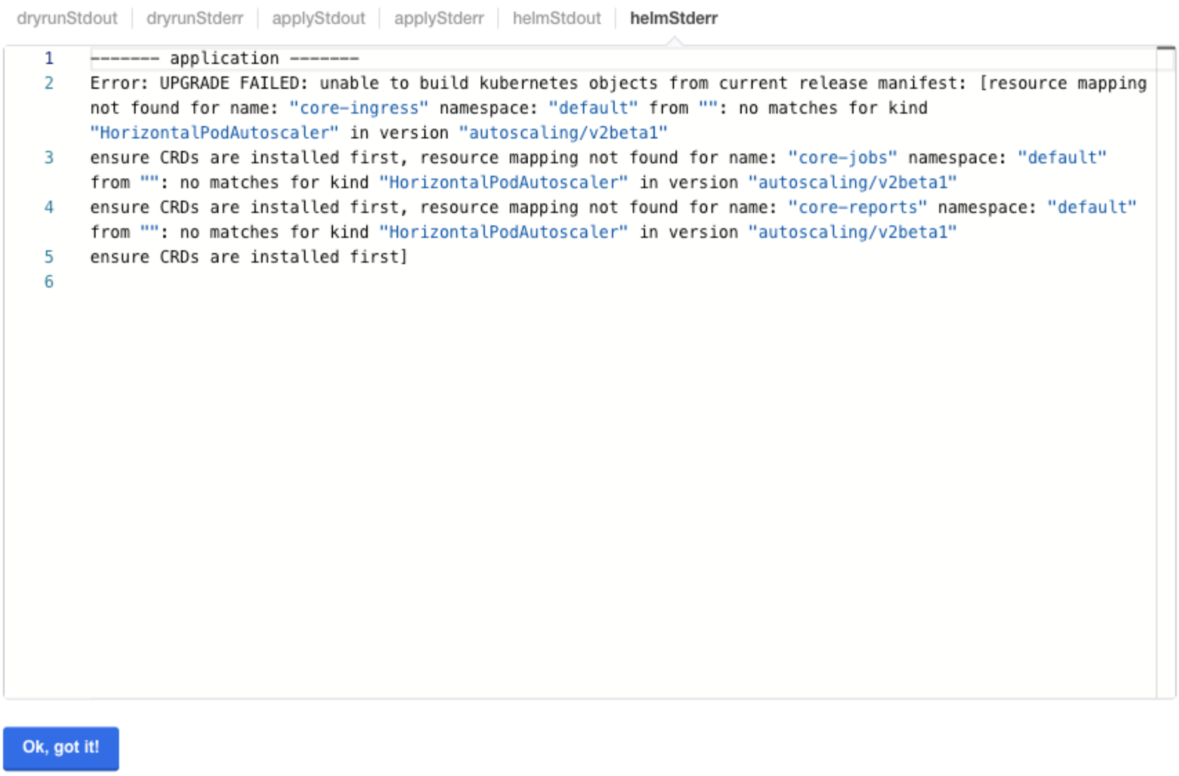
If you run into problems with your KOTS upgrade, here are some resources that might help.

- 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.



```
------- application -------
Error: UPGRADE FAILED: unable to build kubernetes objects from current release manifest: [resource mapping not found for name: "core-ingress" namespace: "default" from "": no matches for kind "HorizontalPodAutoscaler" in version "autoscaling/v2beta1"
ensure CRDs are installed first, resource mapping not found for name: "core-jobs" namespace: "default" from "": no matches for kind "HorizontalPodAutoscaler" in version "autoscaling/v2beta1"
ensure CRDs are installed first, resource mapping not found for name: "core-reports" namespace: "default" from "": no matches for kind "HorizontalPodAutoscaler" in version "autoscaling/v2beta1"
ensure CRDs are installed first]
```

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:

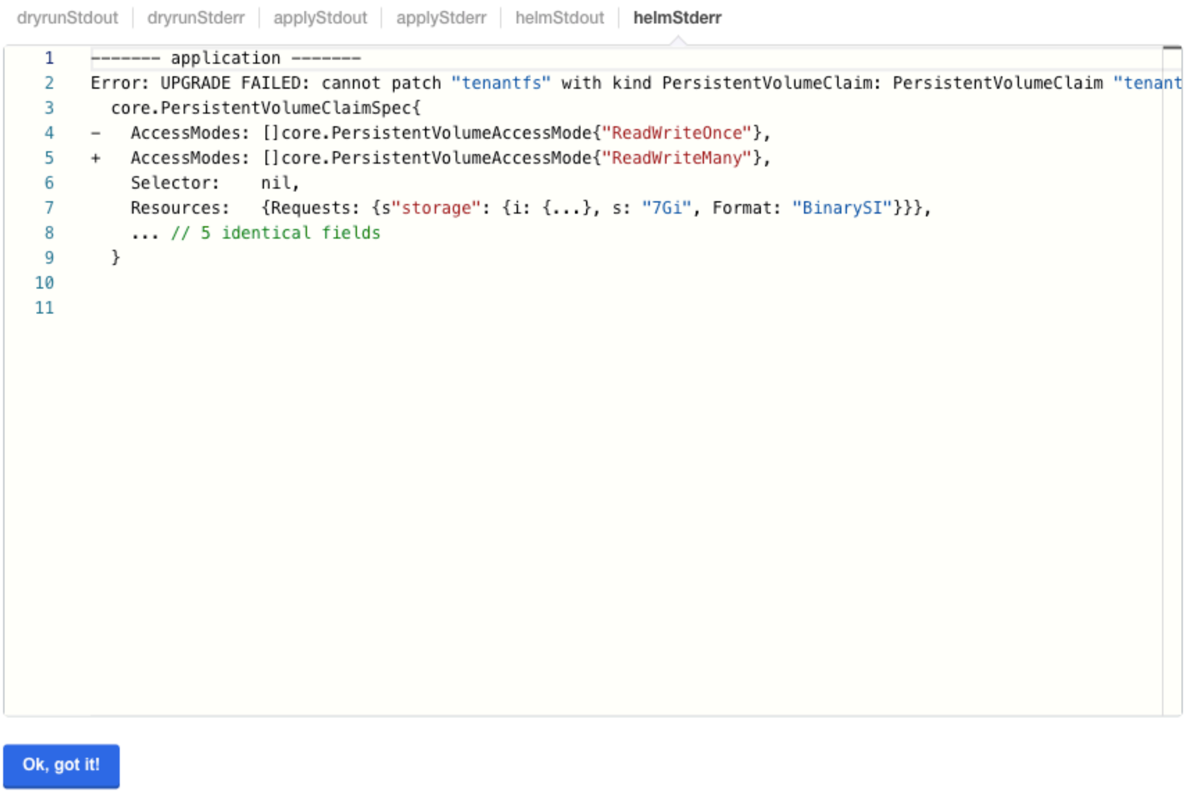
   ```
   cat release.yaml | grep -oP '(?<=release: ).*' | base64 -d | base64 -d |
   gzip -d > release.data.decoded
   ```
5. Using an editor tool, edit the release object data by changing all occurrences that reference the deprecated API version (autoscaling/v2beta1) with the new value (autoscaling/v2)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

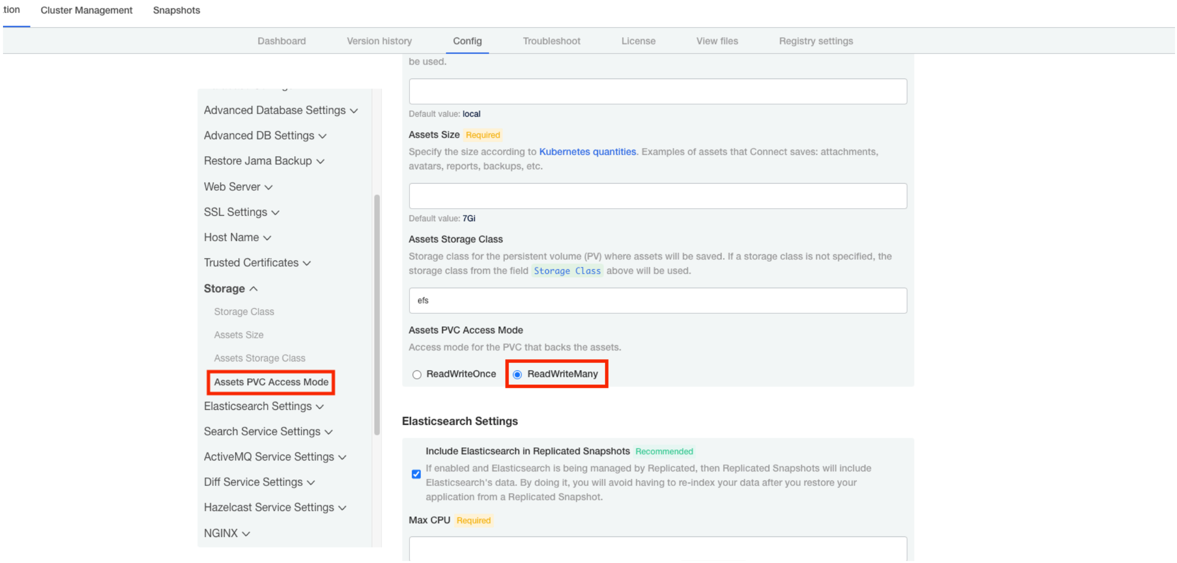
If you configured Jama Connect to use a third-party storage class to save assets, you might get the following deployment error.



```
------- application -------
Error: UPGRADE FAILED: cannot patch "tenantfs" with kind PersistentVolumeClaim: PersistentVolumeClaim "tenant
  core.PersistentVolumeClaimSpec{
-   AccessModes: []core.PersistentVolumeAccessMode{"ReadWriteOnce"},
+   AccessModes: []core.PersistentVolumeAccessMode{"ReadWriteMany"},
    Selector:    nil,
    Resources:   {Requests: {s"storage": {i: {...}, s: "7Gi", Format: "BinarySI"}}},
    ... // 5 identical fields
  }
```

To modify Assets PVC Access Mode:

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



be used.

Default value: local

Assets Size Required

Specify the size according to Kubernetes quantities. Examples of assets that Connect saves: attachments, avatars, reports, backups, etc.

Default value: 7Gi

Assets Storage Class

Storage class for the persistent volume (PV) where assets will be saved. If a storage class is not specified, the storage class from the field `Storage Class` above will be used.

efs

Assets PVC Access Mode

Access mode for the PVC that backs the assets.

○ ReadWriteOnce ● ReadWriteMany

Elasticsearch Settings

☑ Include Elasticsearch in Replicated Snapshots Recommended

If enabled and Elasticsearch is being managed by Replicated, then Replicated Snapshots will include Elasticsearch's data. By doing it, you will avoid having to re-index your data after you restore your application from a Replicated Snapshot.

Max CPU Required

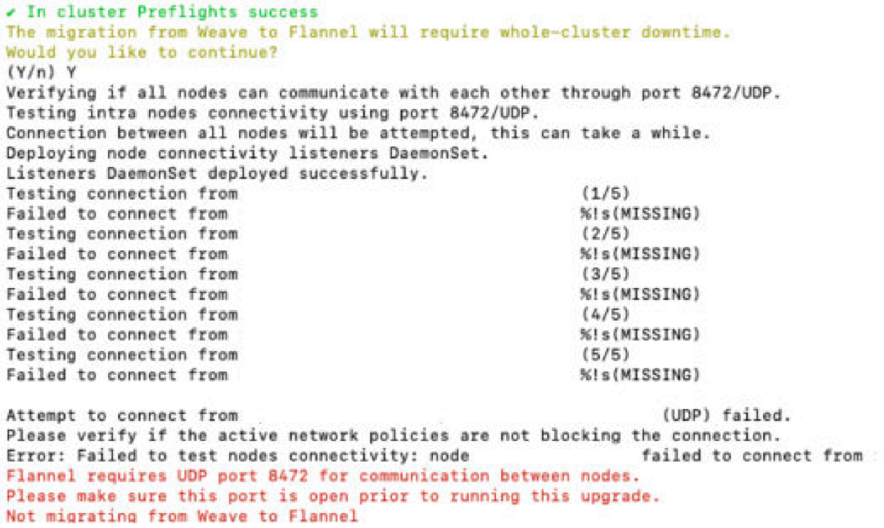
2. Deploy Jama Connect.

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**.