

# Upgrade

This document will provide all the information regarding the upgrading of ACP.

---

[Overview](#)

[Pre-Upgrade Preparation](#)

[Upgrade the global cluster](#)

[Upgrade Workload Clusters](#)

# Overview

ACP upgrades consist of three parts:

1. Upgrading the **global cluster**
2. Upgrading **workload clusters**
3. Upgrading **Operators** and **Cluster Plugins**

Users must upgrade the **global cluster** before upgrading any **workload clusters**.

Upgrading Operators and Cluster Plugins is optional. These can be upgraded during the cluster upgrade process, or separately after the cluster upgrade is complete.

---

## TOC

Cluster Upgrade Feature

Notes

---

## Cluster Upgrade Feature

In the cluster details page, under **Functional Components**, the platform lists all components provided by the system. When a new version is available, the **Upgrade** button becomes active, allowing the user to perform the upgrade.

## Notes

---

- **Kubernetes Version:** Kubernetes upgrades are supported only for **On-Premises Clusters**. For **Managed Clusters** (e.g., Amazon EKS, Azure AKS), Kubernetes upgrades must be performed through the respective cloud provider. See [Cluster Type](#) for more information on the differences between **On-Premises Clusters** and **Managed Clusters**.
- **Operator:** Only Operators with **Source = Alauda** are listed and can be upgraded via the cluster upgrade feature. All other Operators are managed through the **OLM** component in the Marketplace and are not included in this process.
- **Cluster Plugin:** Platform-provided plugins can be upgraded through the cluster upgrade feature on **both** On-Premises and Managed Clusters, provided they are already installed.
- **DR (Disaster Recovery Environment):** A DR environment contains both a **primary global cluster** and a **standby global cluster**, whereas a standard ACP environment includes only one global cluster.
- **Primary global cluster:** Defined as the global cluster that the ACP access domain name resolves to.
- **Standby global cluster:** Defined as the global cluster that the ACP access domain name **does not** resolve to.

# Pre-Upgrade Preparation

## Supported upgrade paths:

- From 4.0 → 4.2
- From 4.1 → 4.2

Before starting, ensure your current platform version is within the supported upgrade range.

## TOC

Important Notes

Kubernetes prerequisites

Run the Checklist

Download the Packages

## Important Notes

- Ensure the directory `/cpaas/minio` on the control plane nodes of global cluster has at least **120 GB** of free disk space.
- Make sure that *ALL* clusters are running Kubernetes versions within the supported range. See details below.

## Kubernetes prerequisites

### Upgrading from 4.1

- *ALL* clusters **MUST** be running a Kubernetes version newer than **1.32**.

### Upgrading from 4.0

- *ALL* clusters **MUST** be running a Kubernetes version newer than **1.31**.

## Run the Checklist

Contact technical support to obtain the **checklist script** and run it against the target platform to verify upgrade readiness.

## Download the Packages

From the **Alauda Customer Portal**, download the **ACP Core Package**.

If you want to upgrade cluster **Extensions** during the upgrade, follow these steps:

1. Navigate to the following path: [Marketplace - Batch Download - Upgrade - Post-ACP v4.0 Upgrades]
2. Download the `ac-get-app.sh` .
3. Upload the script to the control node of **Global** cluster in your environment.
4. Run the script with `bash ac-get-app.sh` .
5. After it finishes, import the generated `apps.yaml` back into the Alauda Customer Portal to align the extensions list.

In addition, navigate to the **CLI Tools** section in the **Alauda Customer Portal** and download the `violet` tool. This tool is required for uploading Extensions. For more information about `violet` , see [Upload Packages](#).

### WARNING

If any cluster installed the **Alauda Build of TopoLVM**, upload that plugin before you proceed with the upgrade. Run the following command:

```
violet push <path/to/directory/only_put_topolvm_plugin_here> \  
  --target-catalog-source "platform" \  
  --platform-address "https://example.com" \  
  --platform-username "<platform_user>" \  
  --platform-password "<platform_password>" \  
  --clusters "<cluster_name>"
```

### WARNING

Starting with v4.2, we introduced a new plugin named **Alauda Container Platform Log Essentials**. If you previously installed the log storage plugin, you also have to upload that plugin before starting the upgrade.

### WARNING

Before upgrading, please verify whether the nodes in the global cluster have **enabled [global Cluster Platform Node Isolation](#)** , You need to contact technical support to get the corresponding solution.

# Upgrade the global cluster

ACP consists of a **global cluster** and one or more **workload clusters**. The global cluster **must** be upgraded before any workload clusters.

This document walks you through the upgrade procedure for the global cluster.

If the global cluster is configured with the **global DR (Disaster Recovery)** solution, follow the [global DR procedure](#) strictly. Otherwise, follow the [Standard procedure](#) .

---

## TOC

### Standard procedure

- Upload images

- Trigger the upgrade

- Upgrade the global cluster

- Install Alauda Container Platform Cluster Enhancer Plugin

- (Conditional) Upgrade Alauda Service Mesh Essentials

### Post-upgrade

### global DR procedure

- Verify data consistency

- Uninstall the etcd sync plugin

- Upload images

- Upgrade the standby cluster

- Upgrade the primary cluster

- Reinstall the etcd sync plugin

- Check Synchronization Status

# Standard procedure

## 1 Upload images

Copy the core package to **any control plane node** of the global cluster. Extract the package and `cd` into the extracted directory.

- If the global cluster uses the **built-in registry**, run:

```
bash upgrade.sh --only-sync-image=true
```

- If the global cluster uses an **external registry**, you also need to provide the registry address:

```
bash upgrade.sh --only-sync-image=true --registry <registry-address> --username  
<username> --password <password>
```

If you plan to upgrade the **Operator** and **Cluster Plugin** while upgrading the global cluster, please push the corresponding packages to the corresponding cluster's registry in advance. For bulk upload instructions, see [Push all packages at once](#).

### INFO

Uploading images typically takes about 2 hours, depending on your network and disk performance.

If your platform is configured for global disaster recovery (DR), remember that the **standby global cluster also requires image upload**. Be sure to plan your maintenance window accordingly.

### WARNING

When using `violet` to upload packages to a standby cluster, the parameter `--dest-repo <VIP addr of standby cluster>` must be specified.



Otherwise, the packages will be uploaded to the image repository of the **primary cluster**, preventing the standby cluster from installing or upgrading extensions.

Also be aware that either authentication info of the standby cluster's image registry or `--no-auth` parameter MUST be provided.

For details of the `violet push` subcommand, please refer to [Upload Packages](#).

## 2 Trigger the upgrade

After the image upload is complete, run the following command to start the upgrade process:

```
bash upgrade.sh --skip-sync-image
```

Wait for the script to finish before proceeding. It will take about **10–15 minutes** for the upgrade button of the *Functional Components* tab to be available. You will then be able to upgrade the **Operator** and **Cluster Plugin** in the following upgrade instructions.

## 3 Upgrade the global cluster

1. Log in to the Web Console of the global cluster and switch to **Administrator** view.
2. Navigate to **Clusters > Clusters**.
3. Click on the `global` cluster to open its detail view.
4. Go to the **Functional Components** tab.
5. Click the **Upgrade** button.

Review the available component updates in the dialog, and confirm to proceed.

### INFO

- If the **Alauda Container Platform GitOps** plugin is installed in the global cluster and its pods are running abnormally after the upgrade, refer to [Upgrading Alauda Container Platform GitOps ↗](#).

4

## Install Alauda Container Platform Cluster Enhancer Plugin

### INFO

This step is only to ensure that the cluster enhancer plugin is installed. If you found this cluster plugin already installed, nothing needs to be done.

1. Navigate to **Administrator**.
2. In the left sidebar, click **Marketplace > Cluster Plugins** and select the `global` cluster.
3. Locate the **Alauda Container Platform Cluster Enhancer** plugin and click **Install**.

5

## (Conditional) Upgrade Alauda Service Mesh Essentials

If **Service Mesh v1** is installed, refer to the [Alauda Service Mesh Essentials Cluster Plugin](#) documentation before upgrading the workload clusters.

## Post-upgrade

- [Upgrade Alauda AI](#)
- [Upgrade Alauda DevOps](#)

## global DR procedure

1

### Verify data consistency

Follow your regular global DR inspection procedures to ensure that data in the **standby global cluster** is consistent with the **primary global cluster**.

If inconsistencies are detected, **contact technical support** before proceeding.

On **both** clusters, run the following command to ensure no **Machine** nodes are in a non-running state:

```
kubectl get machines.platform.tkestack.io
```

If any such nodes exist, contact technical support to resolve them before continuing.

## 2 Uninstall the etcd sync plugin

1. Access the Web Console of the **standby cluster** via its IP or VIP.
2. Switch to the **Administrator** view.
3. Navigate to **Marketplace > Cluster Plugins**.
4. MAKE SURE you have switched to the **global** cluster.
5. Find the **Alauda Container Platform etcd Synchronizer** plugin and **Uninstall** it.  
Wait for the uninstallation to complete.

## 3 Upload images

Perform the **Upload images** step on **both** the standby cluster and the primary cluster.

See [Upload images in Standard procedure](#) for details.

## 4 Upgrade the standby cluster

### INFO

Accessing the **standby cluster** Web Console is required to perform the upgrade.

Before proceeding, verify that the **ProductBase** resource of the standby cluster is correctly configured with the cluster VIP under `spec.alternativeURLs`.

If not, update the configuration as follows:

```
apiVersion: product.alauda.io/v1alpha2
kind: ProductBase
metadata:
  name: base
spec:
  alternativeURLs:
    - https://<standby-cluster-vip>
```

On the **standby cluster**, follow the steps in the [Standard procedure](#) to complete the upgrade.

5

## Upgrade the primary cluster

After the standby cluster has been upgraded, proceed with the [Standard procedure](#) on the **primary cluster**.

6

## Reinstall the etcd sync plugin

Before reinstalling, verify that port `2379` is properly forwarded from both global cluster VIPs to their control plane nodes.

To reinstall:

1. Access the Web Console of the **standby global cluster** via its IP or VIP.
2. Switch to **Administrator** view.
3. Go to **Marketplace > Cluster Plugins**.
4. Select the `global` cluster.
5. Locate **Alauda Container Platform etcd Synchronizer**, click **Install**, and provide the required parameters.

To verify installation:

```
kubectl get po -n cpaas-system -l app=etcd-sync # Ensure pod is 1/1 Running

kubectl logs -n cpaas-system $(kubectl get po -n cpaas-system -l app=etcd-sync --
no-headers | awk '{print $1}' | head -1) | grep -i "Start Sync update"
# Wait until the logs contain "Start Sync update"

# Recreate the pod to trigger synchronization of resources with ownerReferences
kubectl delete po -n cpaas-system $(kubectl get po -n cpaas-system -l app=etcd-sync
--no-headers | awk '{print $1}' | head -1)
```

7

## Check Synchronization Status

Run the following to verify the synchronization status:

```
curl "$(kubectl get svc -n cpaas-system etcd-sync-monitor -
ojsonpath='{.spec.clusterIP}')/check"
```

### Explanation of output:

- "LOCAL ETCD missed keys:" – Keys exist in the **primary cluster** but are missing in the standby. This often resolves after a pod restart.
- "LOCAL ETCD surplus keys:" – Keys exist in the **standby cluster** but not in the primary. Review these with your operations team before deletion.

# Upgrade Workload Clusters

After completing the upgrade of the global cluster, you can proceed to upgrade the workload clusters. The workload cluster upgrade process is similar to that of the global cluster but requires attention to the following considerations:

- If your platform uses the **global disaster recovery (DR)** solution, you **must complete the upgrade of both the primary and standby global clusters** before upgrading any workload clusters.
- All **PostgreSQL** instances will be **automatically restarted** during the upgrade.
- For **MySQL-PXC**, **MySQL-MGR**, **Redis**, **Kafka**, and **RabbitMQ** instances configured with an **automatic update strategy**, the upgrade process includes a restart, which may lead to **temporary service disruption**.
- A maximum of **20 workload clusters** can be upgraded concurrently.

## TOC

Upgrade a workload cluster

Upgrade the workload cluster

Post-upgrade

## Upgrade a workload cluster

**WARNING**

1. If you plan to upgrade the **Operator** and **Cluster Plugin** while upgrading the workload cluster, please push the corresponding packages to the corresponding cluster's registry in advance. For bulk upload instructions, see [Push all packages at once](#).

1

## Upgrade the workload cluster

1. Log into the Web Console and switch to the **Administrator** view.
2. Navigate to **Clusters > Clusters**.
3. Select the **workload cluster** you want to upgrade and open its detail page.
4. Go to the **Functional Components** tab.
5. Click the **Upgrade** button.

If the upgrade program detects any custom configuration overrides, you will be prompted to confirm these settings. If you are unsure whether these overridden configurations may impact the upgrade, please contact technical support for assistance.

Once confirmed, a component upgrade dialog will appear. Review the available updates and proceed with the upgrade.

## Post-upgrade

- [Upgrade Alauda AI](#) ↗
- [Upgrade Alauda DevOps](#) ↗
- [Upgrade Alauda Service Mesh](#) ↗