



Use Astra

Astra Control Center

NetApp
February 12, 2024

This PDF was generated from <https://docs.netapp.com/us-en/astra-control-center-2204/use/manage-apps.html> on February 12, 2024. Always check docs.netapp.com for the latest.

Table of Contents

- Use Astra 1
 - Manage apps 1
 - Protect apps 6
 - View app and cluster health 28
 - Manage your account. 30
 - Manage buckets. 41
 - Manage the storage backend. 43
 - Monitor and protect infrastructure 47
 - Unmanage apps and clusters 54
 - Upgrade Astra Control Center 55
 - Uninstall Astra Control Center 66

Use Astra

Manage apps

Start managing apps

After you [add a cluster to Astra Control management](#), you can install apps on the cluster (outside of Astra Control), and then go to the Apps page in Astra Control to start managing the apps and their resources.

For more information, see [App management requirements](#).

Supported app installation methods

Astra Control supports the following application installation methods:

- **Manifest file:** Astra Control supports apps installed from a manifest file using kubectl. For example:

```
kubectl apply -f myapp.yaml
```

- **Helm 3:** If you use Helm to install apps, Astra Control requires Helm version 3. Managing and cloning apps installed with Helm 3 (or upgraded from Helm 2 to Helm 3) are fully supported. Managing apps installed with Helm 2 is not supported.
- **Operator-deployed apps:** Astra Control supports apps installed with namespace-scoped operators. These operators are generally designed with a "pass-by-value" rather than "pass-by-reference" architecture. The following are some operator apps that follow these patterns:
 - [Apache K8ssandra](#)
 - [Jenkins CI](#)
 - [Percona XtraDB Cluster](#)

Note that Astra Control might not be able to clone an operator that is designed with a "pass-by-reference" architecture (for example, the CockroachDB operator). During these types of cloning operations, the cloned operator attempts to reference Kubernetes secrets from the source operator despite having its own new secret as part of the cloning process. The clone operation might fail because Astra Control is unaware of the Kubernetes secrets in the source operator.



An operator and the app it installs must use the same namespace; you might need to modify the deployment .yaml file for the operator to ensure this is the case.

Install apps on your cluster

Now that you've added your cluster to Astra Control, you can install apps or manage existing apps on the cluster. Any app that is scoped to a namespace can be managed. After the pods are online, you can manage the app with Astra Control.

For help with deploying validated apps from Helm charts, refer to the following:

- [Deploy MariaDB from a Helm chart](#)
- [Deploy MySQL from a Helm chart](#)

- [Deploy Postgres from a Helm chart](#)
- [Deploy Jenkins from a Helm chart](#)

Manage apps

Astra Control enables you to manage your apps at the namespace level or by Kubernetes label.



Apps installed with Helm 2 are not supported.

You can perform the following activities to manage apps:

- Manage apps
 - [Manage apps by namespace](#)
 - [Manage apps by Kubernetes label](#)
- [Ignore apps](#)
- [Unmanage apps](#)



Astra Control itself is not a standard app; it is a "system app." You should not try to manage Astra Control itself. Astra Control itself isn't shown by default for management. To see system apps, use the "Show system apps" filter.

For instructions on how to manage apps using the Astra Control API, see the [Astra Automation and API information](#).



After a data protection operation (clone, backup, restore) and subsequent persistent volume resize, there is up to a twenty-minute delay before the new volume size is shown in the UI. The data protection operation is successful within minutes, and you can use the management software for the storage backend to confirm the change in volume size.

Manage apps by namespace

The **Discovered** section of the Apps page shows namespaces and any Helm-installed apps or custom-labeled apps in those namespaces. You can choose to manage each app individually or at the namespace level. It all comes down to the level of granularity that you need for data protection operations.

For example, you might want to set a backup policy for "maria" that has a weekly cadence, but you might need to back up "mariadb" (which is in the same namespace) more frequently than that. Based on those needs, you would need to manage the apps separately and not under a single namespace.

While Astra Control enables you to separately manage both levels of the hierarchy (the namespace and the apps in that namespace), the best practice is to choose one or the other. Actions that you take in Astra Control can fail if the actions take place at the same time at both the namespace and app level.

Steps

1. From the left navigation bar, select **Applications**.
2. Select the **Discovered** filter.



3. View the list of discovered namespaces. Expand the namespace to view the apps and associated resources.

Astra Control shows you the Helm apps and custom-labeled apps in the namespace. If Helm labels are available, they're designated with a tag icon.

4. Look at the **Group** column to see which namespace the application is running in (it's designated with the folder icon).
5. Decide whether you want to manage each app individually or at the namespace level.
6. Find the app you want at the desired level in the hierarchy, and select **Manage** from the Options menu in the **Actions** column.
7. If you don't want to manage an app, select **Ignore** from the Options menu in the **Actions** column.

For example, if you want to manage all apps under the "maria" namespace together so that they have the same snapshot and backup policies, you would manage the namespace and ignore the apps in the namespace.

8. To see the list of managed apps, select **Managed** as the display filter.



The app you just added might have a warning icon under the Protected column, indicating that it is not backed up and not scheduled for backups yet.

9. To see details of a particular app, select the app name.

Result

Apps that you chose to manage are now available from the **Managed** tab. Any ignored apps will move to the **Ignored** tab. Ideally, the Discovered tab will show zero apps, so that as new apps are installed, they are easier to find and manage.

Manage apps by Kubernetes label

Astra Control includes an action at the top of the Apps page named **Define custom app**. You can use this action to manage apps that are identified with a Kubernetes label. [Learn more about defining custom apps by Kubernetes label.](#)

Steps

1. From the left navigation bar, select **Applications**.
2. Select **Define**.
3. In the **Define custom application** dialog box, provide the required information to manage the app:
 - a. **New App**: Enter the display name of the app.
 - b. **Cluster**: Select the cluster where the app resides.
 - c. **Namespace**: Select the namespace for the app.
 - d. **Label**: Enter a label or select a label from the resources below.
 - e. **Selected Resources**: View and manage the selected Kubernetes resources that you'd like to protect (pods, secrets, persistent volumes, and more).
 - View the available labels by expanding a resource and selecting the number of labels.

- Select one of the labels.

After you choose a label, it displays in the **Label** field. Astra Control also updates the **Unselected Resources** section to show the resources that don't match the selected label.

f. **Unselected Resources:** Verify the app resources that you don't want to protect.

4. Select **Define custom application**.

Result

Astra Control enables management of the app. You can now find it in the **Managed** tab.

Ignore apps

If an app has been discovered, it appears in the Discovered list. In this case, you can clean up the Discovered list so that new apps that are newly installed are easier to find. Or, you might have apps that you are managing and later decide you no longer want to manage them. If you don't want to manage these apps, you can indicate that they should be ignored.

Also, you might want to manage apps under one namespace together (Namespace-managed). You can ignore apps that you want to exclude from the namespace.

Steps

1. From the left navigation bar, select **Applications**.
2. Select **Discovered** as the filter.
3. Select the app.
4. From the Options menu in the **Actions** column, select **Ignore**.
5. To unignore, select **Unignore**.

Unmanage apps

When you no longer want to back up, snapshot, or clone an app, you can stop managing it.



If you unmanage an app, any backups or snapshots that were created earlier will be lost.

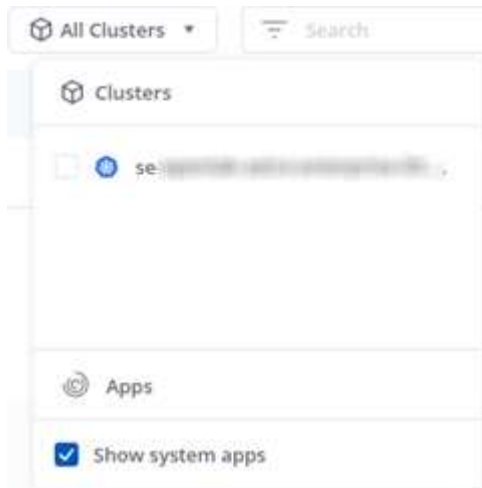
Steps

1. From the left navigation bar, select **Applications**.
2. Select **Managed** as the filter.
3. Select the app.
4. From the Options menu in the **Actions** column, select **Unmanage**.
5. Review the information.
6. Type "unmanage" to confirm.
7. Select **Yes, Unmanage Application**.

What about system apps?

Astra Control also discovers the system apps running on a Kubernetes cluster. We don't show you these system apps by default because it's rare that you'd need to back them up.

You can display system apps from the Applications page by selecting the **Show system apps** check box under the Clusters filter in the toolbar.



Astra Control itself is not a standard app; it is a "system app." You should not try to manage Astra Control itself. Astra Control itself isn't shown by default for management.

Find more information

- [Use the Astra Control API](#)

Define a custom app example

Creating a custom app enables you to group elements of your Kubernetes cluster into a single app. This collection of Kubernetes resources is based on a namespace and a label.

A custom app gives you more granular control over what to include in an Astra Control operation, including:

- Clone
- Snapshot
- Backup
- Protection Policy

In most cases you will want to use Astra Control's features on your entire app. However, you can also create a custom app to use these features by the labels you assign to Kubernetes objects in a namespace.



Custom apps can be created only within a specified namespace on a single cluster. Astra Control does not support the ability for a custom app to span multiple namespaces or clusters.

A label is a key/value pair you can assign to Kubernetes objects for identification. Labels make it easier to sort, organize, and find your Kubernetes objects. To learn more about Kubernetes labels, [see the official Kubernetes documentation](#).



Overlapping policies for the same resource under different names can cause data conflicts. If you create a custom app for a resource, be sure it's not being cloned or backed up under any other policies.

What you'll need

- A cluster added to Astra Control

Steps

1. From the Apps page, select **+ Define**.

The Custom App window shows which resources will be included or excluded from your custom app. This helps you make sure you are choosing the correct criteria for defining your custom app.

2. In the pop-up window, enter the app name, choose the cluster in the **Cluster** drop-down, and choose the app's namespace from the **Namespace** drop-down.
3. From the drop-down **Label** list, select a label for the apps and namespace.
4. After defining the custom app for the one deployment, repeat the process for other deployments, as needed.

When you finish creating the two custom apps, you can treat these resources as any other Astra Control application. They can clone them, create backups and snapshots, and create a custom Protection Policy for each group of resources based on the Kubernetes labels.

Example: Separate Protection Policy for different releases

In this example, the devops team is managing a canary release deployment. Their cluster has three pods running NginX. Two of the pods are dedicated to the stable release. The third pod is for the canary release.

The devops team's Kubernetes admin adds the label `deployment=stable` to the stable release pods. The team adds the label `deployment=canary` to the canary release pod.

The team's stable release includes a requirement for hourly snapshots and daily backups. The canary release is more ephemeral, so they want to create a less aggressive, short-term Protection Policy for anything labeled `deployment=canary`.

In order to avoid possible data conflicts, the admin will create two custom apps: one for the "canary" release, and one for the "stable" release. This keeps the backups, snapshots, and clone operations separate for the two groups of Kubernetes objects.

Protect apps

Protection overview

You can create backups, clones, snapshots, and protection policies for your apps using Astra Control Center. Backing up your apps helps your services and associated data be as available as possible; during a disaster scenario, restoring from backup can ensure full recovery of an app and its associated data with minimal disruption. Backups, clones, and snapshots can help protect against common threats such as ransomware, accidental data loss, and environmental disasters. [Learn about the available types of data protection in Astra Control Center, and when to use them.](#)

App protection workflow

You can use the following example workflow to get started protecting your apps.

[One] Back up all apps

To make sure that your apps are immediately protected, [create a manual backup of all apps](#).

[Two] Configure a protection policy for each app

To automate future backups and snapshots, [configure a protection policy for each app](#). As an example, you can start with weekly backups and daily snapshots, with one month retention for both. Automating backups and snapshots with a protection policy is strongly recommended over manual backups and snapshots.

[Three] Optional: Adjust the protection policies

As apps and their usage patterns change, adjust the protection policies as needed to provide the best protection.

[Four] In case of a disaster, restore your apps

If data loss occurs, you can recover by [restoring the latest backup](#) first for each app. You can then restore the latest snapshot (if available).

Protect apps with snapshots and backups

Protect your apps by taking snapshots and backups using an automated protection policy or on an ad-hoc basis. You can use the Astra UI or [the Astra Control API](#) to protect apps.



If you use Helm to deploy apps, Astra Control Center requires Helm version 3. Managing and cloning apps deployed with Helm 3 (or upgraded from Helm 2 to Helm 3) are fully supported. Apps deployed with Helm 2 are not supported.



When you create a project for hosting an app on an OpenShift cluster, the project (or Kubernetes namespace) is assigned a SecurityContext UID. To enable Astra Control Center to protect your app and move the app to another cluster or project in OpenShift, you need to add policies that enable the app to run as any UID. As an example, the following OpenShift CLI commands grant the appropriate policies to a WordPress app.

```
oc new-project wordpress
oc adm policy add-scc-to-group anyuid system:serviceaccounts:wordpress
oc adm policy add-scc-to-user privileged -z default -n wordpress
```

Configure a protection policy

A protection policy protects an app by creating snapshots, backups, or both at a defined schedule. You can choose to create snapshots and backups hourly, daily, weekly, and monthly, and you can specify the number of copies to retain. As an example, a protection policy might create weekly backups and daily snapshots, and retain the backups and snapshots for one month. How often you create snapshots and backups and how long you retain them depends on the needs of your organization.

Steps

1. Select **Applications** and then select the name of an app.

2. Select **Data Protection**.
3. Select **Configure Protection Policy**.
4. Define a protection schedule by choosing the number of snapshots and backups to keep hourly, daily, weekly, and monthly.

You can define the hourly, daily, weekly, and monthly schedules concurrently. A schedule won't turn active until you set a retention level.

The following example sets four protection schedules: hourly, daily, weekly, and monthly for snapshots and backups.

5. Select **Review**.
6. Select **Set Protection Policy**.

Result

Astra Control Center implements the data protection policy by creating and retaining snapshots and backups using the schedule and retention policy that you defined.

Create a snapshot

You can create an on-demand snapshot at any time.

Steps

1. Select **Applications**.
2. From the Options menu in the **Actions** column for the desired app, select **Snapshot**.
3. Customize the name of the snapshot and then select **Review**.
4. Review the snapshot summary and select **Snapshot**.

Result

The snapshot process begins. A snapshot is successful when the status is **Available** in the **Actions** column on the **Data protection > Snapshots** page.

Create a backup

You can also back up an app at any time.



S3 buckets in Astra Control Center do not report available capacity. Before backing up or cloning apps managed by Astra Control Center, check bucket information in the ONTAP or StorageGRID management system.

Steps

1. Select **Applications**.
2. From the Options menu in the **Actions** column for the desired app, select **Backup**.
3. Customize the name of the backup.
4. Choose whether to back up the app from an existing snapshot. If you select this option, you can choose from a list of existing snapshots.

5. Choose a destination for the backup by selecting from the list of storage buckets.
6. Select **Review**.
7. Review the backup summary and select **Backup**.

Result

Astra Control Center creates a backup of the app.



If your network has an outage or is abnormally slow, a backup operation might time out. This causes the backup to fail.



There is no way to stop a running backup. If you need to delete the backup, wait until it has completed and then use the instructions in [Delete backups](#). To delete a failed backup, [use the Astra Control API](#).



After a data protection operation (clone, backup, restore) and subsequent persistent volume resize, there is up to a twenty-minute delay before the new volume size is shown in the UI. The data protection operation is successful within minutes, and you can use the management software for the storage backend to confirm the change in volume size.

View snapshots and backups

You can view the snapshots and backups of an app from the Data Protection tab.

Steps

1. Select **Applications** and then select the name of an app.
2. Select **Data Protection**.

The snapshots display by default.

3. Select **Backups** to see the list of backups.

Delete snapshots

Delete the scheduled or on-demand snapshots that you no longer need.

Steps

1. Select **Applications** and then select the name of an app.
2. Select **Data Protection**.
3. From the Options menu in the **Actions** column for the desired snapshot, select **Delete snapshot**.
4. Type the word "delete" to confirm deletion and then select **Yes, Delete snapshot**.

Result

Astra Control Center deletes the snapshot.

Delete backups

Delete the scheduled or on-demand backups that you no longer need.



There is no way to stop a running backup. If you need to delete the backup, wait until it has completed and then use these instructions. To delete a failed backup, [use the Astra Control API](#).

1. Select **Applications** and then select the name of an app.
2. Select **Data Protection**.
3. Select **Backups**.
4. From the Options menu in the **Actions** column for the desired backup, select **Delete backup**.
5. Type the word "delete" to confirm deletion and then select **Yes, Delete backup**.

Result

Astra Control Center deletes the backup.

Restore apps

Astra Control can restore your application from a snapshot or backup. Restoring from an existing snapshot will be faster when restoring the application to the same cluster. You can use the Astra Control UI or [the Astra Control API](#) to restore apps.

About this task

- It is strongly recommended to take a snapshot of or back up your application before restoring it. This will enable you to clone from the snapshot or backup in the event that the restore is unsuccessful.
- If you use Helm to deploy apps, Astra Control Center requires Helm version 3. Managing and cloning apps deployed with Helm 3 (or upgraded from Helm 2 to Helm 3) are fully supported. Apps deployed with Helm 2 are not supported.
- If you restore to a different cluster, ensure that the cluster is using the same persistent volume access mode (for example, ReadWriteMany). The restore operation will fail if the destination persistent volume access mode is different.
- Any member user with namespace constraints by namespace name/ID or by namespace labels can clone or restore an app to a new namespace on the same cluster or to any other cluster in their organization's account. However, the same user cannot access the cloned or restored app in the new namespace. After a new namespace is created by a clone or restore operation, the account admin/owner can edit the member user account and update role constraints for the affected user to grant access to the new namespace.
- When you create a project for hosting an app on an OpenShift cluster, the project (or Kubernetes namespace) is assigned a SecurityContext UID. To enable Astra Control Center to protect your app and move the app to another cluster or project in OpenShift, you need to add policies that enable the app to run as any UID. As an example, the following OpenShift CLI commands grant the appropriate policies to a WordPress app.

```
oc new-project wordpress
oc adm policy add-scc-to-group anyuid system:serviceaccounts:wordpress
oc adm policy add-scc-to-user privileged -z default -n wordpress
```

Steps

1. Select **Applications** and then select the name of an app.
2. Select **Data protection**.
3. If you want to restore from a snapshot, keep the **Snapshots** icon selected. Otherwise, select the **Backups** icon to restore from a backup.

4. From the Options menu in the **Actions** column for the snapshot or backup from which you want to restore, select **Restore application**.
5. **Restore details:** Specify details for the restored app. By default, the current cluster and namespace appear. Leave these values intact to restore an app in-place, which reverts the app to an earlier version of itself. Change these values if you want to restore to a different cluster or namespace.
 - Enter a name and namespace for the app.
 - Choose the destination cluster for the app.
 - Select **Review**.



If you restore to a namespace that was previously deleted, a new namespace with the same name is created as part of the restore process. Any users that had rights to manage apps in the previously deleted namespace need to manually restore rights to the newly re-created namespace.

6. **Restore Summary:** Review details about the restore action, type "restore", and select **Restore**.

Result

Astra Control Center restores the app based on the information that you provided. If you restored the app in-place, the contents of any existing persistent volumes are replaced with the contents of persistent volumes from the restored app.



After a data protection operation (clone, backup, restore) and subsequent persistent volume resize, there is a delay of up to twenty minutes before the new volume size is shown in the web UI. The data protection operation is successful within minutes, and you can use the management software for the storage backend to confirm the change in volume size.

Clone and migrate apps

Clone an existing app to create a duplicate app on the same Kubernetes cluster or on another cluster. When Astra Control Center clones an app, it creates a clone of your application configuration and persistent storage.

Cloning can help if you need to move applications and storage from one Kubernetes cluster to another. For example, you might want to move workloads through a CI/CD pipeline and across Kubernetes namespaces. You can use the Astra UI or [the Astra Control API](#) to clone and migrate apps.

What you'll need

To clone apps to a different cluster, you need a default bucket. When you add your first bucket, it becomes the default bucket.

About this task

- If you deploy an app with a StorageClass explicitly set and you need to clone the app, the target cluster must have the originally specified StorageClass. Cloning an application with an explicitly set StorageClass to a cluster that does not have the same StorageClass will fail.
- If you clone an operator-deployed instance of Jenkins CI, you need to manually restore the persistent data. This is a limitation of the app's deployment model.
- S3 buckets in Astra Control Center do not report available capacity. Before backing up or cloning apps managed by Astra Control Center, check bucket information in the ONTAP or StorageGRID management system.

- During an app backup or app restore, you can optionally specify a bucket ID. An app clone operation, however, always uses the default bucket that has been defined. There is no option to change buckets for a clone. If you want control over which bucket is used, you can either [change the bucket default](#) or do a [backup](#) followed by a [restore](#) separately.
- Any member user with namespace constraints by namespace name/ID or by namespace labels can clone or restore an app to a new namespace on the same cluster or to any other cluster in their organization's account. However, the same user cannot access the cloned or restored app in the new namespace. After a new namespace is created by a clone or restore operation, the account admin/owner can edit the member user account and update role constraints for the affected user to grant access to the new namespace.

OpenShift considerations

- If you clone an app between clusters, the source and destination clusters must be the same distribution of OpenShift. For example, if you clone an app from an OpenShift 4.7 cluster, use a destination cluster that is also OpenShift 4.7.
- When you create a project for hosting an app on an OpenShift cluster, the project (or Kubernetes namespace) is assigned a SecurityContext UID. To enable Astra Control Center to protect your app and move the app to another cluster or project in OpenShift, you need to add policies that enable the app to run as any UID. As an example, the following OpenShift CLI commands grant the appropriate policies to a WordPress app.

```
oc new-project wordpress
oc adm policy add-scc-to-group anyuid system:serviceaccounts:wordpress
oc adm policy add-scc-to-user privileged -z default -n wordpress
```

Steps

1. Select **Applications**.
2. Do one of the following:
 - Select the Options menu in the **Actions** column for the desired app.
 - Select the name of the desired app, and select the status drop-down list at the top right of the page.
3. Select **Clone**.
4. **Clone details:** Specify details for the clone:
 - Enter a name.
 - Enter a namespace for the clone.
 - Choose a destination cluster for the clone.
 - Choose whether you want to create the clone from an existing snapshot or backup. If you don't select this option, Astra Control Center creates the clone from the app's current state.
5. **Source:** If you chose to clone from an existing snapshot or backup, choose the snapshot or backup that you'd like to use.
6. Select **Review**.
7. **Clone Summary:** Review the details about the clone and select **Clone**.

Result

Astra Control Center clones that app based on the information that you provided. The clone operation is successful when the new app clone is in the `Available` state on the **Applications** page.



After a data protection operation (clone, backup, restore) and subsequent persistent volume resize, there is up to a twenty-minute delay before the new volume size is shown in the UI. The data protection operation is successful within minutes, and you can use the management software for the storage backend to confirm the change in volume size.

Manage app execution hooks

An execution hook is a custom script that you can run before or after a snapshot of a managed app. For example, if you have a database app, you can use execution hooks to pause all database transactions before a snapshot, and resume transactions after the snapshot is complete. This ensures application-consistent snapshots.

Default execution hooks and regular expressions

For some apps, Astra Control comes with default execution hooks, provided by NetApp, that handle freeze and thaw operations before and after snapshots. Astra Control uses regular expressions to match an app's container image to these apps:

- MariaDB
 - Matching regular expression: `\bmariadb\b`
- MySQL
 - Matching regular expression: `\bmysql\b`
- PostgreSQL
 - Matching regular expression: `\bpostgresql\b`

If there is a match, the NetApp-provided default execution hooks for that app appear in the app's list of active execution hooks, and those hooks run automatically when snapshots of that app are taken. If one of your custom apps has a similar image name that happens to match one of the regular expressions (and you don't want to use the default execution hooks), you can either change the image name, or disable the default execution hook for that app and use a custom hook instead.

You cannot delete or modify the default execution hooks.

Important notes about custom execution hooks

Consider the following when planning execution hooks for your apps.

- Astra Control requires execution hooks to be written in the format of executable shell scripts.
- Script size is limited to 128KB.
- Astra Control uses execution hook settings and any matching criteria to determine which hooks are applicable to a snapshot.
- All execution hook failures are soft failures; other hooks and the snapshot are still attempted even if a hook fails. However, when a hook fails, a warning event is recorded in the **Activity** page event log.
- To create, edit, or delete execution hooks, you must be a user with Owner, Admin, or Member permissions.
- If an execution hook takes longer than 25 minutes to run, the hook will fail, creating an event log entry with a return code of "N/A". Any affected snapshot will time out and be marked as failed, with a resulting event log entry noting the timeout.



Since execution hooks often reduce or completely disable the functionality of the application they are running against, you should always try to minimize the time your custom execution hooks take to run.

When a snapshot is run, execution hook events take place in the following order:

1. Any applicable NetApp-provided default pre-snapshot execution hooks are run on the appropriate containers.
2. Any applicable custom pre-snapshot execution hooks are run on the appropriate containers. You can create and run as many custom pre-snapshot hooks as you need, but the order of execution of these hooks before the snapshot is neither guaranteed nor configurable.
3. The snapshot is performed.
4. Any applicable custom post-snapshot execution hooks are run on the appropriate containers. You can create and run as many custom post-snapshot hooks as you need, but the order of execution of these hooks after the snapshot is neither guaranteed nor configurable.
5. Any applicable NetApp-provided default post-snapshot execution hooks are run on the appropriate containers.



You should always test your execution hook scripts before enabling them in a production environment. You can use the 'kubectl exec' command to conveniently test the scripts. After you enable the execution hooks in a production environment, test the resulting snapshots to ensure they are consistent. You can do this by cloning the app to a temporary namespace, restoring the snapshot, and then testing the app.

View existing execution hooks

You can view existing custom or NetApp-provided default execution hooks for an app.

Steps

1. Go to **Applications** and then select the name of a managed app.
2. Select the **Execution hooks** tab.

You can view all enabled or disabled execution hooks in the resulting list. You can see a hook's status, source, and when it runs (pre- or post-snapshot). To view event logs surrounding execution hooks, go to the **Activity** page in the left-side navigation area.

Create a custom execution hook

You can create a custom execution hook for an app. See [Execution hook examples](#) for hook examples. You need to have Owner, Admin, or Member permissions to create execution hooks.



When you create a custom shell script to use as an execution hook, remember to specify the appropriate shell at the beginning of the file, unless you are running linux commands or providing the full path to an executable.

Steps

1. Select **Applications** and then select the name of a managed app.
2. Select the **Execution hooks** tab.

3. Select **Add a new hook**.
4. In the **Hook Details** area, depending on when the hook should run, choose **Pre-Snapshot** or **Post-Snapshot**.
5. Enter a unique name for the hook.
6. (Optional) Enter any arguments to pass to the hook during execution, pressing the Enter key after each argument you enter to record each one.
7. In the **Container Images** area, if the hook should run against all container images contained within the application, enable the **Apply to all container images** check box. If instead the hook should act only on one or more specified container images, enter the container image names in the **Container image names to match** field.
8. In the **Script** area, do one of the following:
 - Upload a custom script.
 - a. Select the **Upload file** option.
 - b. Browse to a file and upload it.
 - c. Give the script a unique name.
 - d. (Optional) Enter any notes other administrators should know about the script.
 - Paste in a custom script from the clipboard.
 - a. Select the **Paste from clipboard** option.
 - b. Select the text field and paste the script text into the field.
 - c. Give the script a unique name.
 - d. (Optional) Enter any notes other administrators should know about the script.
9. Select **Add hook**.

Disable an execution hook

You can disable an execution hook if you want to temporarily prevent it from running before or after a snapshot of an app. You need to have Owner, Admin, or Member permissions to disable execution hooks.

Steps

1. Select **Applications** and then select the name of a managed app.
2. Select the **Execution hooks** tab.
3. Select the Options menu in the **Actions** column for a hook that you wish to disable.
4. Select **Disable**.

Delete an execution hook

You can remove an execution hook entirely if you no longer need it. You need to have Owner, Admin, or Member permissions to delete execution hooks.

Steps

1. Select **Applications** and then select the name of a managed app.
2. Select the **Execution hooks** tab.
3. Select the Options menu in the **Actions** column for a hook that you wish to delete.

4. Select **Delete**.

Execution hook examples

Use the following examples to get an idea of how to structure your execution hooks. You can use these hooks as templates, or as test scripts.

Simple success example

This is an example of a simple hook that succeeds and writes a message to standard output and standard error.

```
#!/bin/sh

# success_sample.sh
#
# A simple noop success hook script for testing purposes.
#
# args: None
#

#
# Writes the given message to standard output
#
# $* - The message to write
#
msg() {
    echo "$*"
}

#
# Writes the given information message to standard output
#
# $* - The message to write
#
info() {
    msg "INFO: $*"
}

#
# Writes the given error message to standard error
#
# $* - The message to write
#
error() {
```

```

    msg "ERROR: $" 1>&2
}

#
# main
#

# log something to stdout
info "running success_sample.sh"

# exit with 0 to indicate success
info "exit 0"
exit 0

```

Simple success example (bash version)

This is an example of a simple hook that succeeds and writes a message to standard output and standard error, written for bash.

```

#!/bin/bash

# success_sample.bash
#
# A simple noop success hook script for testing purposes.
#
# args: None

#
# Writes the given message to standard output
#
# $* - The message to write
#
msg() {
    echo "$*"
}

#
# Writes the given information message to standard output
#
# $* - The message to write
#
info() {
    msg "INFO: $"
}

```

```

}

#
# Writes the given error message to standard error
#
# $* - The message to write
#
error() {
    msg "ERROR: $*" 1>&2
}

#
# main
#

# log something to stdout
info "running success_sample.bash"

# exit with 0 to indicate success
info "exit 0"
exit 0

```

Simple success example (zsh version)

This is an example of a simple hook that succeeds and writes a message to standard output and standard error, written for Z shell.

```

#!/bin/zsh

# success_sample.zsh
#
# A simple noop success hook script for testing purposes.
#
# args: None
#

#
# Writes the given message to standard output
#
# $* - The message to write
#
msg() {
    echo "$*"
}

```

```

#
# Writes the given information message to standard output
#
# $* - The message to write
#
info() {
    msg "INFO: $*"
}

#
# Writes the given error message to standard error
#
# $* - The message to write
#
error() {
    msg "ERROR: $*" 1>&2
}

#
# main
#

# log something to stdout
info "running success_sample.zsh"

# exit with 0 to indicate success
info "exit 0"
exit 0

```

Success with arguments example

The following example demonstrates how you can use args in a hook.

```

#!/bin/sh

# success_sample_args.sh
#
# A simple success hook script with args for testing purposes.
#
# args: Up to two optional args that are echoed to stdout
#
#
# Writes the given message to standard output
#

```

```

# $* - The message to write
#
msg() {
    echo "$*"
}

#
# Writes the given information message to standard output
#
# $* - The message to write
#
info() {
    msg "INFO: $*"
}

#
# Writes the given error message to standard error
#
# $* - The message to write
#
error() {
    msg "ERROR: $*" 1>&2
}

#
# main
#

# log something to stdout
info "running success_sample_args.sh"

# collect args
arg1=$1
arg2=$2

# output args and arg count to stdout
info "number of args: $#"
```

```

info "arg1 ${arg1}"
info "arg2 ${arg2}"

# exit with 0 to indicate success
info "exit 0"
exit 0

```

Pre-snapshot / post-snapshot hook example

The following example demonstrates how the same script can be used for both a pre-snapshot and a post-snapshot hook.

```
#!/bin/sh

# success_sample_pre_post.sh
#
# A simple success hook script example with an arg for testing purposes
# to demonstrate how the same script can be used for both a prehook and
# posthook
#
# args: [pre|post]

# unique error codes for every error case
ebase=100
eusage=$((ebase+1))
ebadstage=$((ebase+2))
epre=$((ebase+3))
epost=$((ebase+4))

#
# Writes the given message to standard output
#
# $* - The message to write
#
msg() {
    echo "$*"
}

#
# Writes the given information message to standard output
#
# $* - The message to write
#
info() {
    msg "INFO: $*"
}

#
# Writes the given error message to standard error
#
# $* - The message to write
#
```

```

error() {
    msg "ERROR: $" 1>&2
}

#
# Would run prehook steps here
#
prehook() {
    info "Running noop prehook"
    return 0
}

#
# Would run posthook steps here
#
posthook() {
    info "Running noop posthook"
    return 0
}

#
# main
#

# check arg
stage=$1
if [ -z "${stage}" ]; then
    echo "Usage: $0 <pre|post>"
    exit ${eusage}
fi

if [ "${stage}" != "pre" ] && [ "${stage}" != "post" ]; then
    echo "Invalid arg: ${stage}"
    exit ${ebadstage}
fi

# log something to stdout
info "running success_sample_pre_post.sh"

if [ "${stage}" = "pre" ]; then
    prehook
    rc=$?
    if [ ${rc} -ne 0 ]; then
        error "Error during prehook"
    fi
fi

```



```

    fi
fi

if [ "${stage}" = "post" ]; then
    posthook
    rc=$?
    if [ ${rc} -ne 0 ]; then
        error "Error during posthook"
    fi
fi

exit ${rc}

```

Failure example

The following example demonstrates how you can handle failures in a hook.

```

#!/bin/sh

# failure_sample_arg_exit_code.sh
#
# A simple failure hook script for testing purposes.
#
# args: [the exit code to return]
#

#
# Writes the given message to standard output
#
# $* - The message to write
#
msg() {
    echo "$*"
}

#
# Writes the given information message to standard output
#
# $* - The message to write
#
info() {
    msg "INFO: $*"
}

```

```

#
# Writes the given error message to standard error
#
# $* - The message to write
#
error() {
    msg "ERROR: $" 1>&2
}

#
# main
#

# log something to stdout
info "running failure_sample_arg_exit_code.sh"

argexitcode=$1

# log to stderr
error "script failed, returning exit code ${argexitcode}"

# exit with specified exit code
exit ${argexitcode}

```

Verbose failure example

The following example demonstrates how you can handle failures in a hook, with more verbose logging.

```

#!/bin/sh

# failure_sample_verbose.sh
#
# A simple failure hook script with args for testing purposes.
#
# args: [The number of lines to output to stdout]

#
# Writes the given message to standard output
#
# $* - The message to write
#
msg() {
    echo "$*"
}

```

```

#
# Writes the given information message to standard output
#
# $* - The message to write
#
info() {
    msg "INFO: $*"
}

#
# Writes the given error message to standard error
#
# $* - The message to write
#
error() {
    msg "ERROR: $*" 1>&2
}

#
# main
#

# log something to stdout
info "running failure_sample_verbose.sh"

# output arg value to stdout
linecount=$1
info "line count ${linecount}"

# write out a line to stdout based on line count arg
i=1
while [ "$i" -le ${linecount} ]; do
    info "This is line ${i} from failure_sample_verbose.sh"
    i=$(( i + 1 ))
done

error "exiting with error code 8"
exit 8

```

Failure with an exit code example

The following example demonstrates a hook failing with an exit code.

```
#!/bin/sh

# failure_sample_arg_exit_code.sh
#
# A simple failure hook script for testing purposes.
#
# args: [the exit code to return]
#

#
# Writes the given message to standard output
#
# $* - The message to write
#
msg() {
    echo "$*"
}

#
# Writes the given information message to standard output
#
# $* - The message to write
#
info() {
    msg "INFO: $*"
}

#
# Writes the given error message to standard error
#
# $* - The message to write
#
error() {
    msg "ERROR: $*" 1>&2
}

#
# main
#

# log something to stdout
info "running failure_sample_arg_exit_code.sh"

argexitcode=$1
```

```
# log to stderr
error "script failed, returning exit code ${argexitcode}"

# exit with specified exit code
exit ${argexitcode}
```

Success after failure example

The following example demonstrates a hook failing the first time it is run, but succeeding after the second run.

```
#!/bin/sh

# failure_then_success_sample.sh
#
# A hook script that fails on initial run but succeeds on second run for
# testing purposes.
#
# Helpful for testing retry logic for post hooks.
#
# args: None
#

#
# Writes the given message to standard output
#
# $* - The message to write
#
msg() {
    echo "$*"
}

#
# Writes the given information message to standard output
#
# $* - The message to write
#
info() {
    msg "INFO: $*"
}

#
# Writes the given error message to standard error
#
# $* - The message to write
```

```
#
error() {
    msg "ERROR: $" 1>&2
}

#
# main
#

# log something to stdout
info "running failure_success sample.sh"

if [ -e /tmp/hook-test.junk ] ; then
    info "File does exist. Removing /tmp/hook-test.junk"
    rm /tmp/hook-test.junk
    info "Second run so returning exit code 0"
    exit 0
else
    info "File does not exist. Creating /tmp/hook-test.junk"
    echo "test" > /tmp/hook-test.junk
    error "Failed first run, returning exit code 5"
    exit 5
fi
```

View app and cluster health

View a summary of app and cluster health

Select the **Dashboard** to see a high-level view of your apps, clusters, storage backends, and their health.

These aren't just static numbers or statuses—you can drill down from each. For example, if apps aren't fully protected, you can hover over the icon to identify which apps aren't fully protected, which includes a reason why.

Applications tile

The **Applications** tile helps you identify the following:

- How many apps you're currently managing with Astra.
- Whether those managed apps are healthy.
- Whether the apps are fully protected (they're protected if recent backups are available).
- The number of apps that were discovered, but are not yet managed.

Ideally, this number would be zero because you would either manage or ignore apps after they're

discovered. And then you would monitor the number of discovered apps on the Dashboard to identify when developers add new apps to a cluster.

Clusters tile

The **Clusters** tile provides similar details about the health of the clusters that you are managing by using Astra Control Center, and you can drill down to get more details just like you can with an app.

Storage backends tile

The **Storage backends** tile provides information to help you identify the health of storage backends including:

- How many storage backends are managed
- Whether these managed backends are healthy
- Whether the backends are fully protected
- The number of backends that are discovered, but are not yet managed.

View the health and details of clusters

After you add clusters to be managed by Astra Control Center, you can view details about the cluster, such as its location, the worker nodes, persistent volumes, and storage classes.

Steps

1. In the Astra Control Center UI, select **Clusters**.
2. On the **Clusters** page, select the cluster whose details you want to view.



If a cluster is in `removed` state yet cluster and network connectivity appears healthy (external attempts to access the cluster using Kubernetes APIs are successful), the kubeconfig you provided to Astra Control might no longer be valid. This can be due to certificate rotation or expiration on the cluster. To correct this issue, update the credentials associated with the cluster in Astra Control using the [Astra Control API](#).

3. View the information on the **Overview**, **Storage**, and **Activity** tabs to find the information that you're looking for.
 - **Overview**: Details about the worker nodes, including their state.
 - **Storage**: The persistent volumes associated with the compute, including the storage class and state.
 - **Activity**: Shows the activities related to the cluster.



You can also view cluster information starting from the Astra Control Center **Dashboard**. On the **Clusters** tab under **Resource summary**, you can select the managed clusters, which takes you to the **Clusters** page. After you get to the **Clusters** page, follow the steps outlined above.

View the health and details of an app

After you start managing an app, Astra provides details about the app that enables you to identify its status (whether it's healthy), its protection status (whether it's fully protected in case of failure), the pods, persistent storage, and more.

Steps

1. In the Astra Control Center UI, select **Applications** and then select the name of an app.
2. Find the information that you're looking for:

App Status

Provides a status that reflects the app's state in Kubernetes. For example, are pods and persistent volumes online? If an app is unhealthy, you'll need to go and troubleshoot the issue on the cluster by looking at Kubernetes logs. Astra doesn't provide information to help you fix a broken app.

App Protection Status

Provides a status of how well the app is protected:

- **Fully protected:** The app has an active backup schedule and a successful backup that's less than a week old
- **Partially protected:** The app has an active backup schedule, an active snapshot schedule, or a successful backup or snapshot
- **Unprotected:** Apps that are neither fully protected or partially protected.

You can't be fully protected until you have a recent backup. This is important because backups are stored in an object store away from the persistent volumes. If a failure or accident wipes out the cluster and it's persistent storage, then you need a backup to recover. A snapshot wouldn't enable you to recover.

Overview

Information about the state of the pods that are associated with the app.

Data protection

Enables you to configure a data protection policy and to view the existing snapshots and backups.

Storage

Shows you the app-level persistent volumes. The state of a persistent volume is from the perspective of the Kubernetes cluster.

Resources

Enables you to verify which resources are being backed up and managed.

Activity

Shows the activities related to the app.



You can also view app information starting from the Astra Control Center **Dashboard**. On the **Applications** tab under **Resource summary**, you can select the managed apps, which takes you to the **Applications** page. After you get to the **Applications** page, follow the steps outlined above.

Manage your account

Manage users

You can invite, add, remove, and edit users of your Astra Control Center installation using the Astra Control UI.

You can use the Astra Control UI or [the Astra Control API](#) to manage users.

Invite users

Account Owners and Admins can invite new users to Astra Control Center.

Steps

1. In the **Manage Your Account** navigation area, select **Account**.
2. Select the **Users** tab.
3. Select **Invite User**.
4. Enter the user's name and email address.
5. Select a user role with the appropriate system permissions.

Each role provides the following permissions:

- A **Viewer** can view resources.
 - A **Member** has Viewer role permissions and can manage apps and clusters, unmanage apps, and delete snapshots and backups.
 - An **Admin** has Member role permissions and can add and remove any other users except the Owner.
 - An **Owner** has Admin role permissions and can add and remove any user accounts.
6. To add constraints to a user with a Member or Viewer role, enable the **Restrict role to constraints** check box.

For more information on adding constraints, see [Manage roles](#).

7. Select **Invite users**.

The user receives an email informing them that they've been invited to Astra Control Center. The email includes temporary password, which they'll need to change upon first login.

Add users

Account Owners and Admins can add more users to the Astra Control Center installation.

Steps

1. In the **Manage Your Account** navigation area, select **Account**.
2. Select the **Users** tab.
3. Select **Add User**.
4. Enter the user's name, email address, and a temporary password.

The user will need to change the password upon first login.

5. Select a user role with the appropriate system permissions.

Each role provides the following permissions:

- A **Viewer** can view resources.
- A **Member** has Viewer role permissions and can manage apps and clusters, unmanage apps, and delete snapshots and backups.

- An **Admin** has Member role permissions and can add and remove any other users except the Owner.
 - An **Owner** has Admin role permissions and can add and remove any user accounts.
6. To add constraints to a user with a Member or Viewer role, enable the **Restrict role to constraints** check box.

For more information on adding constraints, see [Manage roles](#).

7. Select **Add**.

Manage passwords

You can manage passwords for user accounts in Astra Control Center.

Change your password

You can change the password of your user account at any time.

Steps

1. Select the User icon at the top right of the screen.
2. Select **Profile**.
3. From the Options menu in the **Actions** column, and select **Change Password**.
4. Enter a password that conforms to the password requirements.
5. Enter the password again to confirm.
6. Select **Change password**.

Reset another user's password

If your account has Admin or Owner role permissions, you can reset passwords for other user accounts as well as your own. When you reset a password, you assign a temporary password that the user will have to change upon logging in.

Steps

1. In the **Manage Your Account** navigation area, select **Account**.
2. Select the **Actions** drop-down list.
3. Select **Reset Password**.
4. Enter a temporary password that conforms to the password requirements.
5. Enter the password again to confirm.



The next time the user logs in, the user will be prompted to change the password.

6. Select **Reset password**.

Change a user's role

Users with the Owner role can change the role of all users, while users with the Admin role can change the role of users who have the Admin, Member, or Viewer role.

Steps

1. In the **Manage Your Account** navigation area, select **Account**.
2. Select the **Actions** drop-down list.
3. Select **Edit role**.
4. Select a new role.
5. To apply constraints to the role, enable the **Restrict role to constraints** check box and select a constraint from the list.

If there are no constraints, you can add a constraint. For more information, see [Manage roles](#).

6. Select **Confirm**.

Result

Astra Control Center updates the user's permissions based on the new role that you selected.

Remove users

Users with the Owner or Admin role can remove other users from the account at any time.

Steps

1. In the **Manage Your Account** navigation area, select **Account**.
2. In the **Users** tab, select the check box in the row of each user that you want to remove.
3. From the Options menu in the **Actions** column, select **Remove user/s**.
4. When you're prompted, confirm deletion by typing the word "remove" and then select **Yes, Remove User**.

Result

Astra Control Center removes the user from the account.

Manage roles

You can manage roles by adding namespace constraints and restricting user roles to those constraints. This enables you to control access to resources within your organization. You can use the Astra Control UI or [the Astra Control API](#) to manage roles.

Add a namespace constraint to a role

An Admin or Owner user can add namespace constraints.

Steps

1. In the **Manage Your Account** navigation area, select **Account**.
2. Select the **Users** tab.
3. In the **Actions** column, select the menu button for a user with the Member or Viewer role.
4. Select **Edit role**.
5. Enable the **Restrict role to constraints** check box.

The check box is only available for Member or Viewer roles. You can select a different role from the **Role** drop-down list.

6. Select **Add constraint**.

You can view the list of available constraints by namespace or by namespace label.

7. In the **Constraint type** drop-down list, select either **Kubernetes namespace** or **Kubernetes namespace label** depending on how your namespaces are configured.
8. Select one or more namespaces or labels from the list to compose a constraint that restricts roles to those namespaces.
9. Select **Confirm**.

The **Edit role** page displays the list of constraints you've chosen for this role.

10. Select **Confirm**.

On the **Account** page, you can view the constraints for any Member or Viewer role in the **Role** column.



If you enable constraints for a role and select **Confirm** without adding any constraints, the role is considered to have full restrictions (the role is denied access to any resources that are assigned to namespaces).

Remove a namespace constraint from a role

An Admin or Owner user can remove a namespace constraint from a role.

Steps

1. In the **Manage Your Account** navigation area, select **Account**.
2. Select the **Users** tab.
3. In the **Actions** column, select the menu button for a user with the Member or Viewer role that has active constraints.
4. Select **Edit role**.

The **Edit role** dialog displays the active constraints for the role.

5. Select the **X** to the right of the constraint you need to remove.
6. Select **Confirm**.

For more information

- [User roles and namespaces](#)

View and manage notifications

Astra notifies you when actions have completed or failed. For example, you'll see a notification if a backup of an app completed successfully.

You can manage these notifications from the top right of the interface:



Steps

1. Select the number of unread notifications in the top right.
2. Review the notifications and then select **Mark as read** or **Show all notifications**.

If you selected **Show all notifications**, the Notifications page loads.

3. On the **Notifications** page, view the notifications, select the ones that you want to mark as read, select **Action** and select **Mark as read**.

Add and remove credentials

Add and remove credentials for local private cloud providers such as ONTAP S3, Kubernetes clusters managed with OpenShift, or unmanaged Kubernetes clusters from your account at any time. Astra Control Center uses these credentials to discover Kubernetes clusters and the apps on the clusters, and to provision resources on your behalf.

Note that all users in Astra Control Center share the same sets of credentials.

Add credentials

You can add credentials to Astra Control Center when you manage clusters. To add credentials by adding a new cluster, see [Add a Kubernetes cluster](#).



If you create your own `kubeconfig` file, you should define only **one** context element in it. See [Kubernetes documentation](#) for information about creating `kubeconfig` files.

Remove credentials

Remove credentials from an account at any time. You should only remove credentials after [unmanaging all associated clusters](#).



The first set of credentials that you add to Astra Control Center is always in use because Astra Control Center uses the credentials to authenticate to the backup bucket. It's best not to remove these credentials.

Steps

1. Select **Account**.
2. Select the **Credentials** tab.
3. Select the Options menu in the **State** column for the credentials that you want to remove.
4. Select **Remove**.
5. Type the word "remove" to confirm deletion and then select **Yes, Remove Credential**.

Result

Astra Control Center removes the credentials from the account.

Monitor account activity

You can view details about the activities in your Astra Control account. For example, when new users were invited, when a cluster was added, or when a snapshot was taken. You also have the ability to export your account activity to a CSV file.

View all account activity in Astra Control

1. Select **Activity**.
2. Use the filters to narrow down the list of activities or use the search box to find exactly what you're looking for.
3. Select **Export to CSV** to download your account activity to a CSV file.

View account activity for a specific app

1. Select **Applications** and then select the name of an app.
2. Select **Activity**.

View account activity for clusters

1. Select **Clusters** and then select the name of the cluster.
2. Select **Activity**.

Take action to resolve events that require attention

1. Select **Activity**.
2. Select an event that requires attention.
3. Select the **Take action** drop-down option.

From this list, you can view possible corrective actions that you can take, view documentation related to the issue, and get support to help resolve the issue.

Update an existing license

You can convert an evaluation license to a full license, or you can update an existing evaluation or full license with a new license. If you don't have a full license, work with your NetApp sales contact to obtain a full license and serial number. You can use the Astra UI or [the Astra Control API](#) to update an existing license.

Steps

1. Log in to the [NetApp Support Site](#).
2. Access the Astra Control Center Download page, enter the serial number, and download the full NetApp license file (NLF).
3. Log in to the Astra Control Center UI.
4. From the left navigation, select **Account > License**.
5. In the **Account > License** page, select the status drop-down menu for the existing license and select **Replace**.
6. Browse to the license file that you downloaded.
7. Select **Add**.

The **Account > Licenses** page displays the license information, expiration date, license serial number, account ID, and CPU units used.

For more information

- [Astra Control Center licensing](#)

Manage repository connections

You can connect repositories to Astra Control to use as a reference for software package installation images and artifacts. When you import software packages, Astra Control references installation images in the image repository and binaries and other artifacts in the artifact repository.

What you'll need

- Kubernetes cluster with Astra Control Center installed
- A running Docker repository that you can access
- A running artifact repository (such as Artifactory) that you can access

Connect a Docker image repository

You can connect a Docker image repository to hold package installation images, such as those for Astra Data Store. When you install packages, Astra Control imports the package image files from the image repository.

Steps

1. In the **Manage Your Account** navigation area, select **Account**.
2. Select the **Connections** tab.
3. In the **Docker Image Repository** section, select the menu at the top right.
4. Select **Connect**.
5. Add the URL and port for the repository.
6. Enter the credentials for the repository.
7. Select **Connect**.

Result

The repository is connected. In the **Docker Image Repository** section, the repository should show a connected status.

Disconnect a Docker image repository

You can remove the connection to a Docker image repository if it is no longer needed.

Steps

1. In the **Manage Your Account** navigation area, select **Account**.
2. Select the **Connections** tab.
3. In the **Docker Image Repository** section, select the menu at the top right.
4. Select **Disconnect**.
5. Select **Yes, disconnect Docker image repository**.

Result

The repository is disconnected. In the **Docker Image Repository** section, the repository should show a disconnected status.

Connect an artifact repository

You can connect an artifact repository to host artifacts such as software package binaries. When you install packages, Astra Control imports the artifacts for the software packages from the image repository.

Steps

1. In the **Manage Your Account** navigation area, select **Account**.
2. Select the **Connections** tab.
3. In the **Artifact Repository** section, select the menu at the top right.
4. Select **Connect**.
5. Add the URL and port for the repository.
6. If authentication is required, enable the **Use authentication** check box and enter the credentials for the repository.
7. Select **Connect**.

Result

The repository is connected. In the **Artifact Repository** section, the repository should show a connected status.

Disconnect an artifact repository

You can remove the connection to an artifact repository if it is no longer needed.

Steps

1. In the **Manage Your Account** navigation area, select **Account**.
2. Select the **Connections** tab.
3. In the **Artifact Repository** section, select the menu at the top right.
4. Select **Disconnect**.
5. Select **Yes, disconnect artifact repository**.

Result

The repository is disconnected. In the **Artifact Repository** section, the repository should show a disconnected status.

Find more information

- [Manage software packages](#)

Manage software packages

NetApp delivers additional capabilities for Astra Control Center with software packages that you can download from the NetApp Support Site. After you connect Docker and artifact repositories, you can upload and import packages to add this functionality to Astra Control Center. You can use the CLI or the Astra Control Center web UI to manage software packages.

What you'll need

- Kubernetes cluster with Astra Control Center installed
- A connected Docker image repository to hold software package images. For more information, see [Manage repository connections](#).
- A connected artifact repository to hold software package binaries and artifacts. For more information, see [Manage repository connections](#).
- A software package from the NetApp Support Site

Upload software package images to the repositories

Astra Control Center references package images and artifacts in connected repositories. You can upload images and artifacts to the repositories using the CLI.

Steps

1. Download the software package from the NetApp Support Site, and save it on a machine that has the `kubectl` utility installed.
2. Extract the compressed package file, and change directory to the location of the Astra Control bundle file (for example, `acc.manifest.bundle.yaml`).
3. Push the package images to the Docker repository. Make the following substitutions:
 - Replace `BUNDLE_FILE` with the name of the Astra Control bundle file.
 - Replace `MY_REGISTRY` with the URL of the Docker repository.
 - Replace `MY_REGISTRY_USER` and `MY_REGISTRY_PASSWORD` with the credentials for the repository.

```
kubectl astra packages push-images -m BUNDLE_FILE -r MY_REGISTRY -u  
MY_REGISTRY_USER -p MY_REGISTRY_PASSWORD
```

4. If the package has artifacts, copy the artifacts to the artifact repository. Replace `BUNDLE_FILE` with the name of the Astra Control bundle file, and `NETWORK_LOCATION` with the network location to copy the artifact files to:

```
kubectl astra packages copy-artifacts -m BUNDLE_FILE -n NETWORK_LOCATION
```

Add a software package

You can import software packages using an Astra Control Center bundle file. Doing this installs the package and makes the software available for Astra Control Center to use.

Add a software package using the Astra Control web UI

You can use the Astra Control Center web UI to add a software package that has been uploaded to the connected repositories.

Steps

1. In the **Manage Your Account** navigation area, select **Account**.
2. Select the **Packages** tab.
3. Select the **Add** button.
4. In the file selection dialog, select the upload icon.
5. Choose an Astra Control bundle file, in `.yaml` format, to upload.
6. Select **Add**.

Result

If the bundle file is valid and the package images and artifacts are located in your connected repositories, the

package is added to Astra Control Center. When the status in the **Status** column changes to **Available**, you can use the package. You can hover over the status for a package to get more information.



If one or more images or artifacts for a package are not found in your repository, an error message appears for that package.

Add a software package using the CLI

You can use the CLI to import a software package that you have uploaded to the connected repositories. To do this, you first need to record your Astra Control Center account ID and an API token.

Steps

1. Using a web browser, log in to the Astra Control Center web UI.
2. From the Dashboard, select the user icon at the top right.
3. Select **API access**.
4. Note the Account ID near the top of the screen.
5. Select **Generate API token**.
6. In the resulting dialog, select **Generate API token**.
7. Note the resulting token, and select **Close**.

In the CLI, change directories to the location of the `.yaml` bundle file in the extracted package contents.

8. Import the package using the bundle file, making the following substitutions:
 - Replace `BUNDLE_FILE` with the name of the Astra Control bundle file.
 - Replace `SERVER` with the DNS name of the Astra Control instance.
 - Replace `ACCOUNT_ID` and `TOKEN` with the account ID and API token you recorded earlier.

```
kubectl astra packages import -m BUNDLE_FILE -u SERVER -a ACCOUNT_ID  
-k TOKEN
```

Result

If the bundle file is valid and the package images and artifacts are located in your connected repositories, the package is added to Astra Control Center.



If one or more images or artifacts for a package are not found in your repository, an error message appears for that package.

Remove a software package

You can use the Astra Control Center web UI to remove a software package that you previously imported in Astra Control Center.

Steps

1. In the **Manage Your Account** navigation area, select **Account**.
2. Select the **Packages** tab.

You can see the list of installed packages and their statuses on this page.

3. In the **Actions** column for the package, open the actions menu.
4. Select **Delete**.

Result

The package is deleted from Astra Control Center, but the images and artifacts for the package remain in your repositories.

Find more information

- [Manage repository connections](#)

Manage buckets

An object store bucket provider is essential if you want to back up your applications and persistent storage or if you want to clone applications across clusters. Using Astra Control Center, add an object store provider as your off-cluster, backup destination for your apps.

You don't need a bucket if you are cloning your application configuration and persistent storage to the same cluster.

Use one of the following Amazon Simple Storage Service (S3) bucket providers:

- NetApp ONTAP S3
- NetApp StorageGRID S3
- Generic S3
- Microsoft Azure



Although Astra Control Center supports Amazon S3 as a Generic S3 bucket provider, Astra Control Center might not support all object store vendors that claim Amazon's S3 support.

A bucket can be in one of these states:

- pending: The bucket is scheduled for discovery.
- available: The bucket is available for use.
- removed: The bucket is not currently accessible.

For instructions on how to manage buckets using the Astra Control API, see the [Astra Automation and API information](#).

You can do these tasks related to managing buckets:

- [Add a bucket](#)
- [Edit a bucket](#)
- [Rotate or remove bucket credentials](#)
- [Remove a bucket](#)



S3 buckets in Astra Control Center do not report available capacity. Before backing up or cloning apps managed by Astra Control Center, check bucket information in the ONTAP or StorageGRID management system.

Edit a bucket

You can change the access credential information for a bucket and change whether a selected bucket is the default bucket.



When you add a bucket, select the correct bucket provider and provide the right credentials for that provider. For example, the UI accepts NetApp ONTAP S3 as the type and accepts StorageGRID credentials; however, this will cause all future app backups and restores using this bucket to fail. See the [Release Notes](#).

Steps

1. From the left navigation, select **Buckets**.
2. From the Options menu in the **Actions** column, select **Edit**.
3. Change any information other than the bucket type.



You can't modify the bucket type.

4. Select **Update**.

Rotate or remove bucket credentials

Astra Control uses bucket credentials to gain access and provide secret keys for an S3 bucket so that Astra Control Center can communicate with the bucket.

Rotate bucket credentials

If you rotate credentials, rotate them during a maintenance window when no backups are in progress (scheduled or on-demand).

Steps to edit and rotate credentials

1. From the left navigation, select **Buckets**.
2. From the Options menu in the **Actions** column, select **Edit**.
3. Create the new credential.
4. Select **Update**.

Remove bucket credentials

You should remove bucket credentials only if new credentials have been applied to a bucket, or if the bucket is no longer actively used.



The first set of credentials that you add to Astra Control is always in use because Astra Control uses the credentials to authenticate the backup bucket. Do not remove these credentials if the bucket is in active use as this will lead to backup failures and backup unavailability.



If you do remove active bucket credentials, see [troubleshooting bucket credential removal](#).

For instructions on how to remove S3 credentials using the Astra Control API, see the [Astra Automation and API information](#).

Remove a bucket

You can remove a bucket that is no longer in use or is not healthy. You might want to do this to keep your object store configuration simple and up-to-date.



You cannot remove a default bucket. If you want to remove that bucket, first select another bucket as the default.

What you'll need

- You should check to ensure that there are no running or completed backups for this bucket before you begin.
- You should check to ensure that the bucket is not being used in any active protection policy.

If there are, you will not be able to continue.

Steps

1. From left navigation, select **Buckets**.
2. From the **Actions** menu, select **Remove**.



Astra Control ensures first that there are no schedule policies using the bucket for backups and that there are no active backups in the bucket you are about to remove.

3. Type "remove" to confirm the action.
4. Select **Yes, remove bucket**.

Find more information

- [Use the Astra Control API](#)

Manage the storage backend

Managing storage clusters in Astra Control as a storage backend enables you to get linkages between persistent volumes (PVs) and the storage backend as well as additional storage metrics. You can monitor storage capacity and health details, including performance if Astra Control Center is connected to Cloud Insights.

For instructions on how to manage storage backends using the Astra Control API, see the [Astra Automation and API information](#).

You can complete the following tasks related to managing a storage backend:

- [Add a storage backend](#)
- [View storage backend details](#)
- [Unmanage a storage backend](#)
- [Update a storage backend license](#)
- [Add nodes to a storage backend cluster](#)
- [Remove a storage backend](#)

View storage backend details

You can view storage backend information from the Dashboard or from the Backends option.

In the Storage Backend Details page, for Astra Data Store, you can see the following information:

- Astra Data Store cluster
 - Throughput, IOPS, and latency
 - Used capacity compared to total capacity
- For each Astra Data Store cluster volume
 - Used capacity compared to total capacity
 - Throughput

View storage backend details from the Dashboard

Steps

1. From the left navigation, select **Dashboard**.
2. Review the Storage backend section that shows the state:
 - **Unhealthy**: The storage is not in an optimal state. This could be due to a latency issue or an app is degraded due to a container issue, for example.
 - **All healthy**: The storage has been managed and is in an optimal state.
 - **Discovered**: The storage has been discovered, but not managed by Astra Control.

View storage backend details from the Backends option

View information about the backend health, capacity, and performance (IOPS throughput and/or latency).

With a connection to Cloud Insights, you can see the volumes that the Kubernetes apps are using, which are stored on a selected storage backend.

Steps

1. In the left navigation area, select **Backends**.
2. Select the storage backend.



If you connected to NetApp Cloud Insights, excerpts of data from Cloud Insights appear on the Backends page.

Umeng-Aff300-05-06 Available

Storage backend status: **Healthy**

Capacity (Physical): 37.3% 7.93/21.28 TiB

Performance (Last 24 hrs): Throughput, MB/s

BASIC INFORMATION

Type: ONTAP 9.7.0 Cloud private Credentials Updated 2021/07/28 21:44 UTC

NETWORK

Cluster management IP address: [10.10.10.10](#)

Persistent volumes

Name	Persistent volume	Capacity	App/s	Cluster/s	Cloud
trident_pvc_...	pvc-...	0.04/46.57 GiB: 0.1%	netapp-acc	openshift-cluster010	private
trident_pvc_...	pvc-...	0.34/23.28 GiB: 1.44%	netapp-acc	openshift-cluster010	private
trident_pvc_...	pvc-...	0.02/0.93 GiB: 2.33%	netapp-acc	openshift-cluster010	private
trident_pvc_...	pvc-...	3.02/50.00 GiB: 6.04%	netapp-acc polaris-mongodb-mongodb	openshift-cluster010	private
trident_pvc_...	pvc-...	0.19/8.00 GiB: 2.39%	apps-mysql mysql-mysql	openshift-cluster010	private
trident_pvc_...	pvc-...	0.41/50.00 GiB: 0.81%	netapp-acc polaris-influxdb2-polaris-influxdb2	openshift-cluster010	private
trident_pvc_...	pvc-...	2.93/50.00 GiB: 5.87%	netapp-acc polaris-mongodb-mongodb	openshift-cluster010	private
trident_pvc_...	pvc-...	0.03/10.00 GiB: 0.26%	netapp-acc polaris-consul-consul	openshift-cluster010	private

- To go directly to Cloud Insights, select the **Cloud Insights** icon next to the metrics image.

Unmanage a storage backend

You can unmanage the backend.

Steps

- From the left navigation, select **Backends**.
- Select the storage backend.
- From the Options menu in the **Actions** column, select **Unmanage**.
- Type "unmanage" to confirm the action.
- Select **Yes, unmanage storage backend**.

Remove a storage backend

You can remove a storage backend that is no longer in use. You might want to do this to keep your configuration simple and up-to-date.



If you are removing an Astra Data Store backend, it must not have been created by vCenter.

What you'll need

- Ensure that the storage backend is unmanaged.

- Ensure that the storage backend does not have any volumes associated with the Astra Data Store cluster.

Steps

1. From left navigation, select **Backends**.
2. If the backend is managed, unmanage it.
 - a. Select **Managed**.
 - b. Select the storage backend.
 - c. From the **Actions** option, select **Unmanage**.
 - d. Type "unmanage" to confirm the action.
 - e. Select **Yes, unmanage storage backend**.
3. Select **Discovered**.
 - a. Select the storage backend.
 - b. From the **Actions** option, select **Remove**.
 - c. Type "remove" to confirm the action.
 - d. Select **Yes, remove storage backend**.

Update a storage backend license

You can update the license for an Astra Data Store storage backend to support a larger deployment or enhanced features.

What you'll need

- A deployed and managed Astra Data Store storage backend
- An Astra Data Store license file (contact your NetApp sales representative to purchase an Astra Data Store license)

Steps

1. From the left navigation, select **Backends**.
2. Select the name of a storage backend.
3. Under **Basic Information**, you can see the type of license installed.

If you hover over the license information, a popup appears with more information, such as expiration and entitlement information.

4. Under **License**, select the edit icon next to the license name.
5. In the **Update license** page, do one of the following:

License status	Action
At least one license has been added to Astra Data Store.	Select a license from the list.
No licenses have been added to Astra Data Store.	<ol style="list-style-type: none"> a. Select the Add button. b. Select a license file to upload. c. Select Add to upload the license file.

6. Select **Update**.

Add nodes to a storage backend cluster

You can add nodes to an Astra Data Store cluster, up to the number of nodes supported by the type of license installed for Astra Data Store.

What you'll need

- A deployed and licensed Astra Data Store storage backend
- You have added the Astra Data Store software package in Astra Control Center
- One or more new nodes to add to the cluster

Steps

1. From the left navigation, select **Backends**.
2. Select the name of a storage backend.
3. Under Basic Information, you can see the number of nodes in this storage backend cluster.
4. Under **Nodes**, select the edit icon next to the number of nodes.
5. In the **Add nodes** page, enter information about the new node or nodes:
 - a. Assign a node label for each node.
 - b. Do one of the following:
 - If you want Astra Data Store to always use the maximum available number of nodes according to your license, enable the **Always use up to maximum number of nodes allowed** check box.
 - If you don't want Astra Data Store to always use the maximum available number of nodes, select the desired number of total nodes to use.
 - c. If you deployed Astra Data Store with Protection Domains enabled, assign the new node or nodes to Protection Domains.
6. Select **Next**.
7. Enter IP address and network information for each new node. Enter a single IP address for a single new node, or an IP address pool for multiple new nodes.

If Astra Data Store can use the IP addresses configured during deployment, you don't need to enter any IP address information.

8. Select **Next**.
9. Review the configuration for the new node or nodes.
10. Select **Add nodes**.

Find more information

- [Use the Astra Control API](#)

Monitor and protect infrastructure

You can configure several optional settings to enhance your Astra Control Center experience. If the network where you're running Astra Control Center requires a proxy for connecting to the Internet (to upload support bundles to NetApp Support Site or establish a connection to Cloud Insights), you should configure a proxy

server in Astra Control Center. To monitor and gain insight into your complete infrastructure, create a connection to NetApp Cloud Insights. To collect Kubernetes events from systems monitored by Astra Control Center, add a Fluentd connection.

Add a proxy server

If the network where you're running Astra Control Center requires a proxy for connecting to the Internet (to upload support bundles to NetApp Support Site or establish a connection to Cloud Insights), you should configure a proxy server in Astra Control Center.



Astra Control Center does not validate the details you enter for your proxy server. Ensure that you enter the correct values.

Steps

1. Log in to Astra Control Center using an account with **admin/owner** privilege.
2. Select **Account > Connections**.
3. Select **Connect** from the drop-down list to add a proxy server.



HTTP PROXY

Configure Astra Control to send traffic through a proxy server.

Disconnected

Connect

4. Enter the proxy server name or IP address and the proxy port number.
5. If your proxy server requires authentication, select the check box, and enter the username and password.
6. Select **Connect**.

Result

If the proxy information you entered was saved, the **HTTP Proxy** section of the **Account > Connections** page indicates that it is connected, and displays the server name.



HTTP PROXY ?

Server: proxy.example.com:8888

Authentication: Enabled

Connected



Edit proxy server settings

You can edit the proxy server settings.

Steps

1. Log in to Astra Control Center using an account with **admin/owner** privilege.
2. Select **Account > Connections**.

3. Select **Edit** from the drop-down list to edit the connection.
4. Edit the server details and authentication information.
5. Select **Save**.

Disable proxy server connection

You can disable the proxy server connection. You will be warned before you disable that potential disruption to other connections might occur.

Steps

1. Log in to Astra Control Center using an account with **admin/owner** privilege.
2. Select **Account > Connections**.
3. Select **Disconnect** from the drop-down list to disable the connection.
4. In the dialog box that opens, confirm the operation.

Connect to Cloud Insights

To monitor and gain insight into your complete infrastructure, connect NetApp Cloud Insights with your Astra Control Center instance. Cloud Insights is included in your Astra Control Center license.

Cloud Insights should be accessible from the network that Astra Control Center uses, or indirectly via a proxy server.

When Astra Control Center is connected to Cloud Insights, an Acquisition Unit pod gets created. This pod collects data from the storage backends that are managed by Astra Control Center and pushes it to Cloud Insights. This pod requires 8 GB RAM and 2 CPU cores.



After you enable the Cloud Insights connection, you can view throughput information on the **Backends** page as well as connect to Cloud Insights from here after selecting a storage backend. You can also find the information on the **Dashboard** in the Cluster section, and also connect to Cloud Insights from there.

What you'll need

- An Astra Control Center account with **admin/owner** privileges.
- A valid Astra Control Center license.
- A proxy server if the network where you're running Astra Control Center requires a proxy for connecting to the Internet.



If you are new to Cloud Insights, familiarize yourself with the features and capabilities. See [Cloud Insights documentation](#).

Steps

1. Log in to Astra Control Center using an account with **admin/owner** privilege.
2. Select **Account > Connections**.
3. Select **Connect** where it shows **Disconnected** in the drop-down list to add the connection.



CLOUD INSIGHTS

Monitor and analyze your applications and the infrastructure they run on.

Disconnected

Connect

4. Enter the Cloud Insights API tokens and the tenant URL. The tenant URL has the following format, as an example:

```
https://<environment-name>.c01.cloudinsights.netapp.com/
```

You get the tenant URL when you get the Cloud Insights license. If you do not have the tenant URL, see the [Cloud Insights documentation](#).

- a. To get the [API token](#), log in to your Cloud Insights tenant URL.
- b. In Cloud Insights, generate both a **Read/Write** and a **Read only** API Access token by clicking **Admin** > **API Access**.

The screenshot shows the Cloud Insights Admin interface. The top navigation bar includes the Cloud Insights logo, a 'Tutorial 0% Complete' progress indicator, and a 'Getting Started' dropdown. The left sidebar contains navigation links: MONITOR & OPTIMIZE, HOME, DASHBOARDS, QUERIES, ALERTS, REPORTS, MANAGE, ADMIN (selected), CLOUD SECURE, and HELP. The main content area is titled 'API Access Tokens (4)' and features a '+ API Access Token' button and a 'Bulk Actions' dropdown. Below this is a table with the following data:

<input type="checkbox"/>	Name ↑	Description	Token	API Type	Permission
<input type="checkbox"/>	astra_...		...zBskB1	All Categories	Read/Write
<input type="checkbox"/>	astra_...		...xKOel...	All Categories	Read/Write
<input type="checkbox"/>	astra_...		...2_A6HP	All Categories	Read Only
<input type="checkbox"/>	astra_...		...8BTKYY	All Categories	Read/Write

- c. Copy the **Read only** key. You will need to paste it into the Astra Control Center window for enabling the Cloud Insights connection. For the Read API Access Token key permissions, select: Assets, Alerts, Acquisition Unit, and Data Collection.
- d. Copy the **Read/Write** key. You will need to paste it into the Astra Control Center **Connect Cloud Insights** window. For the Read/Write API Access Token key permissions, select: Assets, Data Ingestion, Log Ingestion, Acquisition Unit, and Data Collection.



We recommend that you generate a **Read only** key and a **Read/Write** key, and not use the same key for both purposes. By default, the token expiry period is set to one year. We recommend that you keep the default selection to give the token the maximum duration before it expires. If your token expires, the telemetry will stop.

e. Paste the keys that you copied from Cloud Insights into Astra Control Center.

5. Select **Connect**.



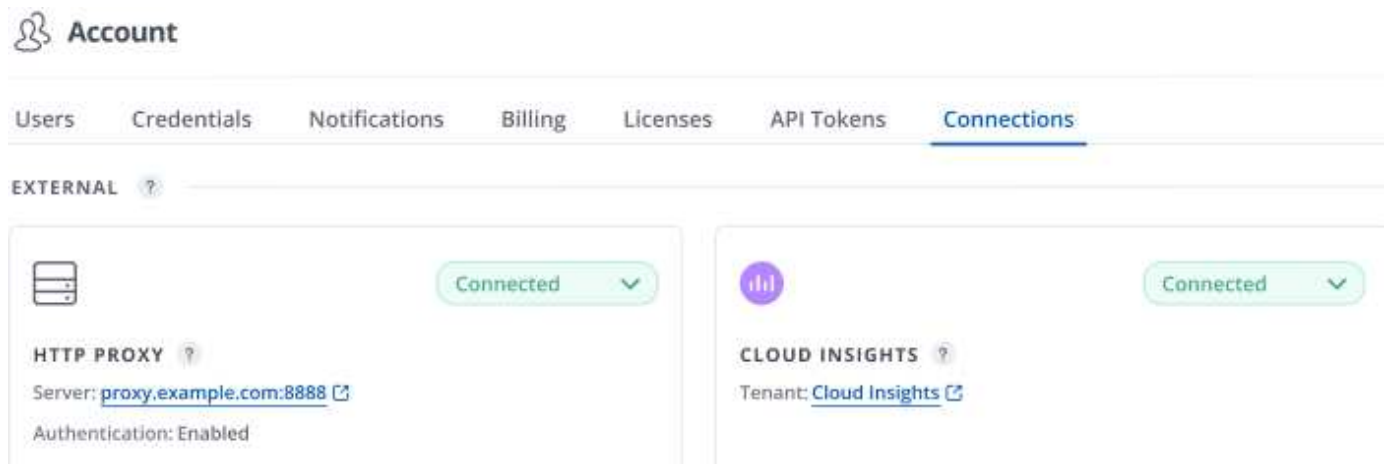
After you select **Connect**, the status of the connection changes to **Pending** in the **Cloud Insights** section of the **Account > Connections** page. It can a few minutes for the connection to be enabled and the status to change to **Connected**.



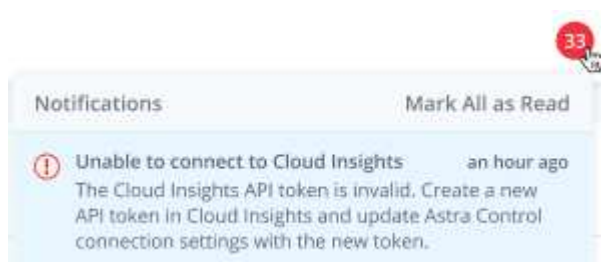
To go back and forth easily between the Astra Control Center and Cloud Insights UIs, ensure that you are logged into both.

View data in Cloud Insights

If the connection was successful, the **Cloud Insights** section of the **Account > Connections** page indicates that it is connected, and displays the tenant URL. You can visit Cloud Insights to see data being successfully received and displayed.

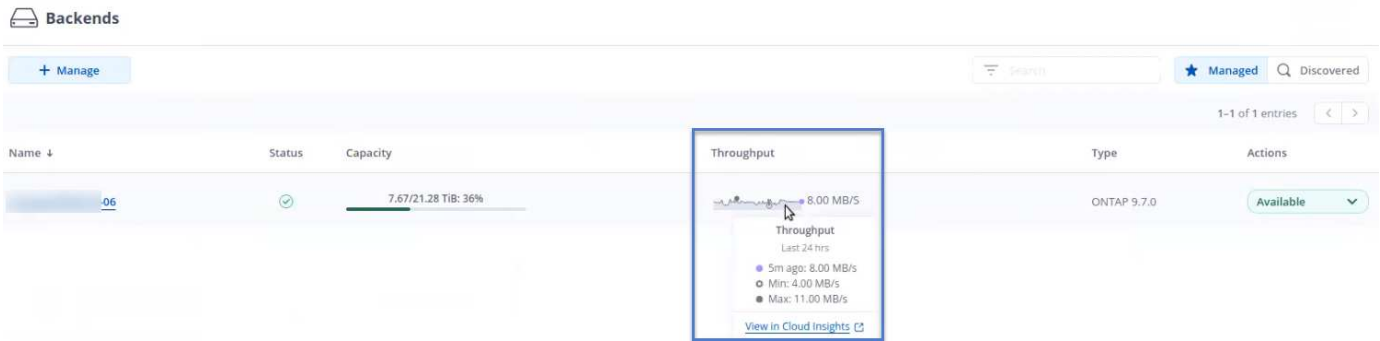


If the connection failed for some reason, the status shows **Failed**. You can find the reason for failure under **Notifications** at the top-right side of the UI.



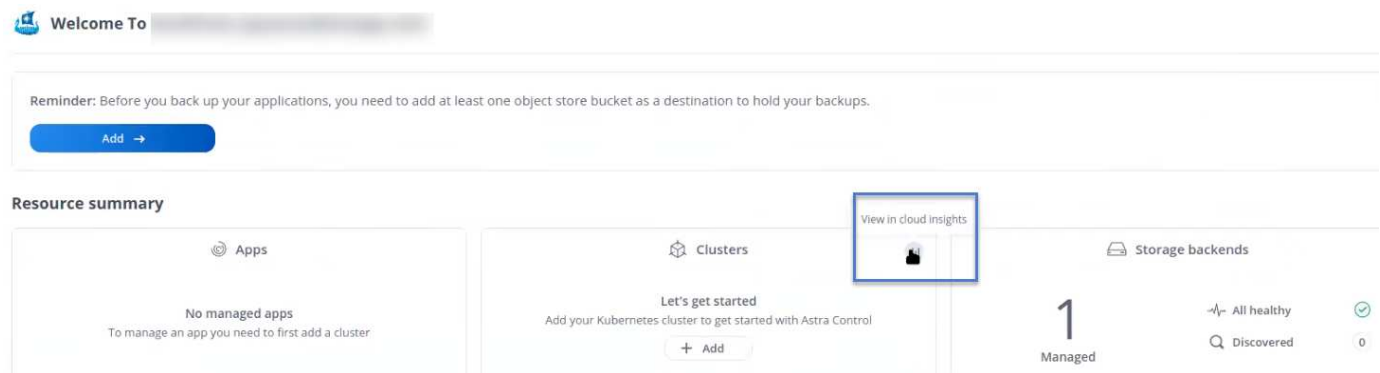
You can also find the same information under **Account > Notifications**.

From Astra Control Center, you can view throughput information on the **Backends** page as well as connect to Cloud Insights from here after selecting a storage backend.



To go directly to Cloud Insights, select the **Cloud Insights** icon next to the metrics image.

You can also find the information on the **Dashboard**.



After enabling the Cloud Insights connection, if you remove the backends that you added in Astra Control Center, the backends stop reporting to Cloud Insights.

Edit Cloud Insights connection

You can edit the Cloud Insights connection.



You can only edit the API keys. To change the Cloud Insights tenant URL, we recommended that you disconnect the Cloud Insights connection, and connect with the new URL.

Steps

1. Log in to Astra Control Center using an account with **admin/owner** privilege.
2. Select **Account > Connections**.
3. Select **Edit** from the drop-down list to edit the connection.
4. Edit the Cloud Insights connection settings.
5. Select **Save**.

Disable Cloud Insights connection

You can disable the Cloud Insights connection for a Kubernetes cluster managed by Astra Control Center. Disabling the Cloud Insights connection does not delete the telemetry data already uploaded to Cloud Insights.

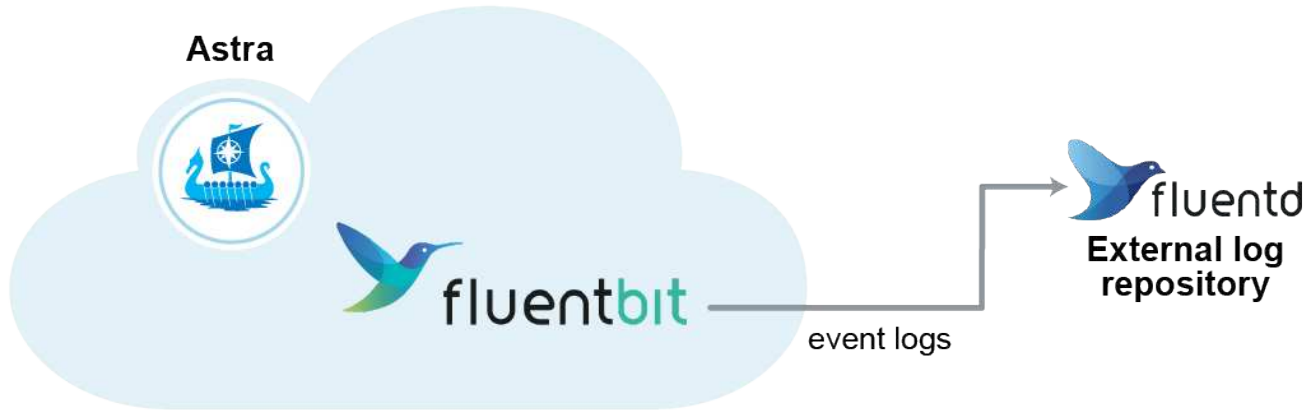
Steps

1. Log in to Astra Control Center using an account with **admin/owner** privilege.

2. Select **Account > Connections**.
3. Select **Disconnect** from the drop-down list to disable the connection.
4. In the dialog box that opens, confirm the operation.
After you confirm the operation, on the **Account > Connections** page, the Cloud Insights status changes to **Pending**. It take a few minutes for the status to change to **Disconnected**.

Connect to Fluentd

You can send logs (Kubernetes events) from Astra Control Center to your Fluentd endpoint. The Fluentd connection is disabled by default.



Only the event logs from managed clusters are forwarded to Fluentd.

What you'll need

- An Astra Control Center account with **admin/owner** privileges.
- Astra Control Center installed and running on a Kubernetes cluster.



Astra Control Center does not validate the details you enter for your Fluentd server. Ensure that you enter the correct values.

Steps

1. Log in to Astra Control Center using an account with **admin/owner** privilege.
2. Select **Account > Connections**.
3. Select **Connect** from the drop-down list where it shows **Disconnected** to add the connection.



FLUENTD

Connect Astra Control logs to Fluentd for use by your log analysis software.

4. Enter the host IP address, the port number, and shared key for your Fluentd server.

5. Select **Connect**.

Result

If the details you entered for your Fluentd server were saved, the **Fluentd** section of the **Account > Connections** page indicates that it is connected. Now you can visit the Fluentd server that you connected and view the event logs.

If the connection failed for some reason, the status shows **Failed**. You can find the reason for failure under **Notifications** at the top-right side of the UI.

You can also find the same information under **Account > Notifications**.



If you are having trouble with log collection, you should log in to your worker node and ensure that your logs are available in `/var/log/containers/`.

Edit the Fluentd connection

You can edit the Fluentd connection to your Astra Control Center instance.

Steps

1. Log in to Astra Control Center using an account with **admin/owner** privilege.
2. Select **Account > Connections**.
3. Select **Edit** from the drop-down list to edit the connection.
4. Change the Fluentd endpoint settings.
5. Select **Save**.

Disable the Fluentd connection

You can disable the Fluentd connection to your Astra Control Center instance.

Steps

1. Log in to Astra Control Center using an account with **admin/owner** privilege.
2. Select **Account > Connections**.
3. Select **Disconnect** from the drop-down list to disable the connection.
4. In the dialog box that opens, confirm the operation.

Unmanage apps and clusters

Remove any apps or clusters that you no longer want to manage from Astra Control Center.

Unmanage an app

Stop managing apps that you no longer want to back up, snapshot, or clone from Astra Control Center.

- Any existing backups and snapshots will be deleted.
- Applications and data remain available.

Steps

1. From the left navigation bar, select **Applications**.
2. Select the check box for the apps that you no longer want to manage.
3. From the **Action** menu, select **Unmanage**.
4. Type "unmanage" to confirm.
5. Confirm that you want to unmanage the apps and then select **Yes, unmanage Application**.

Result

Astra Control Center stops managing the app.

Unmanage a cluster

Unmanage the cluster that you no longer want to manage from Astra Control Center.

- This action stops your cluster from being managed by Astra Control Center. It doesn't make any changes to the cluster's configuration and it doesn't delete the cluster.
- Trident won't be uninstalled from the cluster. [Learn how to uninstall Trident](#).



Before you unmanage the cluster, you should unmanage the apps associated with the cluster.

Steps

1. From the left navigation bar, select **Clusters**.
2. Select the check box for the cluster that you no longer want to manage in Astra Control Center.
3. From the Options menu in the **Actions** column, select **Unmanage**.
4. Confirm that you want to unmanage the cluster and then select **Yes, unmanage cluster**.

Result

The status of the cluster changes to **Removing** and after that the cluster will be removed from the **Clusters** page, and it is no longer managed by Astra Control Center.



If Astra Control Center and Cloud Insights are not connected, unmanaging the cluster removes all the resources that were installed for sending telemetry data. **If Astra Control Center and Cloud Insights are connected**, unmanaging the cluster deletes only the `fluentbit` and `event-exporter` pods.

Upgrade Astra Control Center

To upgrade Astra Control Center, download the installation bundle from the NetApp Support Site and complete these instructions to upgrade the Astra Control Center components in your environment. You can use this procedure to upgrade Astra Control Center in internet-connected or air-gapped environments.

What you'll need

- [Before you begin upgrade, ensure your environment still meets the minimum requirements for Astra Control Center deployment](#).
- Ensure all cluster operators are in a healthy state and available.

OpenShift example:

```
oc get clusteroperators
```

- Ensure all API services are in a healthy state and available.

OpenShift example:

```
oc get apiservices
```

- Log out of your Astra Control Center.

About this task

The Astra Control Center upgrade process guides you through the following high-level steps:

- [Download the Astra Control Center bundle](#)
- [Unpack the bundle and change directory](#)
- [Add the images to your local registry](#)
- [Install the updated Astra Control Center operator](#)
- [Upgrade Astra Control Center](#)
- [Upgrade third-party services \(Optional\)](#)
- [Verify system status](#)
- [Set up ingress for load balancing](#)



Do not execute the following command during the entirety of the upgrade process to avoid deleting all Astra Control Center pods: `kubectl delete -f astra_control_center_operator_deploy.yaml`



Perform upgrades in a maintenance window when schedules, backups, and snapshots are not running.



Podman commands can be used in place of Docker commands if you are using Red Hat's Podman instead of Docker Engine.

Download the Astra Control Center bundle

1. Download the Astra Control Center upgrade bundle (`astra-control-center-[version].tar.gz`) from the [NetApp Support Site](#).
2. (Optional) Use the following command to verify the signature of the bundle:

```
openssl dgst -sha256 -verify astra-control-center[version].pub  
-signature <astra-control-center[version].sig astra-control-  
center[version].tar.gz
```

Unpack the bundle and change directory

1. Extract the images:

```
tar -vxzf astra-control-center-[version].tar.gz
```

2. Change to the Astra directory.

```
cd astra-control-center-[version]
```

Add the images to your local registry

1. Add the files in the Astra Control Center image directory to your local registry.



See a sample script for the automatic loading of images below.

- a. Log in to your Docker registry:

```
docker login [your_registry_path]
```

- b. Load the images into Docker.
- c. Tag the images.
- d. Push the images to your local registry.

```
export REGISTRY=[your_registry_path]
for astraImageFile in $(ls images/*.tar)
  # Load to local cache. And store the name of the loaded image
  trimming the 'Loaded images: '
  do astraImage=$(docker load --input ${astraImageFile} | sed
  's/Loaded image: //' )
  astraImage=$(echo ${astraImage} | sed 's!localhost/!!')
  # Tag with local image repo.
  docker tag ${astraImage} ${REGISTRY}/${astraImage}
  # Push to the local repo.
  docker push ${REGISTRY}/${astraImage}
done
```

Install the updated Astra Control Center operator

1. Edit the Astra Control Center operator deployment yaml
(astra_control_center_operator_deploy.yaml) to refer to your local registry and secret.

```
vim astra_control_center_operator_deploy.yaml
```

- a. If you use a registry that requires authentication, replace the default line of `imagePullSecrets: []` with the following:

```
imagePullSecrets:  
- name: <name_of_secret_with_creds_to_local_registry>
```

- b. Change `[your_registry_path]` for the `kube-rbac-proxy` image to the registry path where you pushed the images in a [previous step](#).
- c. Change `[your_registry_path]` for the `acc-operator-controller-manager` image to the registry path where you pushed the images in a [previous step](#).
- d. Add the following values to the `env` section:

```
- name: ACCOP_HELM_UPGRADE_TIMEOUT  
  value: 300m
```

```

apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    control-plane: controller-manager
  name: acc-operator-controller-manager
  namespace: netapp-acc-operator
spec:
  replicas: 1
  selector:
    matchLabels:
      control-plane: controller-manager
  template:
    metadata:
      labels:
        control-plane: controller-manager
    spec:
      containers:
        - args:
            - --secure-listen-address=0.0.0.0:8443
            - --upstream=http://127.0.0.1:8080/
            - --logtostderr=true
            - --v=10
          image: [your_registry_path]/kube-rbac-proxy:v4.8.0
          name: kube-rbac-proxy
          ports:
            - containerPort: 8443
              name: https
        - args:
            - --health-probe-bind-address=:8081
            - --metrics-bind-address=127.0.0.1:8080
            - --leader-elect
          command:
            - /manager
          env:
            - name: ACCOP_LOG_LEVEL
              value: "2"
            - name: ACCOP_HELM_UPGRADE_TIMEOUT
              value: 300m
          image: [your_registry_path]/acc-operator:[version x.y.z]
          imagePullPolicy: IfNotPresent
      imagePullSecrets: []

```

2. Install the updated Astra Control Center operator:

```
kubectl apply -f astra_control_center_operator_deploy.yaml
```

Sample response:

```
namespace/netapp-acc-operator unchanged
customresourcedefinition.apiextensions.k8s.io/astracontrolcenters.astra.
netapp.io configured
role.rbac.authorization.k8s.io/acc-operator-leader-election-role
unchanged
clusterrole.rbac.authorization.k8s.io/acc-operator-manager-role
configured
clusterrole.rbac.authorization.k8s.io/acc-operator-metrics-reader
unchanged
clusterrole.rbac.authorization.k8s.io/acc-operator-proxy-role unchanged
rolebinding.rbac.authorization.k8s.io/acc-operator-leader-election-
rolebinding unchanged
clusterrolebinding.rbac.authorization.k8s.io/acc-operator-manager-
rolebinding configured
clusterrolebinding.rbac.authorization.k8s.io/acc-operator-proxy-
rolebinding unchanged
configmap/acc-operator-manager-config unchanged
service/acc-operator-controller-manager-metrics-service unchanged
deployment.apps/acc-operator-controller-manager configured
```

Upgrade Astra Control Center

1. Edit the Astra Control Center custom resource (CR) (`astra_control_center_min.yaml`) and change the Astra version (`astraVersion` inside of `Spec`) number to the latest:

```
kubectl edit acc -n [netapp-acc or custom namespace]
```



Your registry path must match the registry path where you pushed the images in a [previous step](#).

2. Add the following lines within `additionalValues` inside of `Spec` in the Astra Control Center CR:

```
additionalValues:
  nautilus:
    startupProbe:
      periodSeconds: 30
      failureThreshold: 600
```

3. Do one of the following:

- a. If you don't have your own IngressController or ingress and have been using the Astra Control Center with its Traefik gateway as a LoadBalancer type service and would like to continue with that setup, specify another field `ingressType` (if not already present) and set it to `AccTraefik`.

```
ingressType: AccTraefik
```

- b. If you want to switch to the default Astra Control Center generic ingress deployment, provide your own IngressController/Ingress setup (with TLS termination, etc.), open up a route to Astra Control Center, and set `ingressType` to `Generic`.

```
ingressType: Generic
```



If you omit the field, the process becomes the generic deployment. If you don't want the generic deployment, be sure to add the field.

4. (Optional) Verify that the pods terminate and become available again:

```
watch kubectl get po -n [netapp-acc or custom namespace]
```

5. Wait for the Astra status conditions to indicate that the upgrade is complete and ready:

```
kubectl get -o yaml -n [netapp-acc or custom namespace]  
astracontrolcenters.astra.netapp.io astra
```

Response:

```
conditions:  
  - lastTransitionTime: "2021-10-25T18:49:26Z"  
    message: Astra is deployed  
    reason: Complete  
    status: "True"  
    type: Ready  
  - lastTransitionTime: "2021-10-25T18:49:26Z"  
    message: Upgrading succeeded.  
    reason: Complete  
    status: "False"  
    type: Upgrading
```

6. Log back in and verify that all managed clusters and apps are still present and protected.
7. If the operator did not update the Cert-manager, upgrade third-party services, next.

Upgrade third-party services (Optional)

The third-party services Traefik and Cert-manager are not upgraded during earlier upgrade steps. You can optionally upgrade them using the procedure described here or retain existing service versions if your system requires it.

- **Traefik:** By default, Astra Control Center manages the lifecycle of the Traefik deployment. Setting `externalTraefik` to `false` (default) indicates that no external Traefik exists in the system and Traefik is being installed and managed by Astra Control Center. In this case, `externalTraefik` is set to `false`.

On the other hand, if you have your own Traefik deployment, set `externalTraefik` to `true`. In this case, you maintain the deployment and Astra Control Center will not upgrade the CRDs, unless `shouldUpgrade` is set to `true`.

- **Cert-manager:** By default, Astra Control Center installs the cert-manager (and CRDs) unless you set `externalCertManager` to `true`. Set `shouldUpgrade` to `true` to have Astra Control Center upgrade the CRDs.

Traefik is upgraded if any of the following conditions are met:

- `externalTraefik: false`
OR
- `externalTraefik: true` AND `shouldUpgrade: true`.

Steps

1. Edit the `acc` CR:

```
kubectl edit acc -n [netapp-acc or custom namespace]
```

2. Change the `externalTraefik` field and the `shouldUpgrade` field to either `true` or `false` as needed.

```
crds:
  externalTraefik: false
  externalCertManager: false
  shouldUpgrade: false
```

Verify system status

1. Log in to Astra Control Center.
2. Verify that all your managed clusters and apps are still present and protected.

Set up ingress for load balancing

You can set up a Kubernetes ingress object that manages external access to the services, such as load balancing in a cluster.

- Default upgrade uses the generic ingress deployment. In this case, you will also need to set up an ingress

controller or ingress resource.

- If you don't want an ingress controller and want to retain what you already have, set `ingressType` to `AccTraefik`.



For additional details about the service type of "LoadBalancer" and ingress, see [Requirements](#).

The steps differ depending on the type of ingress controller you use:

- Nginx ingress controller
- OpenShift ingress controller

What you'll need

- In the CR spec,
 - If `crd.externalTraefik` is present, it should be set to `false` OR
 - If `crd.externalTraefik` is `true`, `crd.shouldUpgrade` should also be `true`.
- The required [ingress controller](#) should already be deployed.
- The [ingress class](#) corresponding to the ingress controller should already be created.
- You are using Kubernetes versions between and including v1.19 and v1.21.

Steps for Nginx ingress controller

1. Use the existing secret `secure-testing-cert` or create a secret of type `kubernetes.io/tls` for a TLS private key and certificate in `netapp-acc` (or custom-named) namespace as described in [TLS secrets](#).
2. Deploy an ingress resource in `netapp-acc` (or custom-named) namespace for either a deprecated or a new schema:
 - a. For a deprecated schema, follow this sample:

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: ingress-acc
  namespace: [netapp-acc or custom namespace]
  annotations:
    kubernetes.io/ingress.class: nginx
spec:
  tls:
    - hosts:
        - <ACC address>
      secretName: [tls secret name]
  rules:
    - host: [ACC address]
      http:
        paths:
          - backend:
              serviceName: traefik
              servicePort: 80
            pathType: ImplementationSpecific
```

b. For a new schema, follow this example:

```

apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: netapp-acc-ingress
  namespace: [netapp-acc or custom namespace]
spec:
  ingressClassName: [class name for nginx controller]
  tls:
  - hosts:
    - <ACC address>
    secretName: [tls secret name]
  rules:
  - host: <ACC address>
    http:
      paths:
      - path:
        backend:
          service:
            name: traefik
            port:
              number: 80
        pathType: ImplementationSpecific

```

Steps for OpenShift ingress controller

1. Procure your certificate and get the key, certificate, and CA files ready for use by the OpenShift route.
2. Create the OpenShift route:

```

oc create route edge --service=traefik
--port=web -n [netapp-acc or custom namespace]
--insecure-policy=Redirect --hostname=<ACC address>
--cert=cert.pem --key=key.pem

```

Verify ingress set up

You can verify the ingress set up before you continue.

1. Ensure that Traefik has changed to `clusterIP` from `Loadbalancer`:

```

kubectl get service traefik -n [netapp-acc or custom namespace]

```

2. Verify routes in Traefik:

```
Kubectl get ingressroute ingressroutetls -n [netapp-acc or custom namespace]
-o yaml | grep "Host("
```



The result should be empty.

Uninstall Astra Control Center

You might need to remove Astra Control Center components if you are upgrading from a trial to a full version of the product. To remove Astra Control Center and the Astra Control Center Operator, run the commands described in this procedure in sequence.

If you have any issues with the uninstall, see [Troubleshooting uninstall issues](#).

What you'll need

- Use Astra Control Center UI to unmanage all [clusters](#).

Steps

1. Delete Astra Control Center. The following sample command is based upon a default installation. Modify the command if you made custom configurations.

```
kubectl delete -f astra_control_center_min.yaml -n netapp-acc
```

Result:

```
astracontrolcenter.astra.netapp.io "astra" deleted
```

2. Use the following command to delete the `netapp-acc` namespace:

```
kubectl delete ns netapp-acc
```

Result:

```
namespace "netapp-acc" deleted
```

3. Use the following command to delete Astra Control Center operator system components:

```
kubectl delete -f astra_control_center_operator_deploy.yaml
```

Result:

```
namespace "netapp-acc-operator" deleted
customresourcedefinition.apiextensions.k8s.io
"astracontrolcenters.astra.netapp.io" deleted
role.rbac.authorization.k8s.io "acc-operator-leader-election-role"
deleted
clusterrole.rbac.authorization.k8s.io "acc-operator-manager-role"
deleted
clusterrole.rbac.authorization.k8s.io "acc-operator-metrics-reader"
deleted
clusterrole.rbac.authorization.k8s.io "acc-operator-proxy-role" deleted
rolebinding.rbac.authorization.k8s.io "acc-operator-leader-election-
rolebinding" deleted
clusterrolebinding.rbac.authorization.k8s.io "acc-operator-manager-
rolebinding" deleted
clusterrolebinding.rbac.authorization.k8s.io "acc-operator-proxy-
rolebinding" deleted
configmap "acc-operator-manager-config" deleted
service "acc-operator-controller-manager-metrics-service" deleted
deployment.apps "acc-operator-controller-manager" deleted
```

Troubleshooting uninstall issues

Use the following workarounds to address any problems you have with uninstalling Astra Control Center.

Uninstall of Astra Control Center fails to clean up the monitoring-operator pod on the managed cluster

If you did not unmanage your clusters before you uninstalled Astra Control Center, you can manually delete the pods in the netapp-monitoring namespace and the namespace with the following commands:

Steps

1. Delete acc-monitoring agent:

```
kubectl delete agents acc-monitoring -n netapp-monitoring
```

Result:

```
agent.monitoring.netapp.com "acc-monitoring" deleted
```

2. Delete the namespace:

```
kubectl delete ns netapp-monitoring
```

Result:

```
namespace "netapp-monitoring" deleted
```

3. Confirm resources removed:

```
kubectl get pods -n netapp-monitoring
```

Result:

```
No resources found in netapp-monitoring namespace.
```

4. Confirm monitoring agent removed:

```
kubectl get crd|grep agent
```

Sample result:

```
agents.monitoring.netapp.com                2021-07-21T06:08:13Z
```

5. Delete custom resource definition (CRD) information:

```
kubectl delete crds agents.monitoring.netapp.com
```

Result:

```
customresourcedefinition.apiextensions.k8s.io  
"agents.monitoring.netapp.com" deleted
```

Uninstall of Astra Control Center fails to clean up Traefik CRDs

You can manually delete the Traefik CRDs. CRDs are global resources, and deleting them might impact other applications on the cluster.

Steps

1. List Traefik CRDs installed on the cluster:

```
kubectl get crds |grep -E 'traefik'
```

Response

<code>ingressroutes.traefik.containo.us</code>	<code>2021-06-23T23:29:11Z</code>
<code>ingressroutetcps.traefik.containo.us</code>	<code>2021-06-23T23:29:11Z</code>
<code>ingressrouteudps.traefik.containo.us</code>	<code>2021-06-23T23:29:12Z</code>
<code>middlewares.traefik.containo.us</code>	<code>2021-06-23T23:29:12Z</code>
<code>middlewareetcps.traefik.containo.us</code>	<code>2021-06-23T23:29:12Z</code>
<code>serverstransports.traefik.containo.us</code>	<code>2021-06-23T23:29:13Z</code>
<code>tlsoptions.traefik.containo.us</code>	<code>2021-06-23T23:29:13Z</code>
<code>tlsstores.traefik.containo.us</code>	<code>2021-06-23T23:29:14Z</code>
<code>traefikservices.traefik.containo.us</code>	<code>2021-06-23T23:29:15Z</code>

2. Delete the CRDs:

```
kubectl delete crd ingressroutes.traefik.containo.us
ingressroutetcps.traefik.containo.us
ingressrouteudps.traefik.containo.us middlewares.traefik.containo.us
serverstransports.traefik.containo.us tlsoptions.traefik.containo.us
tlsstores.traefik.containo.us traefikservices.traefik.containo.us
middlewareetcps.traefik.containo.us
```

Find more information

- [Known issues for uninstall](#)

Copyright information

Copyright © 2024 NetApp, Inc. All Rights Reserved. Printed in the U.S. No part of this document covered by copyright may be reproduced in any form or by any means—graphic, electronic, or mechanical, including photocopying, recording, taping, or storage in an electronic retrieval system—without prior written permission of the copyright owner.

Software derived from copyrighted NetApp material is subject to the following license and disclaimer:

THIS SOFTWARE IS PROVIDED BY NETAPP “AS IS” AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WHICH ARE HEREBY DISCLAIMED. IN NO EVENT SHALL NETAPP BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

NetApp reserves the right to change any products described herein at any time, and without notice. NetApp assumes no responsibility or liability arising from the use of products described herein, except as expressly agreed to in writing by NetApp. The use or purchase of this product does not convey a license under any patent rights, trademark rights, or any other intellectual property rights of NetApp.

The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

LIMITED RIGHTS LEGEND: Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (b)(3) of the Rights in Technical Data -Noncommercial Items at DFARS 252.227-7013 (FEB 2014) and FAR 52.227-19 (DEC 2007).

Data contained herein pertains to a commercial product and/or commercial service (as defined in FAR 2.101) and is proprietary to NetApp, Inc. All NetApp technical data and computer software provided under this Agreement is commercial in nature and developed solely at private expense. The U.S. Government has a non-exclusive, non-transferrable, nonsublicensable, worldwide, limited irrevocable license to use the Data only in connection with and in support of the U.S. Government contract under which the Data was delivered. Except as provided herein, the Data may not be used, disclosed, reproduced, modified, performed, or displayed without the prior written approval of NetApp, Inc. United States Government license rights for the Department of Defense are limited to those rights identified in DFARS clause 252.227-7015(b) (FEB 2014).

Trademark information

NETAPP, the NETAPP logo, and the marks listed at <http://www.netapp.com/TM> are trademarks of NetApp, Inc. Other company and product names may be trademarks of their respective owners.