Use Astra Control Center

Astra Control Center

NetApp
April 25, 2024

Table of Contents

Use Astra Control Center

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 Applications page in Astra Control to define the apps and their resources.

You can define and manage apps that include storage resources with running pods, or apps that include storage resources without any running pods. Apps that have no running pods are known as data-only applications.

Application management requirements

Astra Control has the following application management requirements:

- **Licensing**: To manage applications using Astra Control Center, you need either the embedded Astra Control Center evaluation license or a full license.
- **Namespaces**: Apps can be defined within one or more specified namespaces on a single cluster using Astra Control. An app can contain resources spanning multiple namespaces within the same cluster. Astra Control does not support the ability for apps to be defined across multiple clusters.
- **Storage class**: If you install an application with a storage class explicitly set and you need to clone the app, the target cluster for the clone operation must have the originally specified storage class. Cloning an application with an explicitly set storage class to a cluster that does not have the same storage class will fail.
- **Kubernetes resources**: Applications that use Kubernetes resources not collected by Astra Control might not have full app data management capabilities. Astra Control collects the following Kubernetes resources:

`ClusterRole`	`ClusterRoleBinding`	`ConfigMap`
`CronJob`	`CustomResourceDefinition`	`CustomResource`
`DaemonSet`	`DeploymentConfig`	`HorizontalPodAutoscaler`
`Ingress`	`MutatingWebhook`	`NetworkPolicy`
`PersistentVolumeClaim`	`Pod`	`PodDisruptionBudget`
`PodTemplate`	`ReplicaSet`	`Role`
`RoleBinding`	`Route`	`Secret`
`Service`	`ServiceAccount`	`StatefulSet`
`ValidatingWebhook`		

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) is fully supported. Managing apps installed with Helm 2 is not supported.
- **Operator-deployed apps**: Astra Control supports apps installed with namespace-scoped operators that are, in general, designed with a "pass-by-value" rather than "pass-by-reference" architecture. 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.

  The following are some operator apps that follow these patterns:

  - Apache K8ssandra

    For K8ssandra, in-place restore operations are supported. A restore operation to a new namespace or cluster requires that the original instance of the application to be taken down. This is to ensure that the peer group information carried over does not lead to cross-instance communication. Cloning of the app is not supported.

  - Jenkins CI
  - Percona XtraDB Cluster

  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.

Install apps on your cluster

After 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 one or more namespaces can be managed.

Define apps

After Astra Control discovers namespaces on your clusters, you can define applications that you want to manage. You can choose to manage an app spanning one or more namespaces or manage an entire namespace as a single application. It all comes down to the level of granularity that you need for data protection operations.

Although Astra Control enables you to separately manage both levels of the hierarchy (the namespace and the apps in that namespace or spanning namespaces), 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.

As an 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 as a single-namespace app.

Before you begin

- A Kubernetes cluster added to Astra Control.
- One or more installed apps on the cluster. Read more about supported app installation methods.
- Existing namespaces on the Kubernetes cluster that you added to Astra Control.
- (Optional) A Kubernetes label on any supported Kubernetes resources.

> 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.

About this task

- Before you begin, you should also understand managing standard and system namespaces.
- If you plan to use multiple namespaces with your apps in Astra Control, modify user roles with namespace constraints after you upgrade to an Astra Control Center version with multiple namespace support.
- For instructions on how to manage apps using the Astra Control API, see the Astra Automation and API information.

App management options

- Define resources to manage as an app
- Define a namespace to manage as an app
- (Tech preview) Define an application using a Kubernetes custom resource

Define resources to manage as an app

You can specify the Kubernetes resources that make up an app that you want to manage with Astra Control. Defining an app enables you to group elements of your Kubernetes cluster into a single app. This collection of Kubernetes resources is organized by namespace and label selector criteria.

Defining an app gives you more granular control over what to include in an Astra Control operation, including clone, snapshot, and backups.

> When defining apps, ensure that you do not include a Kubernetes resource in multiple apps with protection policies. Overlapping protection policies on Kubernetes resources can cause data conflicts. Read more in an example.

Expand for more about adding cluster-scoped resources to your app namespaces.

You can import cluster resources that are associated with the namespace resources in addition to those Astra Control included automatically. You can add a rule that will include resources of a specific group, kind, version and optionally, label. You might want to do this if there are resources that Astra Control does not include automatically.

You cannot exclude any of the cluster-scoped resources that are automatically included by Astra Control.

You can add the following `apiVersions` (which are the groups combined with the API version):

Resource kind	apiVersions (group + version)
`ClusterRole`	rbac.authorization.k8s.io/v1
`ClusterRoleBinding`	rbac.authorization.k8s.io/v1
`CustomResource`	apiextensions.k8s.io/v1, apiextensions.k8s.io/v1beta1
`CustomResourceDefinition`	apiextensions.k8s.io/v1, apiextensions.k8s.io/v1beta1
`MutatingWebhookConfiguration`	admissionregistration.k8s.io/v1
`ValidatingWebhookConfiguration`	admissionregistration.k8s.io/v1

Steps

1. From the Applications page, select **Define**.
2. In the **Define application** window, enter the app name.
3. Choose the cluster on which your application is running in the **Cluster** drop-down list.
4. Choose a namespace for your application from the **Namespace** drop-down list.

   Apps can be defined within one or more specified namespaces on a single cluster using Astra Control. An app can contain resources spanning multiple namespaces within the same cluster. Astra Control does not support the ability for apps to be defined across multiple clusters.

5. (Optional) Enter a label for the Kubernetes resources in each namespace. You can specify a single label or label selector criteria (query).

   To learn more about Kubernetes labels, see the official Kubernetes documentation.

6. (Optional) Add additional namespaces for the app by selecting **Add namespace** and choosing the namespace from the drop-down list.
7. (Optional) Enter single label or label selector criteria for any additional namespaces you add.
8. (Optional) To include cluster-scoped resources in addition to those that Astra Control automatically includes, check **Include additional cluster-scoped resources** and complete the following:
   a. Select **Add include rule**.
   b. **Group**: From the drop-down list, select the API group of resources.

c. **Kind**: From the drop-down list, select the name of the object schema.

d. **Version**: Enter the API version.

e. **Label selector**: Optionally, include a label to add to the rule. This label is used to retrieve only those resources matching this label. If you don't provide a label, Astra Control collects all instances of the resource kind specified for that cluster.

f. Review the rule that is created based on your entries.

g. Select **Add**.

> You can create as many cluster-scoped resource rules as you want. The rules appear in the Define application Summary.

9. Select **Define**.
10. After you select **Define**, repeat the process for other apps, as needed.

After you finish defining an app, the app appears in `Healthy` state in the list of apps on the Applications page. You are now able to clone it and create backups and snapshots.

> 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.

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

To see the resources added to this app, select the **Resources** tab. Select the number after the resource name in the Resource column or enter the resource name in the Search to see the additional cluster-scoped resources included.

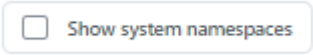
Define a namespace to manage as an app

You can add all Kubernetes resources in a namespace to Astra Control management by defining the resources of that namespace as an application. This method is preferable to defining apps individually if you intend to manage and protect all resources in a particular namespace in a similar way and at common intervals.

Steps

1. From the Clusters page, select a cluster.
2. Select the **Namespaces** tab.
3. Select the Actions menu for the namespace that contains the app resources you want to manage and select **Define as application**.

> If you want to define multiple applications, select from the namespaces list and select the **Actions** button in the upper-left corner and select **Define as application**. This will define multiple individual applications in their individual namespaces. For multi-namespace applications, see Define resources to manage as an app.

> Select the **Show system namespaces** checkbox to reveal system namespaces that are usually not used in app management by default. 

Read more.

After the process completes, the applications that are associated with the namespace appear in the

`Associated applications` column.

[Tech preview] Define an application using a Kubernetes custom resource

You can specify the Kubernetes resources that you want to manage with Astra Control by defining them as an application using a custom resource (CR). You can add cluster-scoped resources if you want to manage those resources individually or all Kubernetes resources in a namespace if, for example, you intend to manage and protect all resources in a particular namespace in a similar way and at common intervals.

Steps

1. Create the custom resource (CR) file and name it (for example, `astra_mysql_app.yaml`).
2. Name the application in `metadata.name`.
3. Define application resources to be managed:

spec.includedClusterScopedResources

Include cluster-scoped resource types in addition to those that Astra Control automatically includes:

- **spec.includedClusterScopedResources**: *(Optional)* A list of cluster-scoped resource types to be included.
  - **groupVersionKind**: *(Optional)* Unambiguously identifies a kind.
    - **group**: *(Required if groupVersionKind is used)* API group of the resource to include.
    - **version**: *(Required if groupVersionKind is used)* API version of the resource to include.
    - **kind**: *(Required if groupVersionKind is used)* Kind of the resource to include.
  - **labelSelector**: *(Optional)* A label query for a set of resources. It's used to retrieve only those resources matching the label. If you don't provide a label, Astra Control collects all instances of the resource kind specified for that cluster. The result of matchLabels and matchExpressions are ANDed.
    - **matchLabels**: *(Optional)* A map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions that has a key field of "key", operator as "In", and values array containing only "value". The requirements are ANDed.
    - **matchExpressions**: *(Optional)* A list of label selector requirements. The requirements are ANDed.
      - **key**: *(Required if matchExpressions is used)* The label key associated with the label selector.
      - **operator**: *(Required if matchExpressions is used)* Represents a key's relationship to a set of values. Valid operators are `In`, `NotIn`, `Exists` and `DoesNotExist`.
      - **values**: *(Required if matchExpressions is used)_An array of string values. If the operator is `In` or `NotIn`, the values array must _not* be empty. If the operator is `Exists` or `DoesNotExist`, the values array must be empty.

spec.includedNamespaces

Include namespaces and resources within those resources in the application:

- **spec.includedNamespaces**: _(Required)_Defines the namespace and optional filters for resource selection.
  - **namespace**: *(Required)* The namespace that contains the app resources you want to manage with Astra Control.
  - **labelSelector**: *(Optional)* A label query for a set of resources. It's used to retrieve only those resources matching the label. If you don't provide a label, Astra Control collects all instances of the resource kind specified for that cluster. The result of matchLabels and matchExpressions are ANDed.
    - **matchLabels**: *(Optional)* A map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of matchExpressions that has a key field of "key", operator as "In", and values array containing only "value". The requirements are ANDed.
    - **matchExpressions**: *(Optional)* A list of label selector requirements. `key` and `operator` are required. The requirements are ANDed.
      - **key**: *(Required if matchExpressions is used)* The label key associated with the label selector.

- **operator**: *(Required if matchExpressions is used)* Represents a key's relationship to a set of values. Valid operators are `In`, `NotIn`, `Exists` and `DoesNotExist`.
- **values**: *(Required if matchExpressions is used)* An array of string values. If the operator is `In` or `NotIn`, the values array must *not* be empty. If the operator is `Exists` or `DoesNotExist`, the values array must be empty.

Example YAML:

```yaml
apiVersion: astra.netapp.io/v1
kind: Application
metadata:
  name: astra_mysql_app
spec:
  includedNamespaces:
    - namespace: astra_mysql_app
      labelSelector:
        matchLabels:
          app: nginx
          env: production
        matchExpressions:
          - key: tier
            operator: In
            values:
              - frontend
              - backend
```

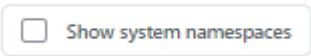
4. After you populate the `astra_mysql_app.yaml` file with the correct values, apply the CR:

```
kubectl apply -f astra_mysql_app.yaml -n astra-connector
```

What about system namespaces?

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

You can display system namespaces from the Namespaces tab for a selected cluster by selecting the **Show system namespaces** check box.



Astra Control Center is not shown by default as an application that you can manage, but you can back up and restore an Astra Control Center instance using another Astra Control Center instance.

Example: Separate Protection Policy for different releases

In this example, the devops team is managing a "canary" release deployment. The team's 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 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.

Find more information

- Use the Astra Control API
- Unmanage an app

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.

Additionally, you can replicate applications to a remote cluster in preparation for disaster recovery.

App protection workflow

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

[One] Protect 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] Adjust the protection policies

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

[Four] Replicate apps to a remote cluster

Replicate applications to a remote cluster by using NetApp SnapMirror technology. Astra Control replicates Snapshots to a remote cluster, providing asynchronous, disaster recovery capability.

[Five] In case of a disaster, restore your apps with the latest backup or replication to remote system

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). Or, you can use the replication to a remote system.

Protect apps with snapshots and backups

Protect all apps by taking snapshots and backups using an automated protection policy or on an ad-hoc basis. You can use the Astra Control Center UI or the Astra Control API to protect apps.

About this task

- **Helm deployed 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.
- **(OpenShift clusters only) Add policies**: 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
  ```

You can do the following tasks related to protecting your app data:

- Configure a protection policy
- Create a snapshot
- Create a backup
- Enable backup and restore for ontap-nas-economy operations
- Create an immutable backup
- View snapshots and backups
- Delete snapshots
- Cancel backups
- Delete backups

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. You can define a protection policy using either the Astra Control web UI or a custom resource (CR) file.

If you need backups or snapshots to run more frequently than once per hour, you can use the Astra Control REST API to create snapshots and backups.

If you are defining a protection policy that creates immutable backups to write once read many (WORM) buckets, ensure that the retention time for the backups is not shorter than the retention period configured for the bucket.

Offset backup and replication schedules to avoid schedule overlaps. For example, perform backups at the top of the hour every hour and schedule replication to start with a 5-minute offset and a 10-minute interval.

Configure a protection policy using the Web UI

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.

   When you set a retention level for backups, you can choose the bucket where you'd like to store the backups.

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

5. [**Tech preview**] Choose a destination bucket for the backups or snapshots from the list of storage buckets.
6. Select **Review**.
7. Select **Set Protection Policy.**

[Tech preview] Configure a protection policy using a CR

Steps

1. Create the custom resource (CR) file and name it `astra-control-schedule-cr.yaml`. Update the values in brackets <> to match your Astra Control environment, cluster configuration, and data protection needs:
   - <CR_NAME>: The name of this custom resource; choose a unique and sensible name for your environment.
   - <APPLICATION_NAME>: The Kubernetes name of the application to back up.
   - <APPVAULT_NAME>: The name of the AppVault where the backup contents should be stored.
   - <BACKUPS_RETAINED>: The number of backups to retain. Zero indicates that no backups should be created.
   - <SNAPSHOTS_RETAINED>: The number of snapshots to retain. Zero indicates that no snapshots should be created.
   - <GRANULARITY>: The frequency at which the schedule should run. Possible values, along with required associated fields:
     - `hourly` (requires that you specify `spec.minute`)
     - `daily` (requires that you specify `spec.minute` and `spec.hour`)
     - `weekly` (requires that you specify `spec.minute`, `spec.hour`, and `spec.dayOfWeek`)
     - `monthly` (requires that you specify `spec.minute`, `spec.hour`, and `spec.dayOfMonth`)
   - <DAY_OF_MONTH>: *(Optional)* The day of the month (1 - 31) that the schedule should run. This field is required if the granularity is set to `monthly`.

- <DAY_OF_WEEK>: *(Optional)* The day of the week (0 - 7) that the schedule should run. Values of 0 or 7 indicate Sunday. This field is required if the granularity is set to `weekly`.
- <HOUR_OF_DAY>: *(Optional)* The hour of the day (0 - 23) that the schedule should run. This field is required if the granularity is set to `daily`, `weekly`, or `monthly`.
- <MINUTE_OF_HOUR>: *(Optional)* The minute of the hour (0 - 59) that the schedule should run. This field is required if the granularity is set to `hourly`, `daily`, `weekly`, or `monthly`.

```
apiVersion: astra.netapp.io/v1
kind: Schedule
metadata:
  namespace: astra-connector
  name: <CR_NAME>
spec:
  applicationRef: <APPLICATION_NAME>
  appVaultRef: <APPVAULT_NAME>
  backupRetention: "<BACKUPS_RETAINED>"
  snapshotRetention: "<SNAPSHOTS_RETAINED>"
  granularity: <GRANULARITY>
  dayOfMonth: "<DAY_OF_MONTH>"
  dayOfWeek: "<DAY_OF_WEEK>"
  hour: "<HOUR_OF_DAY>"
  minute: "<MINUTE_OF_HOUR>"
```

2. After you populate the `astra-control-schedule-cr.yaml` file with the correct values, apply the CR:

```
kubectl apply -f astra-control-schedule-cr.yaml
```

Result

Astra Control 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.

About this task

Astra Control supports snapshot creation using storage classes backed by the following drivers:

- `ontap-nas`
- `ontap-san`
- `ontap-san-economy`

If your app uses a storage class backed by the `ontap-nas-economy` driver, snapshots can't be created. Use an alternate storage class for snapshots.

Create a snapshot using the Web UI

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 **Next**.
4. [**Tech preview**] Choose a destination bucket for the snapshot from the list of storage buckets.
5. Review the snapshot summary and select **Snapshot**.

[Tech preview] Create a snapshot using a CR

Steps

1. Create the custom resource (CR) file and name it `astra-control-snapshot-cr.yaml`. Update the values in brackets <> to match your Astra Control environment and cluster configuration:
   - <CR_NAME>: The name of this custom resource; choose a unique and sensible name for your environment.
   - <APPLICATION_NAME>: The Kubernetes name of the application to snapshot.
   - <APPVAULT_NAME>: The name of the AppVault where the snapshot contents should be stored.
   - <RECLAIM_POLICY>: *(Optional)* Defines what happens to a snapshot when the snapshot CR is deleted. Valid options:
     - `Retain`
     - `Delete` (default)

```
apiVersion: astra.netapp.io/v1
kind: Snapshot
metadata:
  namespace: astra-connector
  name: <CR_NAME>
spec:
  applicationRef: <APPLICATION_NAME>
  appVaultRef: <APPVAULT_NAME>
  reclaimPolicy: <RECLAIM_POLICY>
```

2. After you populate the `astra-control-snapshot-cr.yaml` file with the correct values, apply the CR:

```
kubectl apply -f astra-control-snapshot-cr.yaml
```

Result

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

Create a backup

You can back up an app at any time.

About this task

Buckets in Astra Control do not report available capacity. Before backing up or cloning apps managed by Astra Control, check bucket information in the appropriate storage management system.

If your app uses a storage class backed by the `ontap-nas-economy` driver, you need to enable backup and restore functionality. Be sure that you have defined a `backendType` parameter in your Kubernetes storage object with a value of `ontap-nas-economy` before performing any protection operations.

> Astra Control supports backup creation using storage classes backed by the following drivers:
>
> - `ontap-nas`
> - `ontap-nas-economy`
> - `ontap-san`
> - `ontap-san-economy`

Create a backup using the Web UI

Steps

1. Select **Applications**.
2. From the Options menu in the **Actions** column for the desired app, select **Back up**.
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. [**Tech preview**] Choose a destination bucket for the backup from the list of storage buckets.
6. Select **Next**.
7. Review the backup summary and select **Back up**.

[Tech preview] Create a backup using a CR

Steps

1. Create the custom resource (CR) file and name it `astra-control-backup-cr.yaml`. Update the values in brackets <> to match your Astra Control environment and cluster configuration:
   - <CR_NAME>: The name of this custom resource; choose a unique and sensible name for your environment.
   - <APPLICATION_NAME>: The Kubernetes name of the application to back up.
   - <APPVAULT_NAME>: The name of the AppVault where the backup contents should be stored.

   ```
   apiVersion: astra.netapp.io/v1
   kind: Backup
   metadata:
     namespace: astra-connector
     name: <CR_NAME>
   spec:
     applicationRef: <APPLICATION_NAME>
     appVaultRef: <APPVAULT_NAME>
   ```

2. After you populate the `astra-control-backup-cr.yaml` file with the correct values, apply the CR:

   ```
   kubectl apply -f astra-control-backup-cr.yaml
   ```

Result

Astra Control 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.
- If you need to cancel a running backup, use the instructions in Cancel backups. To delete the backup, wait until it has completed and then use the instructions in Delete backups.
- 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.

Enable backup and restore for ontap-nas-economy operations

Astra Control Provisioner provides backup and restore functionality that can be enabled for storage backends that are using the `ontap-nas-economy` storage class.

Before you begin

- You have enabled Astra Control Provisioner.
- You have defined an application in Astra Control. This application will have limited protection functionality until you complete this procedure.
- You have `ontap-nas-economy` selected as the default storage class for your storage backend.

Steps

1. Do the following on the ONTAP storage backend:
   a. Find the SVM that is hosting the `ontap-nas-economy`-based volumes of the application.
   b. Log in to a terminal connected to ONTAP where the volumes are created.
   c. Hide the snapshot directory for the SVM:

      This change affects the entire SVM. The hidden directory will continue to be accessible.

      ```
      nfs modify -vserver <svm name> -v3-hide-snapshot enabled
      ```

      Verify that the snapshot directory on the ONTAP storage backend is hidden. Failure to hide this directory might lead to loss of access to your application, especially if it is using NFSv3.

2. Do the following in Astra Control Provisioner:
   a. Enable the snapshot directory for each PV that is `ontap-nas-economy` based and associated with the application:

      ```
      tridentctl update volume <pv name> --snapshot-dir=true --pool-level
      =true -n trident
      ```

   b. Confirm that the snapshot directory has been enabled for each associated PV:

```
tridentctl get volume <pv name> -n trident -o yaml | grep snapshotDir
```

Response:

```
snapshotDirectory: "true"
```

3. In Astra Control, refresh the application after enabling all associated snapshot directories so that Astra Control recognizes the changed value.

Result

The application is ready to backup and restore using Astra Control. Each PVC is also available to be used by other applications for backups and restores.

Create an immutable backup

An immutable backup cannot be modified, deleted, or overwritten as long as the retention policy on the bucket that stores the backup forbids it. You can create immutable backups by backing up applications to buckets that have a retention policy configured. Refer to Data protection for important information about working with immutable backups.

Before you begin

You need to configure the destination bucket with a retention policy. How you do this will differ depending on which storage provider you use. Refer to the storage provider documentation for more information:

- **Amazon Web Services**: Enable S3 Object Lock when creating the bucket and set a default retention mode of "governance" with a default retention period.
- **NetApp StorageGRID**: Enable S3 Object Lock when creating the bucket and set a default retention mode of "compliance" with a default retention period.

> Buckets in Astra Control do not report available capacity. Before backing up or cloning apps managed by Astra Control, check bucket information in the appropriate storage management system.

> If your app uses a storage class backed by the `ontap-nas-economy` driver, be sure that you have defined a `backendType` parameter in your Kubernetes storage object with a value of `ontap-nas-economy` before performing any protection operations.

Steps

1. Select **Applications**.
2. From the Options menu in the **Actions** column for the desired app, select **Back up**.
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 bucket for the backup from the list of storage buckets. A write once read many (WORM) bucket is indicated with a status of "Locked" next to the bucket name.

> If the bucket is an unsupported type, this is indicated when you hover over or select the bucket.

6. Select **Next**.
7. Review the backup summary and select **Back up**.

Result

Astra Control creates an immutable 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.
> - If you try to create two immutable backups of the same app to the same bucket at the same time, Astra Control prevents the second backup from starting. Wait until the first backup is complete before starting another.
> - You cannot cancel a running immutable backup.
> - 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.

> An immutable backup is indicated with a status of "Locked" next to the bucket it is using.

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.

> You cannot delete a snapshot that currently is being replicated.

Steps

1. Select **Applications** and then select the name of a managed 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 deletes the snapshot.

Cancel backups

You can cancel a backup that is in progress.

> To cancel a backup, the backup must be in `Running` state. You cannot cancel a backup that is in `Pending` state.

> You cannot cancel a running immutable backup.

Steps

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 **Cancel**.
5. Type the word "cancel" to confirm the operation and then select **Yes, cancel backup**.

Delete backups

Delete the scheduled or on-demand backups that you no longer need. You cannot delete a backup made to an immutable bucket until the bucket's retention policy enables you to do so.

> You cannot delete an immutable backup before the retention period expires.

> If you need to cancel a running backup, use the instructions in Cancel backups. To delete the backup, wait until it has completed and then use these instructions.

Steps

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 deletes the backup.

[Tech preview] Protect an entire cluster

You can create a scheduled, automatic backup of any or all unmanaged namespaces on a cluster. These workflows are provided by NetApp as a Kubernetes service account, role bindings, and a cron job, orchestrated with a Python script.

How it works

When you configure and install the full-cluster backup workflow, a cron job runs periodically and protects any namespace that is not already managed, automatically creating protection policies based on schedules that you choose during installation.

If you don't want to protect every unmanaged namespace on the cluster with the full cluster backup workflow, you can instead utilize the label-based backup workflow. The label-based backup workflow also uses a cron task, but instead of protecting all unmanaged namespaces, it identifies namespaces by labels you provide to optionally protect the namespaces based on bronze, silver, or gold backup policies.

When a new namespace is created that falls within the scope of your chosen workflow, it is automatically protected, without any administrator action. These workflows are implemented on a per-cluster basis, so different clusters can make use of either workflow with unique protection levels, depending on cluster importance.

Example: Full cluster protection

As an example, when you configure and install the full cluster backup workflow, any apps in any namespace are periodically managed and protected without further effort by the administrator. The namespace doesn't have to exist at the time you install the workflow; if a namespace is added in the future, it will be protected.

Example: Label-based protection

For more granularity, you can use the label-based workflow. For example, you can install this workflow and tell your users to apply one of several labels to any namespaces they want to protect, depending on the level of protection they need. This enables users to create the namespace with one of those labels, and they don't have to notify an administrator. Their new namespace and all apps within it are automatically protected.

Create a scheduled backup of all namespaces

You can create a scheduled backup of all namespaces on a cluster using the full cluster backup workflow.

Steps

1. Download the following files to a machine that has network access to your cluster:
    - components.yaml CRD file
    - protectCluster.py Python script
2. To configure and install the toolkit, follow the included instructions.

Create a scheduled backup of specific namespaces

You can create a scheduled backup of specific namespaces by their labels using the label-based backup workflow.

Steps

1. Download the following files to a machine that has network access to your cluster:
    - components.yaml CRD file
    - protectCluster.py Python script
2. To configure and install the toolkit, follow the included instructions.

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 Astra Control API to restore apps.

Before you begin

- **Protect your apps first**: It is strongly recommended that you take a snapshot or backup of your application before restoring it. This will enable you to clone from the snapshot or backup if the restore is unsuccessful.
- **Check destination volumes**: If you restore to a different storage class, ensure that the storage class uses the same persistent volume access mode (for example, ReadWriteMany). The restore operation will fail if the destination persistent volume access mode is different. For example, if your source persistent volume uses the RWX access mode, selecting a destination storage class that is not able to provide RWX, such as Azure Managed Disks, AWS EBS, Google Persistent Disk or `ontap-san`, will cause the restore operation to fail. For more information about persistent volume access modes, refer to the Kubernetes documentation.
- **Plan for space needs**: When you perform an in-place restore of an application that uses NetApp ONTAP storage, the space used by the restored app can double. After performing an in-place restore, remove any unwanted snapshots from the restored application to free up storage space.
- **(Red Hat OpenShift clusters only) Add policies**: 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
  ```

- **Supported storage class drivers**: Astra Control supports restoring backups using storage classes backed by the following drivers:
  - `ontap-nas`
  - `ontap-nas-economy`
  - `ontap-san`
  - `ontap-san-economy`
- **(ontap-nas-economy driver only) Backups and restores**: Before backing up or restoring an app that uses a storage class backed by the `ontap-nas-economy` driver, verify that the snapshot directory on the ONTAP storage backend is hidden. Failure to hide this directory might lead to loss of access to your application, especially if it is using NFSv3.
- **Helm deployed apps**: Apps deployed with Helm 3 (or upgraded from Helm 2 to Helm 3) are fully supported. Apps deployed with Helm 2 are not supported.

> Performing an in-place restore operation on an app that shares resources with another app can have unintended results. Any resources that are shared between the apps are replaced when an in-place restore is performed on one of the apps. For more information, see this example.

Perform the following steps, depending on the type of archive you want to restore:

Restore data from backup or snapshot using the web UI

You can restore data using the Astra Control web UI.

Steps

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

2. From the Options menu in the Actions column, select **Restore**.
3. Choose the restore type:
   - **Restore to original namespaces**: Use this procedure to restore the app in-place to the original cluster.

     > If your app uses a storage class backed by the `ontap-nas-economy` driver, you must restore the app using the original storage classes. You cannot specify a different storage class if you are restoring the app to the same namespace.

     a. Select the snapshot or backup to use to restore the app in-place, which reverts the app to an earlier version of itself.
     b. Select **Next**.

     > 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.

   - **Restore to new namespaces**: Use this procedure to restore the app to another cluster or with different namespaces from the source.
     a. Specify the name for the restored app.
     b. Choose the destination cluster for the app you intend to restore.
     c. Enter a destination namespace for each source namespace associated with the app.

     > Astra Control creates new destination namespaces as part of this restore option. Destination namespaces that you specify must not be already present on the destination cluster.

     d. Select **Next**.
     e. Select the snapshot or backup to use to restore the app.
     f. Select **Next**.
     g. Choose one of the following:
        - **Restore using original storage classes**: The application uses the originally associated storage class unless it does not exist on the target cluster. In this case, the default storage class for the cluster will be used.
        - **Restore using a different storage class**: Select a storage class that exists on the target cluster. All application volumes, regardless of their originally associated storage classes, will be migrated to this different storage class as part of the restore.
     h. Select **Next**.
4. Choose any resources to filter:
   - **Restore all resources**: Restore all resources associated with the original app.
   - **Filter resources**: Specify rules to restore a sub-set of the original application resources:
     i. Choose to include or exclude resources from the restored application.
     ii. Select either **Add include rule** or **Add exclude rule** and configure the rule to filter the correct resources during application restore. You can edit a rule or remove it and create a rule again until the configuration is correct.

To learn about configuring include and exclude rules, see Filter resources during an application restore.

5. Select **Next**.
6. Review details about the restore action carefully, type "restore" (if prompted), and select **Restore**.

[Tech preview] Restore from backup using a custom resource (CR)

You can restore data from a backup using a custom resource (CR) file either to a different namespace or the original source namespace.

Restore from backup using a CR

Steps

1. Create the custom resource (CR) file and name it `astra-control-backup-restore-cr.yaml`. Update the values in brackets <> to match your Astra Control environment and cluster configuration:
   - <CR_NAME>: The name of this CR operation; choose a sensible name for your environment.
   - <APPVAULT_NAME>: The name of the AppVault where the backup contents are stored.
   - <BACKUP_PATH>: The path inside AppVault where the backup contents are stored. For example:

     ```
     ONTAP-S3_1343ff5e-4c41-46b5-af00/backups/schedule-
     20231213023800_94347756-9d9b-401d-a0c3
     ```

   - <SOURCE_NAMESPACE>: The source namespace of the restore operation.
   - <DESTINATION_NAMESPACE>: The destination namespace of the restore operation.

     ```
     apiVersion: astra.netapp.io/v1
     kind: BackupRestore
     metadata:
       name: <CR_NAME>
       namespace: astra-connector
     spec:
       appVaultRef: <APPVAULT_NAME>
       appArchivePath: <BACKUP_PATH>
       namespaceMapping: [{"source": "<SOURCE_NAMESPACE>",
     "destination": "<DESTINATION_NAMESPACE>"}]
     ```

2. (Optional) If you need to select only certain resources of the application to restore, add filtering that includes or excludes resources marked with particular labels:
   - "<INCLUDE-EXCLUDE>": *(Required for filtering)* Use `include` or `exclude` to include or exclude a resource defined in resourceMatchers. Add the following resourceMatchers parameters to define the resources to be included or excluded:
     - <GROUP>: *(Optional)* Group of the resource to be filtered.
     - <KIND>: *(Optional)* Kind of the resource to be filtered.
     - <VERSION>: *(Optional)* Version of the resource to be filtered.
     - <NAMES>: *(Optional)* Names in the Kubernetes metadata.name field of the resource to be filtered.
     - <NAMESPACES>: *(Optional)* Namespaces in the Kubernetes metadata.name field of the resource to be filtered.
     - <SELECTORS>: *(Optional)* Label selector string in the Kubernetes metadata.name field of the resource as defined in Kubernetes documentation. Example: `"trident.netapp.io/os=linux"`.

       Example:

```
spec:
    resourceFilter:
        resourceSelectionCriteria: "<INCLUDE-EXCLUDE>"
        resourceMatchers:
           group: <GROUP>
           kind: <KIND>
           version: <VERSION>
           names: <NAMES>
           namespaces: <NAMESPACES>
           labelSelectors: <SELECTORS>
```

3. After you populate the `astra-control-backup-restore-cr.yaml` file with the correct values, apply the CR:

```
kubectl apply -f astra-control-backup-restore-cr.yaml
```

Restore from backup to the original namespace using a CR

Steps

1. Create the custom resource (CR) file and name it `astra-control-backup-ipr-cr.yaml`. Update the values in brackets <> to match your Astra Control environment and cluster configuration:
   - <CR_NAME>: The name of this CR operation; choose a sensible name for your environment.
   - <APPVAULT_NAME>: The name of the AppVault where the backup contents are stored.
   - <BACKUP_PATH>: The path inside AppVault where the backup contents are stored. For example:

```
ONTAP-S3_1343ff5e-4c41-46b5-af00/backups/schedule-
20231213023800_94347756-9d9b-401d-a0c3
```

```
apiVersion: astra.netapp.io/v1
kind: BackupInplaceRestore
metadata:
  name: <CR_NAME>
  namespace: astra-connector
spec:
  appVaultRef: <APPVAULT_NAME>
  appArchivePath: <BACKUP_PATH>
```

2. (Optional) If you need to select only certain resources of the application to restore, add filtering that includes or excludes resources marked with particular labels:
   - "<INCLUDE-EXCLUDE>": *(Required for filtering)* Use `include` or `exclude` to include or exclude a resource defined in resourceMatchers. Add the following resourceMatchers parameters to define the resources to be included or excluded:

- <GROUP>: *(Optional)* Group of the resource to be filtered.
- <KIND>: *(Optional)* Kind of the resource to be filtered.
- <VERSION>: *(Optional)* Version of the resource to be filtered.
- <NAMES>: *(Optional)* Names in the Kubernetes metadata.name field of the resource to be filtered.
- <NAMESPACES>: *(Optional)* Namespaces in the Kubernetes metadata.name field of the resource to be filtered.
- <SELECTORS>: *(Optional)* Label selector string in the Kubernetes metadata.name field of the resource as defined in Kubernetes documentation. Example: `"trident.netapp.io/os=linux"`.

  Example:

```
spec:
    resourceFilter:
        resourceSelectionCriteria: "<INCLUDE-EXCLUDE>"
        resourceMatchers:
           group: <GROUP>
           kind: <KIND>
           version: <VERSION>
           names: <NAMES>
           namespaces: <NAMESPACES>
           labelSelectors: <SELECTORS>
```

3. After you populate the `astra-control-backup-ipr-cr.yaml` file with the correct values, apply the CR:

```
kubectl apply -f astra-control-backup-ipr-cr.yaml
```

[Tech preview] Restore from snapshot using a custom resource (CR)

You can restore data from a snapshot using a custom resource (CR) file either to a different namespace or the original source namespace.

Restore from snapshot using a CR

Steps

1. Create the custom resource (CR) file and name it `astra-control-snapshot-restore-cr.yaml`. Update the values in brackets <> to match your Astra Control environment and cluster configuration:
   - <CR_NAME>: The name of this CR operation; choose a sensible name for your environment.
   - <APPVAULT_NAME>: The name of the AppVault where the backup contents are stored.
   - <BACKUP_PATH>: The path inside AppVault where the backup contents are stored. For example:

     ```
     ONTAP-S3_1343ff5e-4c41-46b5-af00/backups/schedule-
     20231213023800_94347756-9d9b-401d-a0c3
     ```

   - <SOURCE_NAMESPACE>: The source namespace of the restore operation.
   - <DESTINATION_NAMESPACE>: The destination namespace of the restore operation.

     ```
     apiVersion: astra.netapp.io/v1
     kind: SnapshotRestore
     metadata:
       name: <CR_NAME>
       namespace: astra-connector
     spec:
       appArchivePath: <BACKUP_PATH>
       appVaultRef: <APPVAULT_NAME>
       namespaceMapping: [{"source": "<SOURCE_NAMESPACE>",
     "destination": "<DESTINATION_NAMESPACE>"}]
     ```

2. (Optional) If you need to select only certain resources of the application to restore, add filtering that includes or excludes resources marked with particular labels:
   - "<INCLUDE-EXCLUDE>": *(Required for filtering)* Use `include` or `exclude` to include or exclude a resource defined in resourceMatchers. Add the following resourceMatchers parameters to define the resources to be included or excluded:
     - <GROUP>: *(Optional)* Group of the resource to be filtered.
     - <KIND>: *(Optional)* Kind of the resource to be filtered.
     - <VERSION>: *(Optional)* Version of the resource to be filtered.
     - <NAMES>: *(Optional)* Names in the Kubernetes metadata.name field of the resource to be filtered.
     - <NAMESPACES>: *(Optional)* Namespaces in the Kubernetes metadata.name field of the resource to be filtered.
     - <SELECTORS>: *(Optional)* Label selector string in the Kubernetes metadata.name field of the resource as defined in Kubernetes documentation. Example: `"trident.netapp.io/os=linux"`.

Example:

```
spec:
    resourceFilter:
        resourceSelectionCriteria: "<INCLUDE-EXCLUDE>"
        resourceMatchers:
           group: <GROUP>
           kind: <KIND>
           version: <VERSION>
           names: <NAMES>
           namespaces: <NAMESPACES>
           labelSelectors: <SELECTORS>
```

3. After you populate the `astra-control-snapshot-restore-cr.yaml` file with the correct values, apply the CR:

```
kubectl apply -f astra-control-snapshot-restore-cr.yaml
```

Restore from snapshot to the original namespace using a CR

Steps

1. Create the custom resource (CR) file and name it `astra-control-snapshot-ipr-cr.yaml`. Update the values in brackets <> to match your Astra Control environment and cluster configuration:
   - <CR_NAME>: The name of this CR operation; choose a sensible name for your environment.
   - <APPVAULT_NAME>: The name of the AppVault where the backup contents are stored.
   - <BACKUP_PATH>: The path inside AppVault where the backup contents are stored. For example:

```
ONTAP-S3_1343ff5e-4c41-46b5-af00/backups/schedule-
20231213023800_94347756-9d9b-401d-a0c3
```

```
apiVersion: astra.netapp.io/v1
kind: SnapshotInplaceRestore
metadata:
  name: <CR_NAME>
  namespace: astra-connector
spec:
  appArchivePath: <BACKUP_PATH>
  appVaultRef: <APPVAULT_NAME>
```

2. (Optional) If you need to select only certain resources of the application to restore, add filtering that includes or excludes resources marked with particular labels:
   - "<INCLUDE-EXCLUDE>": *(Required for filtering)* Use `include` or `exclude` to include or exclude

a resource defined in resourceMatchers. Add the following resourceMatchers parameters to define the resources to be included or excluded:

- <GROUP>: *(Optional)* Group of the resource to be filtered.
- <KIND>: *(Optional)* Kind of the resource to be filtered.
- <VERSION>: *(Optional)* Version of the resource to be filtered.
- <NAMES>: *(Optional)* Names in the Kubernetes metadata.name field of the resource to be filtered.
- <NAMESPACES>: *(Optional)* Namespaces in the Kubernetes metadata.name field of the resource to be filtered.
- <SELECTORS>: *(Optional)* Label selector string in the Kubernetes metadata.name field of the resource as defined in Kubernetes documentation. Example: `"trident.netapp.io/os=linux"`.

  Example:

```
spec:
    resourceFilter:
        resourceSelectionCriteria: "<INCLUDE-EXCLUDE>"
        resourceMatchers:
           group: <GROUP>
           kind: <KIND>
           version: <VERSION>
           names: <NAMES>
           namespaces: <NAMESPACES>
           labelSelectors: <SELECTORS>
```

3. After you populate the `astra-control-snapshot-ipr-cr.yaml` file with the correct values, apply the CR:

```
kubectl apply -f astra-control-snapshot-ipr-cr.yaml
```

Result

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

> After a data protection operation (clone, backup, or 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.

> 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 clone or restore operation creates a new namespace, 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.

Filter resources during an application restore

You can add a filter rule to a restore operation that will specify existing application resources to be included or excluded from the restored application. You can include or exclude resources based on a specified namespace, label, or GVK (GroupVersionKind).

Expand for more about include and exclude scenarios

- **You select an include rule with original namespaces (in-place restore)**: Existing application resources that you define in the rule will be deleted and replaced by those from the selected snapshot or backup you are using for the restore. Any resources that you do not specify in the include rule will remain unchanged.
- **You select an include rule with new namespaces**: Use the rule to select the specific resources you want in the restored application. Any resources that you do not specify in the include rule will not be included in the restored application.
- **You select an exclude rule with original namespaces (in-place restore)**: The resources you specify to be excluded will not be restored and remain unchanged. Resources that you do not specify to exclude will be restored from the snapshot or backup. All data on persistent volumes will be deleted and recreated if the corresponding StatefulSet is part of the filtered resources.
- **You select an exclude rule with new namespaces**: Use the rule to select the specific resources you want to remove from the restored application. Resources that you do not specify to exclude will be restored from the snapshot or backup.

Rules are either include or exclude types. Rules combining resource inclusion and exclusion are not available.

Steps

1. After you have chosen to filter resources and selected an include or exclude option in the Restore App wizard, select **Add include rule** or **Add exclude rule**.

   > You cannot exclude any cluster-scoped resources that are automatically included by Astra Control.

2. Configure the filter rule:

   > You must specify at least one namespace, label, or GVK. Ensure that any resources you retain after the filter rules are applied are sufficient to keep the restored application in a healthy state.

   a. Select a specific namespace for the rule. If you don't make a selection, all namespaces will be used in the filter.

      > If your application originally contained multiple namespaces and you restore it to new namespaces, all namespaces will be created even if they don't contain resources.

b. (Optional) Enter a resource name.

c. (Optional) **Label selector**: Include a label selector to add to the rule. The label selector is used to filter only those resources matching the selected label.

d. (Optional) Select **Use GVK (GroupVersionKind) set to filter resources** for additional filtering options.

> If you use a GVK filter, you must specify Version and Kind.

i. (Optional) **Group**: From the drop-down list, select the Kubernetes API group.

ii. **Kind**: From the drop-down list, select the object schema for the Kubernetes resource type to use in the filter.

iii. **Version**: Select the Kubernetes API version.

3. Review the rule that is created based on your entries.
4. Select **Add**.

> You can create as many resource include and exclude rules as you want. The rules appear in the restore application summary before you initiate the operation.

In-place restore complications for an app that shares resources with another app

You can perform an in-place restore operation on an app that shares resources with another app and produce unintended results. Any resources that are shared between the apps are replaced when an in-place restore is performed on one of the apps.

The following is an example scenario that creates an undesirable situation when using NetApp SnapMirror replication for a restore:

1. You define the application `app1` using the namespace `ns1`.
2. You configure a replication relationship for `app1`.
3. You define the application `app2` (on the same cluster) using the namespaces `ns1` and `ns2`.
4. You configure a replication relationship for `app2`.
5. You reverse replication for `app2`. This causes the `app1` app on the source cluster to be deactivated.

Replicate apps between storage backends using SnapMirror technology

Using Astra Control, you can build business continuity for your applications with a low-RPO (Recovery Point Objective) and low-RTO (Recovery Time Objective) using asynchronous replication capabilities of NetApp SnapMirror technology. Once configured, this enables your applications to replicate data and application changes from one storage backend to another, on the same cluster or between different clusters.

For a comparison between backups/restores and replication, refer to Data protection concepts.

You can replicate apps in different scenarios, such as the following on-premises only, hybrid, and multi-cloud scenarios:

- On-premises site A to on-premises site A

- On-premises site A to on-premises site B
- On-premises to cloud with Cloud Volumes ONTAP
- Cloud with Cloud Volumes ONTAP to on-premises
- Cloud with Cloud Volumes ONTAP to cloud (between different regions in the same cloud provider or to different cloud providers)

Astra Control can replicate apps across on-premises clusters, on-premises to cloud (using Cloud Volumes ONTAP) or between clouds (Cloud Volumes ONTAP to Cloud Volumes ONTAP).

> You can simultaneously replicate a different app in the opposite direction. For example, Apps A, B, C can be replicated from Datacenter 1 to Datacenter 2; and Apps X, Y, Z can be replicated from Datacenter 2 to Datacenter 1.

Using Astra Control, you can do the following tasks related to replicating applications:

- Set up a replication relationship
- Bring a replicated app online on the destination cluster (failover)
- Resync a failed over replication
- Reverse application replication
- Fail back applications to the original source cluster
- Delete an application replication relationship

Replication prerequisites

Astra Control application replication requires that the following prerequisites be met before you begin:

ONTAP clusters

- **Astra Control Provisioner or Astra Trident**: Astra Control Provisioner or Astra Trident must exist on both the source and destination Kubernetes clusters that utilize ONTAP as a backend. Astra Control supports replication with NetApp SnapMirror technology using storage classes backed by the following drivers:
  - `ontap-nas`
  - `ontap-san`
- **Licenses**: ONTAP SnapMirror asynchronous licenses using the Data Protection bundle must be enabled on both the source and destination ONTAP clusters. Refer to SnapMirror licensing overview in ONTAP for more information.

Peering

- **Cluster and SVM**: The ONTAP storage backends must be peered. Refer to Cluster and SVM peering overview for more information.

  > Ensure that the SVM names used in the replication relationship between two ONTAP clusters are unique.

- **Astra Control Provisioner or Astra Trident and SVM**: The peered remote SVMs must be available to Astra Control Provisioner or Astra Trident on the destination cluster.

Astra Control Center

Deploy Astra Control Center in a third fault domain or secondary site for seamless disaster recovery.

- **Managed backends**: You need to add and manage ONTAP storage backends in Astra Control Center to create a replication relationship.

  Adding and managing ONTAP storage backends in Astra Control Center is optional if you have enabled Astra Control Provisioner.

- **Managed clusters**: Add and manage the following clusters with Astra Control, ideally at different failure domains or sites:
  - Source Kubernetes cluster
  - Destination Kubernetes cluster
  - Associated ONTAP clusters
- **User accounts**: When you add an ONTAP storage backend to Astra Control Center, apply user credentials with the "admin" role. This role has access methods `http` and `ontapi` enabled on both ONTAP source and destination clusters. Refer to Manage User Accounts in ONTAP documentation for more information.

  With Astra Control Provisioner functionality, you don't need to specifically define an "admin" role to manage clusters in Astra Control Center as these credentials aren't required by Astra Control Center.

Astra Control Center does not support NetApp SnapMirror replication for storage backends that are using the NVMe over TCP protocol.

Astra Trident / ONTAP configuration

Astra Control Center requires that you configure at least one storage backend that supports replication for both the source and destination clusters. If the source and destination clusters are the same, the destination application should use a different storage backend than the source application for the best resiliency.

Astra Control replication supports apps that use a single storage class. When you add an app to a namespace, be sure the app has the same storage class as other apps in the namespace. When you add a PVC to a replicated app, be sure the new PVC has the same storage class as other PVCs in the namespace.

Set up a replication relationship

Setting up a replication relationship involves the following:

- Choosing how frequently you want Astra Control to take an app snapshot (which includes the app's Kubernetes resources as well as the volume snapshots for each of the app's volumes)
- Choosing the replication schedule (included Kubernetes resources as well as persistent volume data)
- Setting the time for the snapshot to be taken

Steps

1. From the Astra Control left navigation, select **Applications**.

2. Select the **Data Protection** > **Replication** tab.
3. Select **Configure replication policy**. Or, from the Application Protection box, select the Actions option and select **Configure replication policy**.
4. Enter or select the following information:
   - **Destination cluster**: Enter a destination cluster (this can be the same as the source cluster).
   - **Destination storage class**: Select or enter the storage class that uses the peered SVM on the destination ONTAP cluster. As a best practice, the destination storage class should point to a different storage backend than the source storage class.
   - **Replication type**: `Asynchronous` is currently the only replication type available.
   - **Destination namespace**: Enter new or existing destination namespaces for the destination cluster.
   - (Optional) Add additional namespaces by selecting **Add namespace** and choosing the namespace from the drop-down list.
   - **Replication frequency**: Set how often you want Astra Control to take a snapshot and replicate it to the destination.
   - **Offset**: Set the number of minutes from the top of the hour that you want Astra Control to take a snapshot. You might want to use an offset so that it doesn't coincide with other scheduled operations.

     Offset backup and replication schedules to avoid schedule overlaps. For example, perform backups at the top of the hour every hour and schedule replication to start with a 5-minute offset and a 10-minute interval.
5. Select **Next**, review the summary, and select **Save**.

   At first, the status displays "app-mirror" before the first schedule occurs.

   Astra Control creates an application snapshot used for replication.
6. To see the application snapshot status, select the **Applications** > **Snapshots** tab.

   The snapshot name uses the format of `replication-schedule-<string>`. Astra Control retains the last snapshot that was used for replication. Any older replication snapshots are deleted after successful completion of replication.

Result

This creates the replication relationship.

Astra Control completes the following actions as a result of establishing the relationship:

- Creates a namespace on the destination (if it doesn't exist)
- Creates a PVC on the destination namespace corresponding to the source app's PVCs.
- Takes an initial app-consistent snapshot.
- Establishes the SnapMirror relationship for persistent volumes using the initial snapshot.

The **Data Protection** page shows the replication relationship state and status:
<Health status> | <Relationship life cycle state>

For example:

Normal | Established

Learn more about replication states and status at the end of this topic.

Bring a replicated app online on the destination cluster (failover)

Using Astra Control, you can fail over replicated applications to a destination cluster. This procedure stops the replication relationship and brings the app online on the destination cluster. This procedure does not stop the app on the source cluster if it was operational.

Steps

1. From the Astra Control left navigation, select **Applications**.
2. Select the **Data Protection** > **Replication** tab.
3. From the Actions menu, select **Fail over**.
4. In the Fail over page, review the information and select **Fail over**.

Result

The following actions occur as a result of the failover procedure:

- The destination app is started based on the latest replicated snapshot.
- The source cluster and app (if operational) are not stopped and will continue to run.
- The replication state changes to "Failing over" and then to "Failed over" when it has completed.
- The source app's protection policy is copied to the destination app based on the schedules present on the source app at the time of the failover.
- If the source app has one or more post-restore execution hooks enabled, those execution hooks are run for the destination app.
- Astra Control shows the app both on the source and destination clusters and its respective health.

Resync a failed over replication

The resync operation re-establishes the replication relationship. You can choose the source of the relationship to retain the data on the source or destination cluster. This operation re-establishes the SnapMirror relationships to start the volume replication in the direction of choice.

The process stops the app on the new destination cluster before re-establishing replication.

> During the resync process, the life cycle state shows as "Establishing."

Steps

1. From the Astra Control left navigation, select **Applications**.
2. Select the **Data Protection** > **Replication** tab.
3. From the Actions menu, select **Resync**.
4. In the Resync page, select either the source or destination app instance containing the data that you want to preserve.

   > Choose the resync source carefully, as the data on the destination will be overwritten.

5. Select **Resync** to continue.

6. Type "resync" to confirm.
7. Select **Yes, resync** to finish.

Result

- The Replication page shows "Establishing" as the replication status.
- Astra Control stops the application on the new destination cluster.
- Astra Control re-establishes the persistent volume replication in the selected direction using SnapMirror resync.
- The Replication page shows the updated relationship.

Reverse application replication

This is the planned operation to move the application to the destination storage backend while continuing to replicate back to the original source storage backend. Astra Control stops the source application and replicates the data to the destination before failing over to the destination app.

In this situation, you are swapping the source and destination.

Steps

1. From the Astra Control left navigation, select **Applications**.
2. Select the **Data Protection** > **Replication** tab.
3. From the Actions menu, select **Reverse replication**.
4. In the Reverse Replication page, review the information and select **Reverse replication** to continue.

Result

The following actions occur as a result of the reverse replication:

- A snapshot is taken of the original source app's Kubernetes resources.
- The original source app's pods are gracefully stopped by deleting the app's Kubernetes resources (leaving PVCs and PVs in place).
- After the pods are shut down, snapshots of the app's volumes are taken and replicated.
- The SnapMirror relationships are broken, making the destination volumes ready for read/write.
- The app's Kubernetes resources are restored from the pre-shutdown snapshot, using the volume data replicated after the original source app was shut down.
- Replication is re-established in the reverse direction.

Fail back applications to the original source cluster

Using Astra Control, you can achieve "fail back" after a failover operation by using the following sequence of operations. In this workflow to restore the original replication direction, Astra Control replicates (resyncs) any application changes back to the original source application before reversing the replication direction.

This process starts from a relationship that has completed a failover to a destination and involves the following steps:

- Start with a failed over state.
- Resync the relationship.

- Reverse the replication.

Steps

1. From the Astra Control left navigation, select **Applications**.
2. Select the **Data Protection** > **Replication** tab.
3. From the Actions menu, select **Resync**.
4. For a fail back operation, choose the failed over app as the source of the resync operation (preserving any data written post failover).
5. Type "resync" to confirm.
6. Select **Yes, resync** to finish.
7. After the resync is complete, in the Data Protection > Replication tab, from the Actions menu, select **Reverse replication**.
8. In the Reverse Replication page, review the information and select **Reverse replication**.

Result

This combines the results from the "resync" and "reverse relationship" operations to bring the application online on the original source cluster with replication resumed to the original destination cluster.

Delete an application replication relationship

Deleting the relationship results in two separate apps with no relationship between them.

Steps

1. From the Astra Control left navigation, select **Applications**.
2. Select the **Data Protection** > **Replication** tab.
3. From the Application Protection box or in the relationship diagram, select **Delete replication relationship**.

Result

The following actions occur as a result of deleting a replication relationship:

- If the relationship is established but the app has not yet been brought online on the destination cluster (failed over), Astra Control retains PVCs created during initialization, leaves an "empty" managed app on the destination cluster, and retains the destination app to keep any backups that might have been created.
- If the app has been brought online on the destination cluster (failed over), Astra Control retains PVCs and destination apps. Source and destination apps are now treated as independent apps. The backup schedules remain on both apps but are not associated with each other.

Replication relationship health status and relationship life cycle states

Astra Control displays the health of the relationship and the states of the life cycle of the replication relationship.

Replication relationship health statuses

The following statuses indicate the health of the replication relationship:

- **Normal**: The relationship is either establishing or has established, and the most recent snapshot transferred successfully.

- **Warning**: The relationship is either failing over or has failed over (and therefore is no longer protecting the source app).
- **Critical**
  - The relationship is establishing or failed over, and the last reconcile attempt failed.
  - The relationship is established, and the last attempt to reconcile the addition of a new PVC is failing.
  - The relationship is established (so a successful snapshot has replicated, and failover is possible), but the most recent snapshot failed or failed to replicate.

Replication life cycle states

The following states states reflect the different stages of the replication life cycle:

- **Establishing**: A new replication relationship is being created. Astra Control creates a namespace if needed, creates persistent volume claims (PVCs) on new volumes on the destination cluster, and creates SnapMirror relationships. This status can also indicate that the replication is resyncing or reversing replication.
- **Established**: A replication relationship exists. Astra Control periodically checks that the PVCs are available, checks the replication relationship, periodically creates snapshots of the app, and identifies any new source PVCs in the app. If so, Astra Control creates the resources to include them in the replication.
- **Failing over**: Astra Control breaks the SnapMirror relationships and restores the app's Kubernetes resources from the last successfully replicated app snapshot.
- **Failed over**: Astra Control stops replicating from the source cluster, uses the most recent (successful) replicated app snapshot on the destination, and restores the Kubernetes resources.
- **Resyncing**: Astra Control resyncs the new data on the resync source to the resync destination by using SnapMirror resync. This operation might overwrite some of the data on the destination based on the direction of the sync. Astra Control stops the app running on the destination namespace and removes the Kubernetes app. During the resyncing process, the status shows as "Establishing."
- **Reversing**: The is the planned operation to move the application to the destination cluster while continuing to replicate back to the original source cluster. Astra Control stops the application on the source cluster, replicates the data to the destination before failing over the app to the destination cluster. During the reverse replication, the status shows as "Establishing."
- **Deleting**:
  - If the replication relationship was established but not failed over yet, Astra Control removes PVCs that were created during replication and deletes the destination managed app.
  - If the replication failed over already, Astra Control retains the PVCs and destination app.

Clone and migrate apps

You can clone an existing app to create a duplicate app on the same Kubernetes cluster or on another cluster. When Astra Control 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 Control Center UI or Astra Control API to clone and migrate apps.

Before you begin

- **Check destination volumes**: If you clone to a different storage class, ensure that the storage class uses

the same persistent volume access mode (for example, ReadWriteMany). The clone operation will fail if the destination persistent volume access mode is different. For example, if your source persistent volume uses the RWX access mode, selecting a destination storage class that is not able to provide RWX, such as Azure Managed Disks, AWS EBS, Google Persistent Disk or `ontap-san`, will cause the clone operation to fail. For more information about persistent volume access modes, refer to the Kubernetes documentation.

- To clone apps to a different cluster, you need to make sure the cloud instances containing the source and destination clusters (if they are not the same) have a default bucket. You'll need to assign a default bucket for each cloud instance.
- During clone operations, apps that need an IngressClass resource or webhooks to function properly must not have those resources already defined on the destination cluster.

> During app cloning in OpenShift environments, Astra Control Center needs to allow OpenShift to mount volumes and change the ownership of files. Because of this, you need to configure an ONTAP volume export policy to allow these operations. You can do so with the following commands:
>
> 1. `export-policy rule modify -vserver <storage virtual machine name> -policyname <policy name> -ruleindex 1 -superuser sys`
> 2. `export-policy rule modify -vserver <storage virtual machine name> -policyname <policy name> -ruleindex 1 -anon 65534`

Clone limitations

- **Explicit storage classes**: If you deploy an app with a storage class explicitly set and you need to clone the app, the target cluster must have the originally specified storage class. Cloning an application with an explicitly set storage class to a cluster that does not have the same storage class will fail.
- **ontap-nas-economy-backed applications**: You can't use clone operations if your application's storage class is backed by the `ontap-nas-economy` driver. You can, however, enable backup and restore for ontap-nas-economy operations.
- **Clones and user constraints**: 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 clone or restore operation creates a new namespace, 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.
- **Clones use default buckets**: 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.
- **With Jenkins CI**: 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.
- **With S3 buckets**: 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.
- **With a specific version of PostgreSQL**: App clones within the same cluster consistently fail with the Bitnami PostgreSQL 11.5.0 chart. To clone successfully, use an earlier or later version of the chart.

OpenShift considerations

- **Clusters and OpenShift versions**: 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.

- **Projects and UIDs**: 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. Specify details for the clone:
   - Enter a name.
   - Choose a destination cluster for the clone.
   - Enter destination namespaces for the clone. Each source namespace associated with the app maps to the destination namespace you define.

     > Astra Control creates new destination namespaces as part of the clone operation. Destination namespaces that you specify must not be already present on the destination cluster.

   - Select **Next**.
   - Choose to keep the original storage class associated with the app or select a different storage class.

     > You can migrate an app's storage class to a native cloud provider storage class or other supported storage class, migrate an app from a storage class backed by `ontap-nas-economy` to a storage class backed by `ontap-nas` on the same cluster, or copy the app to another cluster with a storage class backed by the `ontap-nas-economy` driver.

     > If you select a different storage class and this storage class doesn't exist at the moment of restore, an error will be returned.

5. Select **Next**.
6. Review the information about the clone and select **Clone**.

Result

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

After a clone or restore operation creates a new namespace, 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.

> After a data protection operation (clone, backup, or 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 action that you can configure to run in conjunction with a data protection operation of a managed app. For example, if you have a database app, you can use an execution hook to pause all database transactions before a snapshot, and resume transactions after the snapshot is complete. This ensures application-consistent snapshots.

Types of execution hooks

Astra Control Center supports the following types of execution hooks, based on when they can be run:

- Pre-snapshot
- Post-snapshot
- Pre-backup
- Post-backup
- Post-restore
- Post-failover

Execution hook filters

When you add or edit an execution hook to an application, you can add filters to an execution hook to manage which containers the hook will match. Filters are useful for applications that use the same container image on all containers, but might use each image for a different purpose (such as Elasticsearch). Filters allow you to create scenarios where execution hooks run on some but not necessarily all identical containers. If you create multiple filters for a single execution hook, they are combined with a logical AND operator. You can have up to 10 active filters per execution hook.

Each filter you add to an execution hook uses a regular expression to match containers in your cluster. When a hook matches a container, the hook will run its associated script on that container. Regular expressions for filters use the Regular Expression 2 (RE2) syntax, which does not support creating a filter that excludes containers from the list of matches. For information on the syntax that Astra Control supports for regular expressions in execution hook filters, see Regular Expression 2 (RE2) syntax support.

> If you add a namespace filter to an execution hook that runs after a restore or clone operation and the restore or clone source and destination are in different namespaces, the namespace filter is only applied to the destination namespace.

Important notes about custom execution hooks

Consider the following when planning execution hooks for your apps.

> Because 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.
> If you start a backup or snapshot operation with associated execution hooks but then cancel it, the hooks are still allowed to run if the backup or snapshot operation has already begun. This means that the logic used in a post-backup execution hook cannot assume that the backup was completed.

- The execution hooks feature is disabled by default for new Astra Control deployments.
  - You need to enable the execution hooks feature before you can use execution hooks.
  - Owner or Admin users can enable or disable the execution hooks feature for all users defined in the current Astra Control account. Refer to Enable the execution hooks feature and Disable the execution hooks feature for instructions.
  - The feature enablement status is preserved during Astra Control upgrades.
- An execution hook must use a script to perform actions. Many execution hooks can reference the same script.
- Astra Control requires the scripts that execution hooks use to be written in the format of executable shell scripts.
- Script size is limited to 96KB.
- Astra Control uses execution hook settings and any matching criteria to determine which hooks are applicable to a snapshot, backup, or restore operation.
- All execution hook failures are soft failures; other hooks and the data protection operation 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.
- For on-demand data protection operations, all hook events are generated and saved in the **Activity** page event log. However, for scheduled data protection operations, only hook failure events are recorded in the event log (events generated by the scheduled data protection operations themselves are still recorded).
- If Astra Control Center fails over a replicated source app to the destination app, any post-failover execution hooks that are enabled for the source app are run for the destination app after the failover is complete.

> If you have been running post-restore hooks with Astra Control Center 23.04 and upgraded your Astra Control Center to 23.07 or later, post-restore execution hooks will no longer be executed after a failover replication. You need to create new post-failover execution hooks for your apps. Alternatively, you can change the operation type of existing post-restore hooks intended for failovers from "post-restore" to "post-failover".

Order of execution

When a data protection operation is run, execution hook events take place in the following order:

1. Any applicable custom pre-operation execution hooks are run on the appropriate containers. You can create and run as many custom pre-operation hooks as you need, but the order of execution of these hooks before the operation is neither guaranteed nor configurable.

2. The data protection operation is performed.
3. Any applicable custom post-operation execution hooks are run on the appropriate containers. You can create and run as many custom post-operation hooks as you need, but the order of execution of these hooks after the operation is neither guaranteed nor configurable.

If you create multiple execution hooks of the same type (for example, pre-snapshot), the order of execution of those hooks is not guaranteed. However, the order of execution of hooks of different types is guaranteed. For example, the order of execution of a configuration that has all of the different types of hooks would look like this:

1. Pre-backup hooks executed
2. Pre-snapshot hooks executed
3. Post-snapshot hooks executed
4. Post-backup hooks executed
5. Post-restore hooks executed

You can see an example of this configuration in scenario number 2 from the table in Determine whether a hook will run.

> 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 and backups to ensure they are consistent. You can do this by cloning the app to a temporary namespace, restoring the snapshot or backup, and then testing the app.

Determine whether a hook will run

Use the following table to help determine if a custom execution hook will run for your app.

Note that all high-level app operations consist of running one of the basic operations of snapshot, backup, or restore. Depending on the scenario, a clone operation can consists of various combinations of these operations, so what execution hooks a clone operation runs will vary.

In-place restore operations require an existing snapshot or backup, so these operations don’t run snapshot or backup hooks.

> If you start but then cancel a backup that includes a snapshot and there are associated execution hooks, some hooks might run, and others might not. This means that a post-backup execution hook cannot assume that the backup was completed. Keep in mind the following points for cancelled backups with associated execution hooks:
>
> - The pre-backup and post-backup hooks are always run.
> - If the backup includes a new snapshot and the snapshot has started, the pre-snapshot and post-snapshot hooks are run.
> - If the backup is cancelled prior to the snapshot starting, the pre-snapshot and post-snapshot hooks are not run.

Scenario	Operation	Existing snapshot	Existing backup	Namespace	Cluster	Snapshot hooks run	Backup hooks run	Restore hooks run	Failover hooks run
1	Clone	N	N	New	Same	Y	N	Y	N
2	Clone	N	N	New	Different	Y	Y	Y	N
3	Clone or restore	Y	N	New	Same	N	N	Y	N
4	Clone or restore	N	Y	New	Same	N	N	Y	N
5	Clone or restore	Y	N	New	Different	N	N	Y	N
6	Clone or restore	N	Y	New	Different	N	N	Y	N
7	Restore	Y	N	Existing	Same	N	N	Y	N
8	Restore	N	Y	Existing	Same	N	N	Y	N
9	Snapshot	N/A	N/A	N/A	N/A	Y	N/A	N/A	N
10	Backup	N	N/A	N/A	N/A	Y	Y	N/A	N
11	Backup	Y	N/A	N/A	N/A	N	N	N/A	N
12	Failover	Y	N/A	Created by replication	Different	N	N	N	Y
13	Failover	Y	N/A	Created by replication	Same	N	N	N	Y

Execution hook examples

Visit the NetApp Verda GitHub project to download real execution hooks for popular apps such as Apache Cassandra and Elasticsearch. You can also see examples and get ideas for structuring your own custom execution hooks.

Enable the execution hooks feature

If you are an Owner or Admin user, you can enable the execution hooks feature. When you enable the feature, all users defined in this Astra Control account can use execution hooks and view existing execution hooks and hook scripts.

Steps

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

   The **Account** > **Feature settings** tab appears.

4. In the **Execution hooks** pane, select the settings menu.
5. Select **Enable**.
6. Note the security warning that appears.
7. Select **Yes, enable execution hooks**.

Disable the execution hooks feature

If you are an Owner or Admin user, you can disable the execution hooks feature for all users defined in this Astra Control account. You must delete all existing execution hooks before you can disable the execution hooks feature. Refer to Delete an execution hook for instructions on deleting an existing execution hook.

Steps

1. Go to **Account** and then select the **Feature settings** tab.
2. Select the **Execution hooks** tab.
3. In the **Execution hooks** pane, select the settings menu.
4. Select **Disable**.
5. Note the warning that appears.
6. Type `disable` to confirm that you want to disable the feature for all users.
7. Select **Yes, disable**.

View existing execution hooks

You can view existing custom 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, how many containers it matches, creation time, and when it runs (pre- or post-operation). You can select the + icon next to the hook name to expand the list of containers it will run on. To view event logs surrounding execution hooks for this application, go to the **Activity** tab.

View existing scripts

You can view the existing uploaded scripts. You can also see which scripts are in use, and what hooks are using them, on this page.

Steps

1. Go to **Account**.
2. Select the **Scripts** tab.

   You can see a list of existing uploaded scripts on this page. The **Used by** column shows which execution hooks are using each script.

Add a script

Each execution hook must use a script to perform actions. You can add one or more scripts that execution hooks can reference. Many execution hooks can reference the same script; this allows you to update many execution hooks by only changing one script.

Steps

1. Ensure that the execution hooks feature is enabled.
2. Go to **Account**.
3. Select the **Scripts** tab.
4. Select **Add**.
5. 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.
     e. Select **Save script**.
   - Paste in a custom script from the clipboard.
     a. Select the **Paste or type** 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.
6. Select **Save script**.

Result

The new script appears in the list on the **Scripts** tab.

Delete a script

You can remove a script from the system if it is no longer needed and not used by any execution hooks.

Steps

1. Go to **Account**.
2. Select the **Scripts** tab.
3. Choose a script you want to remove, and select the menu in the **Actions** column.
4. Select **Delete**.

> If the script is associated with one or more execution hooks, the **Delete** action is unavailable. To delete the script, first edit the associated execution hooks and associate them with a different script.

Create a custom execution hook

You can create a custom execution hook for an app and add it to Astra Control. Refer to 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 specific commands or providing the full path to an executable.

Steps

1. Ensure that the execution hooks feature is enabled.
2. Select **Applications** and then select the name of a managed app.
3. Select the **Execution hooks** tab.
4. Select **Add**.
5. In the **Hook Details** area:
   a. Determine when the hook should run by selecting an operation type from the **Operation** drop-down menu.
   b. Enter a unique name for the hook.
   c. (Optional) Enter any arguments to pass to the hook during execution, pressing the Enter key after each argument you enter to record each one.
6. (Optional) In the **Hook Filter Details** area, you can add filters to control which containers the execution hook runs on:
   a. Select **Add filter**.
   b. In the **Hook filter type** column, choose an attribute on which to filter from the drop-down menu.
   c. In the **Regex** column, enter a regular expression to use as the filter. Astra Control uses the Regular Expression 2 (RE2) regex syntax.

      > If you filter on the exact name of an attribute (such as a pod name) with no other text in the regular expression field, a substring match is performed. To match an exact name and only that name, use the exact string match syntax (for example, `^exact_podname$`).

   d. To add more filters, select **Add filter**.

      > Multiple filters for an execution hook are combined with a logical AND operator. You can have up to 10 active filters per execution hook.

7. When done, select **Next**.
8. In the **Script** area, do one of the following:
   - Add a new script.
     a. Select **Add**.
     b. Do one of the following:
       - Upload a custom script.
         i. Select the **Upload file** option.

        ii. Browse to a file and upload it.
        iii. Give the script a unique name.
        iv. (Optional) Enter any notes other administrators should know about the script.
        v. Select **Save script**.
    - Paste in a custom script from the clipboard.
        i. Select the **Paste or type** option.
        ii. Select the text field and paste the script text into the field.
        iii. Give the script a unique name.
        iv. (Optional) Enter any notes other administrators should know about the script.
  - Select an existing script from the list.

    This instructs the execution hook to use this script.

9. Select **Next**.
10. Review the execution hook configuration.
11. Select **Add**.

Check the state of an execution hook

After a snapshot, backup, or restore operation finishes running, you can check the state of execution hooks that ran as part of the operation. You can use this status information to determine if you want to keep the execution hook, modify it, or delete it.

Steps

1. Select **Applications** and then select the name of a managed app.
2. Select the **Data protection** tab.
3. Select **Snapshots** to see running snapshots, or **Backups** to see running backups.

   The **Hook state** shows the status of the execution hook run after the operation is complete. You can hover over the state for more details. For example, if there are execution hook failures during a snapshot, hovering over the hook state for that snapshot gives a list of failed execution hooks. To see reasons for each failure, you can check the **Activity** page in the left-side navigation area.

View script usage

You can see which execution hooks use a particular script in the Astra Control web UI.

Steps

1. Select **Account**.
2. Select the **Scripts** tab.

   The **Used by** column in the list of scripts contains details on which hooks are using each script in the list.

3. Select the information in the **Used by** column for a script you are interested in.

   A more detailed list appears, with the names of hooks that are using the script and the type of operation they are configured to run with.

Edit an execution hook

You can edit an execution hook if you want to change its attributes, filters, or the script that it uses. You need to have Owner, Admin, or Member permissions to edit 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 edit.
4. Select **Edit**.
5. Make any needed changes, selecting **Next** after you complete each section.
6. Select **Save**.

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**.
5. In the resulting dialog, type "delete" to confirm.
6. Select **Yes, delete execution hook**.

For more information

- NetApp Verda GitHub project

Protect Astra Control Center using Astra Control Center

To better ensure resiliency against fatal errors on the Kubernetes cluster where Astra Control Center is running, protect the Astra Control Center application itself. You can backup and restore Astra Control Center using a secondary Astra Control Center instance or use Astra replication if the underlying storage is using ONTAP.

In these scenarios, a second instance of Astra Control Center is deployed and configured in a different fault domain and runs on a different second Kubernetes cluster than the primary Astra Control Center instance. The second Astra Control instance is used to back up and potentially restore the primary Astra Control Center instance. A restored or replicated Astra Control Center instance will continue to provide application data management for the application cluster applications and restore accessibility to backups and snapshots of those applications.

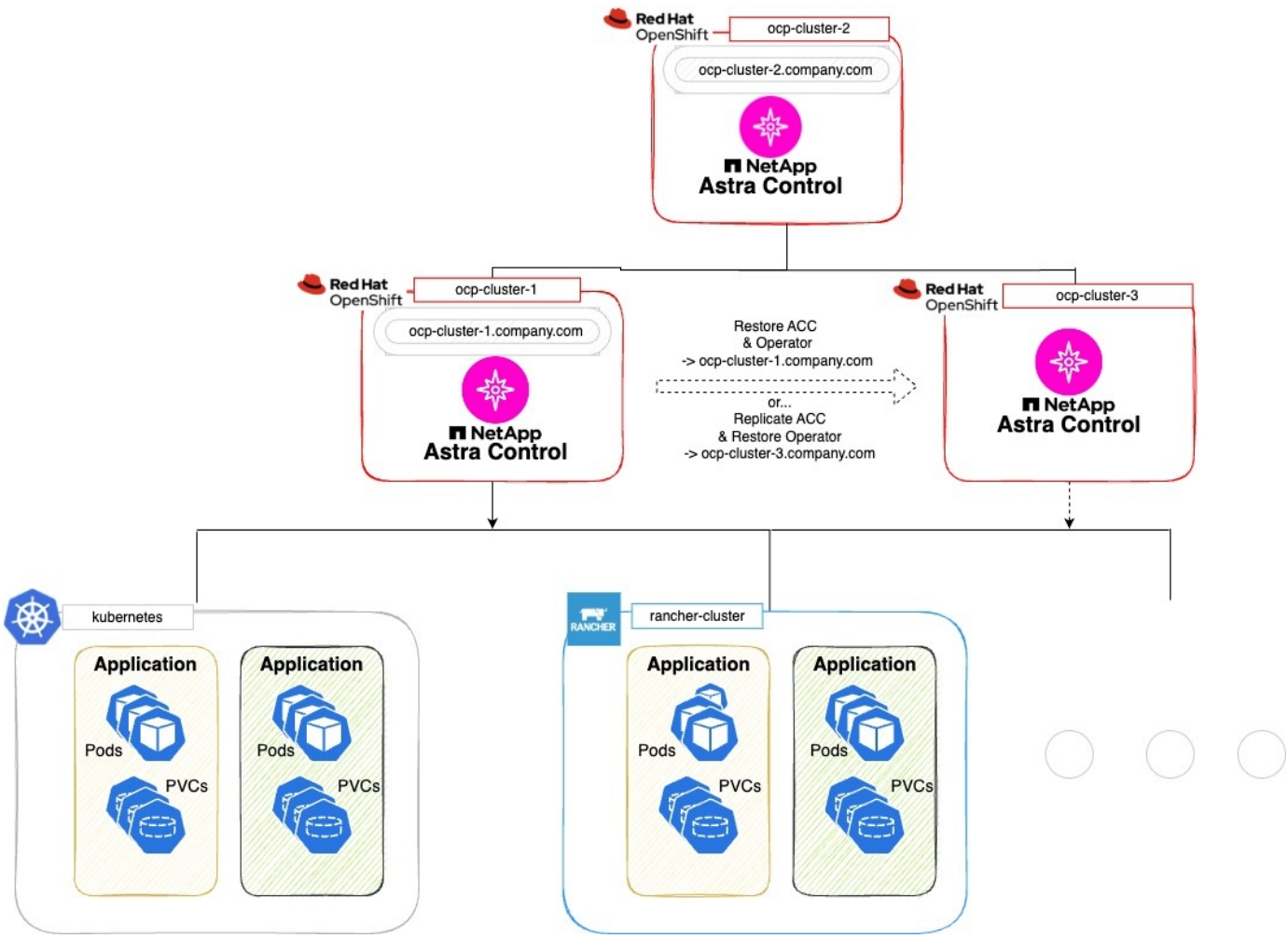
Before you begin

Ensure that you have the following before setting up protection scenarios for Astra Control Center:

- **A Kubernetes cluster running the primary Astra Control Center instance**: This cluster hosts the primary Astra Control Center instance which manages application clusters.
- **A second Kubernetes cluster of the same Kubernetes distribution type as the primary that is running the secondary Astra Control Center instance**: This cluster hosts the Astra Control Center instance that manages the primary Astra Control Center instance.
- **A third Kubernetes cluster of the same Kubernetes distribution type as the primary**: This cluster will host the restored or replicated instance of Astra Control Center. It must have the same Astra Control Center namespace available that is currently deployed on the primary. For example, if Astra Control Center is deployed in namespace `netapp-acc` on the source cluster, the namespace `netapp-acc` must be available and not used by any applications on the destination Kubernetes cluster.
- **S3-compatible buckets**: Each Astra Control Center instance has an accessible S3-compatible object storage bucket.
- **A configured load balancer**: The load balancer provides an IP address for Astra and must have network connectivity to the application clusters and both S3 buckets.
- **Clusters meet Astra Control Center requirements**: Each cluster used in Astra Control Center protection meets general Astra Control Center requirements.

About this task

These procedures describe the necessary steps to restore Astra Control Center to a new cluster using either backup and restore or replication. Steps are based on the example configuration depicted here:



In this example configuration, the following is shown:

- **A Kubernetes cluster running the primary Astra Control Center instance**:
  - OpenShift cluster: `ocp-cluster-1`
  - Astra Control Center primary instance: `ocp-cluster-1.company.com`
  - This cluster manages the application clusters.
- **The second Kubernetes cluster of the same Kubernetes distribution type as the primary that is running the secondary Astra Control Center instance**:
  - OpenShift cluster: `ocp-cluster-2`
  - Astra Control Center secondary instance: `ocp-cluster-2.company.com`
  - This cluster will be used to back up the primary Astra Control Center instance or configure replication to a different cluster (in this example, the `ocp-cluster-3` cluster).
- **A third Kubernetes cluster of the same Kubernetes distribution type as the primary that will be used for restore operations**:
  - OpenShift cluster: `ocp-cluster-3`
  - Astra Control Center third instance: `ocp-cluster-3.company.com`
  - This cluster will be used for Astra Control Center restore or replication failover.

> Ideally, the application cluster should be situated outside of the three Astra Control Center clusters as depicted by the kubernetes and rancher clusters in the image above.

Not depicted in the diagram:

- All the clusters have ONTAP backends with Astra Trident or Astra Control Provisioner installed.
- In this configuration, the OpenShift clusters are using MetalLB as the load balancer.
- The snapshot controller and VolumeSnapshotClass are also installed on all the clusters as outlined in the prerequisites.

Step 1 option: Backup and restore Astra Control Center

This procedure describes the necessary steps to restore Astra Control Center to a new cluster using backup and restore.

In this example, Astra Control Center is always installed under the `netapp-acc` namespace and the operator is installed under the `netapp-acc-operator` namespace.

> Although not described, Astra Control Center operator can also be deployed in the same namespace as the Astra CR.

Before you begin

- You have installed the primary Astra Control Center on a cluster.
- You have installed the secondary Astra Control Center on a different cluster.

Steps

1. Manage the primary Astra Control Center application and destination cluster from the secondary Astra Control Center instance (running on `ocp-cluster-2` cluster):
   a. Log into the secondary Astra Control Center instance.
   b. Add the primary Astra Control Center cluster (`ocp-cluster-1`).
   c. Add the destination third cluster (`ocp-cluster-3`) that will be used for the restore.
2. Manage Astra Control Center and the Astra Control Center operator on the secondary Astra Control Center:
   a. From the Applications page, select **Define**.
   b. In the **Define application** window, enter the new application name (`netapp-acc`).
   c. Choose the cluster that is running the primary Astra Control Center (`ocp-cluster-1`) from the **Cluster** drop-down list.
   d. Choose the `netapp-acc` namespace for Astra Control Center from the **Namespace** drop-down list.
   e. On the Cluster Resources page, check **Include additional cluster-scoped resources**.
   f. Select **Add include rule**.
   g. Select these entries, and select **Add**:
      - Label selector: <label name>
      - Group: apiextensions.k8s.io
      - Version: v1

- Kind: CustomResourceDefinition

h. Confirm the application information.

i. Select **Define**.

After you select **Define**, repeat the Define Application process for the operator (`netapp-acc-operator`) and select the `netapp-acc-operator` namespace in the Define Application wizard.

3. Back up Astra Control Center and the operator:

   a. On the secondary Astra Control Center, navigate to the Applications page by selecting the Applications tab.

   b. Back up the Astra Control Center application (`netapp-acc`).

   c. Back up the operator (`netapp-acc-operator`).

4. After you have backed up Astra Control Center and the operator, simulate a disaster recovery (DR) scenario by uninstalling Astra Control Center from the primary cluster.

> You'll restore Astra Control Center to a new cluster (the third Kubernetes cluster described in this procedure) and use the same DNS as the primary cluster for the newly installed Astra Control Center.

5. Using the secondary Astra Control Center, restore the primary instance of the Astra Control Center application from its backup:

   a. Select **Applications** and then select the name of the Astra Control Center application.

   b. From the Options menu in the Actions column, select **Restore**.

   c. Choose the **Restore to new namespaces** as the restore type.

   d. Enter the restore name (`netapp-acc`).

   e. Choose the destination third cluster (`ocp-cluster-3`).

   f. Update the destination namespace so that it is the same namespace as the original.

   g. On the Restore Source page, select the application backup that will be used as the restore source.

   h. Select **Restore using original storage classes**.

   i. Select **Restore all resources**.

   j. Review restore information, and then select **Restore** to start the restore process that restores Astra Control Center to the destination cluster (`ocp-cluster-3`). The restore is complete when the application enters `available` state.

6. Configure Astra Control Center on the destination cluster:

   a. Open a terminal and connect using kubeconfig to the destination cluster (`ocp-cluster-3`) that contains the restored Astra Control Center.

   b. Confirm that the `ADDRESS` column in the Astra Control Center configuration references the primary system's DNS name:

```
kubectl get acc -n netapp-acc
```

   Response:

```
NAME  UUID                                   VERSION    ADDRESS
READY
astra 89f4fd47-0cf0-4c7a-a44e-43353dc96ba8 24.02.0-69 ocp-cluster-
1.company.com           True
```

c. If the `ADDRESS` field in the above response does not have the FQDN of the primary Astra Control Center instance, update the configuration to reference the Astra Control Center DNS:

```
kubectl edit acc -n netapp-acc
```

i. Change the `astraAddress` under `spec:` to the FQDN (`ocp-cluster-1.company.com` in this example) of the primary Astra Control Center instance.

ii. Save the configuration.

iii. Confirm that the address has been updated:

```
kubectl get acc -n netapp-acc
```

d. Go to the Restore the Astra Control Center Operator section of this document to complete the restore process.

Step 1 option: Protect Astra Control Center using Replication

This procedure describes the necessary steps to configure Astra Control Center replication to protect the primary Astra Control Center instance.

In this example, Astra Control Center is always installed under the `netapp-acc` namespace and the operator is installed under the `netapp-acc-operator` namespace.

Before you begin

- You have installed the primary Astra Control Center on a cluster.
- You have installed the secondary Astra Control Center on a different cluster.

Steps

1. Manage the primary Astra Control Center application and destination cluster from the secondary Astra Control Center instance:
   a. Log into the secondary Astra Control Center instance.
   b. Add the primary Astra Control Center cluster (`ocp-cluster-1`).
   c. Add the destination third cluster (`ocp-cluster-3`) that will be used for the replication.
2. Manage Astra Control Center and the Astra Control Center operator on the secondary Astra Control Center:
   a. Select **Clusters** and select the cluster that contains the primary Astra Control Center (`ocp-cluster-1`).
   b. Select the **Namespaces** tab.

c. Select `netapp-acc` and `netapp-acc-operator` namespaces.

d. Select the Actions menu and select **Define as applications**.

e. Select **View in applications** to see the defined applications.

3. Configure Backends for Replication:

> Replication requires that the primary Astra Control Center cluster and the destination cluster (`ocp-cluster-3`) use different peered ONTAP storage backends.
> After each backend is peered and added to Astra Control, the backend appears in the **Discovered** tab of the Backends page.

a. Add a peered backend to Astra Control Center on the primary cluster.

b. Add a peered backend to Astra Control Center on the destination cluster.

4. Configure replication:

a. On the Applications screen, select the `netapp-acc` application.

b. Select **Configure replication policy**.

c. Select `ocp-cluster-3` as the destination cluster.

d. Select the storage class.

e. Enter `netapp-acc` as the destination namespace.

f. Change the replication frequency if desired.

g. Select **Next**.

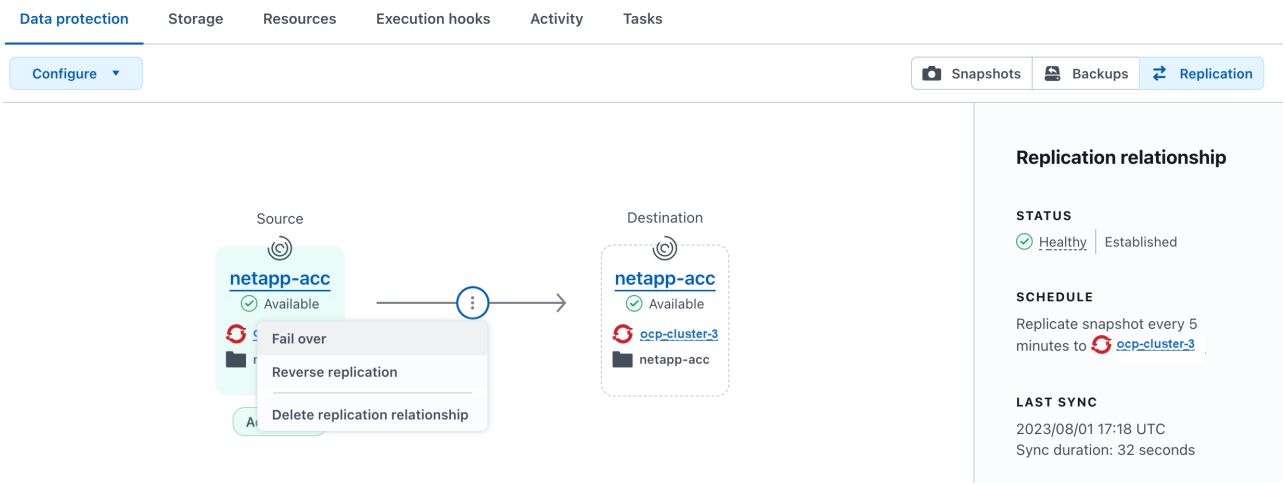
h. Confirm the configuration is correct, and select **Save**.

The replication relationship transitions from `Establishing` to `Established`. When active, this replication will occur every five minutes until the replication configuration is deleted.

5. Failover the replication to the other cluster if the primary system is corrupted or no longer accessible:

> Make sure the destination cluster does not have Astra Control Center installed to ensure a successful failover.

a. Select the vertical ellipses icon and select **Fail over**.



b. Confirm the details and select **Fail over** to begin the failover process.

   The replication relationship status changes to `Failing over` and then `Failed over` when complete.

6. Complete the failover configuration:

   a. Open a terminal and connect using the third cluster's kubeconfig (`ocp-cluster-3`). This cluster now has Astra Control Center installed.

   b. Determine the Astra Control Center FQDN on the third cluster (`ocp-cluster-3`).

   c. Update the configuration to reference the Astra Control Center DNS:

```
kubectl edit acc -n netapp-acc
```

      i. Change the `astraAddress` under `spec:` with the FQDN (`ocp-cluster-3.company.com`) of the destination third cluster.

      ii. Save the configuration.

      iii. Confirm that the address has been updated:

```
kubectl get acc -n netapp-acc
```

   d. Confirm that all required traefik CRDs are present:

```
kubectl get crds | grep traefik
```

   Required traefik CRDS:

```
ingressroutes.traefik.containo.us
ingressroutes.traefik.io
ingressroutetcps.traefik.containo.us
ingressroutetcps.traefik.io
ingressrouteudps.traefik.containo.us
ingressrouteudps.traefik.io
middlewares.traefik.containo.us
middlewares.traefik.io
middlewaretcps.traefik.containo.us
middlewaretcps.traefik.io
serverstransports.traefik.containo.us
serverstransports.traefik.io
tlsoptions.traefik.containo.us
tlsoptions.traefik.io
tIsstores.traefik.containo.us
tIsstores.traefik.io
traefikservices.traefik.containo.us
traefikservices.traefik.io
```

e. If some of the above CRDs are missing:
   i. Go to traefik documentation.
   ii. Copy the "Definitions" area into a file.
   iii. Apply changes:

```
kubectl apply -f <file name>
```

   iv. Restart traefik:

```
kubectl get pods -n netapp-acc | grep -e "traefik" | awk '{print
$1}' | xargs kubectl delete pod -n netapp-acc
```

f. Go to the Restore the Astra Control Center Operator section of this document to complete the restore process.

Step 2: Restore the Astra Control Center Operator

Using the secondary Astra Control Center, restore the primary Astra Control Center operator from backup. The destination namespace must be the same as the source namespace. In the case where Astra Control Center was deleted from the primary source cluster, backups will still exist to perform the same restore steps.

Steps

1. Select **Applications** and then select the name of the operator app (`netapp-acc-operator`).
2. From the Options menu in the Actions column, select **Restore**

3. Choose the **Restore to new namespaces** as the restore type.
4. Choose the destination third cluster (`ocp-cluster-3`).
5. Change the namespace to be the same as the namespace associated with the primary source cluster (`netapp-acc-operator`).
6. Select the backup that was taken earlier as the restore source.
7. Select **Restore using original storage classes**.
8. Select **Restore all resources**.
9. Review the details then click **Restore** to start the restore process.

   The Applications page shows the Astra Control Center operator being restored to the destination third cluster (`ocp-cluster-3`). When the process is complete, the state shows as `Available`. Within ten minutes, the DNS address should resolve on the page.

Result

Astra Control Center, its registered clusters, and managed applications with their snapshots and backups are now available on the destination third cluster (`ocp-cluster-3`). Any protection policies you had on the original are also there on the new instance. You can continue to take scheduled or on-demand backups and snapshots.

Troubleshooting

Determine system health and if protection processes were successful.

- **Pods are not running**: Confirm that all pods are up and running:

  ```
  kubectl get pods -n netapp-acc
  ```

  If some pods are in the `CrashLookBackOff` state, restart them and they should transition to `Running` state.

- **Confirm system status**: Confirm that the Astra Control Center system is in `ready` state:

  ```
  kubectl get acc -n netapp-acc
  ```

  Response:

  ```
  NAME  UUID                                   VERSION    ADDRESS
  READY
  astra 89f4fd47-0cf0-4c7a-a44e-43353dc96ba8 24.02.0-69 ocp-cluster-
  1.company.com           True
  ```

- **Confirm deployment status**: Show Astra Control Center deployment information to confirm that `Deployment State` is `Deployed`.

```
kubectl describe acc astra -n netapp-acc
```

- **Restored Astra Control Center UI returns a 404 error**: If this happens when you have selected `AccTraefik` as an ingress option, check the traefik CRDs to ensure they're all installed.

Monitor 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 cluster health and manage storage classes

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. You can also change the default storage class for managed clusters.

View cluster health and details

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.

Change the default storage class

You can change the default storage class for a cluster. When Astra Control manages a cluster, it keeps track of the cluster's default storage class.

> Do not change the storage class using kubectl commands. Use this procedure instead. Astra Control will revert the changes if made using kubectl.

Steps

1. In the Astra Control Center web UI, select **Clusters**.
2. On the **Clusters** page, select the cluster that you want to change.
3. Select the **Storage** tab.
4. Select the **Storage classes** category.
5. Select the **Actions** menu for the storage class that you want to set as default.
6. Select **Set as default**.

View the health and details of an app

After you start managing an app, Astra Control provides details about the app that enables you to identify its communication status (whether Astra Control can communicate

with the app), 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. Review the information.

   App Status

   Provides a status that reflects whether Astra Control can communicate with the application.

   - **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 local users and roles

You can add, remove, and edit users of your Astra Control Center installation using the Astra Control UI. You can use the Astra Control UI or Astra Control API to manage users.

You can also use LDAP to perform authentication for selected users.

Use LDAP

LDAP is an industry standard protocol for accessing distributed directory information and a popular choice for

enterprise authentication. You can connect Astra Control Center to an LDAP server to perform authentication for selected Astra Control users. At a high level, the configuration involves integrating Astra with LDAP and defining the Astra Control users and groups corresponding to the LDAP definitions. You can use the Astra Control API or web UI to configure LDAP authentication and LDAP users and groups. See the following documentation for more information:

- Use the Astra Control API to manage remote authentication and users
- Use the Astra Control UI to manage remote users and groups
- Use the Astra Control UI to manage remote authentication

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, refer to Manage local users and 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**.

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 Astra Control API to manage roles.

Add a namespace constraint to a role

An Admin or Owner user can add namespace constraints to Member or Viewer roles.

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

Manage remote authentication

LDAP is an industry standard protocol for accessing distributed directory information and a popular choice for enterprise authentication. You can connect Astra Control Center to an LDAP server to perform authentication for selected Astra Control users.

At a high level, the configuration involves integrating Astra with LDAP and defining the Astra Control users and groups corresponding to the LDAP definitions. You can use the Astra Control API or web UI to configure LDAP authentication and LDAP users and groups.

> Astra Control Center uses the user login attribute, configured when remote authentication is enabled, to search for and keep track of remote users. An attribute of an email address ("mail") or user principal name ("userPrincipalName") must exist in this field for any remote user you wish to appear in Astra Control Center. This attribute is used as the username in Astra Control Center for authentication and in searches for remote users.

Add a certificate for LDAPS authentication

Add the private TLS certificate for the LDAP server so that Astra Control Center can authenticate with the LDAP server when you use an LDAPS connection. You only need to do this once, or when the certificate you have installed expires.

Steps

1. Go to **Account**.
2. Select the **Certificates** tab.
3. Select **Add**.
4. Either upload the `.pem` file or paste the contents of the file from your clipboard.
5. Select the **Trusted** check box.
6. Select **Add certificate**.

Enable remote authentication

You can enable LDAP authentication and configure the connection between Astra Control and the remote LDAP server.

Before you begin

If you plan to use LDAPS, ensure that the private TLS certificate for the LDAP server is installed in Astra Control Center so that Astra Control Center can authenticate with the LDAP server. See Add a certificate for LDAPS authentication for instructions.

Steps

1. Go to **Account > Connections**.
2. In the **Remote Authentication** pane, select the configuration menu.
3. Select **Connect**.
4. Enter the server IP address, port, and preferred connection protocol (LDAP or LDAPS).

> As a best practice, use LDAPS when connecting with the LDAP server. You need to install the LDAP server's private TLS certificate in Astra Control Center before you connect with LDAPS.

5. Enter the service account credentials in email format (administrator@example.com). Astra Control will use these credentials when connecting with the LDAP server.
6. In the **User Match** section, do the following:
   a. Enter the base DN and an appropriate user search filter to use when retrieving user information from the LDAP server.
   b. (Optional) If your directory uses the user login attribute `userPrincipalName` instead of `mail`, enter `userPrincipalName` in the correct attribute in the **User login attribute** field.
7. In the **Group Match** section, enter the group search base DN and an appropriate custom group search filter.

> Be sure to use the correct base Distinguished Name (DN) and an appropriate search filter for **User Match** and **Group Match**. The base DN tells Astra Control at what level of the directory tree to start the search, and the search filter limits the parts of the directory tree Astra Control searches from.

8. Select **Submit**.

Result

The **Remote Authentication** pane status moves to **Pending**, and then to **Connected** when the connection to the LDAP server is established.

Disable remote authentication

You can temporarily disable an active connection to the LDAP server.

> When you disable a connection to an LDAP server, all settings are saved, and all remote users and groups that were added to Astra Control from that LDAP server are retained. You can reconnect to this LDAP server at any time.

Steps

1. Go to **Account > Connections**.
2. In the **Remote Authentication** pane, select the configuration menu.
3. Select **Disable**.

Result

The **Remote Authentication** pane status moves to **Disabled**. All remote authentication settings, remote users, and remote groups are preserved, and you can re-enable the connection at any time.

Edit remote authentication settings

If you have disabled the connection to the LDAP server or the **Remote Authentication** pane is in a "Connection error" state, you can edit the configuration settings.

> You cannot edit the LDAP server URL or IP address when the **Remote Authentication** pane is in a "Disabled" state. You need to Disconnect remote authentication first.

Steps

1. Go to **Account > Connections**.
2. In the **Remote Authentication** pane, select the configuration menu.
3. Select **Edit**.
4. Make the necessary changes, and select **Edit**.

Disconnect remote authentication

You can disconnect from an LDAP server and remove the configuration settings from Astra Control.

> If you are an LDAP user and you disconnect, your session will immediately end. When you disconnect from the LDAP server, all configuration settings for that LDAP server are removed from Astra Control, as well as any remote users and groups that were added from that LDAP server.

Steps

1. Go to **Account > Connections**.
2. In the **Remote Authentication** pane, select the configuration menu.
3. Select **Disconnect**.

Result

The **Remote Authentication** pane status moves to **Disconnected**. Remote authentication settings, remote users, and remote groups are removed from Astra Control.

Manage remote users and groups

If you have enabled LDAP authentication on your Astra Control system, you can search for LDAP users and groups, and include them in the approved users of the system.

Add a remote user

Account Owners and Admins can add remote users to Astra Control. Astra Control Center supports up to 10,000 LDAP remote users.

> Astra Control Center uses the user login attribute, configured when remote authentication is enabled, to search for and keep track of remote users. An attribute of an email address ("mail") or user principal name ("userPrincipalName") must exist in this field for any remote user you wish to appear in Astra Control Center. This attribute is used as the username in Astra Control Center for authentication and in searches for remote users.

> You cannot add a remote user if a local user with the same email address (based on the "mail" or "user principal name" attribute) already exists on the system. To add the user as a remote user, delete the local user from the system first.

Steps

1. Go to the **Account** area.
2. Select the **Users & groups** tab.
3. At the far right of the page, select **Remote users**.

4. Select **Add**.
5. Optionally, search for an LDAP user by entering the user's email address in the **Filter by email** field.
6. Select one or more users from the list.
7. Assign a role to the user.

> If you assign different roles to a user and the user's group, the more permissive role takes precedence.

8. Optionally, assign one or more namespace constraints to this user, and select **Restrict role to constraints** to enforce them. You can add a new namespace constraint by selecting **Add constraint**.

> When a user is assigned multiple roles through LDAP group membership, the constraints in the most permissive role are the only ones that take effect. For example, if a user with a local Viewer role joins three groups that are bound to the Member role, the sum of the constraints from the Member roles take effect, and any constraints from the Viewer role are ignored.

9. Select **Add**.

Result

The new user appears in the list of remote users. In this list, you can see active constraints on the user as well as manage the user from the **Actions** menu.

Add a remote group

To add many remote users at once, account Owners and Admins can add remote groups to Astra Control. When you add a remote group, all remote users in that group are available to log in to Astra Control and will inherit the same role as the group.

Astra Control Center supports up to 5,000 LDAP remote groups.

Steps

1. Go to the **Account** area.
2. Select the **Users & groups** tab.
3. At the far right of the page, select **Remote groups**.
4. Select **Add**.

   In this window, you can see a list of the common names and distinguished names of LDAP groups that Astra Control retrieved from the directory.

5. Optionally, search for an LDAP group by entering the group's common name in the **Filter by common name** field.
6. Select one or more groups from the list.
7. Assign a role to the groups.

> The role you select is assigned to all users in this group. If you assign different roles to a user and the user's group, the more permissive role takes precedence.

8. Optionally, assign one or more namespace constraints to this group, and select **Restrict role to**

constraints to enforce them. You can add a new namespace constraint by selecting **Add constraint**.

- **If the resources being accessed belong to clusters that have the latest Astra Connector installed**: When a user is assigned multiple roles through LDAP group membership, the constraints from the roles are combined. For example, if a user with a local Viewer role joins three groups that are bound to the Member role, the user now has Viewer role access to the original resources as well as Member role access to the resources gained through group membership.
- **If the resources being accessed belong to clusters that do not have Astra Connector installed**: When a user is assigned multiple roles through LDAP group membership, the constraints from the most permissive role are the only ones that take effect.

9. Select **Add**.

Result

The new group appears in the list of remote groups. Remote users in this group don't appear in the list of remote users until each remote user logs in. In this list, you can see details about the group as well as manage the group from the **Actions** menu.

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, refer to Add a Kubernetes cluster.

> If you create your own kubeconfig file, you should define only **one** context element in it. Refer to 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 Control Center UI or 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 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

- Microsoft Azure
- Generic S3

> Amazon Web Services (AWS) and Google Cloud Platform (GCP) use the Generic S3 bucket type.

> 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
- Set the default bucket
- Rotate or remove bucket credentials
- Remove a bucket
- [Tech preview] Manage a bucket using a custom resource

> 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 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**.

Set the default bucket

When you perform a clone across clusters, Astra Control requires a default bucket. Follow these steps to set a default bucket for all clusters.

Steps

1. Go to **Cloud instances**.
2. Select the menu in the **Actions** column for the cloud instance in the list.
3. Select **Edit**.
4. In the **Bucket** list, select the bucket you want to be the default.
5. Select **Save**.

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.
> - You cannot remove a write once read many (WORM) bucket before the bucket's cloud provider retention period has expired. WORM buckets are denoted with "Locked" next to the bucket name.

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

Before you begin

- 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'll 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**.

[Tech preview] Manage a bucket using a custom resource

You can add a bucket using the an Astra Control custom resource (CR) on the application cluster. Adding object store bucket providers is essential if you want to back up your applications and persistent storage or if you want to clone applications across clusters. Astra Control stores those backups or clones in the object store buckets that you define. If you are using the custom resource method, application snapshots functionality requires a bucket.

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

The bucket custom resource for Astra Control is known as an AppVault. This CR contains the configurations necessary for a bucket to be used in protection operations.

Before you begin

- Ensure you have a bucket that is reachable from your clusters managed by Astra Control Center.
- Ensure you have credentials for the bucket.
- Ensure the bucket is one of the following types:

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

> Amazon Web Services (AWS) uses the Generic S3 bucket type.

> 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.

Steps

1. Create the custom resource (CR) file and name it (for example, `astra-appvault.yaml`).
2. Configure the following attributes:
   - **metadata.name**: *(Required)* The name of the AppVault custom resource.
   - **spec.prefix**: *(Optional)* A path that is prefixed to the names of all entities stored in the AppVault.
   - **spec.providerConfig**: *(Required)* Stores the configuration necessary to access the AppVault using the specified provider.
   - **spec.providerCredentials**: *(Required)* Stores references to any credential required to access the AppVault using the specified provider.
     - **spec.providerCredentials.valueFromSecret**: *(Optional)* Indicates that the credential value should come from a secret.
       - **key**: *(Required if valueFromSecret is used)* The valid key of the secret to select from.
       - **name**: *(Required if valueFromSecret is used)* Name of the secret containing the value for this field. Must be in the same namespace.
   - **spec.providerType**: *(Required)* Determines what provides the backup; for example, NetApp ONTAP S3 or Microsoft Azure.

     Example YAML:

```yaml
apiVersion: astra.netapp.io/v1
kind: AppVault
metadata:
  name: astra-appvault
spec:
  providerType: generic-s3
  providerConfig:
    path: testpath
    endpoint: 192.168.1.100:80
    bucketName: bucket1
    secure: "false"
  providerCredentials:
    accessKeyID:
      valueFromSecret:
        name: s3-creds
        key: accessKeyID
    secretAccessKey:
      valueFromSecret:
        name: s3-creds
        key: secretAccessKey
```

3. After you populate the `astra-appvault.yaml` file with the correct values, apply the CR:

```
kubectl apply -f astra-appvault.yaml -n astra-connector
```

> When you add a bucket, Astra Control marks one bucket with the default bucket indicator. The first bucket that you create becomes the default bucket. As you add buckets, you can later decide to set another default 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.

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
- Edit storage backend authentication details
- Manage a discovered storage backend
- Unmanage a storage backend
- Remove a storage backend

View storage backend details

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

View storage backend details from the Dashboard

Steps

1. From the left navigation, select **Dashboard**.
2. Review the Storage backend panel of the Dashboard 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).

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.

Edit storage backend authentication details

Astra Control Center offers two modes of authenticating an ONTAP backend.

- **Credential-based authentication**: The username and password to an ONTAP user with the required permissions. You should use a pre-defined security login role, such as admin to ensure maximum compatibility with ONTAP versions.
- **Certificate-based authentication**: Astra Control Center can also communicate with an ONTAP cluster using a certificate installed on the backend. You should use the client certificate, key, and the trusted CA certificate if used (recommended).

You can update existing backends to move from one type of authentication to another method. Only one authentication method is supported at a time.

For details on enabling certificate-based authentication, refer to Enable authentication on the ONTAP storage backend.

Steps

1. From the left navigation, select **Backends**.

2. Select the storage backend.
3. At the Credentials field, select the **Edit** icon.
4. In the Edit page, select one of the following.
   - **Use administrator credentials**: Enter the ONTAP cluster management IP address and admin credentials. The credentials must be cluster-wide credentials.

     > The user whose credentials you enter here must have the `ontapi` user login access method enabled within ONTAP System Manager on the ONTAP cluster. If you plan to use SnapMirror replication, apply user credentials with the "admin" role, which has the access methods `ontapi` and `http`, on both source and destination ONTAP clusters. Refer to Manage User Accounts in ONTAP documentation for more information.

   - **Use a certificate**: Upload the certificate `.pem` file, the certificate key `.key` file, and optionally the certificate authority file.
5. Select **Save**.

Manage a discovered storage backend

You can select to manage an unmanaged, yet discovered storage backend. When you manage a storage backend, Astra Control indicates if a certificate for authentication has expired.

Steps

1. From the left navigation, select **Backends**.
2. Select the **Discovered** option.
3. Select the storage backend.
4. From the Options menu in the **Actions** column, select **Manage**.
5. Make the changes.
6. Select **Save**.

Unmanage a storage backend

You can unmanage the backend.

Steps

1. From the left navigation, select **Backends**.
2. Select the storage backend.
3. From the Options menu in the **Actions** column, select **Unmanage**.
4. Type "unmanage" to confirm the action.
5. 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.

Before you begin

- Ensure that the storage backend is unmanaged.
- Ensure that the storage backend does not have any volumes associated with the 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**.

Find more information

- Use the Astra Control API

Monitor running tasks

You can view details about running tasks and tasks that have completed, failed, or been cancelled in the last 24 hours in Astra Control. For example, you can view the status of a running backup, restore, or clone operation, and see details like percentage completed and estimated time remaining. You can view the status of a scheduled operation that has run or an operation that you started manually.

While viewing a running or completed task, you can expand the task details to see the status of each of the subtasks. The task progress bar is green for ongoing or completed tasks, blue for cancelled tasks, and red for tasks that failed because of an error.

> For clone operations, the task subtasks consist of a snapshot and a snapshot restore operation.

To see more information about failed tasks, refer to Monitor account activity.

Steps

1. While a task is running, go to **Applications**.
2. Select the name of an application from the list.
3. In the details of the application, select the **Tasks** tab.

   You can view details of current or past tasks, and filter by task state.

Tasks are retained in the **Tasks** list for up to 24 hours. You can configure this limit and other task monitor settings using the Astra Control API.

[Tech preview] Manage Astra Control applications using CRs

Manage your Astra Control applications using Kubernetes custom resources (CR). The following options are available:

- Define an application using a Kubernetes custom resource
- Manage a bucket using a custom resource

Monitor infrastructure with Prometheus or Fluentd connections

You can configure several optional settings to enhance your Astra Control Center experience. To monitor and gain insight into your complete infrastructure, configure Prometheus or add a Fluentd connection.

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

- Connect to Prometheus
- Connect to Fluentd

Add a proxy server for connections to the NetApp Support Site

If the network where you're running Astra Control Center requires a proxy for connecting to the Internet (to upload support bundles to the NetApp Support Site), 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 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.

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.

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'll 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 Prometheus

You can monitor Astra Control Center data with Prometheus. You can configure Prometheus to gather metrics from the Kubernetes cluster metrics endpoint, and you can use Prometheus also to visualize the metrics data.

For details about using Prometheus, refer to their documentation at Getting started with Prometheus.

What you'll need

Make sure that you have downloaded and installed the Prometheus package on the Astra Control Center

cluster or a different cluster that can communicate with the Astra Control Center cluster.

Follow the instructions in the official documentation to Install Prometheus.

Prometheus needs to be able to communicate with the Astra Control Center Kubernetes cluster. If Prometheus is not installed on the Astra Control Center cluster, you need to make sure they can communicate with the metrics service running on the Astra Control Center cluster.

Configure Prometheus

Astra Control Center exposes a metrics service on TCP port 9090 in the Kubernetes cluster. You need to configure Prometheus to collect metrics from this service.

Steps

1. Log into the Prometheus server.
2. Add your cluster entry into the `prometheus.yml` file. In the `yml` file, add an entry similar to the following for your cluster in the `scrape_configs section`:

```
job_name: '<Add your cluster name here. You can abbreviate. It just
needs to be a unique name>'
  metrics_path: /accounts/<replace with your account ID>/metrics
  authorization:
     credentials: <replace with your API token>
  tls_config:
     insecure_skip_verify: true
  static_configs:
    - targets: ['<replace with your astraAddress. If using FQDN, the
prometheus server has to be able to resolve it>']
```

> If you set the `tls_config insecure_skip_verify` to `true`, the TLS encryption protocol is not required.

3. Restart the Prometheus service:

```
sudo systemctl restart prometheus
```

Access Prometheus

Access the Prometheus URL.

Steps

1. In a browser, enter the Prometheus URL with port 9090.
2. Verify your connection by selecting **Status** > **Targets**.

View data in Prometheus

You can use Prometheus to view Astra Control Center data.

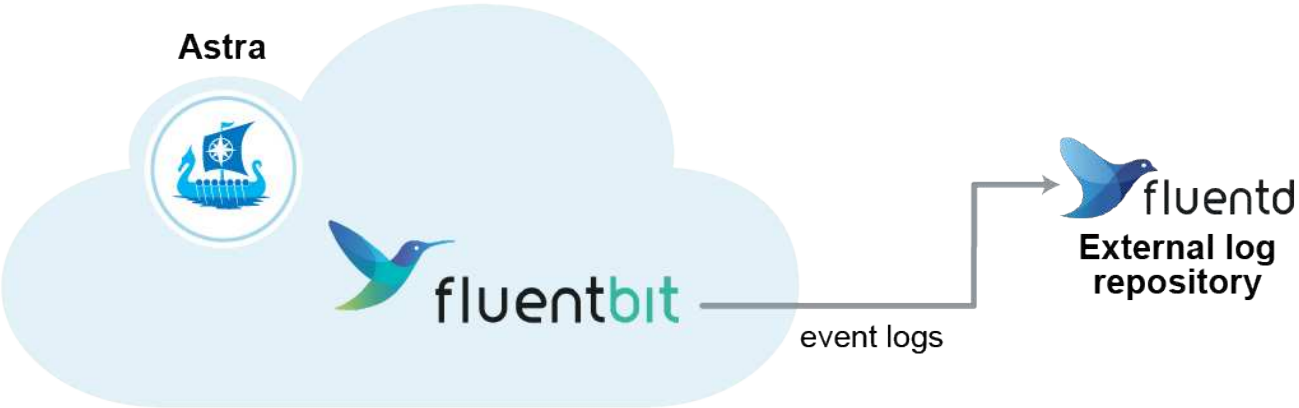
Steps

1. In a browser, enter the Prometheus URL.
2. From the Prometheus menu, select **Graph**.
3. To use the Metrics Explorer, select the icon next to **Execute**.
4. Select `scrape_samples_scraped` and select **Execute**.
5. To see sample scraping over time, select **Graph**.

> If multiple cluster data was collected, each cluster's metrics appear in a different color.

Connect to Fluentd

You can send logs (Kubernetes events) from a system monitored by 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.

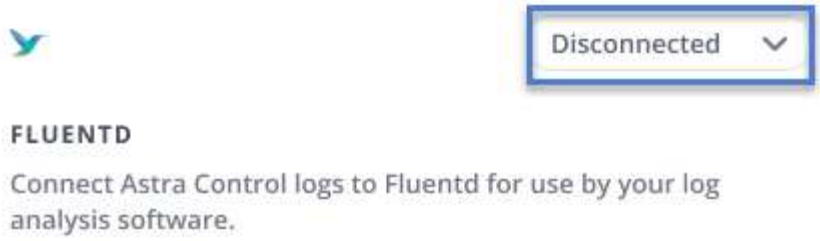
Before you begin

- 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.



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.

When you unmanage an app:

- 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 app.
3. From the Options menu in the Actions column, select **Unmanage**.
4. Review the information.
5. Type "unmanage" to confirm.
6. Select **Yes, unmanage application**.

Result

Astra Control Center stops managing the app.

Unmanage a cluster

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

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

When you unmanage a cluster:

- 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.
- Astra Control Provisioner or Astra Trident won't be uninstalled from the cluster. Learn how to uninstall Astra Trident.

Steps

1. From the left navigation bar, select **Clusters**.
2. Select the check box for the cluster that you no longer want to manage.
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**. After that, the cluster will be removed from the **Clusters** page and it is no longer managed by Astra Control Center.

> Unmanaging the cluster removes all the resources that were installed for sending telemetry data.

Upgrade Astra Control Center

To upgrade Astra Control Center, download the installation images and complete these instructions. You can use this procedure to upgrade Astra Control Center in internet-connected or air-gapped environments.

These instructions describe the upgrade process for Astra Control Center from the second-most recent release to this current release. You cannot upgrade directly from a version that is two or more releases behind the current release. If your installed Astra Control Center version is many versions behind the latest release, you might need to perform chain upgrades to more recent versions until your installed Astra Control Center is only one version behind the latest release. For a complete list of released versions, see the release notes.

Before you begin

Before you upgrade, ensure your environment still meets the minimum requirements for Astra Control Center deployment. Your environment should have the following:

- **An enabled Astra Control Provisioner with Astra Trident running**
  1. Determine the Astra Trident version you are running:

     ```
     kubectl get tridentversion -n trident
     ```

     > If you are running Astra Trident 23.01 or earlier, use these instructions to upgrade to a more recent version of Astra Trident before upgrading to the Astra Control Provisioner. You can perform a direct upgrade to Astra Control Provisioner 24.02 if your Astra Trident is within a four-release window of version 24.02. For example, you can directly upgrade from Astra Trident 23.04 to Astra Control Provisioner 24.02.

  2. Verify that Astra Control Provisioner has been enabled. Astra Control Provisioner will not work with releases of Astra Control Center earlier than 23.10. Upgrade your Astra Control Provisioner so that it has the same version as the Astra Control Center you are upgrading to access the latest functionality.
- **A supported Kubernetes distribution**

  Determine the Kubernetes version you are running:

  ```
  kubectl get nodes -o wide
  ```

- **Sufficient cluster resources**

  Determine available cluster resources:

  ```
  kubectl describe node <node name>
  ```

- **A default storage class**

  Determine your default storage class:

  ```
  kubectl get storageclass
  ```

- **Healthy and available API services**

  Ensure all API services are in a healthy state and available:

  ```
  kubectl get apiservices
  ```

- **(Local registries only) A local registry you can use to push and upload Astra Control Center images**
- **(OpenShift only) Healthy and available cluster operators**

  Ensure all cluster operators are in a healthy state and available.

  ```
  kubectl get clusteroperators
  ```

You should also consider the following:

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

- **Access to the NetApp Astra Control image registry**:
  You have the option to obtain installation images and functionality enhancements for Astra Control, such as Astra Control Provisioner, from the NetApp image registry.

  1. Record your Astra Control account ID that you'll need to log in to the registry.

     You can see your account ID in the Astra Control Service web UI. Select the figure icon at the top right of the page, select **API access**, and write down your account ID.

  2. From the same page, select **Generate API token** and copy the API token string to the clipboard and save it in your editor.
  3. Log into the Astra Control registry:

     ```
     docker login cr.astra.netapp.io -u <account-id> -p <api-token>
     ```

- **Istio service mesh deployments**
  If you installed an Istio service mesh during Astra Control Center installation, this upgrade of Astra Control Center will include Istio service mesh. If you do not yet have a service mesh, you can only install one during an initial deployment of Astra Control Center.

About this task

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

Log out of your Astra Control Center UI before you begin the upgrade.

- Download and extract Astra Control Center
- Complete additional steps if you use a local registry
- Install the updated Astra Control Center operator
- Upgrade Astra Control Center
- Verify system status

Do not delete the Astra Control Center operator (for example, `kubectl delete -f astra_control_center_operator_deploy.yaml`) at any time during the Astra Control Center upgrade or operation to avoid deleting pods.

Download and extract Astra Control Center

Download the Astra Control Center images from one of the following locations:

- **Astra Control Service image registry**: Use this option if you don't use a local registry with the Astra Control Center images or if you prefer this method to the bundle download from the NetApp Support Site.
- **NetApp Support Site**: Use this option if you use a local registry with the Astra Control Center images.

Astra Control image registry

1. Log in to Astra Control Service.
2. On the Dashboard, select **Deploy a self-managed instance of Astra Control**.
3. Follow the instructions to log in to the Astra Control image registry, pull the Astra Control Center installation image, and extract the image.

NetApp Support Site

1. Download the bundle containing Astra Control Center (`astra-control-center-[version].tar.gz`) from the Astra Control Center downloads page.
2. (Recommended but optional) Download the certificates and signatures bundle for Astra Control Center (`astra-control-center-certs-[version].tar.gz`) to verify the signature of the bundle.

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

   ```
   openssl dgst -sha256 -verify certs/AstraControlCenter-public.pub
   -signature certs/astra-control-center-[version].tar.gz.sig astra-
   control-center-[version].tar.gz
   ```

   The output will show `Verified OK` after successful verification.

3. Extract the images from the Astra Control Center bundle:

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

Complete additional steps if you use a local registry

If you are planning to push the Astra Control Center bundle to your local registry, you need to use the NetApp Astra kubectl command line plugin.

Remove the NetApp Astra kubectl plugin and install it again

You need to use the latest version of the NetApp Astra kubectl command line plugin to push images to a local Docker repository.

1. Determine if you have the plug-in installed:

   ```
   kubectl astra
   ```

2. Take one of these actions:
   - If the plugin is installed, the command should return the kubectl plugin help and you can remove the existing version of kubectl-astra: `delete /usr/local/bin/kubectl-astra`.

- If the command returns an error, the plugin is not installed and you can proceed to the next step to install it.

3. Install the plugin:
   a. List the available NetApp Astra kubectl plugin binaries, and note the name of the file you need for your operating system and CPU architecture:

      The kubectl plugin library is part of the tar bundle and is extracted into the folder `kubectl-astra`.

      ```
      ls kubectl-astra/
      ```

   b. Move the correct binary into the current path and rename it to `kubectl-astra`:

      ```
      cp kubectl-astra/<binary-name> /usr/local/bin/kubectl-astra
      ```

Add the images to your registry

1. If you are planning to push the Astra Control Center bundle to your local registry, complete the appropriate step sequence for your container engine:

Docker

a. Change to the root directory of the tarball. You should see the `acc.manifest.bundle.yaml` file and these directories:

```
acc/
kubectl-astra/
acc.manifest.bundle.yaml
```

b. Push the package images in the Astra Control Center image directory to your local registry. Make the following substitutions before running the `push-images` command:

- Replace <BUNDLE_FILE> with the name of the Astra Control bundle file (`acc.manifest.bundle.yaml`).
- Replace <MY_FULL_REGISTRY_PATH> with the URL of the Docker repository; for example, "https://<docker-registry>".
- Replace <MY_REGISTRY_USER> with the user name.
- Replace <MY_REGISTRY_TOKEN> with an authorized token for the registry.

```
kubectl astra packages push-images -m <BUNDLE_FILE> -r
<MY_FULL_REGISTRY_PATH> -u <MY_REGISTRY_USER> -p
<MY_REGISTRY_TOKEN>
```

Podman

a. Change to the root directory of the tarball. You should see this file and directory:

```
acc/
kubectl-astra/
acc.manifest.bundle.yaml
```

b. Log in to your registry:

```
podman login <YOUR_REGISTRY>
```

c. Prepare and run one of the following scripts that is customized for the version of Podman you use. Substitute <MY_FULL_REGISTRY_PATH> with the URL of your repository that includes any sub-directories.

```
<strong>Podman 4</strong>
```

```
export REGISTRY=<MY_FULL_REGISTRY_PATH>
export PACKAGENAME=acc
export PACKAGEVERSION=24.02.0-69
export DIRECTORYNAME=acc
for astraImageFile in $(ls ${DIRECTORYNAME}/images/*.tar) ; do
astraImage=$(podman load --input ${astraImageFile} | sed
's/Loaded image: //')
astraImageNoPath=$(echo ${astraImage} | sed 's:.*/::')
podman tag ${astraImageNoPath} ${REGISTRY}/netapp/astra/
${PACKAGENAME}/${PACKAGEVERSION}/${astraImageNoPath}
podman push ${REGISTRY}/netapp/astra/${PACKAGENAME}/
${PACKAGEVERSION}/${astraImageNoPath}
done
```

```
<strong>Podman 3</strong>
```

```
export REGISTRY=<MY_FULL_REGISTRY_PATH>
export PACKAGENAME=acc
export PACKAGEVERSION=24.02.0-69
export DIRECTORYNAME=acc
for astraImageFile in $(ls ${DIRECTORYNAME}/images/*.tar) ; do
astraImage=$(podman load --input ${astraImageFile} | sed
's/Loaded image: //')
astraImageNoPath=$(echo ${astraImage} | sed 's:.*/::')
podman tag ${astraImageNoPath} ${REGISTRY}/netapp/astra/
${PACKAGENAME}/${PACKAGEVERSION}/${astraImageNoPath}
podman push ${REGISTRY}/netapp/astra/${PACKAGENAME}/
${PACKAGEVERSION}/${astraImageNoPath}
done
```

The image path the script creates should resemble the following, depending on your registry configuration:

```
https://downloads.example.io/docker-astra-control-
prod/netapp/astra/acc/24.02.0-69/image:version
```

2. Change the directory:

```
cd manifests
```

Install the updated Astra Control Center operator

1. (Local registries only) If you are using a local registry, complete these steps:

   a. Open the Astra Control Center operator deployment YAML:

   ```
   vim astra_control_center_operator_deploy.yaml
   ```

   An annotated sample YAML follows these steps.

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

   ```
   imagePullSecrets: [{name: astra-registry-cred}]
   ```

   c. Change `ASTRA_IMAGE_REGISTRY` for the `kube-rbac-proxy` image to the registry path where you pushed the images in a previous step.

   d. Change `ASTRA_IMAGE_REGISTRY` for the `acc-operator` image to the registry path where you pushed the images in a previous step.

   e. Add the following values to the `env` section:

   ```
   - name: ACCOP_HELM_UPGRADETIMEOUT
     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
  strategy:
    type: Recreate
  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: ASTRA_IMAGE_REGISTRY/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
        env:
        - name: ACCOP_LOG_LEVEL
          value: "2"
        - name: ACCOP_HELM_UPGRADETIMEOUT
          value: 300m
        image: ASTRA_IMAGE_REGISTRY/acc-operator:24.02.68
        imagePullPolicy: IfNotPresent
        livenessProbe:
          httpGet:
            path: /healthz
            port: 8081
          initialDelaySeconds: 15
          periodSeconds: 20
        name: manager
        readinessProbe:
          httpGet:
            path: /readyz
            port: 8081
          initialDelaySeconds: 5
          periodSeconds: 10
        resources:
          limits:
            cpu: 300m
            memory: 750Mi
          requests:
            cpu: 100m
            memory: 75Mi
        securityContext:
          allowPrivilegeEscalation: false
      imagePullSecrets: []
      securityContext:
```

```
        runAsUser: 65532
      terminationGracePeriodSeconds: 10
```

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.as
tra.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
```

3. Verify pods are running:

```
kubectl get pods -n netapp-acc-operator
```

Upgrade Astra Control Center

1. Edit the Astra Control Center custom resource (CR):

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

An annotated sample YAML follows these steps.

2. Change the Astra version number (`astraVersion` inside of `spec`) from `23.10.0` to `24.02.0`:

   You cannot upgrade directly from a version that is two or more releases behind the current release. For a complete list of released versions, see the release notes.

```
spec:
  accountName: "Example"
  astraVersion: "[Version number]"
```

3. Change the image registry:
   - (Local registries only) If you are using a local registry, verify that your image registry path matches the registry path you pushed the images to in a previous step. Update `imageRegistry` inside of `spec` if the local registry has changed since your last installation.
   - (Astra Control image registry) Use the Astra Control image registry (`cr.astra.netapp.io`) you used to download the updated Astra Control bundle.

```
imageRegistry:
  name: "[cr.astra.netapp.io or your_registry_path]"
```

4. Add the following to your `crds` configuration inside of `spec`:

```
crds:
  shouldUpgrade: true
```

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

```
additionalValues:
    nautilus:
      startupProbe:
        periodSeconds: 30
        failureThreshold: 600
    keycloak-operator:
      livenessProbe:
        initialDelaySeconds: 180
      readinessProbe:
        initialDelaySeconds: 180
```

6. Save and exit the file editor. The changes will be applied and the upgrade will begin.
7. (Optional) Verify that the pods terminate and become available again:

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

8. Wait for the Astra Control status conditions to indicate that the upgrade is complete and ready (`True`):

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

Response:

```
NAME    UUID                                      VERSION     ADDRESS
READY
astra   9aa5fdae-4214-4cb7-9976-5d8b4c0ce27f      24.02.0-69
10.111.111.111  True
```

To monitor upgrade status during the operation, run the following command: `kubectl get AstraControlCenter -o yaml -n [netapp-acc or custom namespace]`

To inspect the Astra Control Center operator logs, run the following command: `kubectl logs deploy/acc-operator-controller-manager -n netapp-acc-operator -c manager -f`

Verify system status

1. Log in to Astra Control Center.
2. Verify that the version has been upgraded. See the **Support** page in the UI.
3. Verify that all your managed clusters and apps are still present and protected.

Upgrade Astra Control Center using OpenShift OperatorHub

If you installed Astra Control Center using its Red Hat-certified operator, you can upgrade Astra Control Center using an updated operator from OperatorHub. Use this procedure to upgrade Astra Control Center from the Red Hat Ecosystem Catalog or using the Red Hat OpenShift Container Platform.

Before you begin

- **Meet environmental prerequisites**: Before you upgrade, ensure your environment still meets the minimum requirements for Astra Control Center deployment.
- **Ensure that you have enabled Astra Control Provisioner with Astra Trident running**
  1. Determine the Astra Trident version you are running:

```
kubectl get tridentversion -n trident
```

> If you are running Astra Trident 23.01 or earlier, use these instructions to upgrade to a more recent version of Astra Trident before upgrading to the Astra Control Provisioner. You can perform a direct upgrade to Astra Control Provisioner 24.02 if your Astra Trident is within a four-release window of version 24.02. For example, you can directly upgrade from Astra Trident 23.04 to Astra Control Provisioner 24.02.

2. Verify that Astra Control Provisioner has been enabled. Astra Control Provisioner will not work with releases of Astra Control Center earlier than 23.10. Upgrade your Astra Control Provisioner so that it has the same version as the Astra Control Center you are upgrading to access the latest functionality.

- **Ensure healthy cluster operators and API services**:
  - From your OpenShift cluster, ensure all cluster operators are in a healthy state:

    ```
    oc get clusteroperators
    ```

  - From your OpenShift cluster, ensure all API services are in a healthy state:

    ```
    oc get apiservices
    ```

- **OpenShift permissions**: You have all necessary permissions and access to the Red Hat OpenShift Container Platform to perform the upgrade steps described.
- **(ONTAP SAN driver only) Enable multipath**: If you are using an ONTAP SAN driver, be sure that multipath is enabled on all your Kubernetes clusters.

You should also consider the following:

- **Get access to the NetApp Astra Control image registry**:

  You have the option to obtain installation images and functionality enhancements for Astra Control, such as Astra Control Provisioner, from the NetApp image registry.

  1. Record your Astra Control account ID that you'll need to log in to the registry.

     You can see your account ID in the Astra Control Service web UI. Select the figure icon at the top right of the page, select **API access**, and write down your account ID.

  2. From the same page, select **Generate API token** and copy the API token string to the clipboard and save it in your editor.
  3. Log into the Astra Control registry:

     ```
     docker login cr.astra.netapp.io -u <account-id> -p <api-token>
     ```

Steps

- Access the operator install page
- Uninstall the existing operator
- Install the latest operator

- Upgrade Astra Control Center

Access the operator install page

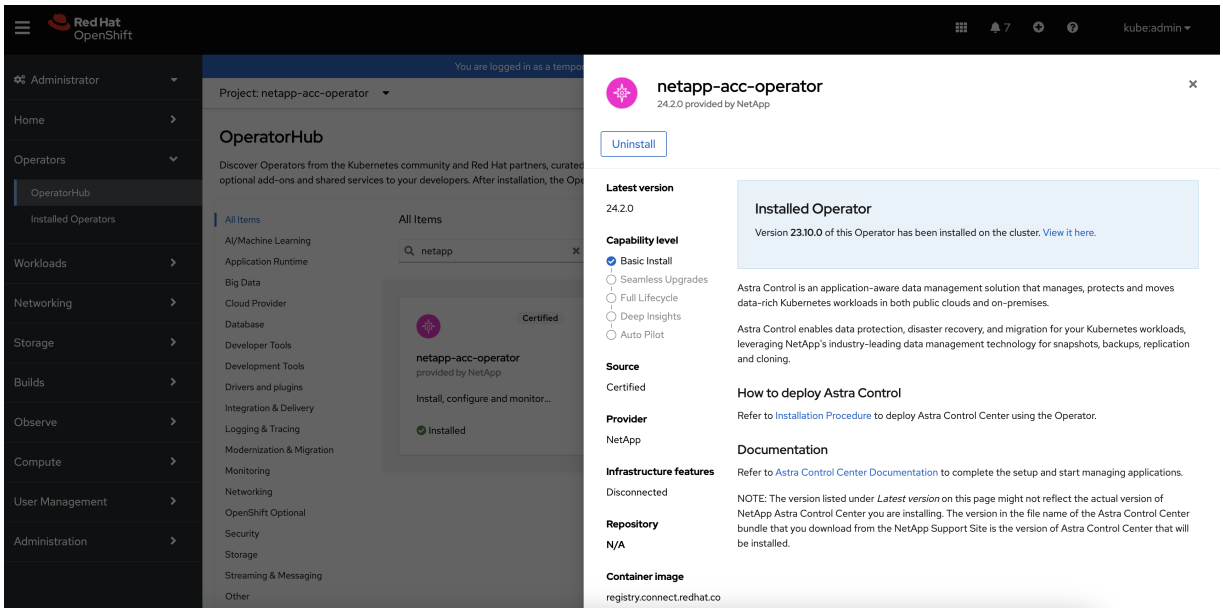
1. Complete the corresponding procedure for either Openshift Container Platform or Ecosystem Catalog:

Red Hat OpenShift web console

1. Log in to the OpenShift Container Platform UI.
2. From the side menu, select **Operators > OperatorHub**.

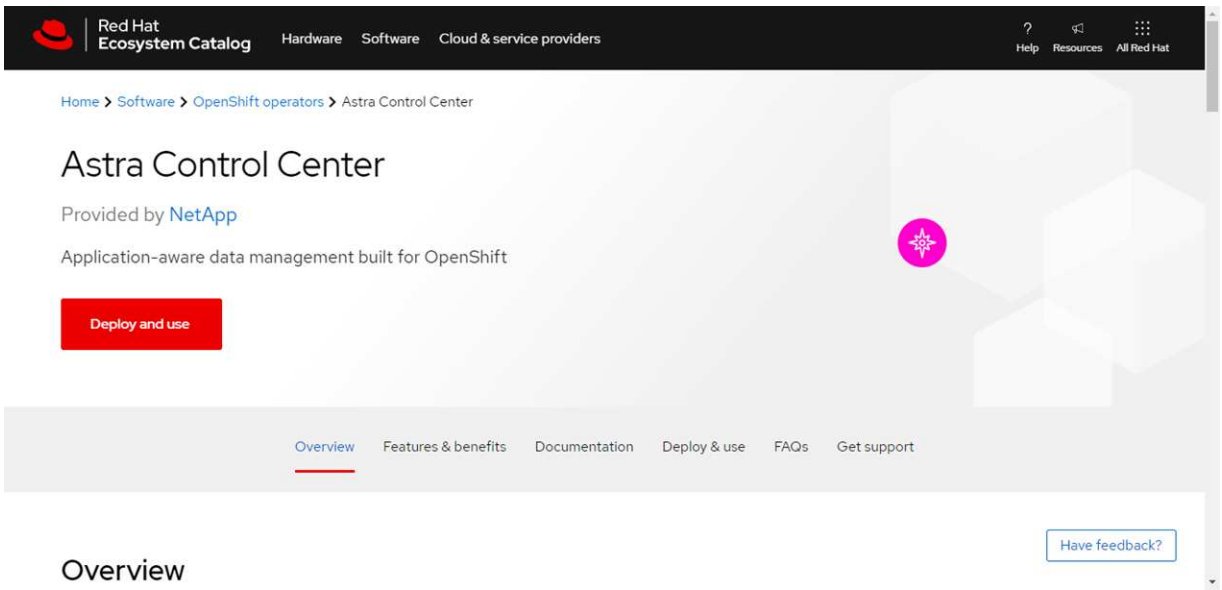
> You can upgrade only to the current version of Astra Control Center using this operator.

3. Search for `netapp-acc` and select the NetApp Astra Control Center operator.



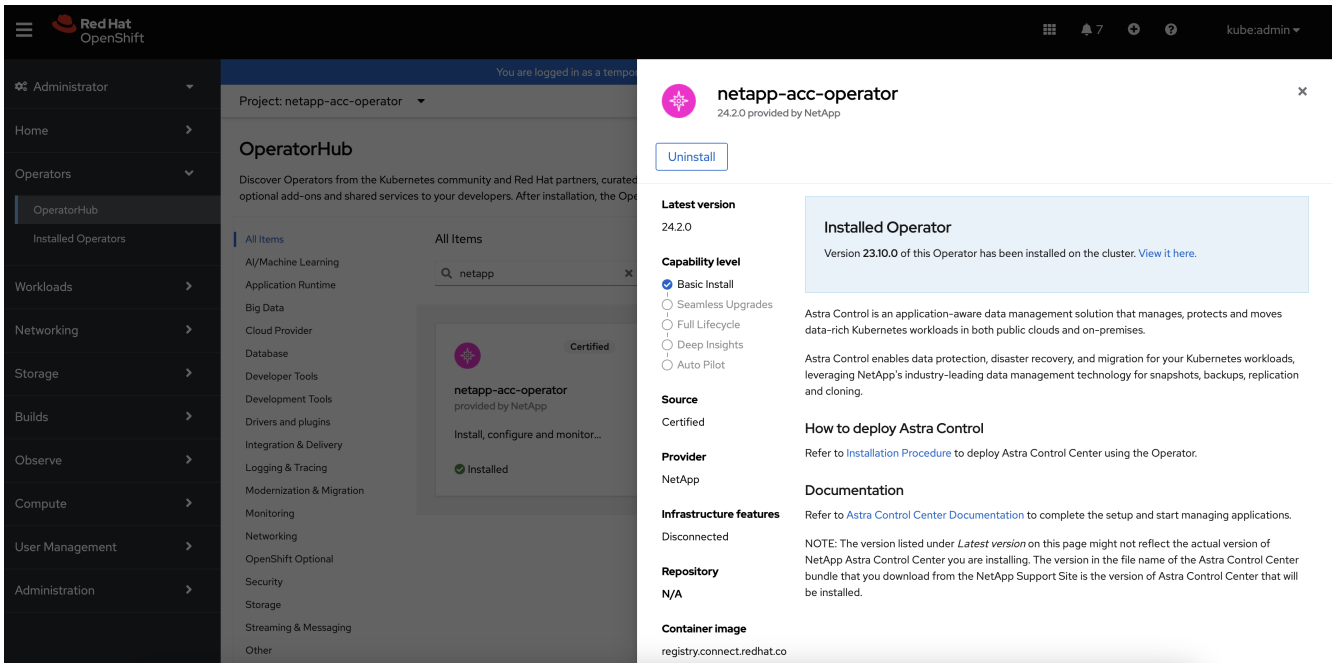
Red Hat Ecosystem Catalog

1. Select the NetApp Astra Control Center operator.
2. Select **Deploy and use**.



Uninstall the existing operator

1. From the **netapp-acc-operator** page, select **Uninstall** to remove your existing operator.

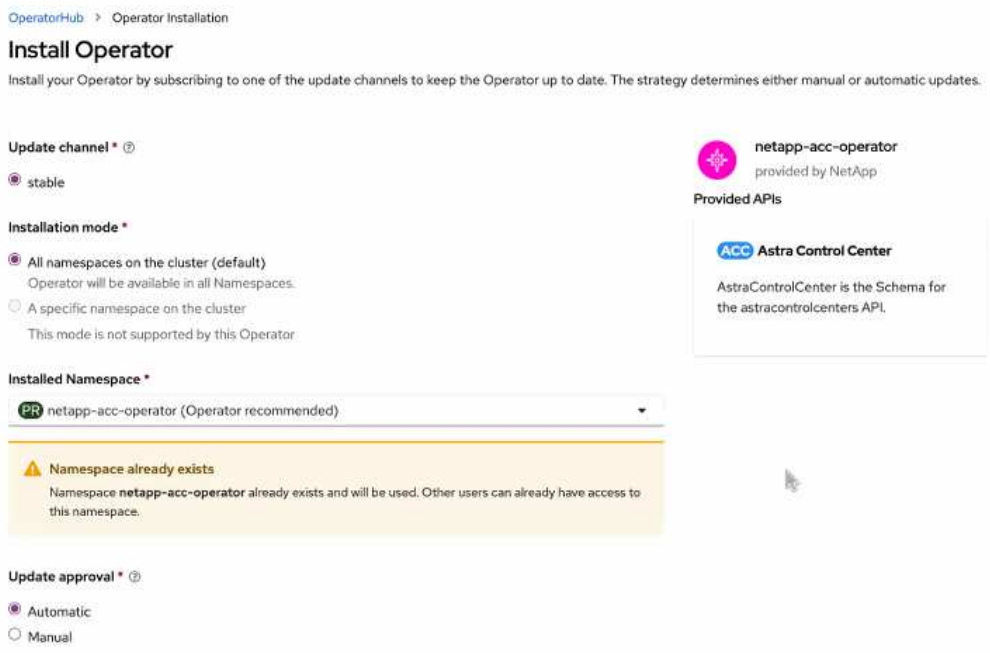


2. Confirm the operation.

> This operation deletes the netapp-acc-operator but preserves the original associated namespace and resources, such as secrets.

Install the latest operator

1. Navigate to the `netapp-acc` operator page again.
2. Complete the **Install Operator** page and install the most recent operator:



> The operator will be available in all cluster namespaces.

a. Select the operator's `netapp-acc-operator` namespace (or custom namespace) that remains from the deleted operator's previous installation.

b. Select a manual or automatic approval strategy.

> Manual approval is recommended. You should only have a single operator instance running per cluster.

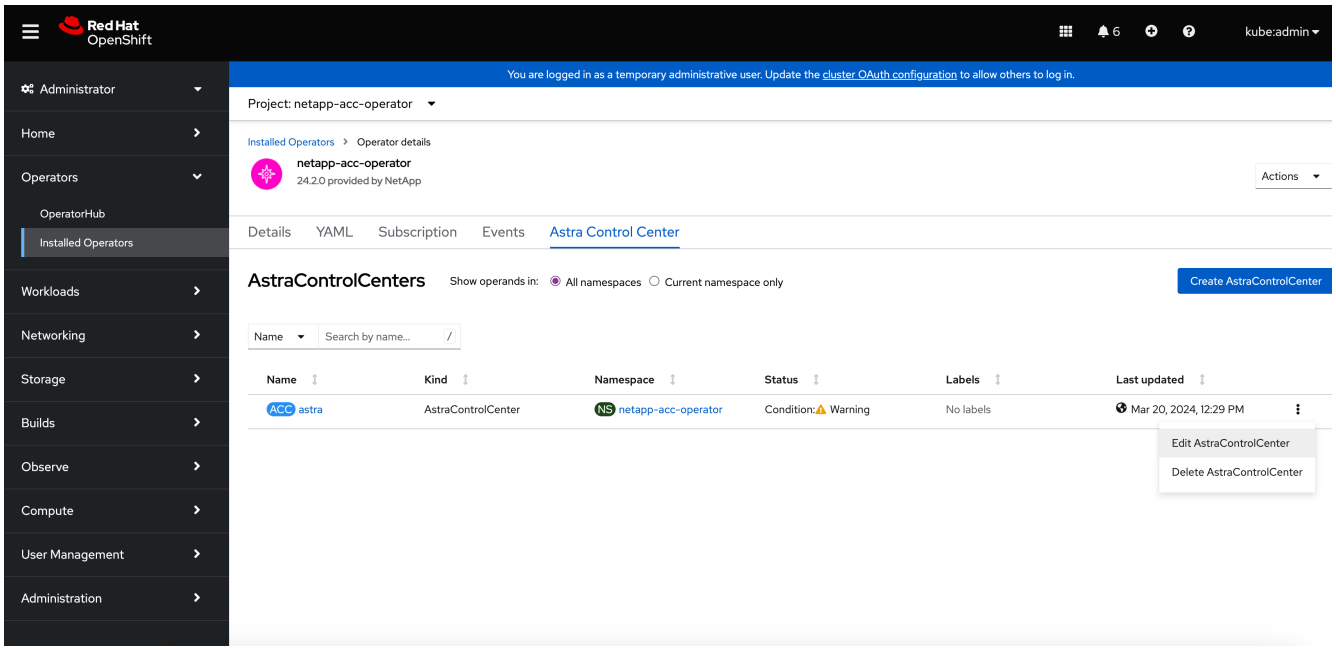
c. Select **Install**.

> If you selected a manual approval strategy, you'll be prompted to approve the manual install plan for this operator.

3. From the console, go to the OperatorHub menu and confirm that the operator installed successfully.

Upgrade Astra Control Center

1. From the Astra Control Center operator tab, select the Astra Control Center that remains from the previous installation and select **Edit AstraControlCenter**.



2. Update the `AstraControlCenter` YAML:
   a. Enter the latest Astra Control Center version; for example, 24.02.0-69.
   b. In `imageRegistry.name`, update the image registry path as needed:
      - If you are using the Astra Control registry option, change the path to `cr.astra.netapp.io`.
      - If you configured a local registry, change or retain the local image registry path where you pushed the images in a previous step.

      > Do not enter `http://` or `https://` in the address field.

   c. Update the `imageRegistry.secret` as needed.

      > The operator uninstall process does not remove existing secrets. You only need to update this field if you create a new secret with a different name from the existing secret.

   d. Add the following to your `crds` configuration:

```
crds:
  shouldUpgrade: true
```

3. Save your changes.
4. The UI confirms that the upgrade was successful.

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.

Before you begin

1. Unmanage all apps on the clusters.
2. 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.yaml -n netapp-acc
   ```

   Result:

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

2. Use the following command to delete the `netapp-acc` (or custom-named) namespace:

   ```
   kubectl delete ns [netapp-acc or custom namespace]
   ```

   Example 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

```
ingressroutes.traefik.containo.us             2021-06-23T23:29:11Z
ingressroutetcps.traefik.containo.us          2021-06-23T23:29:11Z
ingressrouteudps.traefik.containo.us          2021-06-23T23:29:12Z
middlewares.traefik.containo.us               2021-06-23T23:29:12Z
middlewaretcps.traefik.containo.us            2021-06-23T23:29:12Z
serverstransports.traefik.containo.us         2021-06-23T23:29:13Z
tlsoptions.traefik.containo.us                2021-06-23T23:29:13Z
tlsstores.traefik.containo.us                 2021-06-23T23:29:14Z
traefikservices.traefik.containo.us           2021-06-23T23:29:15Z
```

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
middlewaretcps.traefik.containo.us
```

Find more information

- Known issues for uninstall