Protect apps

Astra Control Center

NetApp
May 08, 2024

Table of Contents

Protect apps . . . 1
- Protection overview . . . 1
- Protect apps with snapshots and backups . . . 1
- Restore apps . . . 5
- Replicate apps between storage backends using SnapMirror technology . . . 10
- Clone and migrate apps . . . 16
- Manage app execution hooks . . . 19
- Protect Astra Control Center using Astra Control Center . . . 27

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
- 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.

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.

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.

If your app uses a storage class backed by the `ontap-nas-economy` driver, protection policies cannot be used. Migrate to a storage class supported by Astra Control if you want to schedule backups and snapshots.

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

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.

> 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.

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. Review the snapshot summary and select **Snapshot**.

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 also back up an app at any time.

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

> 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. Backups for apps backed by the `ontap-nas-economy` are disruptive and the app will be unavailable until the backup operation has completed.

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

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.

View snapshots and backups

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

Steps

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

   The snapshots display by default.
3. Select **Backups** to see the list of backups.

Delete snapshots

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

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.

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.

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.

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.

About this task

- **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 in the event that 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.
- **(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
```

- **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.

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.

> You can use this procedure to either 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.

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**.

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 new namespace is created by a clone or restore operation, the account admin/owner can edit the member user account and update role constraints for the affected user to grant access to the new namespace.

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.

Migrate from ontap-nas-economy storage to ontap-nas storage

You can use an Astra Control application restore or application clone operation to migrate application volumes from a storage class backed by `ontap-nas-economy`, which permits limited application protection options, to a storage class backed by `ontap-nas` with its full range of Astra Control protection options. The clone or restore operation migrates Qtree-based volumes that use an `ontap-nas-economy` backend to standard volumes backed by `ontap-nas`. Volumes, regardless of whether they are `ontap-nas-economy` backed only or mixed, will be migrated to the target storage class. After the migration is complete, protection options are no longer limited.

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-premise site A to on-premise site A
- On-premise site A to on-premise site B
- On-premise to cloud with Cloud Volumes ONTAP
- Cloud with Cloud Volumes ONTAP to on-premise
- 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 Trident**: Astra Trident version 22.10 or later must exist on both the source and destination Kubernetes clusters that utilize ONTAP as a backend.

- **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 Trident and SVM**: The peered remote SVMs must be available to 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 clusters**: The following clusters must be added to and managed by 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.
- **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 storage class**: If your app uses a storage class backed by the `ontap-nas-economy` driver, the backup portion of a clone operation is disruptive. The source application is not available until the backup is complete. The restore portion of the clone operation is non-disruptive.
- **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 new namespace is created by a clone or restore operation, the account admin/owner can edit the member user account and update role constraints for the affected user to grant access to the new namespace.
- **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 new namespace is created by a clone or restore operation, the account admin/owner can edit the member user account and update role constraints for the affected user to grant access to the new namespace.

> 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 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.

- 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 ad hoc 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, 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

Scenario	Operation	Existing snapshot	Existing backup	Namespace	Cluster	Snapshot hooks run	Backup hooks run	Restore hooks run	Failover hooks run
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.

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. Go to **Account**.
2. Select the **Scripts** tab.
3. Select **Add**.
4. 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.
5. 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. Select **Applications** and then select the name of a managed app.
2. Select the **Execution hooks** tab.
3. Select **Add**.
4. 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.
5. (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.

6. When done, select **Next**.

7. 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.
8. Select **Next**.
9. Review the execution hook configuration.
10. 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.

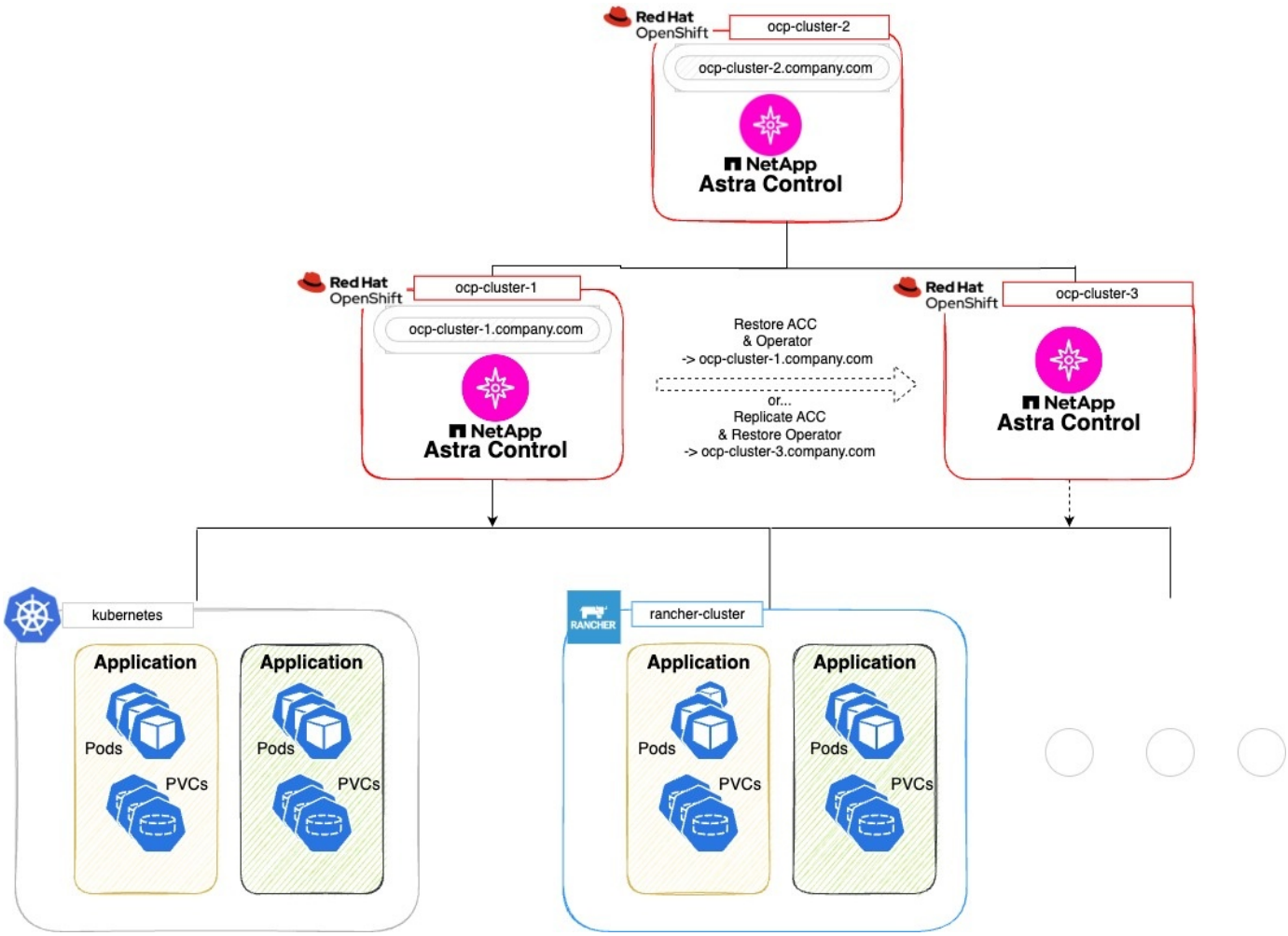
Before you begin

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 Trident 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: acc-crds
      - 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 will 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 23.07.0-24 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**.

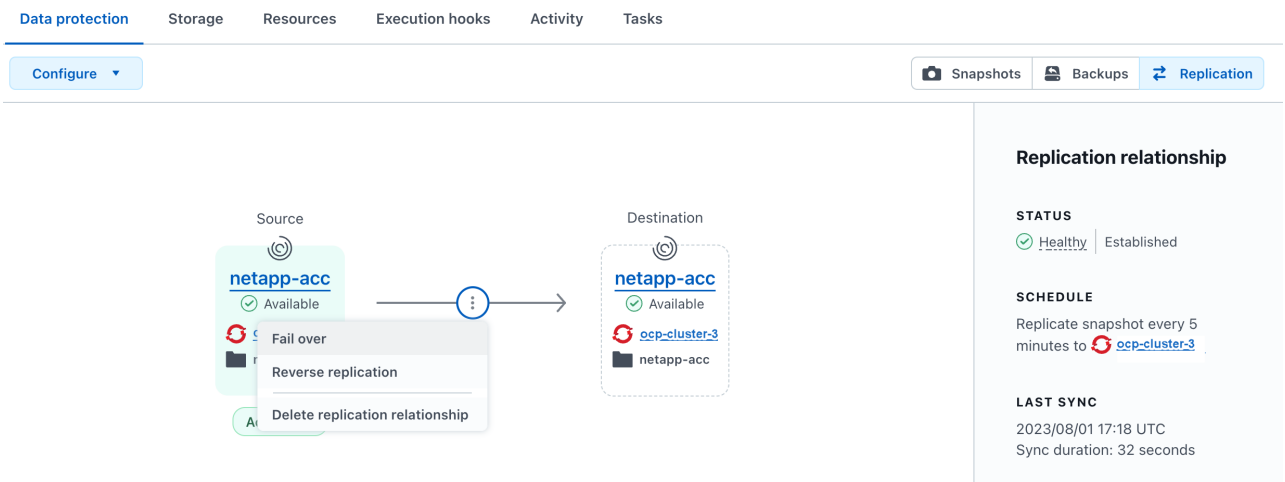
h. Confirm the configuration is correct, and select **Save**.

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 23.07.0-24 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.

Copyright information

Trademark information