Astra Control Center 22.11 documentation

Astra Control Center

NetApp
February 12, 2024

Table of Contents

Astra Control Center 22.11 documentation . . . 1
Release notes . . . 2
- What's new in this release of Astra Control Center . . . 2
- Known issues . . . 4
- Known limitations . . . 6

Get started . . . 11
- Astra Control Center requirements . . . 11
- Quick start for Astra Control Center . . . 15
- Installation overview . . . 17
- Set up Astra Control Center . . . 69
- Frequently asked questions for Astra Control Center . . . 82

Concepts . . . 84
- Architecture and components . . . 84
- Data protection . . . 85
- Licensing . . . 88
- Storage classes and persistent volume size . . . 89
- User roles and namespaces . . . 89

Use Astra Control Center . . . 91
- Start managing apps . . . 91
- Protect apps . . . 96
- Monitor app and cluster health . . . 117
- Manage your account . . . 120
- Manage buckets . . . 130
- Manage the storage backend . . . 133
- Monitor running tasks . . . 135
- Monitor infrastructure with Cloud Insights, Prometheus, or Fluentd connections . . . 136
- Unmanage apps and clusters . . . 145
- Upgrade Astra Control Center . . . 146
- Uninstall Astra Control Center . . . 155

Automate with Astra Control REST API . . . 159
- Automation using the Astra Control REST API . . . 159

Knowledge and support . . . 160
- Troubleshooting . . . 160
- Get help . . . 160

Earlier versions of Astra Control Center documentation . . . 163

Legal notices . . . 164
- Copyright . . . 164
- Trademarks . . . 164
- Patents . . . 164
- Privacy policy . . . 164
- Open source . . . 164
- Astra Control API license . . . 164

Astra Control Center 22.11 documentation

Release notes

We're pleased to announce the latest release of Astra Control Center.

- What's in this release of Astra Control Center
- Known issues
- Known limitations

Follow us on Twitter @NetAppDoc. Send feedback about documentation by becoming a GitHub contributor or sending an email to doccomments@netapp.com.

What's new in this release of Astra Control Center

We're pleased to announce the latest release of Astra Control Center.

22 November 2022 (22.11.0)

New features and support

- Support for applications that span across multiple namespaces
- Support for including cluster resources in an application definition
- Enhanced LDAP authentication with role-based access control (RBAC) integration
- Added support for Kubernetes 1.25 and Pod Security Admission (PSA)
- Enhanced progress reporting for your backup, restore, and clone operations

Known issues and limitations

- Known issues for this release
- Known limitations for this release

8 September 2022 (22.08.1)

This patch release (22.08.1) for Astra Control Center (22.08.0) fixes minor bugs in app replication using NetApp SnapMirror.

10 August 2022 (22.08.0)

Details

New features and support

- App replication using NetApp SnapMirror technology
- Improved app management workflow
- Enhanced provide-your-own execution hooks functionality

> The NetApp provided default pre- and post-snapshot execution hooks for specific applications have been removed in this release. If you upgrade to this release and do not provide your own execution hooks for snapshots, Astra Control will take crash-consistent snapshots only. Visit the NetApp Verda GitHub repository for sample execution hook scripts that you can modify to fit your environment.

- Support for VMware Tanzu Kubernetes Grid Integrated Edition (TKGI)
- Support for Google Anthos
- LDAP configuration (via Astra Control API)

Known issues and limitations

- Known issues for this release
- Known limitations for this release

26 April 2022 (22.04.0)

Details

New features and support

- Namespace role-based access control (RBAC)
- Support for Cloud Volumes ONTAP
- Generic ingress enablement for Astra Control Center
- Bucket removal from Astra Control
- Support for VMware Tanzu Portfolio

Known issues and limitations

- Known issues for this release
- Known limitations for this release

14 December 2021 (21.12)

Details

New features and support

- Application restore
- Execution hooks
- Support for applications deployed with namespace-scoped operators
- Additional support for upstream Kubernetes and Rancher
- Astra Control Center upgrades
- Red Hat OperatorHub option for installation

Resolved issues

- Resolved issues for this release

Known issues and limitations

- Known issues for this release
- Known limitations for this release

5 August 2021 (21.08)

Details

Initial release of Astra Control Center.

- What it is
- Understand architecture and components
- What it takes to get started
- Install and setup
- Manage and protect apps
- Manage buckets and storage backends
- Manage accounts
- Automate with API

Find more information

- Known issues for this release
- Known limitations for this release
- Earlier versions of Astra Control Center documentation

Known issues

Known issues identify problems that might prevent you from using this release of the product successfully.

The following known issues affect the current release:

Apps

- Restore of an app results in PV size larger than original PV
- App clones fail using a specific version of PostgreSQL
- App clones fail when using Service Account level OCP Security Context Constraints (SCC)
- App clones fail after an application is deployed with a set storage class
- App backups and snapshots fail if the volumesnapshotclass is added after a cluster is managed

Clusters

- Managing a cluster with Astra Control Center fails when default kubeconfig file contains more than one context

Other issues

- Managed clusters do not appear in NetApp Cloud Insights when connecting through a proxy
- App data management operations fail with Internal Service Error (500) when Astra Trident is offline

Restore of an app results in PV size larger than original PV

If you resize a persistent volume after creating a backup and then restore from that backup, the persistent volume size will match the new size of the PV instead of using the size of the backup.

App clones fail using 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.

App clones fail when using Service Account level OCP Security Context Constraints (SCC)

An application clone might fail if the original security context constraints are configured at the service account level within the namespace on the OpenShift Container Platform cluster. When the application clone fails, it appears in the Managed Applications area in Astra Control Center with status `Removed`. See the knowledgebase article for more information.

App backups and snapshots fail if the volumesnapshotclass is added after a cluster is managed

Backups and snapshots fail with a `UI 500 error` in this scenario. As a workaround, refresh the app list.

App clones fail after an application is deployed with a set storage class

After an application is deployed with a storage class explicitly set (for example, `helm install …-set global.storageClass=netapp-cvs-perf-extreme`), subsequent attempts to clone the application require that the target cluster 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. There are no recovery steps in this scenario.

Managing a cluster with Astra Control Center fails when default kubeconfig file contains more than one context

You cannot use a kubeconfig with more than one cluster and context in it. See the knowledgebase article for more information.

Managed clusters do not appear in NetApp Cloud Insights when connecting through a proxy

When Astra Control Center connects to NetApp Cloud Insights through a proxy, managed clusters might not appear in Cloud Insights. As a workaround, run the following commands on each managed cluster:

```
kubectl get cm telegraf-conf -o yaml -n netapp-monitoring | sed
'/\[\[outputs.http\]\]/c\    [[outputs.http]]\n    use_system_proxy =
true' | kubectl replace -f -
```

```
kubectl get cm telegraf-conf-rs -o yaml -n netapp-monitoring | sed
'/\[\[outputs.http\]\]/c\    [[outputs.http]]\n    use_system_proxy =
true' | kubectl replace -f -
```

```
kubectl get pods -n netapp-monitoring --no-headers=true | grep 'telegraf-
ds\|telegraf-rs' | awk '{print $1}' | xargs kubectl delete -n netapp-
monitoring pod
```

App data management operations fail with Internal Service Error (500) when Astra Trident is offline

If Astra Trident on an app cluster goes offline (and is brought back online) and 500 internal service errors are encountered when attempting app data management, restart all of the Kubernetes nodes in the app cluster to restore functionality.

Find more information

- Known limitations

Known limitations

Known limitations identify platforms, devices, or functions that are not supported by this release of the product, or that do not interoperate correctly with it. Review these limitations carefully.

Cluster management limitations

- The same cluster cannot be managed by two Astra Control Center instances
- Astra Control Center cannot manage two identically named clusters

Role-based Access Control (RBAC) limitations

- A user with namespace RBAC constraints can add and unmanage a cluster
- A member with namespace constraints cannot access the cloned or restored apps until admin adds the namespace to the constraint

App management limitations

- Multiple applications in a single namespace cannot be restored collectively to a different namespace
- Astra Control does not automatically assign default buckets for cloud instances
- Clones of apps installed using pass-by-reference operators can fail
- In-place restore operations of apps that use a certificate manager are not supported
- OLM-enabled and cluster-scoped operator deployed apps not supported
- Apps deployed with Helm 2 are not supported

General limitations

- S3 buckets in Astra Control Center do not report available capacity
- Astra Control Center does not validate the details you enter for your proxy server
- Existing connections to a Postgres pod causes failures
- Backups and snapshots might not be retained during removal of an Astra Control Center instance
- LDAP user and group limitations

The same cluster cannot be managed by two Astra Control Center instances

If you want to manage a cluster on another Astra Control Center instance, you should first unmanage the cluster from the instance on which it is managed before you manage it on another instance. After you remove the cluster from management, verify that the cluster is unmanaged by executing this command:

```
oc get pods n -netapp-monitoring
```

There should be no pods running in that namespace or the namespace should not exist. If either of those are true, the cluster is unmanaged.

Astra Control Center cannot manage two identically named clusters

If you try to add a cluster with the same name of a cluster that already exists, the operation will fail. This issue most often occurs in a standard Kubernetes environment if you have not changed the cluster name default in Kubernetes configuration files.

As a workaround, do the following:

1. Edit your `kubeadm-config` ConfigMap:

   ```
   kubectl edit configmaps -n kube-system kubeadm-config
   ```

2. Change the `clusterName` field value from `kubernetes` (the Kubernetes default name) to a unique custom name.

3. Edit kubeconfig (`.kube/config`).
4. Update cluster name from `kubernetes` to a unique custom name (`xyz-cluster` is used in the examples below). Make the update in both `clusters` and `contexts` sections as shown in this example:

```
apiVersion: v1
clusters:
- cluster:
    certificate-authority-data:
ExAmPLERb2tCcjZ5K3E2Njk4eQotLExAMpLEORCBDRVJUSUZJQ0FURS0txxxxXX==
    server: https://x.x.x.x:6443
  name: xyz-cluster
contexts:
- context:
    cluster: xyz-cluster
    namespace: default
    user: kubernetes-admin
  name: kubernetes-admin@kubernetes
current-context: kubernetes-admin@kubernetes
```

A user with namespace RBAC constraints can add and unmanage a cluster

A user with namespace RBAC constraints should not be allowed to add or unmanage clusters. Due to a current limitation, Astra does not prevent such users from unmanaging clusters.

A member with namespace constraints cannot access the cloned or restored apps until admin adds the namespace to the constraint

Any `member` user with RBAC constraints by namespace name/ID 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.

Multiple applications in a single namespace cannot be restored collectively to a different namespace

If you manage multiple applications in a single namespace (by creating multiple app definitions in Astra Control), you cannot restore all of the applications to a different single namespace. You need to restore each application to its own separate namespace.

Astra Control does not automatically assign default buckets for cloud instances

Astra Control does not automatically assign a default bucket for any cloud instance. You need to manually set a default bucket for a cloud instance. If a default bucket is not set, you won't be able to perform app clone operations between two clusters.

Clones of apps installed using pass-by-reference operators can fail

Astra Control supports apps installed with namespace-scoped operators. These operators are generally designed with a "pass-by-value" rather than "pass-by-reference" architecture. The following are some operator apps that follow these patterns:

- Apache K8ssandra

> 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.

> 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.

In-place restore operations of apps that use a certificate manager are not supported

This release of Astra Control Center does not support in-place restore of apps with certificate managers. Restore operations to a different namespace and clone operations are supported.

OLM-enabled and cluster-scoped operator deployed apps not supported

Astra Control Center does not support application management activities with cluster-scoped operators.

Apps deployed with Helm 2 are not supported

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) is fully supported. For more information, see Astra Control Center requirements.

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.

Astra Control Center does not validate the details you enter for your proxy server

Ensure that you enter the correct values when establishing a connection.

Existing connections to a Postgres pod causes failures

When you perform operations on Postgres pods, you shouldn't connect directly within the pod to use the psql command. Astra Control requires psql access to freeze and thaw the databases. If there is a pre-existing connection, the snapshot, backup, or clone will fail.

Backups and snapshots might not be retained during removal of an Astra Control Center instance

If you have an evaluation license, be sure you store your account ID to avoid data loss in the event of Astra Control Center failure if you are not sending ASUPs.

LDAP user and group limitations

Astra Control Center supports up to 5,000 remote groups and 10,000 remote users.

Find more information

- Known issues

Get started

Astra Control Center requirements

Get started by verifying the readiness of your operational environment, application clusters, applications, licenses, and web browser.

- Operational environment requirements
- Supported storage backends
- Access to the internet
- License
- Ingress for on-premises Kubernetes clusters
- Networking requirements
- Supported web browsers
- Additional requirements for application clusters
- Google Anthos cluster requirements
- VMware Tanzu Kubernetes Grid cluster requirements

Operational environment requirements

Astra Control Center has been validated on the following types of operational environments:

- Cisco IKS with Kubernetes 1.22
- Google Anthos 1.11 or 1.12 (See Google Anthos cluster requirements)
- Rancher Kubernetes Engine (RKE):
  - RKE 1.3.12 with Rancher 2.6.5 and 2.6.6
  - RKE 1.3.13 with Rancher 2.6.8
  - RKE 2 (v1.23.6+rke2r1) with Rancher 2.6.5 and 2.6.6
  - RKE 2 (v1.24.x) with Rancher 2.6.8
- Red Hat OpenShift Container Platform 4.8 through 4.11
- Upstream Kubernetes 1.23 to 1.25 (Astra Trident 22.10 or newer required for Kubernetes 1.25)
- VMware Tanzu Kubernetes Grid: (See VMware Tanzu Kubernetes Grid cluster requirements)
  - VMware Tanzu Kubernetes Grid 1.5
  - VMware Tanzu Kubernetes Grid Integrated Edition 1.13 and 1.14

Ensure that the operating environment you choose to host Astra Control Center meets the basic resource requirements outlined in the environment's official documentation. Astra Control Center requires the following resources in addition to the environment's resource requirements:

Component	Requirement
CPU extensions	The CPUs in all nodes of the hosting environment must have AVX extensions enabled.

Component	Requirement
Storage backend capacity	At least 500GB available
Worker nodes	At least 3 worker nodes total, with 4 CPU cores and 12GB RAM each
FQDN address	An FQDN address for Astra Control Center
Astra Trident	Astra Trident 22.01 or newer installed and configured Astra Trident 22.07 or newer installed for SnapMirror-based application replication Astra Trident 22.10 or newer installed for Kubernetes 1.25 clusters (you must upgrade to Astra Trident 22.10 prior to upgrading to Kubernetes 1.25)

> These requirements assume that Astra Control Center is the only application running in the operational environment. If the environment is running additional applications, adjust these minimum requirements accordingly.

- **Image registry**: You must have an existing private Docker image registry to which you can push Astra Control Center build images. You need to provide the URL of the image registry where you will upload the images.
- **Astra Trident / ONTAP configuration**:
  - You need to configure at least one Astra Trident storage class on the cluster. If a default storage class is configured, ensure that it is the only storage class with the default designation.
  - Ensure that the worker nodes in your cluster are configured with the appropriate storage drivers so that the pods can interact with the backend storage. Astra Control Center supports the following ONTAP drivers provided by Astra Trident:
    - ontap-nas
    - ontap-san
    - ontap-san-economy (not supported for app replication)

Supported storage backends

Astra Control Center supports the following storage backends.

- NetApp ONTAP 9.5 or newer AFF, FAS, and ASA systems
- NetApp ONTAP 9.8 or newer AFF, FAS, and ASA systems for SnapMirror-based application replication
- NetApp ONTAP Select 9.5 or newer
- NetApp ONTAP Select 9.8 or newer for SnapMirror-based application replication
- NetApp Cloud Volumes ONTAP 9.5 or newer

To use Astra Control Center, verify that you have the following ONTAP licenses, depending on what you need to accomplish:

- FlexClone
- SnapMirror: Optional. Needed only for replication to remote systems using SnapMirror technology. Refer to

SnapMirror license information.

- S3 license: Optional. Needed only for ONTAP S3 buckets

To check whether your ONTAP system has the required licenses, refer to Manage ONTAP licenses.

Access to the internet

You should determine whether you have outside access to the internet. If you do not, some functionality might be limited, such as receiving monitoring and metrics data from NetApp Cloud Insights, or sending support bundles to the NetApp Support Site.

License

Astra Control Center requires an Astra Control Center license for full functionality. Obtain an evaluation license or full license from NetApp. You need a license to protect your applications and data. Refer to Astra Control Center features for details.

You can try Astra Control Center with an evaluation license, which lets you use Astra Control Center for 90 days from the date you download the license. You can sign up for a free trial by registering here.

To set up the license, refer to use a 90-day evaluation license.

To learn more about how licenses work, see Licensing.

For details about licenses needed for ONTAP storage backends, refer to Supported storage backends.

Ingress for on-premises Kubernetes clusters

You can choose the type of network ingress Astra Control Center uses. By default, Astra Control Center deploys the Astra Control Center gateway (service/traefik) as a cluster-wide resource. Astra Control Center also supports using a service load balancer, if they are permitted in your environment. If you would rather use a service load balancer and you don't already have one configured, you can use the MetalLB load balancer to automatically assign an external IP address to the service. In the internal DNS server configuration, you should point the chosen DNS name for Astra Control Center to the load-balanced IP address.

> The load balancer should use an IP address located in the same subnet as the Astra Control Center worker node IP addresses.

> If you are hosting Astra Control Center on a Tanzu Kubernetes Grid cluster, use the `kubectl get nsxlbmonitors -A` command to see if you already have a service monitor configured to accept ingress traffic. If one exists, you should not install MetalLB, because the existing service monitor will override any new load balancer configuration.

For more information, see Set up ingress for load balancing.

Networking requirements

The operational environment that hosts Astra Control Center communicates using the following TCP ports. You should ensure that these ports are allowed through any firewalls, and configure firewalls to allow any HTTPS egress traffic originating from the Astra network. Some ports require connectivity both ways between the environment hosting Astra Control Center and each managed cluster (noted where applicable).

You can deploy Astra Control Center in a dual-stack Kubernetes cluster, and Astra Control Center can manage applications and storage backends that have been configured for dual-stack operation. For more information about dual-stack cluster requirements, see the Kubernetes documentation.

Source	Destination	Port	Protocol	Purpose
Client PC	Astra Control Center	443	HTTPS	UI / API access - Ensure this port is open both ways between the cluster hosting Astra Control Center and each managed cluster
Metrics consumer	Astra Control Center worker node	9090	HTTPS	Metrics data communication - ensure each managed cluster can access this port on the cluster hosting Astra Control Center (two-way communication required)
Astra Control Center	Hosted Cloud Insights service (https://www.netapp.com/cloud-services/cloud-insights/)	443	HTTPS	Cloud Insights communication
Astra Control Center	Amazon S3 storage bucket provider	443	HTTPS	Amazon S3 storage communication
Astra Control Center	NetApp AutoSupport (https://support.netapp.com)	443	HTTPS	NetApp AutoSupport communication

Supported web browsers

Astra Control Center supports recent versions of Firefox, Safari, and Chrome with a minimum resolution of 1280 x 720.

Additional requirements for application clusters

Keep in mind these requirements if you plan to use these Astra Control Center features:

- **Application cluster requirements**: Cluster management requirements
  - **Managed application requirements**: Application management requirements
  - **Additional requirements for app replication**: Replication prerequisites

Google Anthos cluster requirements

When hosting Astra Control Center on a Google Anthos cluster, note that Google Anthos includes the MetalLB load balancer and the Istio ingress gateway service by default, enabling you to simply use the generic ingress capabilities of Astra Control Center during installation. See Configure Astra Control Center for details.

VMware Tanzu Kubernetes Grid cluster requirements

When hosting Astra Control Center on a VMware Tanzu Kubernetes Grid (TKG) or Tanzu Kubernetes Grid Integrated Edition (TKGi) cluster, keep in mind the following considerations.

- Disable the TKG or TKGi default storage class enforcement on any application clusters intended to be managed by Astra Control. You can do this by editing the `TanzuKubernetesCluster` resource on the namespace cluster.
- Be aware of specific requirements for Astra Trident when you deploy Astra Control Center in a TKG or TKGi environment. For more information, see the Astra Trident documentation.

> The default VMware TKG and TKGi configuration file token expires ten hours after deployment. If you use Tanzu portfolio products, you must generate a Tanzu Kubernetes Cluster configuration file with a non-expiring token to prevent connection issues between Astra Control Center and managed application clusters. For instructions, visit the VMware NSX-T Data Center Product Documentation.

What's next

View the quick start overview.

Quick start for Astra Control Center

Here's an overview of the steps needed to get started with Astra Control Center. The links within each step take you to a page that provides more details.

1 Review Kubernetes cluster requirements

Ensure that your environment meets these requirements.

Kubernetes cluster

- ☐ Ensure your environment meets operational environment requirements
- ☐ Configure ingress for load balancing on-premises Kubernetes clusters

Storage integration

- ☐ Ensure your environment includes the Astra Trident supported version
- ☐ Prepare the worker nodes
- ☐ Configure Astra Trident storage backend
- ☐ Configure Astra Trident storage classes
- ☐ Install Astra Trident volume snapshot controller

- ☐ Create a volume snapshot class

ONTAP credentials

- ☐ Configure ONTAP credentials

2 Download and install Astra Control Center

Complete these installation tasks.

- ☐ Download Astra Control Center from the NetApp Support Site Evaluation downloads page
- ☐ Obtain the NetApp license file:
  - If you are evaluating Astra Control Center, download the evaluation license file
  - If you already purchased Astra Control Center, generate your license file
- ☐ Install Astra Control Center
- ☐ Perform additional optional configuration steps

3 Complete some initial setup tasks

Complete some basic tasks to get started.

- ☐ Add a license
- ☐ Prepare your environment for cluster management
- ☐ Add a cluster
- ☐ Add a storage backend
- ☐ Add a bucket

4 Use Astra Control Center

After you finish setting up Astra Control Center, here's what you might do next.
You can use the Astra Control user interface (UI) or the Astra Control API.

- ☐ Manage apps
- ☐ Protect apps: Configure protection policies and replicate, clone, and migrate apps.
- ☐ Manage accounts: Users, roles, LDAP, credentials, and more
- ☐ Optionally, connect to Cloud Insights: View metrics on the health of your system.

For more information

- Astra Control API
- Upgrade Astra Control Center
- Get help with Astra Control

Installation overview

Choose and complete one of the following Astra Control Center installation procedures:

- Install Astra Control Center using the standard process
- (If you use Red Hat OpenShift) Install Astra Control Center using OpenShift OperatorHub
- Install Astra Control Center with a Cloud Volumes ONTAP storage backend

Depending on your environment, there might be additional configuration needed after you install Astra Control Center:

- Configure Astra Control Center after installation

Install Astra Control Center using the standard process

To install Astra Control Center, download the installation bundle from the NetApp Support Site and perform the following steps. You can use this procedure to install Astra Control Center in internet-connected or air-gapped environments.

Other installation procedures

- **Install with RedHat Openshift OperatorHub**: Use this alternative procedure to install Astra Control Center on Openshift using OperatorHub.
- **Install in the public cloud with Cloud Volumes ONTAP backend**: Use these procedures to install Astra Control Center in Amazon Web Services (AWS), Google Cloud Platform (GCP), or Microsoft Azure with a Cloud Volumes ONTAP storage backend.

For a demonstration of the Astra Control Center installation process, see this video.

What you'll need

- Before you begin installation, prepare your environment for Astra Control Center deployment.
- If you have configured or want to configure pod security policies in your environment, familiarize yourself with pod security policies and how they affect Astra Control Center installation. See Understand pod security policy restrictions.
- Ensure all API services are in a healthy state and available:

```
kubectl get apiservices
```

- Ensure the Astra FQDN you plan to use is routable to this cluster. This means that you either have a DNS entry in your internal DNS server or you are using a core URL route that is already registered.
- If a cert manager already exists in the cluster, you need to perform some prerequisite steps so that Astra Control Center does not attempt to install its own cert manager. By default, Astra Control Center installs its own cert manager during installation.

About this task

The Astra Control Center installation process helps you to do the following:

- Install the Astra components into the `netapp-acc` (or custom-named) namespace.
- Create a default Astra Control Owner admin account.
- Establish an administrative user email address and default initial setup password. This user is assigned the Owner role that is needed for first time login to the UI.
- Determine that all Astra Control Center pods are running.
- Install the Astra Control Center UI.

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

Steps

To install Astra Control Center, do the following steps:

- Download and extract Astra Control Center
- Install the NetApp Astra kubectl plugin
- Add the images to your local registry
- Set up namespace and secret for registries with auth requirements
- Install the Astra Control Center operator
- Configure Astra Control Center
- Complete Astra Control Center and operator installation
- Verify system status
- Set up ingress for load balancing
- Log in to the Astra Control Center UI

Download and extract Astra Control Center

1. Go to the Astra Control Center Evaluation downloads page on the NetApp Support Site.
2. Download the bundle containing Astra Control Center (`astra-control-center-[version].tar.gz`).
3. (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.

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

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

Install the NetApp Astra kubectl plugin

The NetApp Astra kubectl command line plugin saves time when performing common tasks associated with deploying and upgrading Astra Control Center.

What you'll need

NetApp provides plugin binaries for different CPU architectures and operating systems. You need to know which CPU and operating system you have before you perform this task.

Steps

1. 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/
   ```

2. 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 local registry

1. Complete the appropriate step sequence for your container engine:

Docker

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

```
acc.manifest.bundle.yaml
acc/
```

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.manifest.bundle.yaml
acc/
```

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=22.11.0-82
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=22.11.0-82
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://netappdownloads.jfrog.io/docker-astra-control-prod/netapp/astra/acc/22.11.0-82/image:version

Set up namespace and secret for registries with auth requirements

1. Export the KUBECONFIG for the Astra Control Center host cluster:

```
export KUBECONFIG=[file path]
```

> Before you complete the installation, be sure your KUBECONFIG is pointing to the cluster where you want to install Astra Control Center. The KUBECONFIG can contain only one context.

2. If you use a registry that requires authentication, you need to do the following:
   a. Create the `netapp-acc-operator` namespace:

```
kubectl create ns netapp-acc-operator
```

   Response:

```
namespace/netapp-acc-operator created
```

   b. Create a secret for the `netapp-acc-operator` namespace. Add Docker information and run the following command:

> The placeholder `your_registry_path` should match the location of the images that you uploaded earlier (for example, `[Registry_URL]/netapp/astra/astracc/22.11.0-82`).

```
kubectl create secret docker-registry astra-registry-cred -n netapp-
acc-operator --docker-server=[your_registry_path] --docker
-username=[username] --docker-password=[token]
```

   Sample response:

```
secret/astra-registry-cred created
```

> If you delete the namespace after the secret is generated, recreate the namespace and then regenerate the secret for the namespace.

   c. Create the `netapp-acc` (or custom-named) namespace.

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

   Sample response:

```
namespace/netapp-acc created
```

   d. Create a secret for the `netapp-acc` (or custom-named) namespace. Add Docker information and run the following command:

```
kubectl create secret docker-registry astra-registry-cred -n [netapp-acc or custom namespace] --docker-server=[your_registry_path] --docker-username=[username] --docker-password=[token]
```

Response

```
secret/astra-registry-cred created
```

Install the Astra Control Center operator

1. Change the directory:

```
cd manifests
```

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

```
vim astra_control_center_operator_deploy.yaml
```

> An annotated sample YAML follows these steps.

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

```
imagePullSecrets:
- name: astra-registry-cred
```

b. Change `[your_registry_path]` for the `kube-rbac-proxy` image to the registry path where you pushed the images in a previous step.

c. Change `[your_registry_path]` for the `acc-operator-controller-manager` image to the registry path where you pushed the images in a previous step.

```
<strong>astra_control_center_operator_deploy.yaml</strong>
```

```
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: [your_registry_path]/kube-rbac-proxy:v4.8.0
        name: kube-rbac-proxy
        ports:
        - containerPort: 8443
          name: https
      - args:
        - --health-probe-bind-address=:8081
        - --metrics-bind-address=127.0.0.1:8080
        - --leader-elect
        env:
        - name: ACCOP_LOG_LEVEL
          value: "2"
        - name: ACCOP_HELM_INSTALLTIMEOUT
          value: 5m
        image: [your_registry_path]/acc-operator:[version x.y.z]
        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
```

3. Install the Astra Control Center operator:

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

Sample response:

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

4. Verify pods are running:

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

Configure Astra Control Center

1. Edit the Astra Control Center custom resource (CR) file (`astra_control_center.yaml`) to make account, support, registry, and other necessary configurations:

```
vim astra_control_center.yaml
```

> An annotated sample YAML follows these steps.

2. Modify or confirm the following settings:

`accountName`

Setting	Guidance	Type	Example
`accountName`	Change the `accountName` string to the name you want to associate with the Astra Control Center account. There can be only one accountName.	string	`Example`

`astraVersion`

Setting	Guidance	Type	Example
`astraVersion`	The version of Astra Control Center to deploy. No action is needed for this setting as the value will be pre-populated.	string	`22.11.0-82`

`astraAddress`

Setting	Guidance	Type	Example
`astraAddress`	Change the `astraAddress` string to the FQDN (recommended) or IP address you want to use in your browser to access Astra Control Center. This address defines how Astra Control Center will be found in your data center and is the same FQDN or IP address you provisioned from your load balancer when you completed Astra Control Center requirements. NOTE: Do not use `http://` or `https://` in the address. Copy this FQDN for use in a later step.	string	`astra.example.com`

`autoSupport`

Your selections in this section determine whether you will participate in NetApp's pro-active support application, NetApp Active IQ, and where data is sent. An internet connection is required (port 442), and all support data is anonymized.

Setting	Use	Guidance	Type	Example
`autoSupport.enrolled`	Either `enrolled` or `url` fields must be selected	Change `enrolled` for AutoSupport to `false` for sites without internet connectivity or retain `true` for connected sites. A setting of `true` enables anonymous data to be sent to NetApp for support purposes. The default election is `false` and indicates no support data will be sent to NetApp.	Boolean	`false` (this value is the default)
`autoSupport.url`	Either `enrolled` or `url` fields must be selected	This URL determines where the anonymous data will be sent.	string	`https://support.netapp.com/asupprod/post/1.0/postAsup`

`email`

Setting	Guidance	Type	Example
`email`	Change the `email` string to the default initial administrator address. Copy this email address for use in a later step. This email address will be used as the username for the initial account to log in to the UI and will be notified of events in Astra Control.	string	`admin@example.com`

`firstName`

Setting	Guidance	Type	Example
`firstName`	The first name of the default initial administrator associated with the Astra account. The name used here will be visible in a heading in the UI after your first login.	string	`SRE`

`LastName`

Setting	Guidance	Type	Example
`lastName`	The last name of the default initial administrator associated with the Astra account. The name used here will be visible in a heading in the UI after your first login.	string	`Admin`

`imageRegistry`

Your selections in this section define the container image registry that is hosting the Astra application images, Astra Control Center Operator, and Astra Control Center Helm repository.

Setting	Use	Guidance	Type	Example
`imageRegistry.name`	Required	The name of the image registry where you pushed the images in the previous step. Do not use `http://` or `https://` in the registry name.	string	`example.registry.com/astra`
`imageRegistry.secret`	Required if the string you entered for `imageRegistry.name' requires a secret.` `IMPORTANT: If you are using a registry that does not require authorization, you must delete this` `secret` line within `imageRegistry` or the installation will fail.	The name of the Kubernetes secret used to authenticate with the image registry.	string	`astra-registry-cred`

`storageClass`

Setting	Guidance	Type	Example
`storageClass`	Change the `storageClass` value from `ontap-gold` to another Trident storageClass resource as required by your installation. Run the command `kubectl get sc` to determine your existing configured storage classes. One of the Trident-based storage classes must be entered in the manifest file (`astra-control-center-<version>.manifest`) and will be used for Astra PVs. If it is not set, the default storage class will be used. NOTE: If a default storage class is configured, ensure that it is the only storage class that has the default annotation.	string	`ontap-gold`

`volumeReclaimPolicy`

Setting	Guidance	Type	Options
`volumeReclaimPolicy`	This sets the reclaim policy for Astra's PVs. Setting this policy to `Retain` retains persistent volumes after Astra is deleted. Setting this policy to `Delete` deletes persistent volumes after astra is deleted. If this value is not set, the PVs are retained.	string	• `Retain` (This is the default value) • `Delete`

ingressType

Setting	Guidance	Type	Options
`ingressType`	Use one of the following ingress types: **`Generic`** (`ingressType: "Generic"`) (Default) Use this option when you have another ingress controller in use or would prefer to use your own ingress controller. After Astra Control Center is deployed, you will need to configure the ingress controller to expose Astra Control Center with a URL. **`AccTraefik`** (`ingressType: "AccTraefik"`) Use this option when you would prefer not to configure an ingress controller. This deploys the Astra Control Center `traefik` gateway as a Kubernetes LoadBalancer type service. Astra Control Center uses a service of the type "LoadBalancer" (`svc/traefik` in the Astra Control Center namespace), and requires that it be assigned an accessible external IP address. If load balancers are permitted in your environment and you don't already have one configured, you can use MetalLB or another external service load balancer to assign an external IP address to the service. In the internal DNS server configuration, you should point the chosen	string	• `Generic` (this is the default value) • `AccTraefik`

`astraResourcesScaler`

Setting	Guidance	Type	Options
`astraResourcesScaler`	Scaling options for AstraControlCenter Resource limits. By default, Astra Control Center deploys with resource requests set for most of the components within Astra. This configuration allows the Astra Control Center software stack to perform better in environments under increased application load and scale. However, in scenarios using smaller development or test clusters, the CR field `astraResourcesScalar` may be set to `Off`. This disables resource requests and allows for deployment on smaller clusters.	string	• `Default` (This is the default value) • `Off`

`crds`

Your selections in this section determine how Astra Control Center should handle CRDs.

Setting	Guidance	Type	Example
`crds.externalCertManager`	If you use an external cert manager, change `externalCertManager` to `true`. The default `false` causes Astra Control Center to install its own cert manager CRDs during installation. CRDs are cluster-wide objects and installing them might have an impact on other parts of the cluster. You can use this flag to signal to Astra Control Center that these CRDs will be installed and managed by the cluster administrator outside of Astra Control Center.	Boolean	`False` (this value is the default)
`crds.externalTraefik`	By default, Astra Control Center will install required Traefik CRDs. CRDs are cluster-wide objects and installing them might have an impact on other parts of the cluster. You can use this flag to signal to Astra Control Center that these CRDs will be installed and managed by the cluster administrator outside of Astra Control Center.	Boolean	`False` (this value is the default)

```
<strong>astra_control_center.yaml</strong>
```

```
apiVersion: astra.netapp.io/v1
kind: AstraControlCenter
metadata:
  name: astra
spec:
  accountName: "Example"
  astraVersion: "ASTRA_VERSION"
  astraAddress: "astra.example.com"
  autoSupport:
    enrolled: true
  email: "[admin@example.com]"
  firstName: "SRE"
  lastName: "Admin"
  imageRegistry:
    name: "[your_registry_path]"
    secret: "astra-registry-cred"
  storageClass: "ontap-gold"
  volumeReclaimPolicy: "Retain"
  ingressType: "Generic"
  astraResourcesScaler: "Default"
  additionalValues: {}
  crds:
    externalTraefik: false
    externalCertManager: false
```

Complete Astra Control Center and operator installation

1. If you didn't already do so in a previous step, create the `netapp-acc` (or custom) namespace:

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

   Sample response:

   ```
   namespace/netapp-acc created
   ```

2. Install Astra Control Center in the `netapp-acc` (or your custom) namespace:

   ```
   kubectl apply -f astra_control_center.yaml -n [netapp-acc or custom
   namespace]
   ```

   Sample response:

```
astracontrolcenter.astra.netapp.io/astra created
```

Verify system status

You can verify system status using kubectl commands. If you prefer to use OpenShift, you can use comparable oc commands for verification steps.

Steps

1. Verify that all system components installed successfully.

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

   Each pod should have a status of `Running`. It may take several minutes before the system pods are deployed.

Sample response

```
NAME                                          READY   STATUS
RESTARTS        AGE
acc-helm-repo-76d8d845c9-ggds2                1/1     Running   0
14m
activity-6cc67ff9f4-z48mr                     1/1     Running   2
(8m32s ago)   9m
api-token-authentication-7s67v                1/1     Running   0
8m56s
api-token-authentication-bplb4                1/1     Running   0
8m56s
api-token-authentication-p2c9z                1/1     Running   0
8m56s
asup-6cdfbc6795-md8vn                         1/1     Running   0
9m14s
authentication-9477567db-8hnc9                1/1     Running   0
7m4s
bucketservice-f4dbdfcd6-wqzkw                 1/1     Running   0
8m48s
cert-manager-bb756c7c4-wm2cv                  1/1     Running   0
14m
cert-manager-cainjector-c9bb86786-8wrf5       1/1     Running   0
14m
cert-manager-webhook-dd465db99-j2w4x          1/1     Running   0
14m
certificates-68dff9cdd6-kcvml                 1/1     Running   2
(8m43s ago)   9m2s
certificates-68dff9cdd6-rsnsb                 1/1     Running   0
9m2s
cloud-extension-69d48c956c-2s8dt              1/1     Running   3
(8m43s ago)   9m24s
cloud-insights-service-7c4f48b978-7gvlh       1/1     Running   3
(8m50s ago)   9m28s
composite-compute-7d9ff5f68-nxbhl             1/1     Running   0
8m51s
composite-volume-57b4756d64-nl66d             1/1     Running   0
9m13s
credentials-6dbc55f89f-qpzff                  1/1     Running   0
11m
entitlement-67bfb6d7-gl6kp                    1/1     Running   4
(8m33s ago)   9m38s
features-856cc4dccc-mxbdb                     1/1     Running   0
9m20s
fluent-bit-ds-4rtsp                           1/1     Running   0
6m54s
```

```
fluent-bit-ds-9rqll                        1/1     Running   0
6m54s
fluent-bit-ds-w5mp7                        1/1     Running   0
6m54s
graphql-server-7c7cc49776-jz2kn            1/1     Running   0
2m29s
identity-87c59c975-9jpnf                   1/1     Running   0
9m6s
influxdb2-0                                1/1     Running   0
13m
keycloak-operator-84ff6d59d4-qcnmc         1/1     Running   0
7m1s
krakend-cbf6c7df9-mdtzv                    1/1     Running   0
2m30s
license-5b888b78bf-plj6j                   1/1     Running   0
9m32s
login-ui-846b4664dd-fz8hv                  1/1     Running   0
2m24s
loki-0                                     1/1     Running   0
13m
metrics-facade-779cc9774-n26rw             1/1     Running   0
9m18s
monitoring-operator-974db78f-pkspq         2/2     Running   0
6m58s
nats-0                                     1/1     Running   0
13m
nats-1                                     1/1     Running   0
13m
nats-2                                     1/1     Running   0
13m
nautilus-7bdc7ddc54-49tfn                  1/1     Running   0
7m50s
nautilus-7bdc7ddc54-cwc79                  1/1     Running   0
9m36s
openapi-5584ff9f46-gbrdj                   1/1     Running   0
9m17s
openapi-5584ff9f46-z9mzk                   1/1     Running   0
9m17s
packages-bfc58cc98-lpxq9                   1/1     Running   0
8m58s
polaris-consul-consul-server-0             1/1     Running   0
13m
polaris-consul-consul-server-1             1/1     Running   0
13m
polaris-consul-consul-server-2             1/1     Running   0
13m
```

```
polaris-keycloak-0                          1/1     Running   3
(6m15s ago)   6m56s
polaris-keycloak-1                          1/1     Running   0
4m22s
polaris-keycloak-2                          1/1     Running   0
3m41s
polaris-keycloak-db-0                       1/1     Running   0
6m56s
polaris-keycloak-db-1                       1/1     Running   0
4m23s
polaris-keycloak-db-2                       1/1     Running   0
3m36s
polaris-mongodb-0                           2/2     Running   0
13m
polaris-mongodb-1                           2/2     Running   0
13m
polaris-mongodb-2                           2/2     Running   0
12m
polaris-ui-5ccff47897-8rzgh                 1/1     Running   0
2m33s
polaris-vault-0                             1/1     Running   0
13m
polaris-vault-1                             1/1     Running   0
13m
polaris-vault-2                             1/1     Running   0
13m
public-metrics-6cb7bfc49b-p54xm             1/1     Running   1
(8m29s ago)   9m31s
storage-backend-metrics-5c77994586-kjn48    1/1     Running   0
8m52s
storage-provider-769fdc858c-62w54           1/1     Running   0
8m54s
task-service-9ffc484c5-kx9f4                1/1     Running   3
(8m44s ago)   9m34s
telegraf-ds-bphb9                           1/1     Running   0
6m54s
telegraf-ds-rtsm2                           1/1     Running   0
6m54s
telegraf-ds-s9h5h                           1/1     Running   0
6m54s
telegraf-rs-lbpv7                           1/1     Running   0
6m54s
telemetry-service-57cfb998db-zjx78          1/1     Running   1
(8m40s ago)   9m26s
tenancy-5d5dfbcf9f-vmbxh                    1/1     Running   0
9m5s
```

```
traefik-7b87c4c474-jmgp2                    1/1     Running   0
2m24s
traefik-7b87c4c474-t9k8x                    1/1     Running   0
2m24s
trident-svc-c78f5b6bd-nwdsq                 1/1     Running   0
9m22s
vault-controller-55bbc96668-c6425           1/1     Running   0
11m
vault-controller-55bbc96668-lq9n9           1/1     Running   0
11m
vault-controller-55bbc96668-rfkgg           1/1     Running   0
11m
```

2. (Optional) To ensure the installation is completed, you can watch the `acc-operator` logs using the following command.

```
kubectl logs deploy/acc-operator-controller-manager -n netapp-acc-
operator -c manager -f
```

> `accHost` cluster registration is one of the last operations, and if it fails it will not cause deployment to fail. In the event of a cluster registration failure indicated in the logs, you can attempt registration again through the Add cluster workflow in the UI or API.

3. When all the pods are running, verify that the installation was successful (`READY` is `True`) and get the initial setup password you will use when you log in to Astra Control Center:

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

Response:

```
NAME    UUID                                   VERSION     ADDRESS
READY
astra   9aa5fdae-4214-4cb7-9976-5d8b4c0ce27f  22.11.0-82  10.111.111.111
True
```

> Copy the UUID value. The password is `ACC-` followed by the UUID value (`ACC-[UUID]` or, in this example, `ACC-9aa5fdae-4214-4cb7-9976-5d8b4c0ce27f`).

Set up ingress for load balancing

You can set up a Kubernetes ingress controller that manages external access to services. These procedures give setup examples for an ingress controller if you used the default of `ingressType: "Generic"` in the

Astra Control Center custom resource (`astra_control_center.yaml`). You do not need to use this procedure if you specified `ingressType: "AccTraefik"` in the Astra Control Center custom resource (`astra_control_center.yaml`).

After Astra Control Center is deployed, you will need to configure the ingress controller to expose Astra Control Center with a URL.

Setup steps differ depending on the type of ingress controller you use. Astra Control Center supports many ingress controller types. These setup procedures provide example steps for the following ingress controller types:

- Istio ingress
- Nginx ingress controller
- OpenShift ingress controller

What you'll need

- The required ingress controller should already be deployed.
- The ingress class corresponding to the ingress controller should already be created.

Steps for Istio ingress

1. Configure Istio ingress.

   This procedure assumes that Istio is deployed using the "default" configuration profile.

2. Gather or create the desired certificate and private key file for the Ingress Gateway.

   You can use a CA-signed or self-signed certificate. The common name must be the Astra address (FQDN).

   Sample command:

   ```
   openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout tls.key -out
   tls.crt
   ```

3. Create a secret `tls secret name` of type `kubernetes.io/tls` for a TLS private key and certificate in the `istio-system namespace` as described in TLS secrets.

   Sample command:

   ```
   kubectl create secret tls [tls secret name] --key="tls.key"
   --cert="tls.crt" -n istio-system
   ```

   The name of the secret should match the `spec.tls.secretName` provided in `istio-ingress.yaml` file.

4. Deploy an ingress resource in the `netapp-acc` (or custom-named) namespace using the v1 resource type for a schema (`istio-Ingress.yaml` is used in this example):

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

5. Apply the changes:

```
kubectl apply -f istio-Ingress.yaml
```

6. Check the status of the ingress:

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

Response:

```
NAME    CLASS HOSTS             ADDRESS         PORTS   AGE
ingress istio astra.example.com 172.16.103.248  80, 443 1h
```

7. Finish Astra Control Center installation.

Steps for Nginx ingress controller

1. Create a secret of type `kubernetes.io/tls` for a TLS private key and certificate in `netapp-acc` (or custom-named) namespace as described in TLS secrets.
2. Deploy an ingress resource in `netapp-acc` (or custom-named) namespace using the v1 resource type for a schema (`nginx-Ingress.yaml` is used in this example):

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

3. Apply the changes:

```
kubectl apply -f nginx-Ingress.yaml
```

NetApp recommends installing the nginx controller as a deployment rather than a `daemonSet`.

Steps for OpenShift ingress controller

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

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

Log in to the Astra Control Center UI

After installing Astra Control Center, you will change the password for the default administrator and log in to the Astra Control Center UI dashboard.

Steps

1. In a browser, enter the FQDN (including the `https://` prefix) you used in the `astraAddress` in the `astra_control_center.yaml` CR when you installed Astra Control Center.
2. Accept the self-signed certificates if prompted.

   > You can create a custom certificate after login.

3. At the Astra Control Center login page, enter the value you used for `email` in `astra_control_center.yaml` CR when you installed Astra Control Center, followed by the initial setup password (`ACC-[UUID]`).

   > If you enter an incorrect password three times, the admin account will be locked for 15 minutes.

4. Select **Login**.
5. Change the password when prompted.

   > If this is your first login and you forget the password and no other administrative user accounts have yet been created, contact NetApp Support for password recovery assistance.

6. (Optional) Remove the existing self-signed TLS certificate and replace it with a custom TLS certificate signed by a Certificate Authority (CA).

Troubleshoot the installation

If any of the services are in `Error` status, you can inspect the logs. Look for API response codes in the 400 to 500 range. Those indicate the place where a failure happened.

Steps

1. To inspect the Astra Control Center operator logs, enter the following:

```
kubectl logs deploy/acc-operator-controller-manager -n netapp-acc-
operator -c manager -f
```

What's next

- (Optional) Depending on your environment, complete post-installation configuration steps.
- Complete the deployment by performing setup tasks.

Install Astra Control Center using OpenShift OperatorHub

If you use Red Hat OpenShift, you can install Astra Control Center using the Red Hat certified operator. Use this procedure to install Astra Control Center from the Red Hat Ecosystem Catalog or using the Red Hat OpenShift Container Platform.

After you complete this procedure, you must return to the installation procedure to complete the remaining steps to verify installation success and log on.

What you'll need

- **Environmental prerequisites met**: Before you begin installation, prepare your environment for Astra Control Center deployment.
- **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
    ```

- **FQDN address**: Obtain an FQDN address for Astra Control Center in your data center.
- **Openshift Permissions**: Obtain the necessary permissions and access to the Red Hat OpenShift Container Platform to perform the installation steps described.
- **cert manager configured**: If a cert manager already exists in the cluster, you need to perform some prerequisite steps so that Astra Control Center does not install its own cert manager. By default, Astra Control Center installs its own cert manager during installation.
- **Kubernetes ingress controller**: If you have a Kubernetes ingress controller that manages external access to services, such as load balancing in a cluster, you need to set it up for use with Astra Control Center:
  a. Create the operator namespace:

     ```
     oc create namespace netapp-acc-operator
     ```

  b. Complete setup for your ingress controller type.

Steps

- Download and extract Astra Control Center
- Install the NetApp Astra kubectl plugin
- Add the images to your local registry

- Find the operator install page
- Install the operator
- Install Astra Control Center

Download and extract Astra Control Center

1. Go to the Astra Control Center Evaluation downloads page on the NetApp Support Site.
2. Download the bundle containing Astra Control Center (`astra-control-center-[version].tar.gz`).
3. (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.

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

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

Install the NetApp Astra kubectl plugin

The NetApp Astra kubectl command line plugin saves time when performing common tasks associated with deploying and upgrading Astra Control Center.

What you'll need

NetApp provides plugin binaries for different CPU architectures and operating systems. You need to know which CPU and operating system you have before you perform this task.

Steps

1. 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/
```

2. 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 local registry

1. Complete the appropriate step sequence for your container engine:

Docker

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

```
acc.manifest.bundle.yaml
acc/
```

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.manifest.bundle.yaml
acc/
```

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=22.11.0-82
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>
```

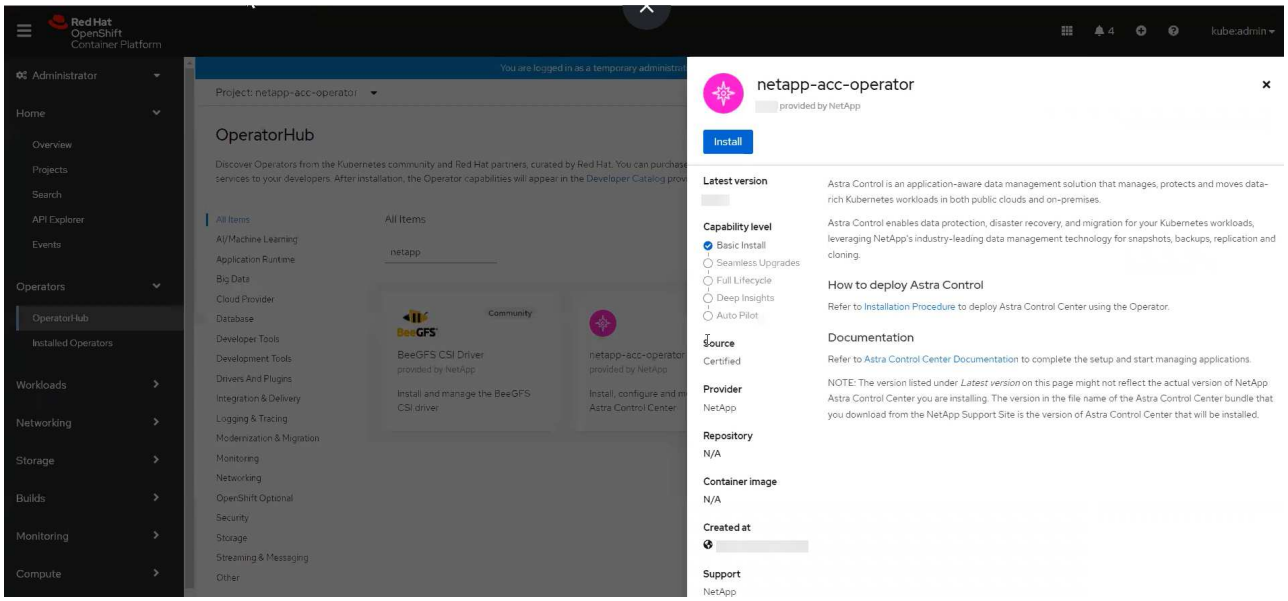
```
export REGISTRY=<MY_FULL_REGISTRY_PATH>
export PACKAGENAME=acc
export PACKAGEVERSION=22.11.0-82
export DIRECTORYNAME=acc
for astraImageFile in $(ls ${DIRECTORYNAME}/images/*.tar) ; do
astraImage=$(podman load --input ${astraImageFile} | sed 's/Loaded
image: //')
astraImageNoPath=$(echo ${astraImage} | 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://netappdownloads.jfrog.io/docker-astra-control-prod/netapp/astra/acc/22.11.0-82/image:version`

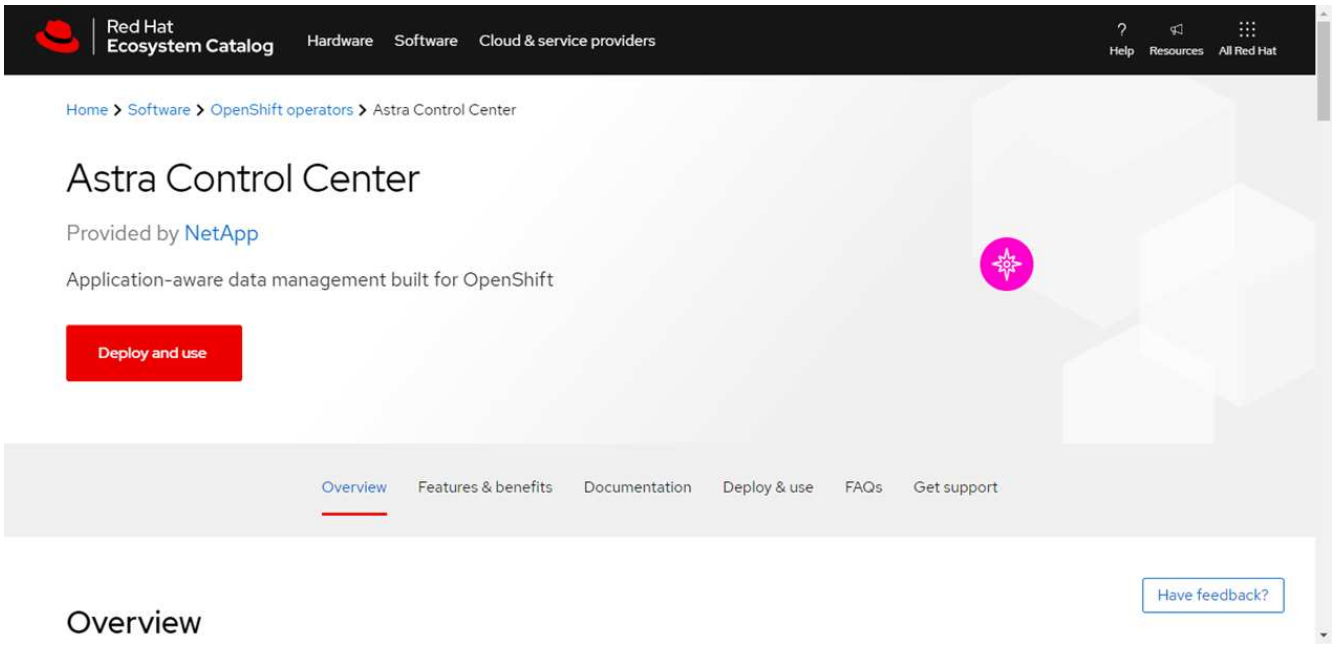
Find the operator install page

1. Complete one of the following procedures to access the operator install page:
   - From Red Hat Openshift web console:
     a. Log in to the OpenShift Container Platform UI.
     b. From the side menu, select **Operators > OperatorHub**.

c. Search for and select the NetApp Astra Control Center operator.



- From Red Hat Ecosystem Catalog:
  a. Select the NetApp Astra Control Center operator.
  b. Select **Deploy and Use**.



Install the operator

1. Complete the **Install Operator** page and install the operator:

> The operator will be available in all cluster namespaces.

a. Select the operator namespace or `netapp-acc-operator` namespace will be created automatically as part of the operator installation.

b. Select a manual or automatic approval strategy.

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

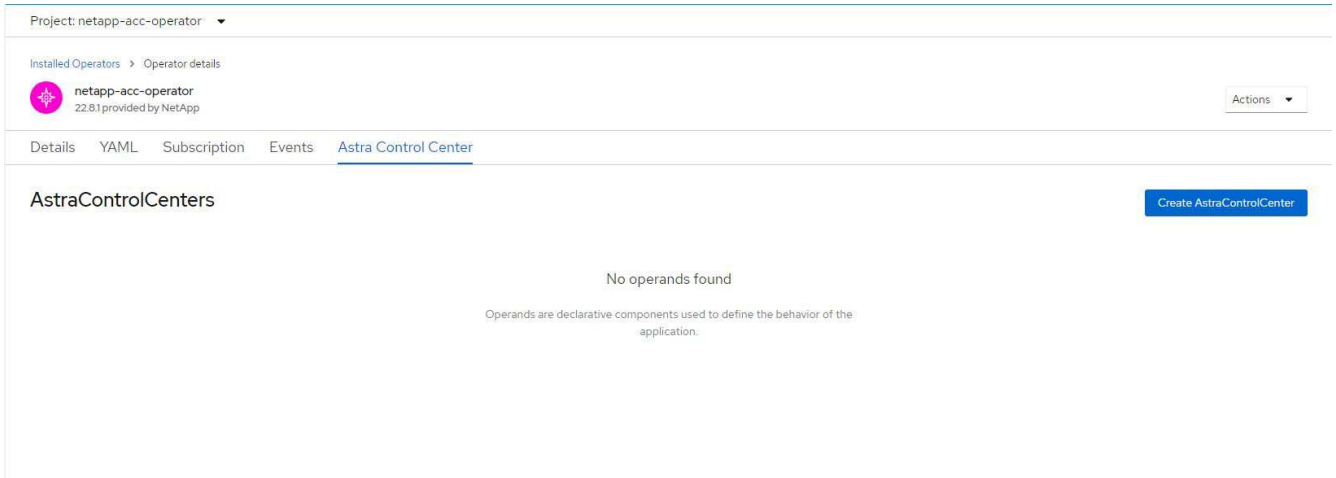
c. Select **Install**.

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

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

Install Astra Control Center

1. From the console within the **Astra Control Center** tab of the Astra Control Center operator, select **Create AstraControlCenter**.



2. Complete the `Create AstraControlCenter` form field:
   a. Keep or adjust the Astra Control Center name.
   b. Add labels for the Astra Control Center.
   c. Enable or disable Auto Support. Retaining Auto Support functionality is recommended.
   d. Enter the Astra Control Center FQDN or IP address. Do not enter `http://` or `https://` in the address field.
   e. Enter the Astra Control Center version; for example, 22.04.1.
   f. Enter an account name, email address, and admin last name.
   g. Choose a volume reclaim policy of `Retain`, `Recycle`, or `Delete`. The default value is `Retain`.
   h. Select the ingress type:
      - **`Generic`** (`ingressType: "Generic"`) (Default)

        Use this option when you have another ingress controller in use or would prefer to use your own ingress controller. After Astra Control Center is deployed, you will need to configure the ingress controller to expose Astra Control Center with a URL.

      - **`AccTraefik`** (`ingressType: "AccTraefik"`)

        Use this option when you would prefer not to configure an ingress controller. This deploys the Astra Control Center `traefik` gateway as a Kubernetes "LoadBalancer" type service.

Astra Control Center uses a service of the type "LoadBalancer" (`svc/traefik` in the Astra Control Center namespace), and requires that it be assigned an accessible external IP address. If load balancers are permitted in your environment and you don't already have one configured, you can use MetalLB or another external service load balancer to assign an external IP address to the service. In the internal DNS server configuration, you should point the chosen DNS name for Astra Control Center to the load-balanced IP address.

> For details about the service type of "LoadBalancer" and ingress, see Requirements.

i. In **Image Registry**, enter your local container image registry path. Do not enter `http://` or `https://` in the address field.

j. If you use an image registry that requires authentication, enter the image secret.

> If you use a registry that requires authentication, create a secret on the cluster.

k. Enter the admin first name.

l. Configure resources scaling.

m. Provide the default storage class.

> If a default storage class is configured, ensure that it is the only storage class that has the default annotation.

n. Define CRD handling preferences.

3. Select the YAML view to review the settings you have selected.
4. Select `Create`.

Create a registry secret

If you use a registry that requires authentication, create a secret on the Openshift cluster and enter the secret name in the `Create AstraControlCenter` form field.

1. Create a namespace for the Astra Control Center operator:

```
oc create ns [netapp-acc-operator or custom namespace]
```

2. Create a secret in this namespace:

```
oc create secret docker-registry astra-registry-cred n [netapp-acc-
operator or custom namespace] --docker-server=[your_registry_path]
--docker username=[username] --docker-password=[token]
```

> Astra Control supports Docker registry secrets only.

3. Complete the remaining fields in the Create AstraControlCenter form field.

What's next

Complete the remaining steps to verify that Astra Control Center installed successfully, set up an ingress controller (optional), and log in to the UI. Additionally, you will need to perform setup tasks after completing installation.

Install Astra Control Center with a Cloud Volumes ONTAP storage backend

With Astra Control Center, you can manage your apps in a hybrid cloud environment with self-managed Kubernetes clusters and Cloud Volumes ONTAP instances. You can deploy Astra Control Center in your on-premise Kubernetes clusters or in one of the self-managed Kubernetes clusters in the cloud environment.

With one of these deployments, you can perform app data management operations using Cloud Volumes ONTAP as a storage backend. You can also configure an S3 bucket as the backup target.

To install Astra Control Center in Amazon Web Services (AWS), Google Cloud Platform (GCP) and Microsoft Azure with a Cloud Volumes ONTAP storage backend, perform the following steps depending on your cloud environment.

- Deploy Astra Control Center in Amazon Web Services
- Deploy Astra Control Center in Google Cloud Platform
- Deploy Astra Control Center in Microsoft Azure

You can manage your apps in distributions with self-managed Kubernetes clusters, such with OpenShift Container Platform (OCP). Only self-managed OCP clusters are validated for deploying Astra Control Center.

Deploy Astra Control Center in Amazon Web Services

You can deploy Astra Control Center on a self-managed Kubernetes cluster hosted on an Amazon Web Services (AWS) public cloud.

What you'll need for AWS

Before you deploy Astra Control Center in AWS, you will need the following items:

- Astra Control Center license. See Astra Control Center licensing requirements.
- Meet Astra Control Center requirements.
- NetApp Cloud Central account
- If using OCP, Red Hat OpenShift Container Platform (OCP) permissions (on namespace level to create pods)
- AWS credentials, Access ID and Secret Key with permissions that enable you to create buckets and connectors
- AWS account Elastic Container Registry (ECR) access and login
- AWS hosted zone and Route 53 entry required to access the Astra Control UI

Operational environment requirements for AWS

Astra Control Center requires the following operational environment for AWS:

- Red Hat OpenShift Container Platform 4.8

> Ensure that the operating environment you choose to host Astra Control Center meets the basic resource requirements outlined in the environment's official documentation.

Astra Control Center requires the following resources in addition to the environment's resource requirements:

Component	Requirement
Backend NetApp Cloud Volumes ONTAP storage capacity	At least 300GB available
Worker nodes (AWS EC2 requirement)	At least 3 worker nodes total, with 4 vCPU cores and 12GB RAM each
Load balancer	Service type "LoadBalancer" available for ingress traffic to be sent to services in the operational environment cluster
FQDN	A method for pointing the FQDN of Astra Control Center to the load balanced IP address
Astra Trident (installed as part of the Kubernetes cluster discovery in NetApp BlueXP, formerly Cloud Manager)	Astra Trident 21.04 or newer installed and configured and NetApp ONTAP version 9.5 or newer as a storage backend
Image registry	You must have an existing private registry, such as AWS Elastic Container Registry, to which you can push Astra Control Center build images. You need to provide the URL of the image registry where you will upload the images. The Astra Control Center hosted cluster and the managed cluster must have access to the same image registry to be able to back up and restore apps using the Restic-based image.
Astra Trident / ONTAP configuration	Astra Control Center requires that a storage class be created and set as the default storage class. Astra Control Center supports the following ONTAP Kubernetes storage classes that are created when you import your Kubernetes cluster into NetApp BlueXP (formerly Cloud Manager). These are provided by Astra Trident: • `vsaworkingenvironment-<>-ha-nas csi.trident.netapp.io` • `vsaworkingenvironment-<>-ha-san csi.trident.netapp.io` • `vsaworkingenvironment-<>-single-nas csi.trident.netapp.io` • `vsaworkingenvironment-<>-single-san csi.trident.netapp.io`

These requirements assume that Astra Control Center is the only application running in the operational environment. If the environment is running additional applications, adjust these minimum requirements accordingly.

The AWS registry token expires in 12 hours, after which you will have to renew the Docker image registry secret.

Overview of deployment for AWS

Here is an overview of the process to install Astra Control Center for AWS with Cloud Volumes ONTAP as a storage backend.

Each of these steps is explained in more detail below.

1. Ensure that you have sufficient IAM permissions.
2. Install a RedHat OpenShift cluster on AWS.
3. Configure AWS.
4. Configure NetApp BlueXP for AWS.
5. Install Astra Control Center for AWS.

Ensure that you have sufficient IAM permissions

Ensure that you have sufficient IAM roles and permissions that enable you to install a RedHat OpenShift cluster and a NetApp BlueXP (formerly Cloud Manager) Connector.

See Initial AWS credentials.

Install a RedHat OpenShift cluster on AWS

Install a RedHat OpenShift Container Platform cluster on AWS.

For installation instructions, see Installing a cluster on AWS in OpenShift Container Platform.

Configure AWS

Next, configure AWS to create a virtual network, set up EC2 compute instances, create an AWS S3 bucket, create an Elastic Container Register (ECR) to host the Astra Control Center images, and push the images to this registry.

Follow the AWS documentation to complete the following steps. See AWS installation documentation.

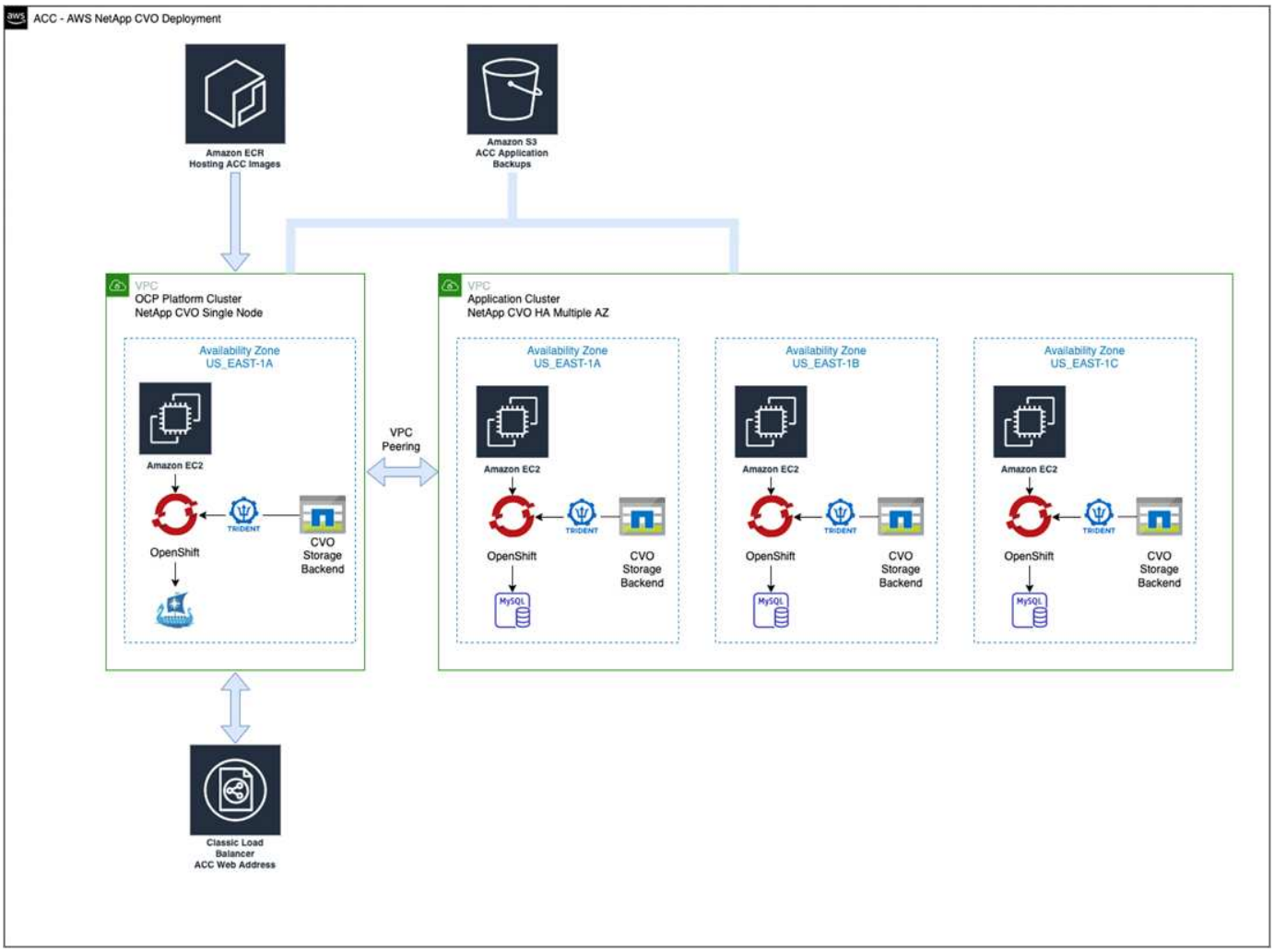
1. Create an AWS virtual network.
2. Review the EC2 compute instances. This can be a bare metal server or VMs in AWS.
3. If the instance type does not already match the Astra minimum resource requirements for master and worker nodes, change the instance type in AWS to meet the Astra requirements. See Astra Control Center requirements.
4. Create at least one AWS S3 bucket to store your backups.
5. Create an AWS Elastic Container Registry (ECR) to host all the ACC images.

> If you do not create the ECR, Astra Control Center cannot access monitoring data from a cluster containing Cloud Volumes ONTAP with an AWS backend. The issue is caused when the cluster you try to discover and manage using Astra Control Center does not have AWS ECR access.

6. Push the ACC images to your defined registry.

> The AWS Elastic Container Registry (ECR) token expires after 12 hours and causes cross-cluster clone operations to fail. This issue occurs when managing a storage backend from Cloud Volumes ONTAP configured for AWS. To correct this issue, authenticate with the ECR again and generate a new secret for clone operations to resume successfully.

Here's an example of an AWS deployment:



Configure NetApp BlueXP for AWS

Using NetApp BlueXP (formerly Cloud Manager), create a workspace, add a connector to AWS, create a working environment, and import the cluster.

Follow the BlueXP documentation to complete the following steps. See the following:

- Getting started with Cloud Volumes ONTAP in AWS.

- Create a connector in AWS using BlueXP

Steps

1. Add your credentials to BlueXP.
2. Create a workspace.
3. Add a connector for AWS. Choose AWS as the Provider.
4. Create a working environment for your cloud environment.
   a. Location: "Amazon Web Services (AWS)"
   b. Type: "Cloud Volumes ONTAP HA"
5. Import the OpenShift cluster. The cluster will connect to the working environment you just created.
   a. View the NetApp cluster details by selecting **K8s** > **Cluster list** > **Cluster Details**.
   b. In the upper right corner, note the Trident version.
   c. Note the Cloud Volumes ONTAP cluster storage classes showing NetApp as the provisioner.

      This imports your Red Hat OpenShift cluster and assigns it a default storage class. You select the storage class.
      Trident is automatically installed as part of the import and discovery process.
6. Note all the persistent volumes and volumes in this Cloud Volumes ONTAP deployment.

> Cloud Volumes ONTAP can operate as a single node or in High Availability. If HA is enabled, note the HA status and node deployment status running in AWS.

Install Astra Control Center for AWS

Follow the standard Astra Control Center installation instructions.

> AWS uses the Generic S3 bucket type.

Deploy Astra Control Center in Google Cloud Platform

You can deploy Astra Control Center on a self-managed Kubernetes cluster hosted on a Google Cloud Platform (GCP) public cloud.

What you'll need for GCP

Before you deploy Astra Control Center in GCP, you will need the following items:

- Astra Control Center license. See Astra Control Center licensing requirements.
- Meet Astra Control Center requirements.
- NetApp Cloud Central account
- If using OCP, Red Hat OpenShift Container Platform (OCP) 4.10
- If using OCP, Red Hat OpenShift Container Platform (OCP) permissions (on namespace level to create pods)
- GCP Service Account with permissions that enable you to create buckets and connectors

Operational environment requirements for GCP

> Ensure that the operating environment you choose to host Astra Control Center meets the basic resource requirements outlined in the environment's official documentation.

Astra Control Center requires the following resources in addition to the environment's resource requirements:

Component	Requirement
Backend NetApp Cloud Volumes ONTAP storage capacity	At least 300GB available
Worker nodes (GCP compute requirement)	At least 3 worker nodes total, with 4 vCPU cores and 12GB RAM each
Load balancer	Service type "LoadBalancer" available for ingress traffic to be sent to services in the operational environment cluster
FQDN (GCP DNS zone)	A method for pointing the FQDN of Astra Control Center to the load balanced IP address
Astra Trident (installed as part of the Kubernetes cluster discovery in NetApp BlueXP, formerly Cloud Manager)	Astra Trident 21.04 or newer installed and configured and NetApp ONTAP version 9.5 or newer as a storage backend
Image registry	You must have an existing private registry, such as Google Container Registry, to which you can push Astra Control Center build images. You need to provide the URL of the image registry where you will upload the images. You need to enable anonymous access to pull Restic images for backups.
Astra Trident / ONTAP configuration	Astra Control Center requires that a storage class be created and set as the default storage class. Astra Control Center supports the following ONTAP Kubernetes storage classes that are created when you import your Kubernetes cluster into NetApp BlueXP. These are provided by Astra Trident: • `vsaworkingenvironment-<>-ha-nas csi.trident.netapp.io` • `vsaworkingenvironment-<>-ha-san csi.trident.netapp.io` • `vsaworkingenvironment-<>-single-nas csi.trident.netapp.io` • `vsaworkingenvironment-<>-single-san csi.trident.netapp.io`

These requirements assume that Astra Control Center is the only application running in the operational environment. If the environment is running additional applications, adjust these minimum requirements accordingly.

Overview of deployment for GCP

Here is an overview of the process to install Astra Control Center on a self-managed OCP cluster in GCP with Cloud Volumes ONTAP as a storage backend.

Each of these steps is explained in more detail below.

1. Install a RedHat OpenShift cluster on GCP.
2. Create a GCP Project and Virtual Private Cloud.
3. Ensure that you have sufficient IAM permissions.
4. Configure GCP.
5. Configure NetApp BlueXP for GCP.
6. Install Astra Control Center for GCP.

Install a RedHat OpenShift cluster on GCP

The first step is to install a RedHat OpenShift cluster on GCP.

For installation instructions, see the following:

- Installing an OpenShift cluster in GCP
- Creating a GCP Service Account

Create a GCP Project and Virtual Private Cloud

Create at least one GCP Project and Virtual Private Cloud (VPC).

OpenShift might create its own resource groups. In addition to these, you should also define a GCP VPC. Refer to OpenShift documentation.

You might want to create a platform cluster resource group and a target app OpenShift cluster resource group.

Ensure that you have sufficient IAM permissions

Ensure that you have sufficient IAM roles and permissions that enable you to install a RedHat OpenShift cluster and a NetApp BlueXP (formerly Cloud Manager) Connector.

See Initial GCP credentials and permissions.

Configure GCP

Next, configure GCP to create a VPC, set up compute instances, create a Google Cloud Object Storage, create an Google Container Register to host the Astra Control Center images, and push the images to this registry.

Follow the GCP documentation to complete the following steps. See Installing OpenShift cluster in GCP.

1. Create a GCP Project and VPC in the GCP that you plan on using for the OCP cluster with CVO backend.
2. Review the compute instances. This can be a bare metal server or VMs in GCP.
3. If the instance type does not already match the Astra minimum resource requirements for master and worker nodes, change the instance type in GCP to meet the Astra requirements. See Astra Control Center requirements.
4. Create at least one GCP Cloud Storage Bucket to store your backups.
5. Create a secret, which is required for bucket access.
6. Create a Google Container Registry to host all the Astra Control Center images.
7. Set up Google Container Registry access for Docker push/pull for all the Astra Control Center images.

   Example: ACC images can be pushed to this registry by entering the following script:

   ```
   gcloud auth activate-service-account <service account email address>
   --key-file=<GCP Service Account JSON file>
   ```

   This script requires an Astra Control Center manifest file and your Google Image Registry location.

   Example:

   ```
   manifestfile=astra-control-center-<version>.manifest
   GCP_CR_REGISTRY=<target image repository>
   ASTRA_REGISTRY=<source ACC image repository>

   while IFS= read -r image; do
       echo "image: $ASTRA_REGISTRY/$image $GCP_CR_REGISTRY/$image"
       root_image=${image%:*}
       echo $root_image
       docker pull $ASTRA_REGISTRY/$image
       docker tag $ASTRA_REGISTRY/$image $GCP_CR_REGISTRY/$image
       docker push $GCP_CR_REGISTRY/$image
   done < astra-control-center-22.04.41.manifest
   ```

8. Set up DNS zones.

Configure NetApp BlueXP for GCP

Using NetApp BlueXP (formerly Cloud Manager), create a workspace, add a connector to GCP, create a working environment, and import the cluster.

Follow the BlueXP documentation to complete the following steps. See Getting started with Cloud Volumes ONTAP in GCP.

What you'll need

- Access to the GCP Service Account with the required IAM permissions and roles

Steps

1. Add your credentials to BlueXP. See Adding GCP accounts.
2. Add a connector for GCP.
   a. Choose "GCP" as the Provider.
   b. Enter GCP credentials. See Creating a connector in GCP from BlueXP.
   c. Ensure that the connector is running and switch to that connector.
3. Create a working environment for your cloud environment.
   a. Location: "GCP"
   b. Type: "Cloud Volumes ONTAP HA"
4. Import the OpenShift cluster. The cluster will connect to the working environment you just created.
   a. View the NetApp cluster details by selecting **K8s** > **Cluster list** > **Cluster Details**.
   b. In the upper right corner, note the Trident version.
   c. Note the Cloud Volumes ONTAP cluster storage classes showing "NetApp" as the provisioner.

      This imports your Red Hat OpenShift cluster and assigns it a default storage class. You select the storage class.
      Trident is automatically installed as part of the import and discovery process.

5. Note all the persistent volumes and volumes in this Cloud Volumes ONTAP deployment.

> Cloud Volumes ONTAP can operate as a single node or in High Availability (HA). If HA is enabled, note the HA status and node deployment status running in GCP.

Install Astra Control Center for GCP

Follow the standard Astra Control Center installation instructions.

> GCP uses the Generic S3 bucket type.

1. Generate the Docker Secret to pull images for the Astra Control Center installation:

```
kubectl create secret docker-registry <secret name> --docker
-server=<Registry location> --docker-username=_json_key --docker
-password="$(cat <GCP Service Account JSON file>)" --namespace=pcloud
```

Deploy Astra Control Center in Microsoft Azure

You can deploy Astra Control Center on a self-managed Kubernetes cluster hosted on a Microsoft Azure public cloud.

What you'll need for Azure

Before you deploy Astra Control Center in Azure, you will need the following items:

- Astra Control Center license. See Astra Control Center licensing requirements.
- Meet Astra Control Center requirements.

- NetApp Cloud Central account
- If using OCP, Red Hat OpenShift Container Platform (OCP) 4.8
- If using OCP, Red Hat OpenShift Container Platform (OCP) permissions (on namespace level to create pods)
- Azure credentials with permissions that enable you to create buckets and connectors

Operational environment requirements for Azure

Ensure that the operating environment you choose to host Astra Control Center meets the basic resource requirements outlined in the environment's official documentation.

Astra Control Center requires the following resources in addition to the environment's resource requirements:

See Astra Control Center operational environment requirements.

Component	Requirement
Backend NetApp Cloud Volumes ONTAP storage capacity	At least 300GB available
Worker nodes (Azure compute requirement)	At least 3 worker nodes total, with 4 vCPU cores and 12GB RAM each
Load balancer	Service type "LoadBalancer" available for ingress traffic to be sent to services in the operational environment cluster
FQDN (Azure DNS zone)	A method for pointing the FQDN of Astra Control Center to the load balanced IP address
Astra Trident (installed as part of the Kubernetes cluster discovery in NetApp BlueXP)	Astra Trident 21.04 or newer installed and configured and NetApp ONTAP version 9.5 or newer will be used as a storage backend
Image registry	You must have an existing private registry, such as Azure Container Registry (ACR), to which you can push Astra Control Center build images. You need to provide the URL of the image registry where you will upload the images. You need to enable anonymous access to pull Restic images for backups.

Component	Requirement
Astra Trident / ONTAP configuration	Astra Control Center requires that a storage class be created and set as the default storage class. Astra Control Center supports the following ONTAP Kubernetes storage classes that are created when you import your Kubernetes cluster into NetApp BlueXP. These are provided by Astra Trident: • `vsaworkingenvironment-<>-ha-nas csi.trident.netapp.io` • `vsaworkingenvironment-<>-ha-san csi.trident.netapp.io` • `vsaworkingenvironment-<>-single-nas csi.trident.netapp.io` • `vsaworkingenvironment-<>-single-san csi.trident.netapp.io`

These requirements assume that Astra Control Center is the only application running in the operational environment. If the environment is running additional applications, adjust these minimum requirements accordingly.

Overview of deployment for Azure

Here is an overview of the process to install Astra Control Center for Azure.

Each of these steps is explained in more detail below.

1. Install a RedHat OpenShift cluster on Azure.
2. Create Azure resource groups.
3. Ensure that you have sufficient IAM permissions.
4. Configure Azure.
5. Configure NetApp BlueXP (formerly Cloud Manager) for Azure.
6. Install and configure Astra Control Center for Azure.

Install a RedHat OpenShift cluster on Azure

The first step is to install a RedHat OpenShift cluster on Azure.

For installation instructions, see the following:

- Installing OpenShift cluster on Azure.
- Installing an Azure account.

Create Azure resource groups

Create at least one Azure resource group.

> OpenShift might create its own resource groups. In addition to these, you should also define Azure resource groups. Refer to OpenShift documentation.

You might want to create a platform cluster resource group and a target app OpenShift cluster resource group.

Ensure that you have sufficient IAM permissions

Ensure that you have sufficient IAM roles and permissions that enable you to install a RedHat OpenShift cluster and a NetApp BlueXP Connector.

See Azure credentials and permissions.

Configure Azure

Next, configure Azure to create a virtual network, set up compute instances, create an Azure Blob container, create an Azure Container Register (ACR) to host the Astra Control Center images, and push the images to this registry.

Follow the Azure documentation to complete the following steps. See Installing OpenShift cluster on Azure.

1. Create an Azure virtual network.
2. Review the compute instances. This can be a bare metal server or VMs in Azure.
3. If the instance type does not already match the Astra minimum resource requirements for master and worker nodes, change the instance type in Azure to meet the Astra requirements. See Astra Control Center requirements.
4. Create at least one Azure Blob container to store your backups.
5. Create a storage account. You will need a storage account to create a container to be used as a bucket in Astra Control Center.
6. Create a secret, which is required for bucket access.
7. Create an Azure Container Registry (ACR) to host all the Astra Control Center images.
8. Set up ACR access for Docker push/pull all the Astra Control Center images.
9. Push the ACC images to this registry by entering the following script:

   ```
   az acr login -n <AZ ACR URL/Location>
   This script requires ACC manifest file and your Azure ACR location.
   ```

   Example:

```
manifestfile=astra-control-center-<version>.manifest
AZ_ACR_REGISTRY=<target image repository>
ASTRA_REGISTRY=<source ACC image repository>

while IFS= read -r image; do
    echo "image: $ASTRA_REGISTRY/$image $AZ_ACR_REGISTRY/$image"
    root_image=${image%:*}
    echo $root_image
    docker pull $ASTRA_REGISTRY/$image
    docker tag $ASTRA_REGISTRY/$image $AZ_ACR_REGISTRYY/$image
    docker push $AZ_ACR_REGISTRY/$image
done < astra-control-center-22.04.41.manifest
```

10. Set up DNS zones.

Configure NetApp BlueXP (formerly Cloud Manager) for Azure

Using BlueXP (formerly Cloud Manager), create a workspace, add a connector to Azure, create a working environment, and import the cluster.

Follow the BlueXP documentation to complete the following steps. See Getting started with BlueXP in Azure.

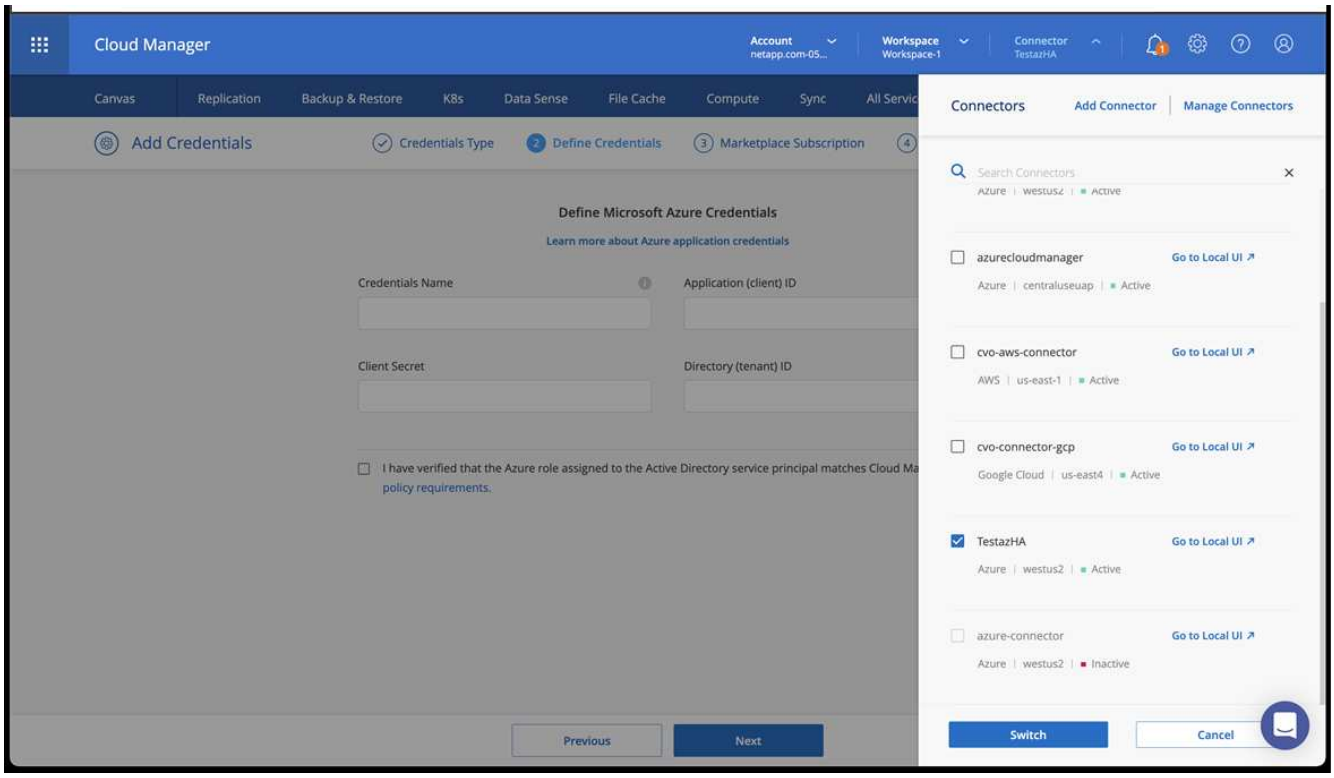
What you'll need

Access to the Azure account with the required IAM permissions and roles

Steps

1. Add your credentials to BlueXP.
2. Add a connector for Azure. See BlueXP policies.
   a. Choose **Azure** as the Provider.
   b. Enter Azure credentials, including the application ID, client secret, and directory (tenant) ID.

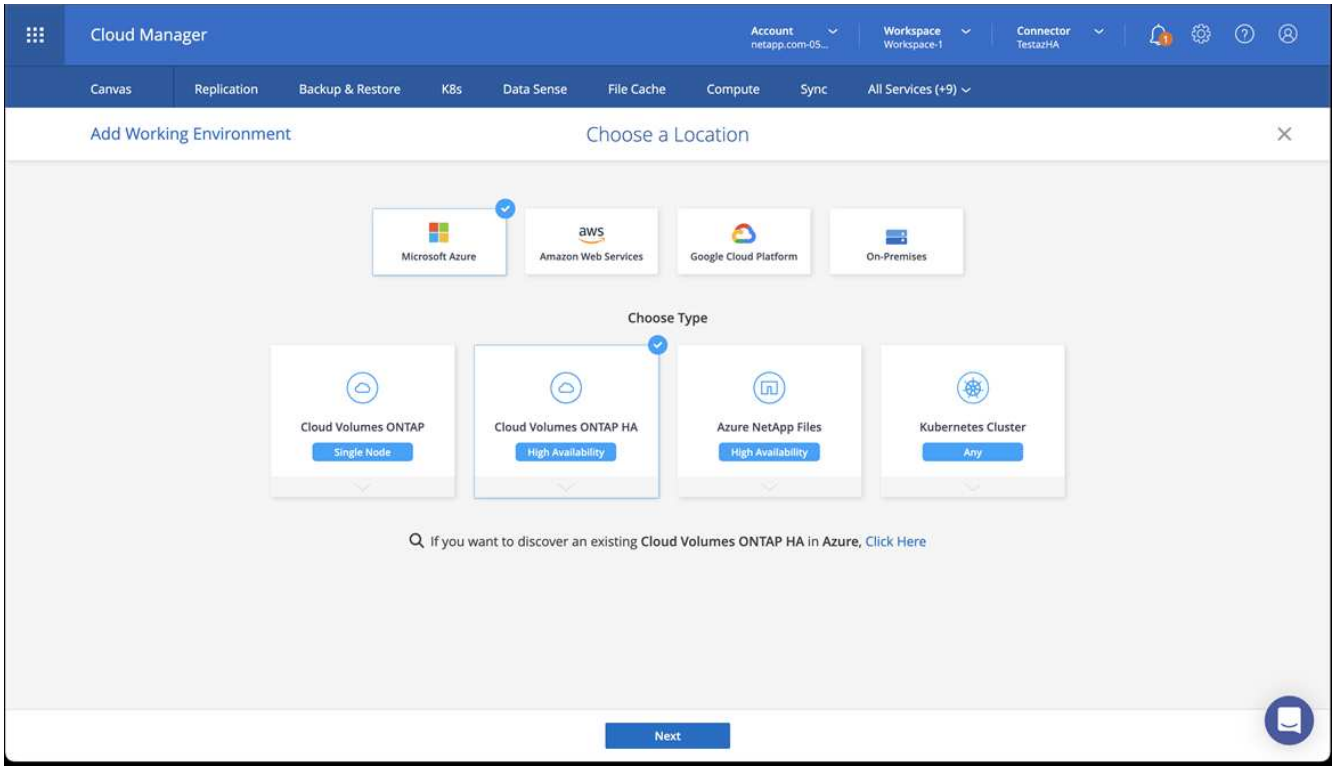
      See Creating a connector in Azure from BlueXPr.
3. Ensure that the connector is running and switch to that connector.



4. Create a working environment for your cloud environment.

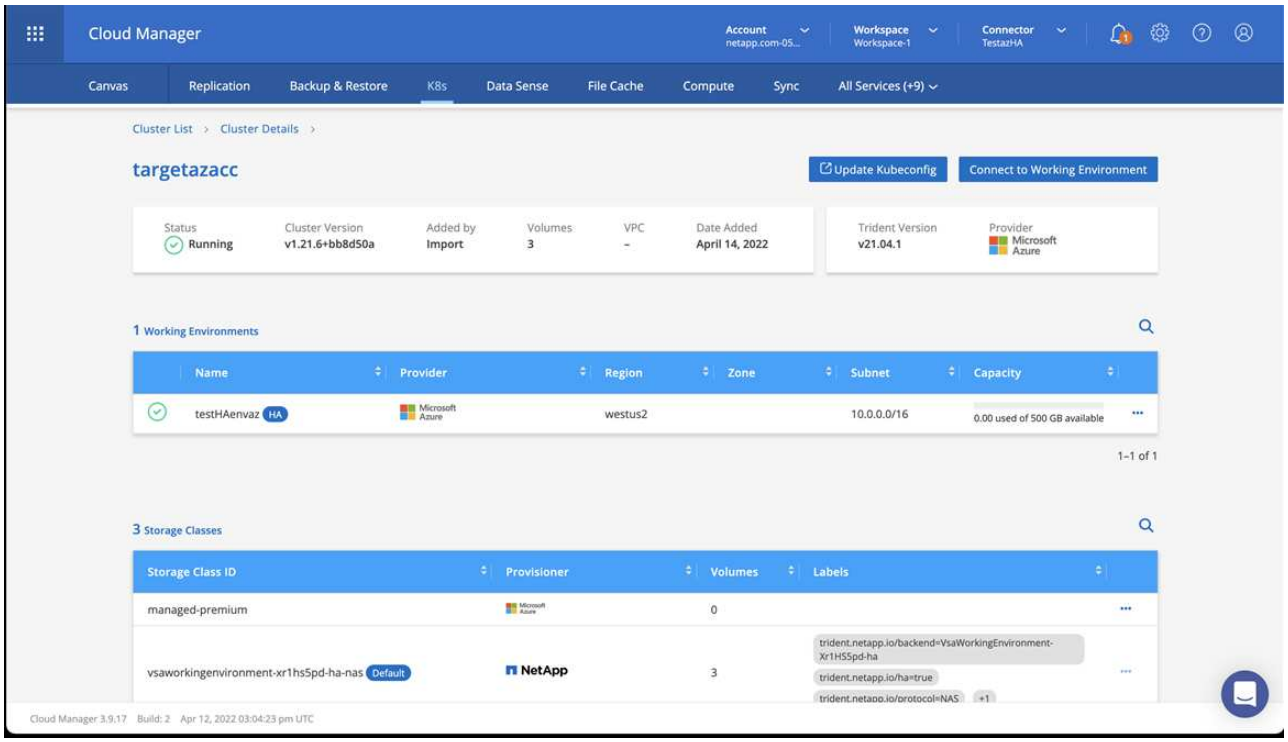
   a. Location: "Microsoft Azure".

   b. Type: "Cloud Volumes ONTAP HA".



5. Import the OpenShift cluster. The cluster will connect to the working environment you just created.

   a. View the NetApp cluster details by selecting **K8s** > **Cluster list** > **Cluster Details**.



b. In the upper right corner, note the Trident version.

c. Note the Cloud Volumes ONTAP cluster storage classes showing NetApp as the provisioner.

This imports your Red Hat OpenShift cluster and assigns a default storage class. You select the storage class.
Trident is automatically installed as part of the import and discovery process.

6. Note all the persistent volumes and volumes in this Cloud Volumes ONTAP deployment.
7. Cloud Volumes ONTAP can operate as a single node or in High Availability. If HA is enabled, note the HA status and node deployment status running in Azure.

Install and configure Astra Control Center for Azure

Install Astra Control Center with the standard installation instructions.

Using Astra Control Center, add an Azure bucket. See Set up Astra Control Center and add buckets.

Set up Astra Control Center

After you install Astra Control Center, log in to the UI, and change your password, you'll want to set up a license, add clusters, manage storage, and add buckets.

Tasks

- Add a license for Astra Control Center
- Prepare your environment for cluster management using Astra Control
- Add cluster
- Add a storage backend
- Add a bucket

Add a license for Astra Control Center

You can add a new license using the Astra Control UI or API to gain full Astra Control Center functionality. Without a license, your usage of Astra Control Center is limited to managing users and adding new clusters.

Astra Control Center licenses measure CPU resources using Kubernetes CPU units and account for the CPU resources assigned to the worker nodes of all the managed Kubernetes clusters. Licenses are based on vCPU usage. For more information on how licenses are calculated, refer to Licensing.

> If your installation grows to exceed the licensed number of CPU units, Astra Control Center prevents you from managing new applications. An alert is displayed when capacity is exceeded.

> To update an existing evaluation or full license, refer to Update an existing license.

What you'll need

- Access to a newly installed Astra Control Center instance.
- Administrator role permissions.
- A NetApp License File (NLF).

Steps

1. Log in to the Astra Control Center UI.
2. Select **Account** > **License**.
3. Select **Add License**.
4. Browse to the license file (NLF) that you downloaded.
5. Select **Add License**.

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

> If you have an evaluation license and are not sending data to AutoSupport, be sure that you store your account ID to avoid data loss in the event of Astra Control Center failure.

Prepare your environment for cluster management using Astra Control

You should ensure that the following prerequisite conditions are met before you add a cluster. You should also run eligibility checks to ensure that your cluster is ready to be added to Astra Control Center and create roles for cluster management.

What you'll need

- Ensure that the worker nodes in your cluster are configured with the appropriate storage drivers so that the pods can interact with the backend storage.
- Your environment meets the operational environment requirements for Astra Trident and Astra Control Center.
- A version of Astra Trident that is supported by Astra Control Center is installed:

> You can deploy Astra Trident using either Trident operator (manually or using Helm chart) or `tridentctl`. Prior to installing or upgrading Astra Trident, review the supported frontends, backends, and host configurations.

- **Trident storage backend configured**: At least one Astra Trident storage backend must be configured on the cluster.
- **Trident storage classes configured**: At least one Astra Trident storage class must be configured on the cluster. If a default storage class is configured, ensure that it is the only storage class that has the default annotation.
- **Astra Trident volume snapshot controller and volume snapshot class installed and configured**: The volume snapshot controller must be installed so that snapshots can be created in Astra Control. At least one Astra Trident `VolumeSnapshotClass` has been set up by an administrator.

- **Kubeconfig accessible**: You have access to the cluster kubeconfig that includes only one context element.
- **ONTAP credentials**: You need ONTAP credentials and a superuser and user ID set on the backing ONTAP system to back up and restore apps with Astra Control Center.

  Run the following commands in the ONTAP command line:

  ```
  export-policy rule modify -vserver <storage virtual machine name>
  -policyname <policy name> -ruleindex 1 -superuser sys
  export-policy rule modify -vserver <storage virtual machine name>
  -policyname <policy name> -ruleindex 1 -anon 65534
  ```

- **Rancher only**: When managing application clusters in a Rancher environment, modify the application cluster's default context in the kubeconfig file provided by Rancher to use a control plane context instead of the Rancher API server context. This reduces load on the Rancher API server and improves performance.

Run eligibility checks

Run the following eligibility checks to ensure that your cluster is ready to be added to Astra Control Center.

Steps

1. Check the Trident version.

   ```
   kubectl get tridentversions -n trident
   ```

   If Trident exists, you see output similar to the following:

   ```
   NAME      VERSION
   trident   22.10.0
   ```

   If Trident does not exist, you see output similar to the following:

   ```
   error: the server doesn't have a resource type "tridentversions"
   ```

   > If Trident is not installed or the installed version is not the latest, you need to install the latest version of Trident before proceeding. Refer to the Trident documentation for instructions.

2. Ensure that the pods are running:

```
kubectl get pods -n trident
```

3. Determine if the storage classes are using the supported Trident drivers. The provisioner name should be `csi.trident.netapp.io`. See the following example:

```
kubectl get sc
```

Sample response:

```
NAME                   PROVISIONER            RECLAIMPOLICY
VOLUMEBINDINGMODE  ALLOWVOLUMEEXPANSION  AGE
ontap-gold (default)  csi.trident.netapp.io  Delete         Immediate
true                   5d23h
```

Create a limited cluster role kubeconfig

You can optionally create a limited administrator role for Astra Control Center. This is not a required procedure for Astra Control Center setup. This procedure helps create a separate kubeconfig that limits Astra Control permissions on the clusters it manages.

What you'll need

Ensure that you have the following for the cluster you intend to manage before completing the procedure steps:

- kubectl v1.23 or later installed
- kubectl access to the cluster that you intend to add and manage with Astra Control Center

> For this procedure, you do not need kubectl access to the cluster that is running Astra Control Center.

- An active kubeconfig for the cluster you intend to manage with cluster admin rights for the active context

Steps

1. Create a service account:

   a. Create a service account file called `astracontrol-service-account.yaml`.

      Adjust the name and namespace as needed. If changes are made here, you should apply the same changes in the following steps.

      ```
      <strong>astracontrol-service-account.yaml</strong>
      ```

      ```yaml
      apiVersion: v1
      kind: ServiceAccount
      metadata:
        name: astracontrol-service-account
        namespace: default
      ```

   b. Apply the service account:

      ```
      kubectl apply -f astracontrol-service-account.yaml
      ```

2. Create a limited cluster role with the minimum permissions necessary for a cluster to be managed by Astra Control:

   a. Create a `ClusterRole` file called `astra-admin-account.yaml`.

      Adjust the name and namespace as needed. If changes are made here, you should apply the same changes in the following steps.

      ```
      <strong>astra-admin-account.yaml</strong>
      ```

      ```yaml
      apiVersion: rbac.authorization.k8s.io/v1
      kind: ClusterRole
      metadata:
        name: astra-admin-account
      rules:

      # Get, List, Create, and Update all resources
      # Necessary to backup and restore all resources in an app
      - apiGroups:
        - '*'
        resources:
        - '*'
        verbs:
      ```

```
    - get
    - list
    - create
    - patch

  # Delete Resources
  # Necessary for in-place restore and AppMirror failover
  - apiGroups:
    - ""
    - apps
    - autoscaling
    - batch
    - crd.projectcalico.org
    - extensions
    - networking.k8s.io
    - policy
    - rbac.authorization.k8s.io
    - snapshot.storage.k8s.io
    - trident.netapp.io
    resources:
    - configmaps
    - cronjobs
    - daemonsets
    - deployments
    - horizontalpodautoscalers
    - ingresses
    - jobs
    - namespaces
    - networkpolicies
    - persistentvolumeclaims
    - poddisruptionbudgets
    - pods
    - podtemplates
    - podsecuritypolicies
    - replicasets
    - replicationcontrollers
    - replicationcontrollers/scale
    - rolebindings
    - roles
    - secrets
    - serviceaccounts
    - services
    - statefulsets
    - tridentmirrorrelationships
    - tridentsnapshotinfos
    - volumesnapshots
```

```
  - volumesnapshotcontents
  verbs:
  - delete

# Watch resources
# Necessary to monitor progress
- apiGroups:
  - ""
  resources:
  - pods
  - replicationcontrollers
  - replicationcontrollers/scale
  verbs:
  - watch

# Update resources
- apiGroups:
  - ""
  - build.openshift.io
  - image.openshift.io
  resources:
  - builds/details
  - replicationcontrollers
  - replicationcontrollers/scale
  - imagestreams/layers
  - imagestreamtags
  - imagetags
  verbs:
  - update

# Use PodSecurityPolicies
- apiGroups:
  - extensions
  - policy
  resources:
  - podsecuritypolicies
  verbs:
  - use
```

b. Apply the cluster role:

```
kubectl apply -f astra-admin-account.yaml
```

3. Create the cluster role binding for the cluster role to the service account:

   a. Create a `ClusterRoleBinding` file called `astracontrol-clusterrolebinding.yaml`.

Adjust any names and namespaces modified when creating the service account as needed.

```
<strong>astracontrol-clusterrolebinding.yaml</strong>
```

```yaml
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: astracontrol-admin
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: astra-admin-account
subjects:
- kind: ServiceAccount
  name: astracontrol-service-account
  namespace: default
```

b. Apply the cluster role binding:

```
kubectl apply -f astracontrol-clusterrolebinding.yaml
```

4. List the service account secrets, replacing `<context>` with the correct context for your installation:

```
kubectl get serviceaccount astracontrol-service-account --context
<context> --namespace default -o json
```

The end of the output should look similar to the following:

```
"secrets": [
{ "name": "astracontrol-service-account-dockercfg-vhz87"},
{ "name": "astracontrol-service-account-token-r59kr"}
]
```

The indices for each element in the `secrets` array begin with 0. In the above example, the index for `astracontrol-service-account-dockercfg-vhz87` would be 0 and the index for `astracontrol-service-account-token-r59kr` would be 1. In your output, make note of the index for the service account name that has the word "token" in it.

5. Generate the kubeconfig as follows:

a. Create a `create-kubeconfig.sh` file. Replace `TOKEN_INDEX` in the beginning of the following script with the correct value.

```
<strong>create-kubeconfig.sh</strong>
```

```
# Update these to match your environment.
# Replace TOKEN_INDEX with the correct value
# from the output in the previous step. If you
# didn't change anything else above, don't change
# anything else here.

SERVICE_ACCOUNT_NAME=astracontrol-service-account
NAMESPACE=default
NEW_CONTEXT=astracontrol
KUBECONFIG_FILE='kubeconfig-sa'

CONTEXT=$(kubectl config current-context)

SECRET_NAME=$(kubectl get serviceaccount ${SERVICE_ACCOUNT_NAME}
\
  --context ${CONTEXT} \
  --namespace ${NAMESPACE} \
  -o jsonpath='{.secrets[TOKEN_INDEX].name}')
TOKEN_DATA=$(kubectl get secret ${SECRET_NAME} \
  --context ${CONTEXT} \
  --namespace ${NAMESPACE} \
  -o jsonpath='{.data.token}')

TOKEN=$(echo ${TOKEN_DATA} | base64 -d)

# Create dedicated kubeconfig
# Create a full copy
kubectl config view --raw > ${KUBECONFIG_FILE}.full.tmp

# Switch working context to correct context
kubectl --kubeconfig ${KUBECONFIG_FILE}.full.tmp config use-
context ${CONTEXT}

# Minify
kubectl --kubeconfig ${KUBECONFIG_FILE}.full.tmp \
  config view --flatten --minify > ${KUBECONFIG_FILE}.tmp

# Rename context
kubectl config --kubeconfig ${KUBECONFIG_FILE}.tmp \
  rename-context ${CONTEXT} ${NEW_CONTEXT}

# Create token user
```

```
kubectl config --kubeconfig ${KUBECONFIG_FILE}.tmp \
  set-credentials ${CONTEXT}-${NAMESPACE}-token-user \
  --token ${TOKEN}

# Set context to use token user
kubectl config --kubeconfig ${KUBECONFIG_FILE}.tmp \
  set-context ${NEW_CONTEXT} --user ${CONTEXT}-${NAMESPACE}-token
-user

# Set context to correct namespace
kubectl config --kubeconfig ${KUBECONFIG_FILE}.tmp \
  set-context ${NEW_CONTEXT} --namespace ${NAMESPACE}

# Flatten/minify kubeconfig
kubectl config --kubeconfig ${KUBECONFIG_FILE}.tmp \
  view --flatten --minify > ${KUBECONFIG_FILE}

# Remove tmp
rm ${KUBECONFIG_FILE}.full.tmp
rm ${KUBECONFIG_FILE}.tmp
```

b. Source the commands to apply them to your Kubernetes cluster.

```
source create-kubeconfig.sh
```

6. (Optional) Rename the kubeconfig to a meaningful name for your cluster.

```
mv kubeconfig-sa YOUR_CLUSTER_NAME_kubeconfig
```

What's next?

Now that you've verified that the prerequisites are met, you're ready to add a cluster.

Add cluster

To begin managing your apps, add a Kubernetes cluster and manage it as a compute resource. You have to add a cluster for Astra Control Center to discover your Kubernetes applications.

> We recommend that Astra Control Center manage the cluster it is deployed on first before you add other clusters to Astra Control Center to manage. Having the initial cluster under management is necessary to send Kubemetrics data and cluster-associated data for metrics and troubleshooting.

What you'll need

- Before you add a cluster, review and perform the necessary prerequisite tasks.

Steps

1. Navigate from either the Dashboard or the Clusters menu:
   - From **Dashboard** in the Resource Summary, select **Add** from the Clusters pane.
   - In the left navigation area, select **Clusters** and then select **Add Cluster** from the Clusters page.
2. In the **Add Cluster** window that opens, upload a `kubeconfig.yaml` file or paste the contents of a `kubeconfig.yaml` file.

   > The `kubeconfig.yaml` file should include **only the cluster credential for one 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. If you created a kubeconfig for a limited cluster role using the process above, be sure to upload or paste that kubeconfig in this step.

3. Provide a credential name. By default, the credential name is auto-populated as the name of the cluster.
4. Select **Next**.
5. Select the default storage class to be used for this Kubernetes cluster, and select **Next**.

   > You should select a Trident storage class backed by ONTAP storage.

6. Review the information, and if everything looks good, select **Add**.

Result

The cluster enters **Discovering** state and then changes to **Healthy**. You are now managing the cluster with Astra Control Center.

> After you add a cluster to be managed in Astra Control Center, it might take a few minutes to deploy the monitoring operator. Until then, the Notification icon turns red and logs a **Monitoring Agent Status Check Failed** event. You can ignore this, because the issue resolves when Astra Control Center obtains the correct status. If the issue does not resolve in a few minutes, go to the cluster, and run `oc get pods -n netapp-monitoring` as the starting point. You will need to look into the monitoring operator logs to debug the problem.

Add a storage backend

You can add an existing ONTAP storage backend to Astra Control Center to manage its resources.

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.

Steps

1. From the Dashboard in the left-navigation area, select **Backends**.
2. Do one of the following:
   - **New backends**: Select **Add** to manage an existing backend, select **ONTAP**, and select **Next**.
   - **Discovered backends**: From the Actions menu, select **Manage** on a discovered backend from the

managed cluster.

3. 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.

4. Select **Next**.
5. Confirm the backend details and select **Manage**.

Result

The backend appears in the `Healthy` state in the list with summary information.

> You might need to refresh the page for the backend to appear.

Add a bucket

You can add a bucket using the Astra Control UI or API. 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.

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

What you'll need

- A bucket that is reachable from your clusters managed by Astra Control Center.
- Credentials for the bucket.
- A bucket of the following types:
  - 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.

Steps

1. In the left navigation area, select **Buckets**.
2. Select **Add**.
3. Select the bucket type.

> 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.

4. Enter an existing bucket name and optional description.

> The bucket name and description appear as a backup location that you can choose later when you're creating a backup. The name also appears during protection policy configuration.

5. Enter the name or IP address of the S3 endpoint.
6. Under **Select Credentials**, choose either the **Add** or **Use existing** tab.
   - If you chose **Add**:
     a. Enter a name for the credential that distinguishes it from other credentials in Astra Control.
     b. Enter the access ID and secret key by pasting the contents from your clipboard.
   - If you chose **Use existing**:
     a. Select the existing credentials you want to use with the bucket.
7. Select `Add`.

> 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.

What's next?

Now that you've logged in and added clusters to Astra Control Center, you're ready to start using Astra Control Center's application data management features.

- Manage local users and roles
- Start managing apps
- Protect apps
- Manage notifications
- Connect to Cloud Insights
- Add a custom TLS certificate
- Change the default storage class

Find more information

- Use the Astra Control API
- Known issues

Frequently asked questions for Astra Control Center

This FAQ can help if you're just looking for a quick answer to a question.

Overview

The following sections provide answers to some additional questions that you might come across as you use Astra Control Center. For additional clarifications, please reach out to astra.feedback@netapp.com

Access to Astra Control Center

What's the Astra Control URL?

Astra Control Center uses local authentication and a URL specific to each environment.

For the URL, in a browser, enter the Fully Qualified Domain Name (FQDN) you set in the spec.astraAddress field in the astra_control_center.yaml custom resource (CR) file when you installed Astra Control Center. The email is the value that you set in the spec.email field in the astra_control_center.yaml CR.

Licensing

I am using the Evaluation license. How to I change to the full license?

You can easily change to a full license by obtaining the NetApp license file (NLF).

Steps

1. From the left navigation, select **Account** > **License**.
2. Select **Add license**.
3. Browse to the license file you downloaded and select **Add**.

I am using the Evaluation license. Can I still manage apps?

Yes, you can test out the managing apps functionality with the Evaluation license.

Registering Kubernetes clusters

I need to add worker nodes to my Kubernetes cluster after adding to Astra Control. What should I do?

New worker nodes can be added to existing pools. These will be automatically discovered by Astra Control. If the new nodes are not visible in Astra Control, check if the new worker nodes are running the supported image type. You can also verify the health of the new worker nodes by using the `kubectl get nodes` command.

How do I properly unmanage a cluster?

1. Unmanage the applications from Astra Control.
2. Unmanage the cluster from Astra Control.

What happens to my applications and data after removing the Kubernetes cluster from Astra Control?

Removing a cluster from Astra Control will not make any changes to the cluster's configuration (applications and persistent storage). Any Astra Control snapshots or backups taken of applications on that cluster will be

unavailable to restore. Persistent storage backups created by Astra Control remain within Astra Control, but they are unavailable for restore.

> Always remove a cluster from Astra Control before you delete it through any other methods. Deleting a cluster using another tool while it's still being managed by Astra Control can cause problems for your Astra Control account.

Is NetApp Trident automatically uninstalled from a cluster when I unmanage it?
When you unmanage a cluster from Astra Control Center, Trident isn't automatically uninstalled from the cluster. To uninstall Trident, you'll need to follow these steps in the Trident documentation.

Managing applications

Can Astra Control deploy an application?

Astra Control doesn't deploy applications. Applications must be deployed outside of Astra Control.

What happens to applications after I stop managing them from Astra Control?

Any existing backups or snapshots will be deleted. Applications and data remain available. Data management operations will not be available for unmanaged applications or any backups or snapshots that belong to it.

Can Astra Control manage an application that is on non-NetApp storage?

No. While Astra Control can discover applications that are using non-NetApp storage, it can't manage an application that's using non-NetApp storage.

Should I manage Astra Control itself?
No, you should not manage Astra Control itself because it is a "system app."

Do unhealthy pods affect app management?
If a managed app has pods in an unhealthy state, Astra Control can't create new backups and clones.

Data management operations

My application uses several PVs. Will Astra Control take snapshots and backups of these PVs?

Yes. A snapshot operation on an application by Astra Control includes snapshot of all the PVs that are bound to the application's PVCs.

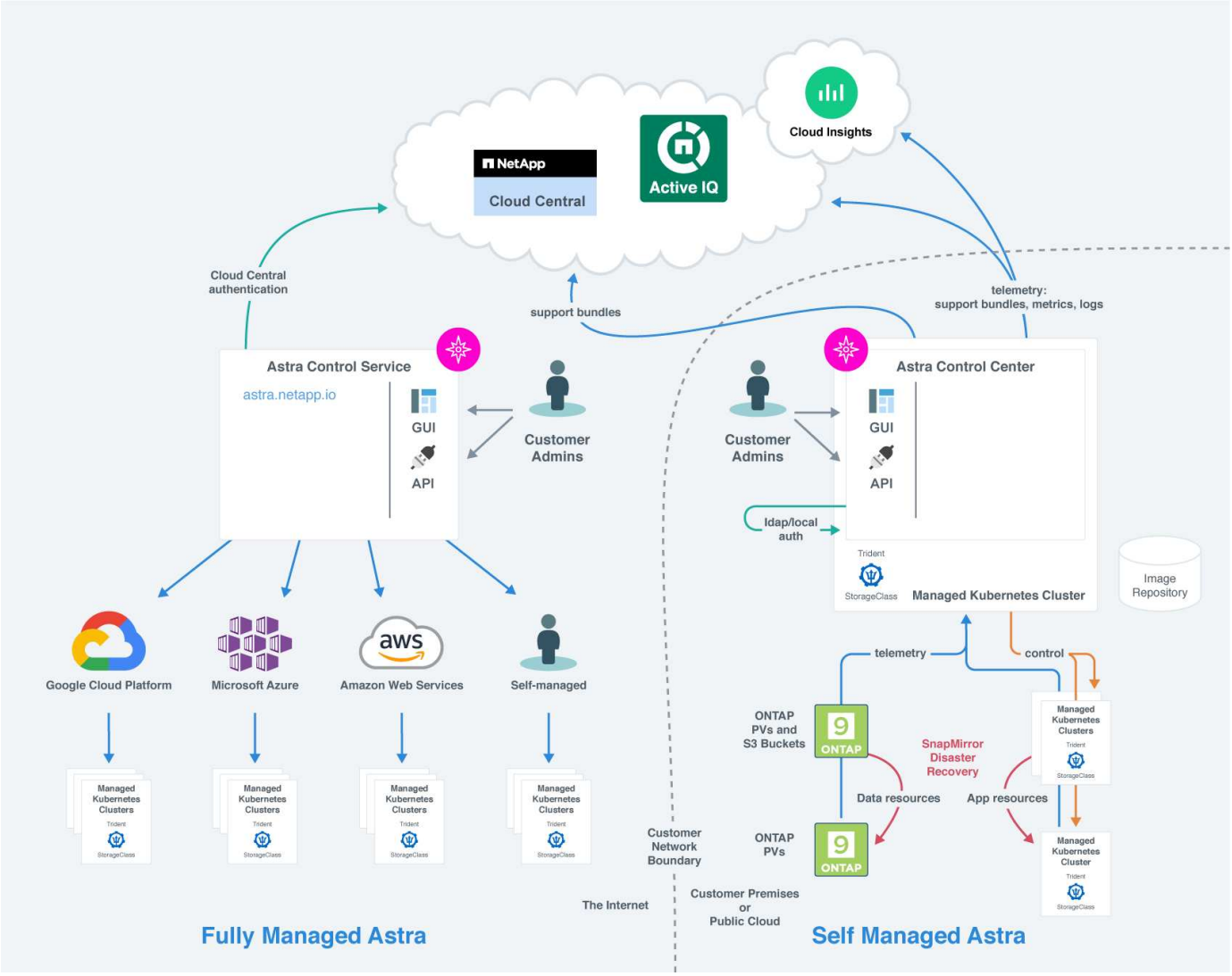
Can I manage snapshots taken by Astra Control directly through a different interface or object storage?

No. Snapshots and backups taken by Astra Control can only be managed with Astra Control.

Concepts

Architecture and components

Here is an overview of the various components of the Astra Control environment.



Astra Control components

- **Kubernetes clusters**: Kubernetes is a portable, extensible, open-source platform for managing containerized workloads and services, that facilitates both declarative configuration and automation. Astra provides management services for applications hosted in a Kubernetes cluster.
- **Astra Trident**: As a fully supported open source storage provisioner and orchestrator maintained by NetApp, Trident enables you to create storage volumes for containerized applications managed by Docker and Kubernetes. When deployed with Astra Control Center, Trident includes a configured ONTAP storage backend.
- **Storage backend**:
  - Astra Control Service uses the following storage backends:
    - NetApp Cloud Volumes Service for Google Cloud or Google Persistent Disk as the storage

backend for GKE clusters
    - Azure NetApp Files or Azure Managed Disks as the storage backend for AKS clusters.
    - Amazon Elastic Block Store (EBS) or Amazon FSx for NetApp ONTAP as backend storage options for EKS clusters.
  - Astra Control Center uses the following storage backends:
    - ONTAP AFF, FAS, and ASA. As a storage software and hardware platform, ONTAP provides core storage services, support for multiple storage access protocols, and storage management functionality, such as snapshots and mirroring.
    - Cloud Volumes ONTAP
- **Cloud Insights**: A NetApp cloud infrastructure monitoring tool, Cloud Insights enables you to monitor performance and utilization for your Kubernetes clusters managed by Astra Control Center. Cloud Insights correlates storage usage to workloads. When you enable the Cloud Insights connection in Astra Control Center, telemetry information shows in Astra Control Center UI pages.

Astra Control interfaces

You can complete tasks using different interfaces:

- **Web user interface (UI)**: Both Astra Control Service and Astra Control Center use the same web-based UI where you can manage, migrate and protect apps. Use the UI also to manage user accounts and configuration settings.
- **API**: Both Astra Control Service and Astra Control Center use the same Astra Control API. Using the API, you can perform the same tasks that you would using the UI.

Astra Control Center also enables you to manage, migrate, and protect Kubernetes clusters running within VM environments.

For more information

- Astra Control Service documentation
- Astra Control Center documentation
- Astra Trident documentation
- Use the Astra Control API
- Cloud Insights documentation
- ONTAP documentation

Data protection

Learn about the available types of data protection in Astra Control Center, and how best to use them to protect your apps.

Snapshots, backups, and protection policies

Both snapshots and backups protect the following types of data:

- The application itself
- Any persistent data volumes associated with the application

- Any resource artifacts belonging to the application

A *snapshot* is a point-in-time copy of an app that's stored on the same provisioned volume as the app. They are usually fast. You can use local snapshots to restore the application to an earlier point in time. Snapshots are useful for fast clones; snapshots include all of the Kubernetes objects for the app, including configuration files. Snapshots are useful for cloning or restoring an app within the same cluster.

A *backup* is based on a snapshot. It is stored in the external object store, and because of this, can be slower to take compared to local snapshots. You can restore an app backup to the same cluster, or you can migrate an app by restoring its backup to a different cluster. You can also choose a longer retention period for backups. Because they are stored in the external object store, backups generally offer you better protection than snapshots in cases of server failure or data loss.

A *protection policy* is a way to protect an app by automatically creating snapshots, backups, or both according to a schedule that you define for that app. A protection policy also enables you to choose how many snapshots and backups to retain in the schedule, and set different schedule granularity levels. Automating your backups and snapshots with a protection policy is the best way to ensure each app is protected according to the needs of your organization and service level agreement (SLA) requirements.

> *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 its associated persistent storage, then you need a backup to recover. A snapshot would not enable you to recover.

Clones

A *clone* is an exact duplicate of an app, its configuration, and its persistent data volumes. You can manually create a clone on either the same Kubernetes cluster or on another cluster. Cloning an app can be useful if you need to move applications and storage from one Kubernetes cluster to another.

Replication to a remote cluster

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 cluster to another.

Astra Control asynchronously replicates app Snapshot copies to a remote cluster. The replication process includes data in the persistent volumes replicated by SnapMirror and the app metadata protected by Astra Control.

App replication is different from app backup and restore in the following ways:

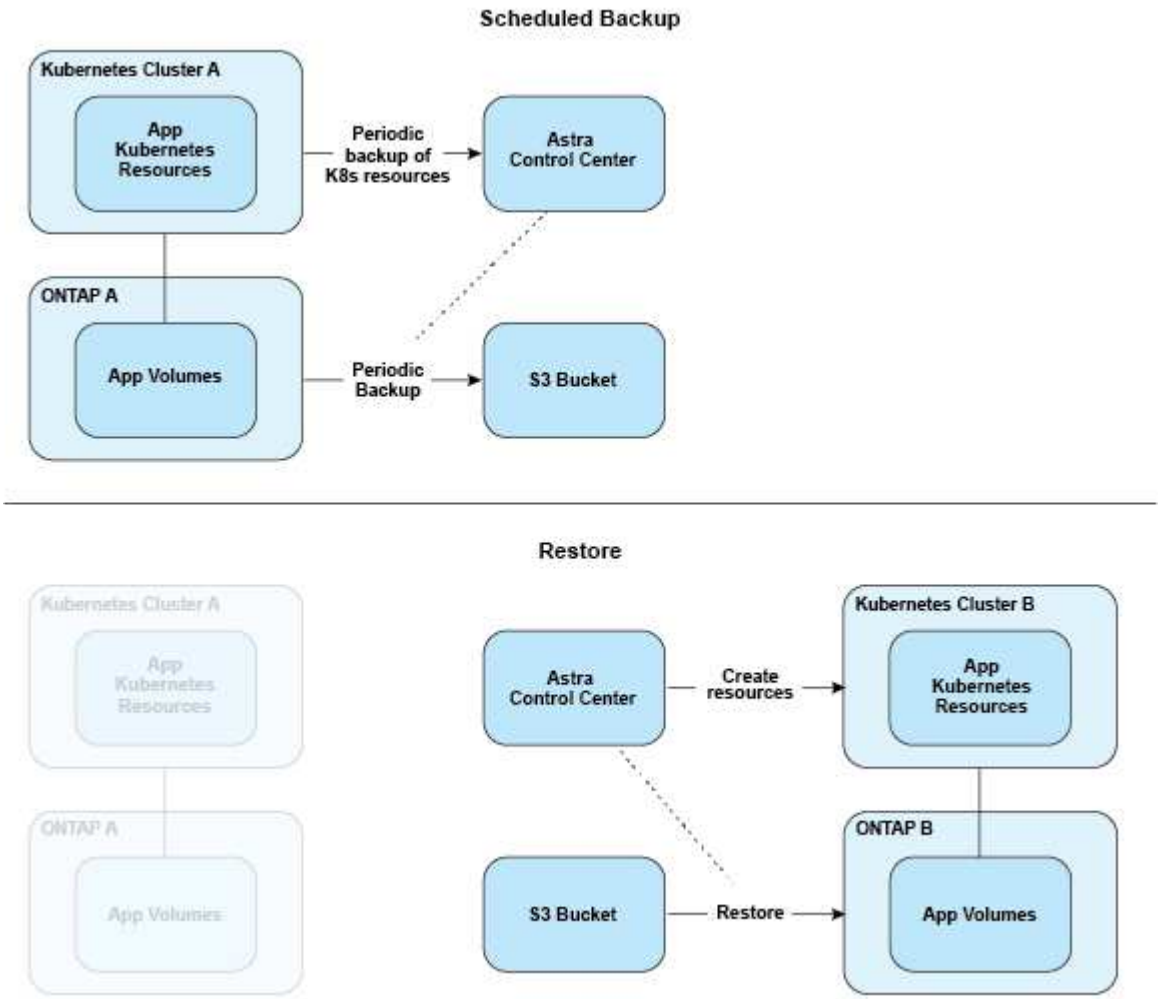
- **App replication**: Astra Control requires the source and destination Kubernetes clusters to be available and managed with their respective ONTAP storage backends configured to enable NetApp SnapMirror. Astra Control takes the policy-driven application Snapshot and replicates it to the remote cluster. NetApp SnapMirror technology is used to replicate the persistent volume data. To fail over, Astra Control can bring the replicated app online by recreating the app objects on the destination Kubernetes cluster with the replicated volumes on the destination ONTAP cluster. Since the persistent volume data is already present on the destination ONTAP cluster, Astra Control can offer quick recovery times for failover.
- **App backup and restore**: When backing up applications, Astra Control creates a Snapshot of the app data and stores it in an object storage bucket. When a restore is needed, the data in the bucket must be copied to a persistent volume on the ONTAP cluster. The backup/restore operation does not require the

secondary Kubernetes/ONTAP cluster to be available and managed, but the additional data copy can result in longer restore times.

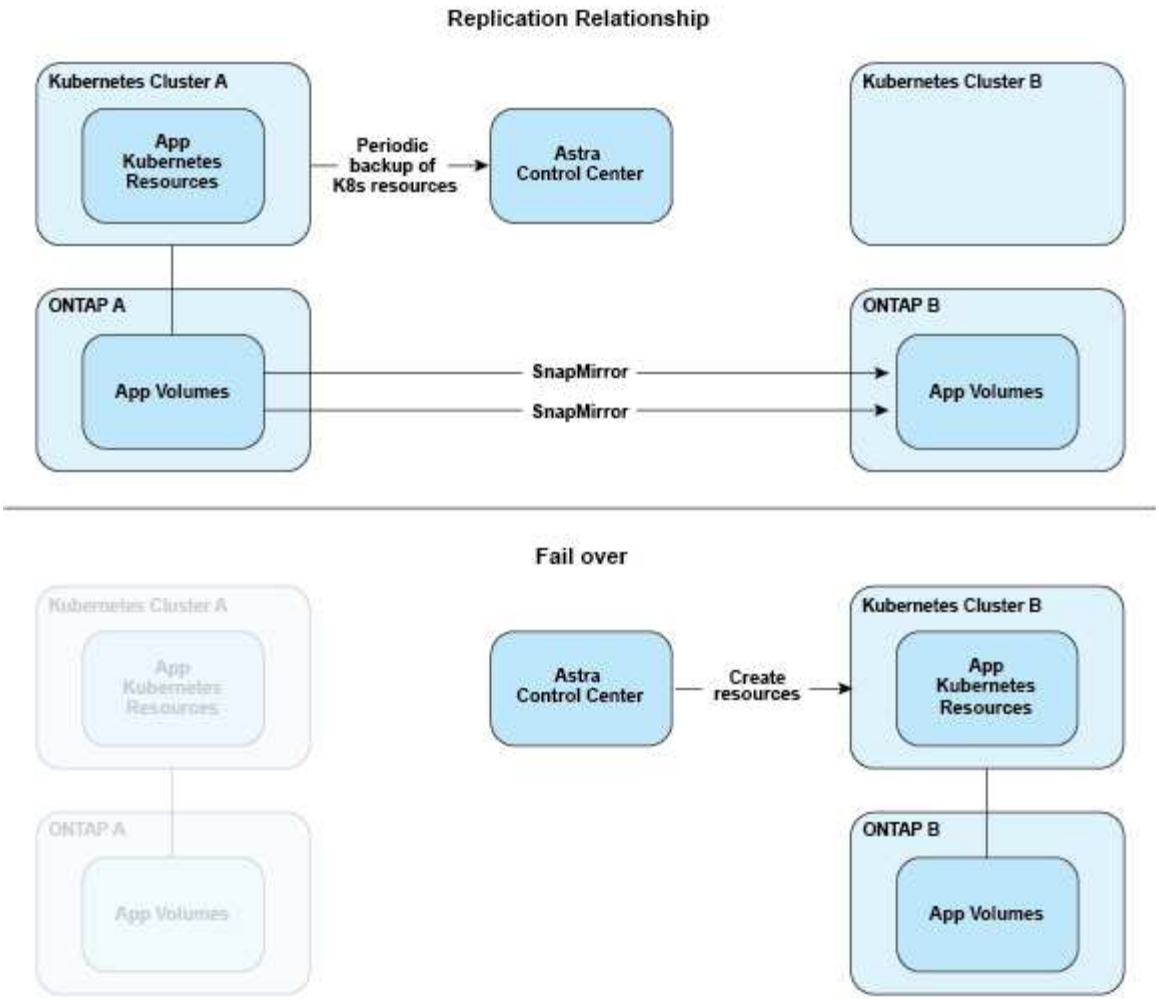
To learn how to replicate apps, see Replicate apps to a remote system using SnapMirror technology.

The following images show the scheduled backup and restore process compared to the replication process.

The backup process copies data to S3 buckets and restores from S3 buckets:



On the other hand, replication is done by replicating to ONTAP and then a fail over creates the Kubernetes resources:



Licensing

Astra Control Center requires a license to be installed for the full app data management functionality to be enabled. When you deploy Astra Control Center without a license, a banner is displayed in the web UI, warning that system functionality is limited.

You obtain a license in one of the following ways:

- If you are evaluating Astra Control Center, download the evaluation license file. An evaluation license lets you use Astra Control Center for 90 days from the date you download the license.
- If you already purchased Astra Control Center, generate your NetApp license file (NLF) from the NetApp Support Site. After you purchase the product, you receive a serial number and license, which you use on the Support Site.

For details about licenses needed for ONTAP storage backends, refer to supported storage backends.

> You can add a cluster, add a bucket, and manage a storage backend without a license.

How license consumption is calculated

When you add a new cluster to Astra Control Center, it doesn't count toward consumed licenses until at least one application running on the cluster is managed by Astra Control Center.

When you start managing an app on a cluster, all of that cluster's CPU units are included in the Astra Control Center license consumption.

Find more information

- Add a license when you first set up Astra Control Center
- Update an existing license

Storage classes and persistent volume size

Astra Control Center supports ONTAP as the storage backend.

Overview

Astra Control Center supports the following:

- **Trident storage classes backed by ONTAP storage**: If you are using an ONTAP backend, Astra Control Center offers the ability to import the ONTAP backend to report various monitoring information.

> Trident storage classes should be preconfigured outside of Astra Control Center.

Storage classes

When you add a cluster to Astra Control Center, you're prompted to select one previously configured storage class on that cluster as the default storage class. This storage class will be used when no storage class is specified in a persistent volume claim (PVC). The default storage class can be changed at any time within Astra Control Center and any storage class can be used at any time by specifying the name of the storage class within the PVC or Helm chart. Ensure that you have only a single default storage class defined for your Kubernetes cluster.

For more information

- Astra Trident documentation

User roles and namespaces

Learn about user roles and namespaces in Astra Control, and how you can use them to control access to resources in your organization.

User roles

You can use roles to control the access users have to resources or capabilities of Astra Control. The following are the user roles in Astra Control:

- 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.

You can add constraints to a Member or Viewer user to restrict the user to one or more Namespaces.

Namespaces

A namespace is a scope you can assign to specific resources within a cluster that is managed by Astra Control. Astra Control discovers a cluster's namespaces when you add the cluster to Astra Control. Once discovered, the namespaces are available to assign as constraints to users. Only members that have access to that namespace are able to use that resource. You can use namespaces to control access to resources using a paradigm that makes sense for your organization; for example, by physical regions or divisions within a company. When you add constraints to a user, you can configure that user to have access to all namespaces or only a specific set of namespaces. You can also assign namespace constraints using namespace labels.

Find more information

Manage local users and roles

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.

Application management requirements

Astra Control has the following application management requirements:

- **Licensing**: To manage applications using Astra Control Center, you need an Astra Control Center 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) are 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.

What you’ll need

- A Kubernetes cluster added to Astra Control.
- One or more installed apps on the cluster. Read more about supported app installation methods.
- One or more active pods.
- 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

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.

> 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 example, the following scenario creates an undesirable situation when using NetApp SnapMirror replication:
>
> 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.

Read 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.

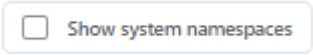
Define a namespace to manage as an app

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.

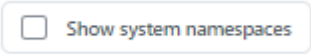
After the process completes, the applications that are associated with the namespace appear in the

`Associated applications` column.

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

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
- 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.

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.

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.

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 the Astra Control API to restore apps.

> 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.

About this task

- **Protect your apps first**: It is strongly recommended to take a snapshot of or back up your application before restoring it. This will enable you to clone from the snapshot or backup in the event that the restore is unsuccessful.
- **Check destination volumes**: If you restore to a different cluster, ensure that the cluster is using the same persistent volume access mode (for example, ReadWriteMany). The restore operation will fail if the destination persistent volume access mode is different.
- **(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**: Cloning apps deployed with Helm 3 (or upgraded from Helm 2 to Helm 3) is fully supported. Apps deployed with Helm 2 are not supported.

Steps

1. Select **Applications** and then select the name of an app.
2. Select **Data protection**.
3. If you want to restore from a snapshot, keep the **Snapshots** icon selected. Otherwise, select the **Backups** icon to restore from a backup.
4. From the Options menu in the **Actions** column for the snapshot or backup from which you want to restore, select **Restore application**.
5. Choose the restore type:
   - **Restore to original namespaces**: Use this procedure to restore the app in-place to the original cluster.

> 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 example, the following scenario creates an undesirable situation when using NetApp SnapMirror replication:
>
> 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.

a. Select the snapshot 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.

c. Review details about the restore action, type "restore", and select **Restore**.

- **Restore to new namespaces**: Use this procedure to restore the app to another cluster or with different namespaces from the source.

  a. Choose the destination cluster for the app you intend to restore.

  b. 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.

  c. Select **Next**.

  d. Select the snapshot to use to restore the app.

  e. Select **Next**.

  f. Review details about the restore action 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.

Replicate apps to a remote system 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 cluster to another.

For a comparison between backups/restores and replication, see 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 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 (running on the other cluster or site) 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 (fail over)
- 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 must be met before you begin:

- To achieve seamless disaster recovery, we recommend that you deploy Astra Control Center in a third fault domain or secondary site.

- The app's host Kubernetes cluster and a destination Kubernetes cluster must be managed along with their ONTAP clusters, ideally at different failure domains or sites.
- ONTAP clusters and the host SVM must be paired. See Cluster and SVM peering overview.
- The paired remote SVM must be available to Astra Trident on the destination cluster.
- Astra Trident version 22.07 or greater must exist on both the source and destination ONTAP clusters.
- ONTAP SnapMirror asynchronous licenses using the Data Protection bundle must be enabled on both the source and destination ONTAP clusters. See SnapMirror licensing overview in ONTAP.
- When you add an ONTAP storage backend to Astra Control Center, apply user credentials with the "admin" role, which has access methods `http` and `ontapi` enabled on both ONTAP source and destination clusters. See Manage User Accounts in ONTAP documentation for more information.
- Both source and destination Kubernetes clusters and ONTAP clusters must be managed by Astra Control.

> You can simultaneously replicate a different app (running on the other cluster or site) 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.

- **Astra Trident / ONTAP configuration**: Astra Control Center requires that a storage class be created and set as the default storage class. Astra Control Center supports the following ONTAP drivers provided by Astra Trident for replication:
  - ontap-nas
  - ontap-nas-flexgroup
  - ontap-san

Learn how to replicate apps to a remote system using SnapMirror technology.

Set up a replication relationship

Setting up a replication relationship involves the following that make up the replication policy;

- 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. In the Application page, select the **Data Protection** > **Replication** tab.
3. In the Data Protection > Replication tab, 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 that is different from the source.
   - **Destination storage class**: Select or enter the storage class that uses the paired SVM on the destination ONTAP cluster.
   - **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 its 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. For example, if you want to take the Snapshot every 5 minutes starting at 10:02, enter "02" as the offset minutes. The result would be 10:02, 10:07, 10:12, etc.

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 (fail over)

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. In the Application page, select the **Data Protection** > **Replication** tab.
3. In the Data Protection > Replication tab, 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 fail over procedure:

- On the destination cluster, the 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 fail over.
- 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. In the Application page, select the **Data Protection** > **Replication** tab.
3. In the Data Protection > Replication tab, 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 cluster while continuing to replicate back to the original source cluster. Astra Control stops the application on the source cluster and replicates the data to the destination before failing over the app to the destination cluster.

In this situation, you are swapping the source and destination. The original source cluster becomes the new

destination cluster, and the original destination cluster becomes the new source cluster.

Steps

1. From the Astra Control left navigation, select **Applications**.
2. In the Application page, select the **Data Protection** > **Replication** tab.
3. In the Data Protection > Replication tab, 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 "fail over" 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 cluster before reversing the replication direction.

This process starts from a relationship that has completed a fail over 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. In the Application page, select the **Data Protection** > **Replication** tab.
3. In the Data Protection > Replication tab, 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 fail over).
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. In the Application page, select the **Data Protection** > **Replication** tab.
3. In the Data Protection > Replication tab, 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 the Astra Control API to clone and migrate apps.

What you'll need

- 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.
- **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.

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 whether you want to create the clone from an existing snapshot or backup. If you don't select this option, Astra Control Center creates the clone from the app's current state.
  - If you chose to clone from an existing snapshot or backup, choose the snapshot or backup that you'd like to use.

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 execution hooks 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

Important notes about custom execution hooks

Consider the following when planning execution hooks for your apps.

- 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 adhoc 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 you create an execution hook for an application that participates in an Istio service mesh, make sure the hook runs against the original application container, and not the service mesh container. You can exclude Istio service mesh containers by applying a filter regex to every execution hook that runs for applications that use an Istio service mesh.
> - Since execution hooks often reduce or completely disable the functionality of the application they are running against, you should always try to minimize the time your custom execution hooks take to run.
> - 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 a post-backup execution hook cannot assume that the backup was completed.

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 five 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
1	Clone	N	N	New	Same	Y	N	Y
2	Clone	N	N	New	Different	Y	Y	Y
3	Clone or restore	Y	N	New	Same	N	N	Y
4	Clone or restore	N	Y	New	Same	N	N	Y
5	Clone or restore	Y	N	New	Different	N	Y	Y
6	Clone or restore	N	Y	New	Different	N	N	Y
7	Restore	Y	N	Existing	Same	N	N	Y
8	Restore	N	Y	Existing	Same	N	N	Y

Scenario	Operation	Existing snapshot	Existing backup	Namespace	Cluster	Snapshot hooks run	Backup hooks run	Restore hooks run
9	Snapshot	N/A	N/A	N/A	N/A	Y	N/A	N/A
10	Backup	N	N/A	N/A	N/A	Y	Y	N/A
11	Backup	Y	N/A	N/A	N/A	N	Y	N/A

Execution hook examples

Visit the NetApp Verda GitHub project to see examples and get an idea of how to structure your execution hooks. You can use these examples as templates or test scripts.

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, source, and when it runs (pre- or post-operation). To view event logs surrounding execution hooks, go to the **Activity** page in the left-side navigation area.

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

You can add one or more scripts that execution hooks can reference. Many execution hooks can reference the same script; this enables 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. See Execution hook examples for hook examples. You need to have Owner, Admin, or Member permissions to create execution hooks.

> When you create a custom shell script to use as an execution hook, remember to specify the appropriate shell at the beginning of the file, unless you are running 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, determine when the hook should run by selecting an operation type from the **Operation** drop-down menu.
5. Enter a unique name for the hook.

6. (Optional) Enter any arguments to pass to the hook during execution, pressing the Enter key after each argument you enter to record each one.
7. In the **Container Images** area, if the hook should run against all container images contained within the application, enable the **Apply to all container images** check box. If instead the hook should act only on one or more specified container images, enter the container image names in the **Container image names to match** field.
8. In the **Script** area, do one of the following:
    - 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 **Add hook**.

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.

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**.

For more information

- NetApp Verda GitHub project

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 status (whether it's healthy), its protection status (whether it's fully protected in case of failure), the pods, persistent storage, and more.

Steps

1. In the Astra Control Center UI, select **Applications** and then select the name of an app.
2. Review the information.
   - **App Status**: Provides a status that reflects the app's state in Kubernetes. For example, are pods and persistent volumes online? If an app is unhealthy, you'll need to go and troubleshoot the issue on the cluster by looking at Kubernetes logs. Astra doesn't provide information to help you fix a broken app.
   - **App Protection Status**: Provides a status of how well the app is protected:

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

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

  - **Overview**: Information about the state of the pods that are associated with the app.
  - **Data protection**: Enables you to configure a data protection policy and to view the existing snapshots and backups.
  - **Storage**: Shows you the app-level persistent volumes. The state of a persistent volume is from the perspective of the Kubernetes cluster.
  - **Resources**: Enables you to verify which resources are being backed up and managed.
  - **Activity**: Shows the activities related to the app.

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

Manage your account

Manage 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 the 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, see 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 the 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 email address in the LDAP "mail" attribute to search for and keep track of remote users. This attribute might be an optional or empty field in your directory. An email address must exist in this field for any remote users you wish to appear in Astra Control Center. This email address is used as the username in Astra Control Center for authentication.

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.

What you'll need

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, enter the base DN and an appropriate user search filter to use when retrieving user information from the LDAP server.
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.

> 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.

> You cannot add remote user if a local user with the same email address already exists on the system. To add the user as a remote user, delete the local user from the system first.

> Astra Control Center uses the email address in the LDAP "mail" attribute to search for and keep track of remote users. This attribute might be an optional or empty field in your directory. An email address must exist in this field for any remote users you wish to appear in Astra Control Center. This email address is used as the username in Astra Control Center for authentication.

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 added to Astra Control and inherit the same role.

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**.

   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 group appears in the list of remote groups, and all remote users in this group appear in the list of remote users. 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, see Add a Kubernetes cluster.

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

Remove credentials

Remove credentials from an account at any time. You should only remove credentials after unmanaging all associated clusters.

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

Steps

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

Result

Astra Control Center removes the credentials from the account.

Monitor account activity

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

> If you manage Kubernetes clusters from Astra Control and Astra Control is connected to Cloud Insights, Astra Control sends event logs to Cloud Insights. The log information, including information about pod deployment and PVC attachments, appears in the Astra Control Activity log. Use this information to identify any issues on the Kubernetes clusters you are managing.

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 the Astra Control API to update an existing license.

Steps

1. Log in to the NetApp Support Site.
2. Access the Astra Control Center Download page, enter the serial number, and download the full NetApp license file (NLF).

3. Log in to the Astra Control Center UI.
4. From the left navigation, select **Account** > **License**.
5. In the **Account** > **License** page, select the status drop-down menu for the existing license and select **Replace**.
6. Browse to the license file that you downloaded.
7. Select **Add**.

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

For more information

- Astra Control Center licensing

Manage 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

> 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.

What you'll need

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

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

Steps

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

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

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

Find more information

- Use the Astra Control API

Manage the storage backend

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

For instructions on how to manage storage backends using the Astra Control API, see the Astra Automation and API information.

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

- Add a storage backend
- View storage backend details
- Unmanage a storage backend
- 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

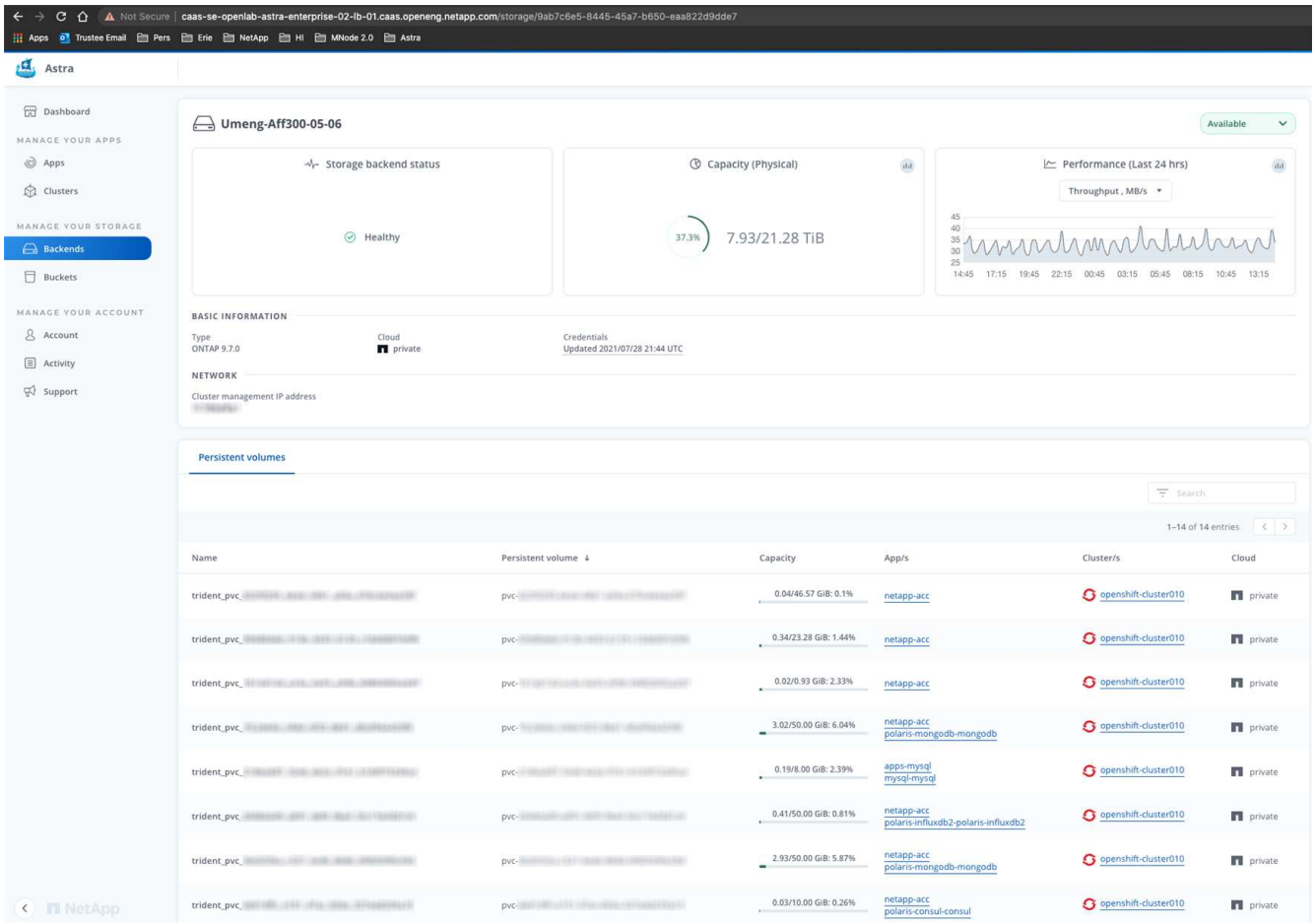
View information about the backend health, capacity, and performance (IOPS throughput and/or latency).

You can see the volumes that the Kubernetes apps are using, which are stored on a selected storage backend. With Cloud Insights, you can see additional information. See Cloud Insights documentation.

Steps

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

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



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

Unmanage a storage backend

You can unmanage the backend.

Steps

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.

What you'll need

- 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, see 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.

Monitor infrastructure with Cloud Insights, 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, create a connection to NetApp Cloud Insights, 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 NetApp Support Site or establish a connection to Cloud Insights), you should configure a proxy server in Astra Control Center.

- Connect to Cloud Insights
- Connect to Prometheus
- Connect to Fluentd

Add a proxy server for connections to Cloud Insights or to 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 NetApp Support Site or establish a connection to Cloud Insights), you should configure a proxy server in Astra Control Center.

> Astra Control Center does not validate the details you enter for your proxy server. Ensure that you enter 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 will be warned before you disable that potential disruption to other connections might occur.

Steps

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

Connect to Cloud Insights

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

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

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

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

What you'll need

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

> If you are new to Cloud Insights, familiarize yourself with the features and capabilities. See Cloud Insights documentation.

Steps

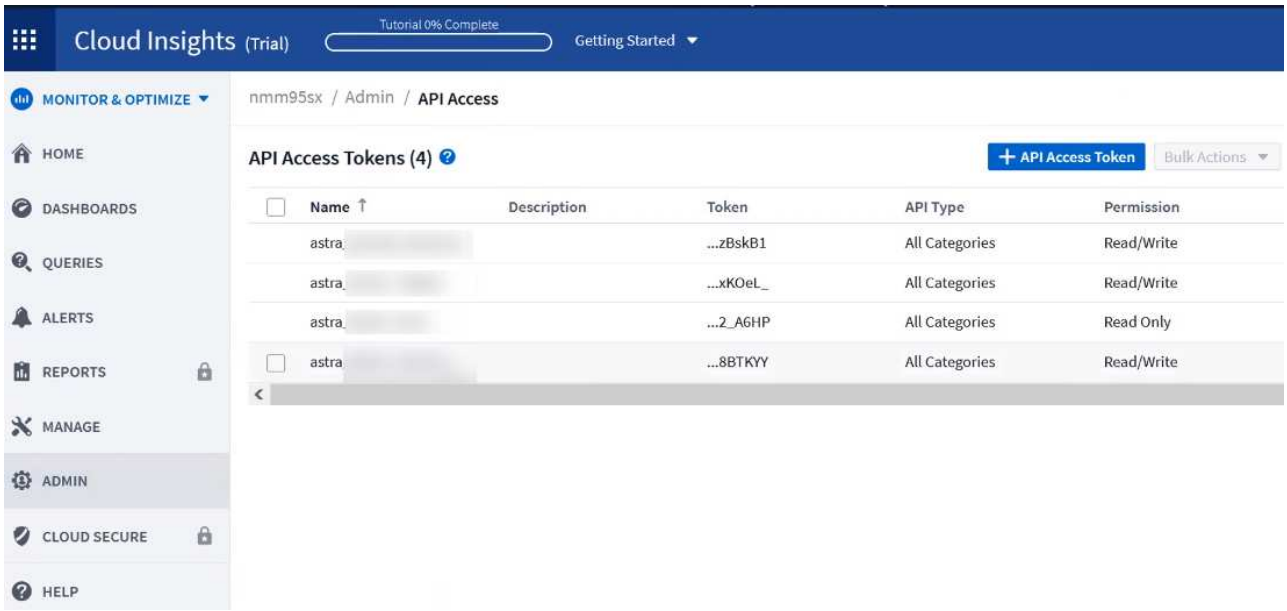
1. Log in to Astra Control Center using an account with **admin**/**owner** privilege.
2. Select **Account** > **Connections**.
3. Select **Connect** where it shows **Disconnected** in the drop-down list to add the connection.

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

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

   You get the tenant URL when you get the Cloud Insights license. If you do not have the tenant URL, see the Cloud Insights documentation.

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



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

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

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

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

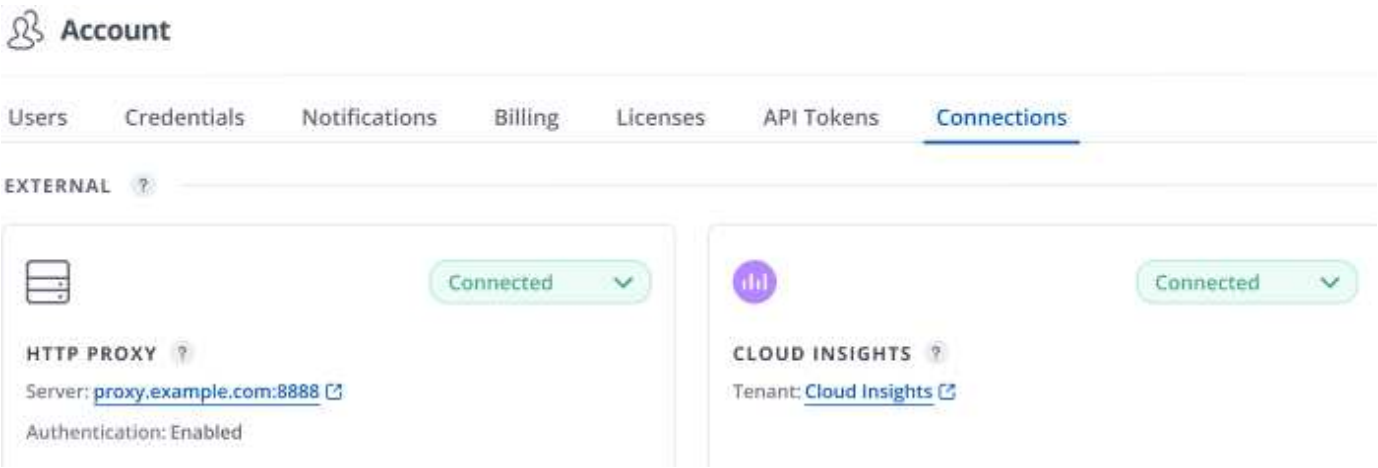
5. Select **Connect**.

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

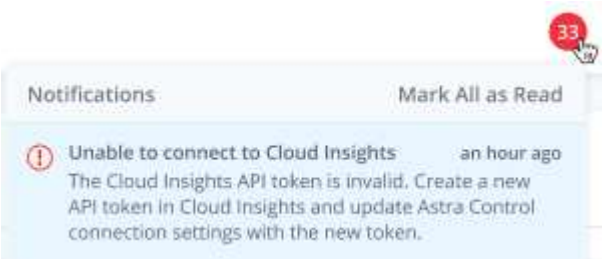
> To go back and forth easily between the Astra Control Center and Cloud Insights UIs, ensure that you are logged into both.

View data in Cloud Insights

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

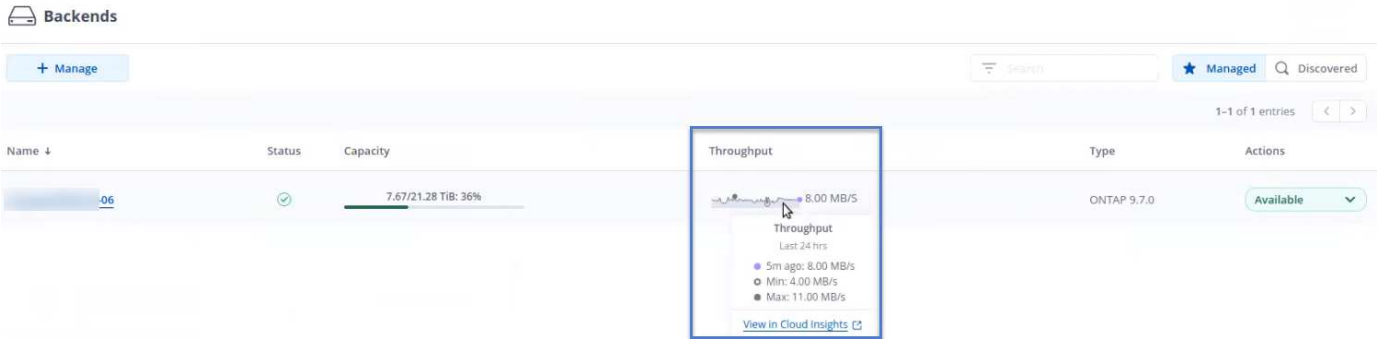


If the connection failed for some reason, the status shows **Failed**. You can find the reason for failure under **Notifications** at the top-right side of the UI.



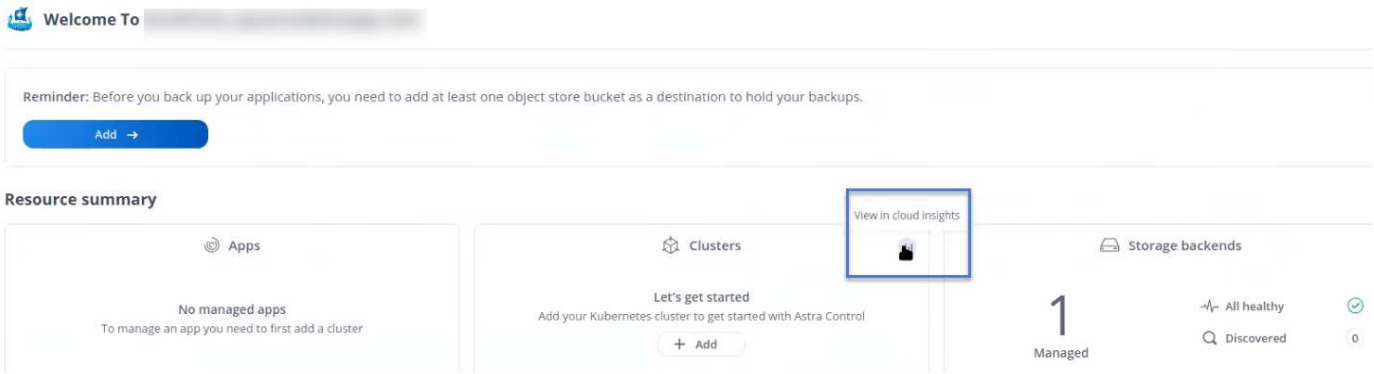
You can also find the same information under **Account** > **Notifications**.

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



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

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



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

Edit Cloud Insights connection

You can edit the Cloud Insights connection.

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

Steps

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

Disable Cloud Insights connection

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

Steps

1. Log in to Astra Control Center using an account with **admin**/**owner** privilege.
2. Select **Account** > **Connections**.
3. Select **Disconnect** from the drop-down list to disable the connection.
4. In the dialog box that opens, confirm the operation.
   After you confirm the operation, on the **Account** > **Connections** page, the Cloud Insights status changes to **Pending**. It take a few minutes for the status to change to **Disconnected**.

Connect to 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.

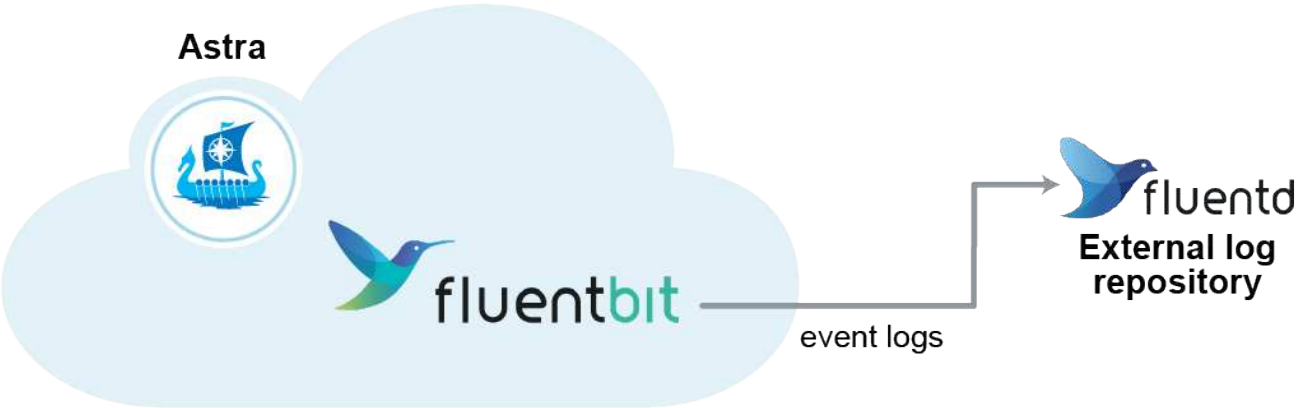
Steps

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 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.

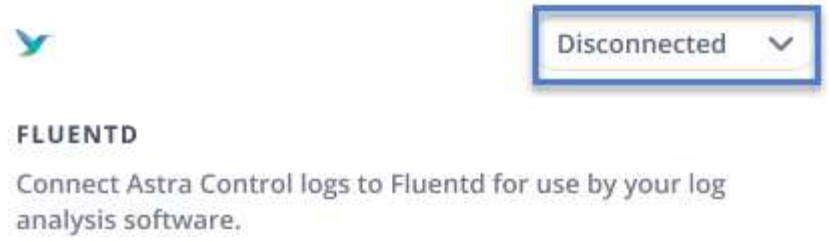
What you'll need

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

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

Steps

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



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.
- Trident won’t be uninstalled from the cluster. Learn how to uninstall 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.

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

Upgrade Astra Control Center

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

What you'll need

- Before you upgrade, refer to Operational environment requirements to ensure your environment still meets the minimum requirements for Astra Control Center deployment. Your environment should have the following:
  - A supported Astra Trident version
    To determine the version you are running, run the following command against your existing Astra Control Center:

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

    Refer to Astra Trident documentation to upgrade from an older version.

    > You must upgrade to Astra Trident 22.10 **PRIOR** to upgrading to Kubernetes 1.25.

  - A supported Kubernetes distribution
    To determine the version you are running, run the following command against your existing Astra Control Center: `kubectl get nodes -o wide`
  - Sufficient cluster resources
    To determine cluster resources, run the following command in your existing Astra Control Center cluster: `kubectl describe node <node name>`
  - A registry you can use to push and upload Astra Control Center images
  - A default storage class
    To determine your default storage class, run the following command against your existing Astra Control Center: `kubectl get storageclass`
- (OpenShift only) Ensure all cluster operators are in a healthy state and available.

  ```
  kubectl get clusteroperators
  ```

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

  ```
  kubectl get apiservices
  ```

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

About this task

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

- Download and extract Astra Control Center
- Remove the NetApp Astra kubectl plugin and install it again
- Add the images to your 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.

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

Download and extract Astra Control Center

1. Go to the Astra Control Center product downloads page on the NetApp Support Site. You can select the latest version or another version you want from the drop-down menu.
2. Download the bundle containing Astra Control Center (`astra-control-center-[version].tar.gz`).
3. (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.

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

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

Remove the NetApp Astra kubectl plugin and install it again

The NetApp Astra kubectl command line plugin saves time when performing common tasks associated with deploying and upgrading Astra Control Center.

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. To remove an existing version of kubectl-astra, run this command: `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 local registry

1. Complete the appropriate step sequence for your container engine:

Docker

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

```
acc.manifest.bundle.yaml
acc/
```

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.manifest.bundle.yaml
acc/
```

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=22.11.0-82
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=22.11.0-82
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://netappdownloads.jfrog.io/docker-astra-control-prod/netapp/astra/acc/22.11.0-82/image:version

Install the updated Astra Control Center operator

1. Change the directory:

```
cd manifests
```

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

```
vim astra_control_center_operator_deploy.yaml
```

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

```
imagePullSecrets:
- name: <astra-registry-cred_or_custom_name_of_secret>
```

b. Change `[your_registry_path]` for the `kube-rbac-proxy` image to the registry path where you pushed the images in a previous step.

c. Change `[your_registry_path]` for the `acc-operator` image to the registry path where you pushed the images in a previous step.

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

```
- name: ACCOP_HELM_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: [your_registry_path]/kube-rbac-proxy:v4.8.0
        name: kube-rbac-proxy
        ports:
        - containerPort: 8443
          name: https
      - args:
        - --health-probe-bind-address=:8081
        - --metrics-bind-address=127.0.0.1:8080
        - --leader-elect
        env:
        - name: ACCOP_LOG_LEVEL
          value: "2"
        - name: ACCOP_HELM_UPGRADETIMEOUT
          value: 300m
        image: [your_registry_path]/acc-operator:[version x.y.z]
        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
```

3. Install the updated Astra Control Center operator:

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

Sample response:

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

4. 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]
```

2. Change the Astra version number (`astraVersion` inside of `Spec`) to the version you are upgrading to:

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

3. 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 registry has changed since your last installation.

```
  imageRegistry:
    name: "[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
```

After you save and exit the file editor, the changes will be applied and the upgrade will begin.

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

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

7. 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      22.11.0-82
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.

Uninstall Astra Control Center

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

If you have any issues with the uninstall, see Troubleshooting uninstall issues.

What you'll need

- Use Astra Control Center UI to unmanage all clusters.

Steps

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

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

   Result:

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

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

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

   Result:

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

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

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

Result:

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

Troubleshooting uninstall issues

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

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

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

Steps

1. Delete `acc-monitoring` agent:

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

Result:

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

2. Delete the namespace:

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

Result:

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

3. Confirm resources removed:

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

Result:

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

4. Confirm monitoring agent removed:

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

Sample result:

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

5. Delete custom resource definition (CRD) information:

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

Result:

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

Uninstall of Astra Control Center fails to clean up Traefik CRDs

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

Steps

1. List Traefik CRDs installed on the cluster:

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

Response

```
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

Automate with Astra Control REST API

Automation using the Astra Control REST API

Astra Control has a REST API that enables you to directly access the Astra Control functionality using a programming language or utility such as Curl. You can also manage Astra Control deployments using Ansible and other automation technologies.

To set up and manage your Kubernetes apps, you can use either the Astra Control Center UI or the Astra Control API.

To learn more, go to the Astra automation docs.

Knowledge and support

Troubleshooting

Learn how to work around some common problems you might encounter.

NetApp Knowledge Base for Astra

Find more information

- How to upload a file to NetApp (login required)
- How to manually upload a file to NetApp (login required)

Get help

NetApp provides support for Astra Control in a variety of ways. Extensive free self-support options are available 24x7, such as knowledgebase (KB) articles and a Discord channel. Your Astra Control account includes remote technical support via web ticketing.

> If you have an evaluation license for Astra Control Center, you can get technical support. However, case creation via NetApp Support Site (NSS) is not available. You can get in touch with Support via the feedback option or use the Discord channel for self service.

You must first activate support for your NetApp serial number in order to use these non self-service support options. A NetApp Support Site (NSS) SSO account is required for chat and web ticketing along with case management.

Self-support options

You can access support options from the Astra Control Center UI by selecting the **Support** tab from the main menu.

These options are available for free, 24x7:

- **Knowledge base** (login required): Search for articles, FAQs, or Break Fix information related to Astra Control.
- **Documentation center**: This is the doc site that you're currently viewing.
- **Get help via Discord**: Go to Astra in The Pub category to connect with peers and experts.
- **Create a support case**: Generate support bundles to provide to NetApp Support for troubleshooting.
- **Give feedback about Astra Control**: Send an email to astra.feedback@netapp.com to let us know your thoughts, ideas, or concerns.

Enable daily scheduled support bundle upload to NetApp Support

During Astra Control Center installation, if you specify `enrolled: true` for `autoSupport` in the Astra Control Center Custom Resource (CR) file (`astra_control_center.yaml`), daily support bundles are automatically uploaded to the NetApp Support Site.

Generate support bundle to provide to NetApp Support

Astra Control Center enables the admin user to generate bundles, which include information useful to NetApp Support, including logs, events for all the components of the Astra deployment, metrics, and topology information about the clusters and apps under management. If you are connected to the Internet, you can upload support bundles to NetApp Support Site (NSS) directly from the Astra Control Center UI.

> The time taken by Astra Control Center to generate the bundle depends on the size of your Astra Control Center installation as well as the parameters of the requested support bundle. The time duration that you specified when requesting a support bundle dictates the time it takes for the bundle to be generated (for example, a shorter time period results in faster bundle generation).

Before you begin

Determine whether a proxy connection will be required to upload bundles to NSS. If a proxy connection is needed, verify that Astra Control Center has been configured to use a proxy server.

1. Select **Accounts** > **Connections**.
2. Check the proxy settings in **Connection settings**.

Steps

1. Create a case on the NSS portal using the license serial number listed on the **Support** page of the Astra Control Center UI.
2. Perform the following steps for generating the support bundle by using the Astra Control Center UI:
   a. On the **Support** page, in the Support bundle tile, select **Generate**.
   b. In the **Generate a Support Bundle** window, select the timeframe.

      You can choose between quick or custom timeframes.

      > You can choose a custom date range as well as specify a custom time period during the date range.

   c. After you make the selections, select **Confirm**.
   d. Select the **Upload the bundle to the NetApp Support Site when generated** check box.
   e. Select **Generate Bundle**.

      When the support bundle is ready, a notification appears on the **Accounts** > **Notification** page in the Alerts area, on the **Activity** page, and also in the notifications list (accessible by selecting the icon in the top-right side of the UI).

      If the generation failed, an icon appears on the Generate Bundle page. Select the icon to see the message.

> The notifications icon at the top-right side of the UI provides information about events related to the support bundle, such as when the bundle is successfully created, when the bundle creation fails, when the bundle could not be uploaded, when the bundle could not be downloaded, and so on.

If you have an air-gapped installation

If you have an air-gapped installation, perform the following steps after the Support bundle is generated. When the bundle is available for download, the Download icon appears next to **Generate** in the **Support Bundles** section of the **Support** page.

Steps

1. Select the Download icon to download the bundle locally.
2. Manually upload the bundle to NSS.

   You can use one of the following methods to do this:

   - Use NetApp Authenticated File Upload (login required).
   - Attach the bundle to the case directly on NSS.
   - Use NetApp Active IQ.

Find more information

- How to upload a file to NetApp (login required)
- How to manually upload a file to NetApp (login required)

Earlier versions of Astra Control Center documentation

Documentation for previous releases is available.

- Astra Control Center 22.08 documentation
- Astra Control Center 22.04 documentation
- Astra Control Center 21.12 documentation
- Astra Control Center 21.08 documentation

Legal notices

Legal notices provide access to copyright statements, trademarks, patents, and more.

Copyright

https://www.netapp.com/company/legal/copyright/

Trademarks

NETAPP, the NETAPP logo, and the marks listed on the NetApp Trademarks page are trademarks of NetApp, Inc. Other company and product names may be trademarks of their respective owners.

https://www.netapp.com/company/legal/trademarks/

Patents

A current list of NetApp owned patents can be found at:

https://www.netapp.com/pdf.html?item=/media/11887-patentspage.pdf

Privacy policy

https://www.netapp.com/company/legal/privacy-policy/

Open source

Notice files provide information about third-party copyright and licenses used in NetApp software.

- Notice for Astra Control Center

Astra Control API license

https://docs.netapp.com/us-en/astra-automation/media/astra-api-license.pdf