Astra Control Center 21.08 documentation

Astra Control Center

NetApp
February 12, 2024

Table of Contents

Astra Control Center 21.08 documentation

Release notes

We’re pleased to announce the initial 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 in this release of Astra Control Center

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

5 August 2021 (21.08)

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

Known issues with this release

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

The following known issues affect the current release:

- Incorrect ClusterRoleBinding created by Astra Control Center CRD during installation
- App with user-defined label goes into "removed" state
- Unable to stop running app backup
- Backup or clone fails for apps using PVCs with decimal units in Astra Control Center
- Astra Control Center UI slow to show changes to app resources such as persistent volume changes
- During app restore from backup Trident creates a larger PV than the original PV

- Clone performance impacted by large persistent volumes
- App clones fail using a specific version of PostgreSQL
- App clones fail when using Service Account level OCP Security Context Constraints (SCC)
- S3 buckets in Astra Control Center do not report available capacity
- Reusing buckets between instances of Astra Control Center causes failures
- Selecting a bucket provider type with credentials for another type causes data protection failures
- Backups and snapshots might not be retained during removal of an Astra Control Center instance
- Extra backups are retained as part of scheduled backup
- Clone operation can't use other buckets besides the default
- Managing a cluster with Astra Control Center fails when default kubeconfig file contains more than one context
- Can't determine ASUP tar bundle status in scaled environment
- Uninstall of Astra Control Center fails to clean up the monitoring-operator pod on the managed cluster
- Uninstall of Astra Control Center fails to clean up Traefik CRDs
- ASUP collection stuck in a generating or uploading state

Incorrect ClusterRoleBinding created by Astra Control Center CRD during installation

Apply the following patch to all Kubernetes clusters where the acc-operator version 21.08.65 has been deployed. It should also be applied if the acc-operator is re-deployed.

To resolve this issue:

1. Replace `ACC_NAMESPACE` in the script below with the namespace you used to deploy Astra Control Center.

```
cat <<EOF | kubectl apply -f -
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: acc-operator-manager-rolebinding
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: acc-operator-manager-role
subjects:
- kind: ServiceAccount
  name: default
  namespace: netapp-acc-operator
- apiGroup: rbac.authorization.k8s.io
  kind: Group
  name: system:serviceaccounts:ACC_NAMESPACE
EOF
```

2. Run the script.

The patch removes the following two subjects from `ClusterRoleBinding: "acc-operator-manager-rolebinding"`

```
- apiGroup: rbac.authorization.k8s.io
  kind: Group
  name: system:serviceaccounts
- apiGroup: ""
  kind: Group
  name: system:serviceaccounts
```

App with user-defined label goes into "removed" state

If you define an app with a non-existent k8s label, Astra Control Center will create, manage, and then immediately remove the app. To avoid this, add the k8s label to pods and resources after the app is managed by Astra Control Center.

Unable to stop running app backup

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

Backup or clone fails for apps using PVCs with decimal units in Astra Control Center

Volumes created with decimal units fail using the Astra Control Center backup or clone process. See the knowledgebase article for more information.

Astra Control Center UI slow to show changes to app resources such as persistent volume changes

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. This delay in the UI can also occur when any app resources are added or modified. In this case, a 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.

During app restore from backup Trident creates a larger PV than the original PV

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

Clone performance impacted by large persistent volumes

Clones of very large and consumed persistent volumes might be intermittently slow, dependent on cluster access to the object store. If the clone is hung and no data has been copied for more than 30 minutes, Astra Control terminates the clone action.

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 OCP 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.

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.

Reusing buckets between instances of Astra Control Center causes failures

If you try to reuse a bucket used by another or previous installation of Astra Control Center, backup and restore will fail. You must use a different bucket or completely clean out the previously used bucket. You can’t share buckets between instances of Astra Control Center.

Selecting a bucket provider type with credentials for another type causes data protection failures

When you add a bucket, select the correct bucket provider type with credentials that are correct for that provider. For example, the UI accepts NetApp ONTAP S3 as the type with StorageGRID credentials; however, this will cause all future app backups and restores using this bucket to 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.

Extra backups are retained as part of scheduled backup

Sometimes one or more backups in Astra Control Center are retained beyond the number specified to be retained in the backup schedule. These extra backups should be deleted as part of a scheduled backup but are not deleted and are stuck in a `pending` state. To resolve the issue, force delete the extra backups.

Clone operation can't use other buckets besides the default

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.

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.

Can't determine ASUP tar bundle status in scaled environment

During ASUP collection, the status of the bundle in the UI is reported as either `collecting` or `done`. Collection can take up to an hour for large environments. During ASUP download the network file transfer speed for the bundle might be insufficient, and the download might time out after 15 minutes without any indication in the UI. Download issues depend on the size of the ASUP, the scaled cluster size, and if collection time goes beyond the seven day limit.

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:

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

   Result:

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

2. Delete the namespace:

```
oc delete ns netapp-monitoring
```

Result:

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

3. Confirm resources removed:

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

Result:

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

4. Confirm monitoring agent removed:

```
oc get crd|grep agent
```

Sample result:

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

5. Delete custom resource definition (CRD) information:

```
oc 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:

Steps

1. Confirm which CRDs were not deleted by the uninstall process:

```
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
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
```

ASUP collection stuck in a generating or uploading state

If an ASUP pod is killed or restarted, an ASUP collection might become stuck in a generating or uploading state. Perform the following Astra Control REST API call to initiate manual collection again:

HTTP method	Path
POST	/accounts/{accountID}/core/v1/asups

This API workaround works only if performed more than 10 minutes after ASUP is started.

Find more information

- Known limitations for this release

Known limitations with this release

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.

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.

Cluster is in `removed` state although cluster and network are otherwise working as expected

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:

1. Run a POST call to add an updated kubeconfig file to the `/credentials` endpoint and retrieve the assigned `id` from the response body.
2. Run a PUT call from the `/clusters` endpoint using the appropriate cluster ID and set the `credentialID` to the `id` value from the previous step.

After you complete these steps, the credential associated with the cluster is updated and the cluster should reconnect and update its state to `available`.

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

Astra Control Center does not support apps that are deployed with Operator Lifecycle Manager (OLM)-enabled operators or cluster-scoped operators.

Cloning apps can only be done with same K8s distribution

If you clone an app between clusters, the source and destination clusters must be the same distribution of Kubernetes. For example, if you clone an app from an OpenShift 4.7 cluster, use a destination cluster that is also OpenShift 4.7.

OpenShift 4.8 is not supported

OpenShift 4.8 is not supported for the July release of Astra Control Center. For more information, see Astra Control Center requirements.

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

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.

Data protection for Astra Control Center as app not yet available

This release does not support the ability to manage Astra as an app using snapshot, backup, or restore options.

Unhealthy pods affect app management

If a managed app has pods in an unhealthy state, Astra Control can’t create new backups and clones.

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.

Trident isn’t uninstalled from a cluster

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.

Find more information

- Known issues for this release

Concepts

Intro to Astra Control

Astra Control is a Kubernetes application data lifecycle management solution that simplifies operations for stateful applications. Easily protect, back up, and migrate Kubernetes workloads, and instantly create working application clones.

Features

Astra Control offers critical capabilities for Kubernetes application data lifecycle management:

- Automatically manage persistent storage
- Create application-aware, on-demand snapshots and backups
- Automate policy-driven snapshot and backup operations
- Migrate applications and data from one Kubernetes cluster to another
- Easily clone an application from production to staging
- Visualize application health and protection status
- Use a user interface or an API to implement your backup and migration workflows

Astra Control continually watches your compute for state changes, so it’s aware of any new apps that you add along the way.

Deployment models

Astra Control is available in two deployment models:

- **Astra Control Service**: A NetApp-managed service that provides application-aware data management of Kubernetes clusters in Google Kubernetes Engine (GKE) and Azure Kubernetes Service (AKS).
- **Astra Control Center**: Self-managed software that provides application-aware data management of Kubernetes clusters running in your on-premises environment.

	Astra Control Service	Astra Control Center
How is it offered?	As a fully managed cloud service from NetApp	As software that you download, install, and manage
Where is it hosted?	On a public cloud of NetApp’s choice	On your provided Kubernetes cluster
How is it updated?	Managed by NetApp	You manage any updates
What are the app data management capabilities?	Same capabilities on both platforms with exceptions to backend storage or to external services	Same capabilities on both platforms with exceptions to backend storage or to external services
What is the backend storage support?	NetApp cloud service offerings	NetApp ONTAP AFF and FAS systems

Supported apps

Astra Control Center does not support apps that are deployed with Operator Lifecycle Manager (OLM)-enabled operators or cluster-scoped operators.

NetApp has validated some apps to ensure the safety and consistency of the snapshots and backups.

- Learn the difference between a validated app and a standard app in Astra Control Center.

No matter which type of app that you use with Astra Control, you should always test the backup and restore workflow yourself to ensure that you can meet your disaster recovery requirements.

How Astra Control Service works

Astra Control Service is a NetApp-managed cloud service that is always on and updated with the latest capabilities. It utilizes several components to enable application data lifecycle management.

At a high level, Astra Control Service works like this:

- You get started with Astra Control Service by setting up your cloud provider and by registering for an Astra account.
    - For GKE clusters, Astra Control Service uses NetApp Cloud Volumes Service for Google Cloud as the backend storage for your persistent volumes.
    - For AKS clusters, Astra Control Service uses Azure NetApp Files as the backend storage for your persistent volumes.
- You add your first Kubernetes compute to Astra Control Service. Astra Control Service then does the following:
    - Creates an object store in your cloud provider account, which is where backup copies are stored.

      In Azure, Astra Control Service also creates a resource group, a storage account, and keys for the Blob container.
    - Creates a new admin role and Kubernetes service account on the cluster.
    - Uses that new admin role to install Astra Trident on the cluster and to create one or more storage classes.
    - Uses Astra Trident to provision persistent volumes for your apps.
- At this point, you can add apps to your cluster. Persistent volumes will be provisioned on the new default storage class.
- You then use Astra Control Service to manage these apps, and start creating snapshots, backups, and clones.

  Astra Control Service continually watches your compute for state changes, so it’s aware of any new apps that you add along the way.

Astra Control’s Free Plan enables you to manage up to 10 apps in your account. If you want to manage more than 10 apps, then you’ll need to set up billing by upgrading from the Free Plan to the Premium Plan.

How Astra Control Center works

Astra Control Center runs locally in your own private cloud.

For the first release, Astra Control Center will support OpenShift Kubernetes clusters and Trident storage backends with ONTAP 9.5 and above.

In a cloud connected environment Astra Control Center uses Cloud Insights to provide advanced monitoring and telemetry. In the absence of a Cloud Insights connection, limited (7-days of metrics) monitoring and telemetry is available in Astra Control Center and also exported to Kubernetes native monitoring tools (such as Prometheus and Grafana) through open metrics end points.

Astra Control Center is fully integrated into the AutoSupport and Active IQ ecosystem to provide users and NetApp support with troubleshooting and usage information.

You can try Astra Control Center out using a 90-day evaluation license. The evaluation version is supported through email and community (Slack channel) options. Additionally, you have access to Knowledgebase articles and documentation from the in-product support dashboard.

To install and use Astra Control Center, you'll need to meet certain requirements.

At a high level, Astra Control Center works like this:

- You install Astra Control Center in your local environment. Learn more about how to install Astra Control Center.
- You complete some setup tasks such as these:
  - Set up licensing.
  - Add your first cluster.
  - Add backend storage that is discovered when you added the cluster.
  - Add an object store bucket that will store your app backups.

Learn more about how to set up Astra Control Center.

Astra Control Center does this:

- Discovers details about the managed Kubernetes clusters.
- Discovers your Astra Trident configuration on the clusters that you choose to manage and lets you monitor the storage backends.
- Discovers apps on those clusters and enables you to manage and protect the apps.

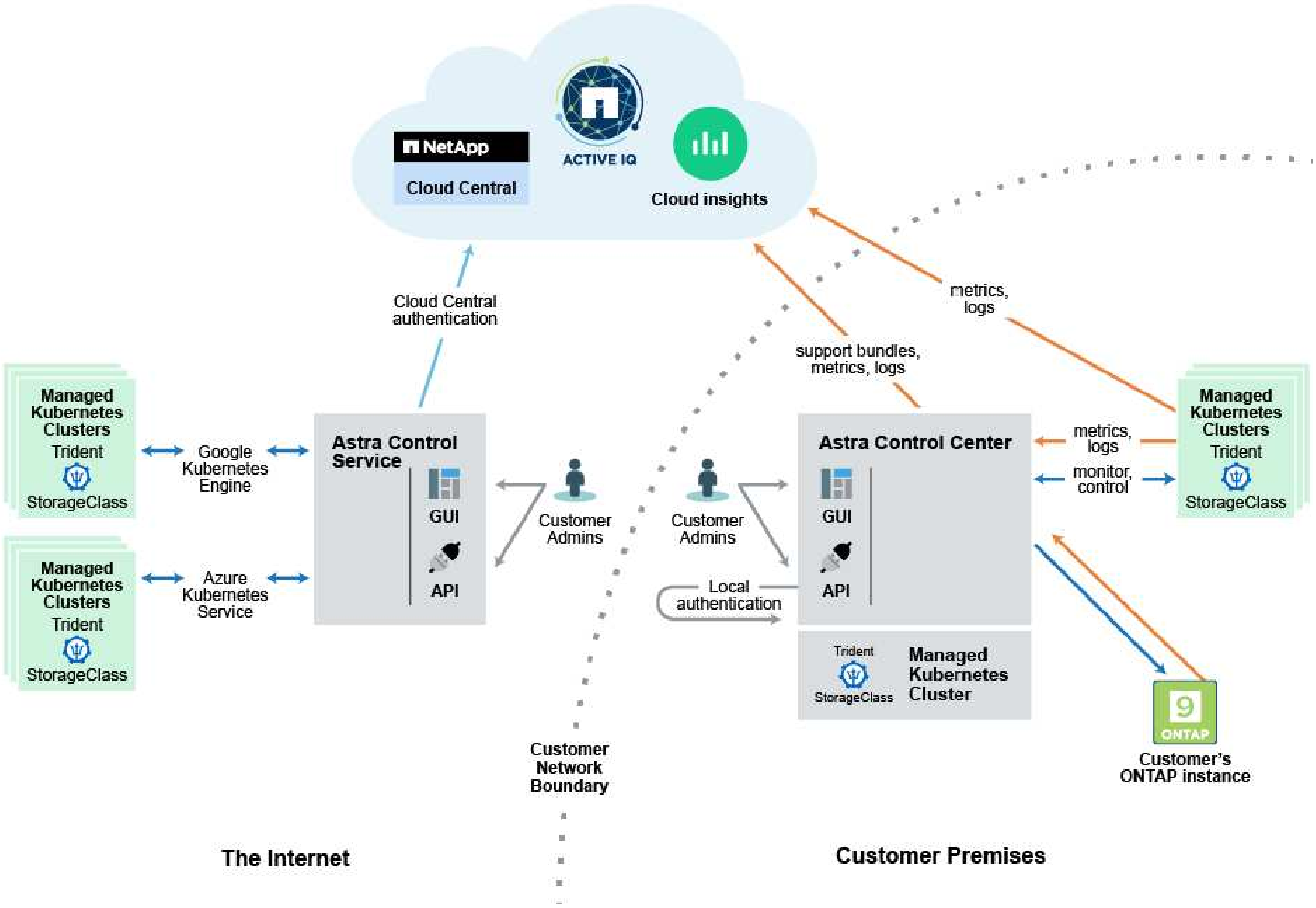
You can add apps to your cluster. Or, if you have some apps already in the cluster being managed, you can use Astra Control Center to discover and manage them. Then, use Astra Control Center to create snapshots, backups, and clones.

For more information

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

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 NetApp Cloud Volumes Service for Google Cloud as the backend storage for GKE clusters and Azure NetApp Files as the backend storage for AKS clusters.

  Astra Control Center uses an ONTAP AFF and FAS storage backend. 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 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 Control documentation
- Astra Trident documentation
- Use the Astra API
- Cloud Insights documentation
- ONTAP documentation

Validated vs standard apps

There are two types of applications you can bring to Astra Control: validated and standard. Learn the difference between these two categories and the potential impacts on your projects and strategy.

> It’s tempting to think of these two categories as "supported" and "unsupported." But as you will see, there is no such thing as an "unsupported" app in Astra Control. You can add any app to Astra Control, although validated apps have more infrastructure built around their Astra Control workflows compared to standard apps.

Validated apps

Validated apps for Astra Control include the following:

- MySQL 8.0.25
- MariaDB 10.5.9
- PostgreSQL 11.12
- Jenkins 2.277.4 LTS and 2.289.1 LTS

The list of validated apps represents applications that Astra Control recognizes. The Astra Control team has analyzed and confirmed these apps to be fully tested to restore. Astra Control executes custom workflows to help ensure application-level consistency of snapshots and backups.

If an app is validated, the Astra Control team has identified and implemented steps that can be taken to quiesce the app before taking a snapshot in order to obtain an application-consistent snapshot. For example, when Astra Control takes a backup of a PostgreSQL database, it first quiesces the database. After the backup is complete, Astra Control restores the database to normal operation.

No matter which type of app you use with Astra Control, always test the backup and restore workflow yourself to ensure that you can meet your disaster recovery requirements.

Standard apps

Other apps, including custom programs, are considered standard apps. You can add and manage standard apps through Astra Control. You can also create basic, crash-consistent snapshots and backups of a standard app. However, these have not been fully tested to restore the app to its original state.

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

Storage classes and persistent volume size

Astra Control Center supports ONTAP as the backend storage. You should understand how storage class and persistent volume (PV) size can help you meet your performance objectives.

Overview

Astra Control Center supports only Trident storage classes backed by ONTAP storage at this time. Astra Control Center discovers and uses the already-deployed resources, including ONTAP, Trident, and associated storage classes.

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

Storage classes

When you add clusters to Astra Control Center, you’re prompted to choose one of the previously discovered storage classes for persistent volumes. Service levels in storage classes are designed for different capacity and bandwidth needs. These discovered storage classes are eligible for use within Astra Control Center.

Persistent volume size and performance

See Trident information that provides cost comparisons and examples that can help you better understand how to couple a service level with volume size to meet your performance objectives.

Find more information

- Trident documentation on storage configuration

Get started

Astra Control Center requirements

Get started by verifying support for your Kubernetes clusters, apps, licenses, and web browser.

Kubernetes cluster general requirements

A Kubernetes cluster must meet the following general requirements so you can discover and manage it from Astra Control Center.

- **Image registry**: You must have an existing private Docker image registry to which you can push Astra Control Center build images. You must have the URL of the image registry where you will upload the images, and you must have tagged the images for the private container registry.
- **Trident / ONTAP storage configuration**: Astra Control Center requires that Trident version 21.01 or 21.04 already be installed and configured to work with NetApp ONTAP version 9.5 or newer as the storage backend. 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 Trident:
  - ontap-nas
  - ontap-nas-flexgroup
  - ontap-san
  - ontap-san-economy

If you are planning to manage the Kubernetes cluster from Astra Control Center as well as use the cluster to host the Astra Control Center installation, the cluster has the following additional requirements:

- The most recent version of the Kubernetes snapshot-controller component is installed
- A Trident volumesnapshotclass object has been defined by an administrator
- A default Kubernetes storage class exists on the cluster
- At least one storage class is configured to use Trident
- A method for pointing the FQDN of Astra Control Center to the external IP address of the Astra Control Center service

OpenShift clusters

Astra Control Center requires a Red Hat OpenShift Container Platform 4.6.8 or 4.7 cluster that has Trident storage classes backed by ONTAP 9.5 or newer, with the following attributes:

- At least 300GB of available ONTAP storage capacity
- 3 controller nodes with 4 CPU cores, 16GB RAM, and 120GB of available storage each
- 3 worker nodes with at least 12 CPU cores, 32GB RAM, and 50GB of available storage each
- Kubernetes version 1.19 or 1.20
- Service type "LoadBalancer" available for ingress traffic to be sent to services in the OpenShift cluster
- A method for pointing the FQDN of Astra Control Center to the load balanced IP address

> These minimum requirements assume that Astra Control Center is the only application running on the OpenShift cluster. If the cluster is running additional applications, you need to adjust these minimum requirements accordingly.

Make sure that your cluster meets the minimum requirements and that you follow Kubernetes best practices so that Astra Control Center is highly available in your Kubernetes cluster.

> OpenShift 4.8 is not supported.

> During app cloning, Astra Control Center needs to allow OpenShift to mount volumes and change the ownership of files. Because of this, ONTAP needs to be configured to allow volume operations to complete successfully using the following commands:
>
> 1. `export-policy rule modify -vserver svm0 -policyname default -ruleindex 1 -superuser sys`
> 2. `export-policy rule modify -policyname default -ruleindex 1 -anon 65534`

> If you plan to add a second OpenShift 4.6 or 4.7 cluster as a managed compute resource, you need to ensure that the Trident Volume Snapshot feature is enabled. See the official Trident instructions to enable and test Volume Snapshots with Trident.

App management requirements

Astra Control Center has the following app management requirements:

- **Licensing**: You need an Astra Control Center license to manage apps using Astra Control Center.
- **Helm 3**: 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.
- **Operator management**: Astra Control Center does not support apps that are deployed with Operator Lifecycle Manager (OLM)-enabled operators or cluster-scoped operators.

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. Without a license, you will be unable to:

- Define custom apps
- Create snapshots or clones of existing apps
- Configure data protection policies

If you want to try Astra Control Center, you can use a 90-day evaluation license.

Service type "LoadBalancer" for on-premises Kubernetes clusters

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. For on-premises OpenShift clusters, you can use MetalLB 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.

Networking requirements

The cluster 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 cluster hosting Astra Control Center and each managed cluster (noted where applicable).

Product	Port	Protocol	Direction	Purpose
Astra Control Center	443	HTTPS	Ingress	UI / API access - Ensure this port is open both ways between the cluster hosting Astra Control Center and each managed cluster
Astra Control Center	9090	HTTPS	• Ingress (to cluster hosting Astra Control Center) • Egress (random port from the node IP address of each worker node of each managed cluster)	Metrics data to metrics consumer - ensure each managed cluster can access this port on the cluster hosting Astra Control Center
Trident	34571	HTTPS	Ingress	Node pod communication
Trident	9220	HTTP	Ingress	Metrics endpoint

Supported web browsers

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

What’s next

View the quick start overview.

Quick start for Astra Control Center

This page provides a high-level 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.

Try it out! If you want to try Astra Control Center, you can use a 90-day evaluation license. See licensing information for details.

❶ Review Kubernetes cluster requirements

- Astra works with Kubernetes clusters with a Trident-configured ONTAP storage backend.
- Clusters must be running in a healthy state, with at least three online worker nodes.
- The cluster must be running Kubernetes.

Learn more about the Astra Control Center requirements.

❷ Download and install Astra Control Center

- Download Astra Control Center from the NetApp Support Site.
- Install Astra Control Center in your local environment.
- Discover your Trident configuration backed by the ONTAP storage backend.

  For our first release, you’ll install the images on an OpenShift registry or use your local registry.

Learn more about installing Astra Control Center.

❸ Complete some initial setup tasks

- Add a license.
- Add a Kubernetes cluster and Astra Control Center discovers details.
- Add an ONTAP storage backend.
- Optionally, add an object store bucket that will store your app backups.

Learn more about the initial setup process.

❹ Use Astra Control Center

After you finish setting up Astra Control Center, here’s what you might do next:

- Manage an app. Learn more about how to manage apps.
- Optionally, connect to NetApp Cloud Insights to display metrics on the health of your system, capacity, and throughput inside the Astra Control Center UI. Learn more about connecting to Cloud Insights.

❺ Continue from this Quick Start

Install Astra Control Center.

Find more information

- Use the Astra API

Install Astra Control Center

To install Astra Control Center, do the following steps:

- Install Astra Control Center
- Log in to the Astra Control Center UI

Install Astra Control Center

To install Astra Control Center, download the installation bundle from the NetApp Support Site and perform a series of commands to install Astra Control Center Operator and Astra Control Center in your environment. You can use this procedure to install Astra Control Center in internet-connected or air-gapped environments.

What you'll need

- Before you begin installation, prepare your environment for Astra Control Center deployment.
- From your OpenShift cluster, ensure all cluster operators are in a healthy state (`available` is `true`):

  ```
  oc get clusteroperators
  ```

- From your OpenShift cluster, ensure all API services are in a healthy state (`available` is `true`):

  ```
  oc get apiservices
  ```

About this task

The Astra Control Center installation process does the following:

- Installs the Astra components into the `netapp-acc` (or custom named) namespace.
- Creates a default account.
- Establishes a default administrative user email address and default one-time password of `ACC-<UUID_of_installation>` for this instance of Astra Control Center. This user is assigned the Owner role in the system and is needed for first time login to the UI.
- Helps you determine that all Astra Control Center pods are running.
- Installs the Astra UI.

> ⓘ Podman commands can be used in place of Docker commands if you are using Red Hat's Podman repository.

Steps

1. Download the Astra Control Center bundle (`astra-control-center-[version].tar.gz`) from the NetApp Support Site.

2. Download the zip of Astra Control Center certificates and keys from NetApp Support Site.
3. (Optional) Use the following command to verify the signature of the bundle:

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

4. Extract the images:

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

5. Change to the Astra directory.

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

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

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

   a. Log in to your Docker registry:

      ```
      docker login [Docker_registry_path]
      ```

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

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

7. (For registries with auth requirements only) 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:

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

   Sample response:

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

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

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

   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] --docker-server=[Docker_registry_path] --docker-username=[username] --docker-password=[token]
   ```

   Response

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

8. Edit the Astra Control Center operator deployment yaml

(astra_control_center_operator_deploy.yaml) to refer to your local registry and secret.

```
vim astra_control_center_operator_deploy.yaml
```

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

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

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

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

```
apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    control-plane: controller-manager
  name: acc-operator-controller-manager
  namespace: netapp-acc-operator
spec:
  replicas: 1
  selector:
    matchLabels:
      control-plane: controller-manager
  template:
    metadata:
      labels:
        control-plane: controller-manager
    spec:
      containers:
      - args:
        - --secure-listen-address=0.0.0.0:8443
        - --upstream=http://127.0.0.1:8080/
        - --logtostderr=true
        - --v=10
        image: [Docker_registry_path]/kube-rbac-proxy:v0.5.0
        name: kube-rbac-proxy
        ports:
        - containerPort: 8443
          name: https
      - args:
        - --health-probe-bind-address=:8081
        - --metrics-bind-address=127.0.0.1:8080
        - --leader-elect
        command:
        - /manager
        env:
        - name: ACCOP_LOG_LEVEL
          value: "2"
        image: [Docker_registry_path]/acc-operator:[version x.y.z]
        imagePullPolicy: IfNotPresent
      imagePullSecrets: []
```

9. Edit the Astra Control Center custom resource (CR) file (`astra_control_center_min.yaml`):

```
vim astra_control_center_min.yaml
```

If additional customizations are required for your environment, you can use `astra_control_center.yaml` as an alternative CR. `astra_control_center_min.yaml` is the default CR and is suitable for most installations.

Properties configured by the CR cannot be changed after initial Astra Control Center deployment.

a. Change `[Docker_registry_path]` to the registry path where you pushed the images in the previous step.

b. Change the `accountName` string to the name you want to associate with the account.

c. Change the `astraAddress` string to the FQDN you want to use in your browser to access Astra. Do not use `http://` or `https://` in the address. Copy this FQDN for use in a later step.

d. Change the `email` string to the default initial administrator address. Copy this email address for use in a later step.

e. Change `enrolled` for autoSupport to `false` for sites without internet connectivity or retain `true` for connected sites.

f. (Optional) Add a first name `firstName` and last name `lastName` of the user associated with the account. You can perform this step now or later within the UI.

g. (Optional) Change the `storageClass` value to another Trident storageClass resource if required by your installation.

h. If you are not using a registry that requires authorization, delete the `secret` line.

```
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: "[Docker_registry_path]"
    secret: "astra-registry-cred"
  storageClass: "ontap-gold"
```

10. 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
```

11. 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]
```

Sample response:

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

12. Run the following patch to correct cluster role binding.
13. Install Astra Control Center in the `netapp-acc` (or your custom) namespace:

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

Sample response:

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

14. Verify that all system components installed successfully.

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

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-5fdfff786f-gkv6z               1/1     Running   0
4m58s
activity-649f869bf7-jn5gs                    1/1     Running   0
3m14s
asup-79846b5fdc-s9s97                        1/1     Running   0
3m10s
authentication-84c78f5cf4-qhx9t              1/1     Running   0
118s
billing-9b8496787-v8rzv                      1/1     Running   0
2m54s
bucketservice-5fb876d9d5-wkfvz               1/1     Running   0
3m26s
cloud-extension-f9f4f59c6-dz6s6              1/1     Running   0
3m
cloud-insights-service-5676b8c6d4-6q7lv      1/1     Running   0
2m52s
composite-compute-7dcc9c6d6c-lxdr6           1/1     Running   0
2m50s
composite-volume-74dbfd7577-cd42b            1/1     Running   0
3m2s
credentials-75dbf46f9d-5qm2b                 1/1     Running   0
3m32s
entitlement-6cf875cb48-gkvhp                 1/1     Running   0
3m12s
features-74fd97bb46-vss2n                    1/1     Running   0
3m6s
fluent-bit-ds-2g9jb                          1/1     Running   0
113s
fluent-bit-ds-5tg5h                          1/1     Running   0
113s
fluent-bit-ds-qfxb8                          1/1     Running   0
113s
graphql-server-7769f98b86-p4qrv              1/1     Running   0
90s
identity-566c566cd5-ntfj6                    1/1     Running   0
3m16s
```

```
influxdb2-0                                  1/1     Running   0
4m43s
krakend-5cb8d56978-44q66                     1/1     Running   0
93s
license-66cbbc6f48-27kgf                     1/1     Running   0
3m4s
login-ui-584f7fd84b-dmdrp                    1/1     Running   0
87s
loki-0                                       1/1     Running   0
4m44s
metrics-ingestion-service-6dcfddf45f-mhnvh   1/1     Running   0
3m8s
monitoring-operator-78d67b4d4-nxs6v          2/2     Running   0
116s
nats-0                                       1/1     Running   0
4m40s
nats-1                                       1/1     Running   0
4m26s
nats-2                                       1/1     Running   0
4m15s
nautilus-9b664bc55-rn9t8                     1/1     Running   0
2m56s
openapi-dc5ddfb7d-6q8vh                      1/1     Running   0
3m20s
polaris-consul-consul-5tjs7                  1/1     Running   0
4m43s
polaris-consul-consul-5wbnx                  1/1     Running   0
4m43s
polaris-consul-consul-bfvl7                  1/1     Running   0
4m43s
polaris-consul-consul-server-0               1/1     Running   0
4m43s
polaris-consul-consul-server-1               1/1     Running   0
4m43s
polaris-consul-consul-server-2               1/1     Running   0
4m43s
polaris-mongodb-0                            2/2     Running   0
4m49s
polaris-mongodb-1                            2/2     Running   0
4m22s
polaris-mongodb-arbiter-0                    1/1     Running   0
4m49s
polaris-ui-6648875998-75d98                  1/1     Running   0
92s
polaris-vault-0                              1/1     Running   0
4m41s
```

```
polaris-vault-1                              1/1     Running   0
4m41s
polaris-vault-2                              1/1     Running   0
4m41s
storage-backend-metrics-69546f4fc8-m7lfj     1/1     Running   0
3m22s
storage-provider-5d46f755b-qfv89             1/1     Running   0
3m30s
support-5dc579865c-z4pwq                     1/1     Running   0
3m18s
telegraf-ds-4452f                            1/1     Running   0
113s
telegraf-ds-gnqxl                            1/1     Running   0
113s
telegraf-ds-jhw74                            1/1     Running   0
113s
telegraf-rs-gg6m4                            1/1     Running   0
113s
telemetry-service-6dcc875f98-zft26           1/1     Running   0
3m24s
tenancy-7f7f77f699-q7l6w                     1/1     Running   0
3m28s
traefik-769d846f9b-c9crt                     1/1     Running   0
83s
traefik-769d846f9b-l9n4k                     1/1     Running   0
67s
trident-svc-8649c8bfc5-pdj79                 1/1     Running   0
2m57s
vault-controller-745879f98b-49c5v            1/1     Running   0
4m51s
```

15. (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
    ```

16. When all the pods are running, verify installation success by retrieving the AstraControlCenter instance installed by the ACC Operator.

    ```
    kubectl get acc -o yaml -n netapp-acc
    ```

17. Check the `status.deploymentState` field in the response for the `Deployed` value. If deployment was unsuccessful, an error message appears instead.

You will use the `uuid` in the next step.

```
apiVersion: v1
items:
- apiVersion: astra.netapp.io/v1
  kind: AstraControlCenter
  metadata:
    creationTimestamp: "2021-07-28T21:36:49Z"
    finalizers:
    - astracontrolcenter.netapp.io/finalizer
   generation: 1
    name: astra
    namespace: netapp-acc
    resourceVersion: "27797604"
    selfLink: /apis/astra.netapp.io/v1/namespaces/netapp-
acc/astracontrolcenters/astra
    uid: 61cd8b65-047b-431a-ba35-510afcb845f1
  spec:
    accountName: Example
    astraAddress: astra.example.com
    astraResourcesScaler: "Off"
    astraVersion: 21.08.52
    autoSupport:
      enrolled: false
    email: admin@example.com
    firstName: SRE
    lastName: Admin
    imageRegistry:
      name: registry_name/astra
  status:
    certManager: deploy
    deploymentState: Deployed
    observedGeneration: 1
    observedVersion: 21.08.52
    postInstall: Complete
    uuid: c49008a5-4ef1-4c5d-a53e-830daf994116
kind: List
metadata:
  resourceVersion: ""
  selfLink: ""
```

18. To get the one-time password you will use when you log in to Astra Control Center, copy the `status.uuid` value from the response in the previous step. The password is `ACC-` followed by the UUID value (`ACC-[UUID]` or, in this example, `ACC-c49008a5-4ef1-4c5d-a53e-830daf994116`).

Log in to the Astra Control Center UI

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

Steps

1. In a browser, enter the FQDN you used in the `astraAddress` in the `astra_control_center_min.yaml` CR when you installed ACC.
2. Accept the self-signed certificates when 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_min.yaml` CR when you installed ACC, followed by the one-time 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 --follow -n netapp-acc-operator $(kubectl get pods -n
   netapp-acc-operator -o name)  -c manager
   ```

What’s next

Complete the deployment by performing setup tasks.

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
- Add cluster
- Add a storage backend
- Add a bucket

Add a license for Astra Control Center

You can add a new license using the 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.

What you'll need

When you downloaded Astra Control Center from the NetApp Support Site, you also downloaded the NetApp license file (NLF). Ensure you have access to this license file.

> ⓘ To update an existing evaluation or full license, see Update an existing license.

Add a full or evaluation license

Astra Control Center licenses measure CPU resources using Kubernetes CPU units. The license needs to account for the CPU resources assigned to the worker nodes of all the managed Kubernetes clusters. Before you add a license, you need to obtain the license file (NLF) from the NetApp Support Site.

You can also 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.

> ⓘ 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.

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, 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.

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. You can use the **Add Cluster** feature to manage a cluster with Astra Control Center.

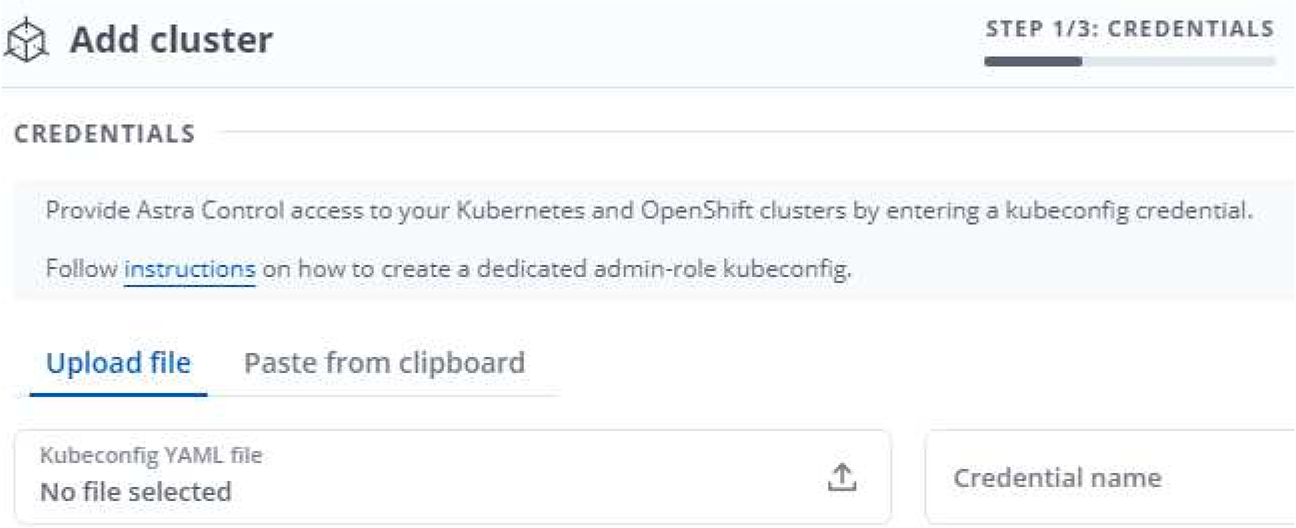
What you'll need

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

Steps

1. From the **Dashboard** in the Astra Control Center UI, select **Add** in the Clusters section.
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**.

   Add cluster

   STEP 1/3: CREDENTIALS

   CREDENTIALS

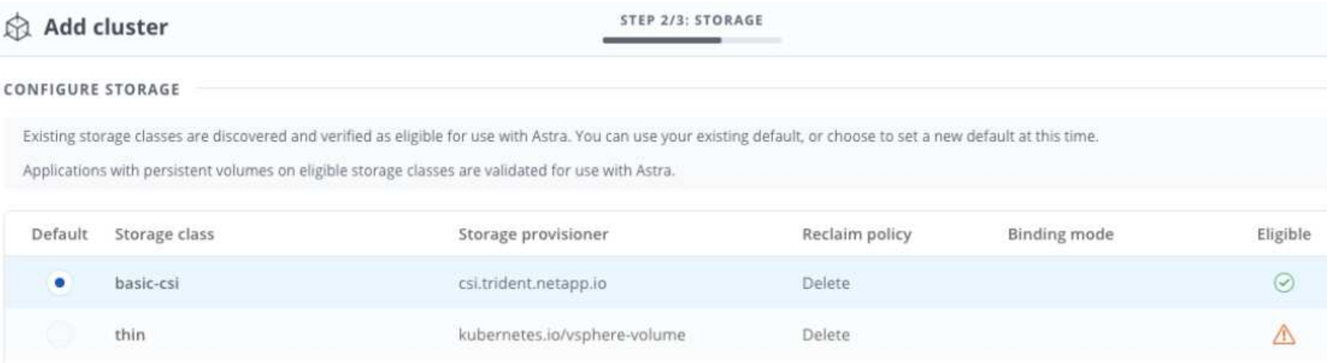
   Provide Astra Control access to your Kubernetes and OpenShift clusters by entering a kubeconfig credential.

   Follow instructions on how to create a dedicated admin-role kubeconfig.

   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.

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

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



Add cluster

STEP 2/3: STORAGE

CONFIGURE STORAGE

Existing storage classes are discovered and verified as eligible for use with Astra. You can use your existing default, or choose to set a new default at this time.

Applications with persistent volumes on eligible storage classes are validated for use with Astra.

Default	Storage class	Storage provisioner	Reclaim policy	Binding mode	Eligible
●	basic-csi	csi.trident.netapp.io	Delete		✓
○	thin	kubernetes.io/vsphere-volume	Delete		⚠

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

Result

The cluster enters the **Discovering** status and then changes to **Running**. You have successfully added a Kubernetes cluster and are now managing it in 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 a storage backend so that Astra Control can 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.

You can add a storage backend in the following ways:

- Configure storage when you are adding a cluster. See Add cluster.
- Add a discovered storage backend using either the Dashboard or the Backends option.

You can add an already discovered storage backend using these options:

- Add storage backend using Dashboard
- Add storage backend using Backends option

Add storage backend using Dashboard

1. From the Dashboard do one of the following:
   a. From the Dashboard Storage backend section, select **Manage**.
   b. From the Dashboard Resource Summary > Storage backends section, select **Add**.
2. Enter the ONTAP admin credentials and select **Review**.
3. Confirm the backend details and select **Manage**.

   The backend appears in the list with summary information.

Add storage backend using Backends option

1. In the left navigation area, select **Backends**.
2. Select **Manage**.
3. Enter the ONTAP admin credentials and select **Review**.
4. Confirm the backend details and select **Manage**.

   The backend appears in the list with summary information.

5. To see details of the backend storage, select it.

   > Persistent volumes used by apps in the managed compute cluster are also displayed.

Add a bucket

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.

When you add a bucket, Astra Control marks one bucket as the default bucket indicator. The first bucket that you create becomes the default bucket.

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

Use any of the following bucket types:

- NetApp ONTAP S3
- NetApp StorageGRID S3
- Generic S3

> 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.

For instructions on how to add buckets using the Astra API, see Astra Automation and API information.

Steps

1. In the left navigation area, select **Buckets**.
   a. Select **Add**.
   b. Select the bucket type.

      > When you add a bucket, select the correct bucket provider type with credentials that are correct for that provider. For example, the UI accepts NetApp ONTAP S3 as the type with StorageGRID credentials; however, this will cause all future app backups and restores using this bucket to fail.

   c. Create a new bucket name or 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.

d. Enter the name or IP address of the S3 server.

e. If you want this bucket to be the default bucket for all backups, check the `Make this bucket the default bucket for this private cloud` option.

This option does not appear for the first bucket you create.

f. Continue by adding credential information.

Add S3 access credentials

Add S3 access credentials at any time.

Steps

1. From the Buckets dialog, select either the **Add** or **Use existing** tab.
   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.

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 users
- Start managing apps
- Protect apps
- Clone apps
- Manage notifications
- Connect to Cloud Insights
- Add a custom TLS certificate

Find more information

- Use the Astra API
- Known issues

Prerequisites for adding a cluster

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

What you'll need before you add a cluster

- A cluster running OpenShift 4.6 or 4.7, which has Trident StorageClasses backed by ONTAP 9.5 or later.

  - One or more worker nodes with at least 1GB RAM available for running telemetry services.

    > If you plan to add a second OpenShift 4.6 or 4.7 cluster as a managed compute resource, you should ensure that the Trident Volume Snapshot feature is enabled. See the official Trident instructions to enable and test Volume Snapshots with Trident.

- The superuser and user ID set on the backing ONTAP system to back up and restore apps with Astra Control Center (ACC). Run the following commands in the ONTAP command line:
  `export policy rule modify -vserver svm0 -policyname default -ruleindex 1 -superuser sys`
  `export-policy rule modify -policyname default -ruleindex 1 -anon 65534` (this is the default value)

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   21.04.0
   ```

   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. See the Trident documentation for instructions.

2. Check 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 storageClass -A
   NAME                   PROVISIONER                    RECLAIMPOLICY
   VOLUMEBINDINGMODE   ALLOWVOLUMEEXPANSION   AGE
   ontap-gold (default)   csi.trident.netapp.io          Delete
   Immediate           true                   5d23h
   thin                   kubernetes.io/vsphere-volume   Delete
   Immediate           false                  6d
   ```

Create an admin-role kubeconfig

Ensure that you have the following on your machine before you do the steps:

- `kubectl` v1.19 or later installed
- An active kubeconfig with cluster admin rights for the active context

Steps

1. Create a service account as follows:
   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>
      ```

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

   b. Apply the service account:

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

2. Grant cluster admin permissions as follows:
   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>
      ```

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

b. Apply the cluster role binding:

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

3. 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.

4. Generate the kubeconfig as follows:

a. Create a `create-kubeconfig.sh` file. If the token index you noted in the previous step was not 0, replace the value for `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 the value for
```

```
TOKEN_INDEX from
# the output in the previous step if it was not 0. 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'
TOKEN_INDEX=0

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
```

5. (**Optional**) Rename the kubeconfig to a meaningful name for your cluster. Protect your cluster credential.

```
chmod 700 create-kubeconfig.sh
mv kubeconfig-sa.txt YOUR_CLUSTER_NAME_kubeconfig
```

What’s next?

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

Find more information

- Trident documentation
- Use the Astra API

Add a custom TLS certificate

You can remove the existing self-signed TLS certificate and replace it with a TLS certificate signed by a Certificate Authority (CA).

What you’ll need

- Kubernetes cluster with Astra Control Center installed
- Administrative access to a command shell on the cluster to run `kubectl` commands
- Private key and certificate files from the CA

Remove the self-signed certificate

1. Using SSH, log in to the Kubernetes cluster that hosts Astra Control Center as an administrative user.

2. Find the TLS secret associated with the current certificate using the following command, replacing `<ACC-deployment-namespace>` with the Astra Control Center deployment namespace:

   ```
   kubectl get certificate -n <ACC-deployment-namespace>
   ```

3. Delete the currently installed secret and certificate using the following commands:

   ```
   kubectl delete cert cert-manager-certificates -n <ACC-deployment-
   namespace>
   kubectl delete secret secure-testing-cert -n <ACC-deployment-namespace>
   ```

Add a new certificate

1. Use the following command to create the new TLS secret with the private key and certificate files from the CA, replacing the arguments in brackets <> with the appropriate information:

   ```
   kubectl create secret tls <secret-name> --key <private-key-filename>
   --cert <certificate-filename> -n <ACC-deployment-namespace>
   ```

2. Use the following command and example to edit the cluster Custom Resource Definition (CRD) file and change the `spec.selfSigned` value to `spec.ca.secretName` to refer to the TLS secret you created earlier:

   ```
   kubectl edit clusterissuers.cert-manager.io/cert-manager-certificates -n
   <ACC-deployment-namespace>
   ....

   #spec:
   #  selfSigned: {}

   spec:
     ca:
       secretName: <secret-name>
   ```

3. Use the following command and example output to validate that the changes are correct and the cluster is ready to validate certificates, replacing `<ACC-deployment-namespace>` with the Astra Control Center deployment namespace:

```
kubectl describe clusterissuers.cert-manager.io/cert-manager-
certificates -n <ACC-deployment-namespace>
....

Status:
  Conditions:
    Last Transition Time:  2021-07-01T23:50:27Z
    Message:               Signing CA verified
    Reason:                KeyPairVerified
    Status:                True
    Type:                  Ready
Events:                    <none>
```

4. Create the `certificate.yaml` file using the following example, replacing the placeholder values in brackets <> with appropriate information:

```
apiVersion: cert-manager.io/v1
kind: Certificate
metadata:
  name: <certificate-name>
  namespace: <ACC-deployment-namespace>
spec:
  secretName: <certificate-secret-name>
  duration: 2160h # 90d
  renewBefore: 360h # 15d
  dnsNames:
  - <astra.dnsname.example.com> #Replace with the correct Astra Control
Center DNS address
  issuerRef:
    kind: ClusterIssuer
    name: cert-manager-certificates
```

5. Create the certificate using the following command:

```
kubectl apply -f certificate.yaml
```

6. Using the following command and example output, validate that the certificate has been created correctly and with the arguments you specified during creation (such as name, duration, renewal deadline, and DNS names).

```
kubectl describe certificate -n <ACC-deployment-namespace>
....

Spec:
  Dns Names:
    astra.example.com
  Duration:  125h0m0s
  Issuer Ref:
    Kind:        ClusterIssuer
    Name:        cert-manager-certificates
  Renew Before:  61h0m0s
  Secret Name:   <certificate-secret-name>
Status:
  Conditions:
    Last Transition Time:  2021-07-02T00:45:41Z
    Message:               Certificate is up to date and has not expired
    Reason:                Ready
    Status:                True
    Type:                  Ready
  Not After:               2021-07-07T05:45:41Z
  Not Before:              2021-07-02T00:45:41Z
  Renewal Time:            2021-07-04T16:45:41Z
  Revision:                1
Events:                    <none>
```

7. Edit the ingress CRD TLS option to point to your new certificate secret using the following command and example, replacing the placeholder values in brackets <> with appropriate information:

```
kubectl edit ingressroutes.traefik.containo.us -n <ACC-deployment-
namespace>
....

# tls:
#    options:
#      name: default
#    secretName: secure-testing-cert
#    store:
#      name: default

 tls:
    options:
      name: default
    secretName: <certificate-secret-name>
    store:
      name: default
```

8. Using a web browser, browse to the deployment IP address of Astra Control Center.
9. Verify that the certificate details match the details of the certificate you installed.
10. Export the certificate and import the result into the certificate manager in your web browser.

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_min.yaml custom resource definition (CRD) 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_min.yaml CRD.

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

- From the left navigation, select **Account** > **License**.
- Select **Add license**.
- 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.

Will NetApp Trident be uninstalled when I remove a Kubernetes cluster from Astra Control?

Trident will not be uninstalled from a cluster when you remove it from Astra Control.

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."

Data management operations

There are snapshots in my account that I didn’t create. Where did they come from?

In some situations, Astra Control will automatically create a snapshot as part of a backup, clone or restore process.

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

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.

Use Astra

Manage apps

Start managing apps

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

Install apps on your cluster

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

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

- Deploy MariaDB from a Helm chart
- Deploy MySQL from a Helm chart
- Deploy Postgres from a Helm chart
- Deploy Jenkins from a Helm chart

Manage apps

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

> ⓘ Apps deployed with Helm 2 are not supported.

You can perform the following activities to manage apps:

- Manage apps
  - Manage apps by namespace
  - Manage apps by Kubernetes label
- Ignore apps
- Unmanage apps

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

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

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

Manage apps by namespace

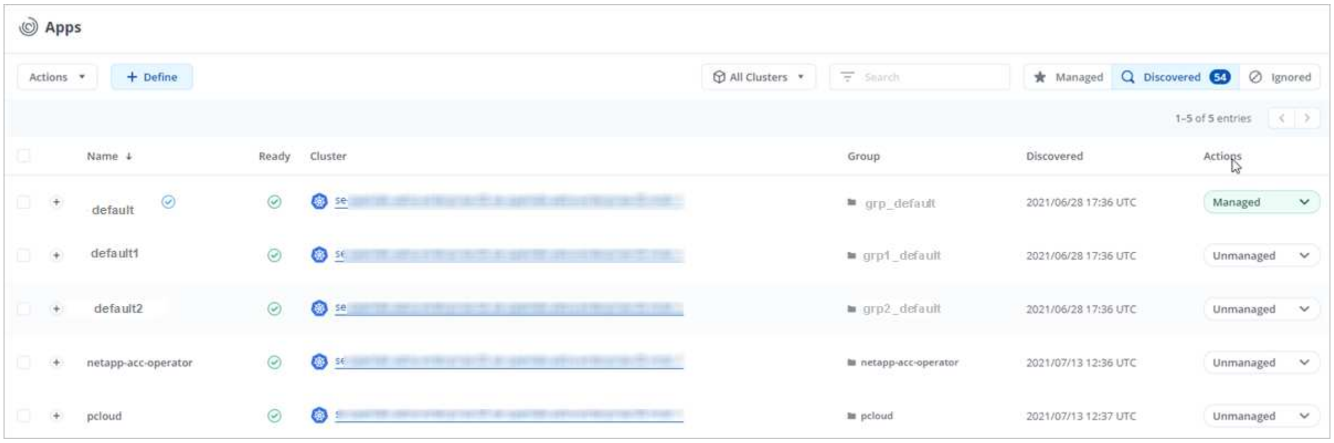
The **Discovered** section of the Apps page shows namespaces and any Helm-installed apps or custom-labeled apps in those namespaces. You can choose to manage each app individually or at the namespace level. It all comes down to the level of granularity that you need for data protection operations.

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

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

Steps

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



Name ↓	Ready	Cluster	Group	Discovered	Actions
default	✓	[redacted]	grp_default	2021/06/28 17:36 UTC	Managed
default1	✓	[redacted]	grp1_default	2021/06/28 17:36 UTC	Unmanaged
default2	✓	[redacted]	grp2_default	2021/06/28 17:36 UTC	Unmanaged
netapp-acc-operator	✓	[redacted]	netapp-acc-operator	2021/07/13 12:36 UTC	Unmanaged
pcloud	✓	[redacted]	pcloud	2021/07/13 12:37 UTC	Unmanaged

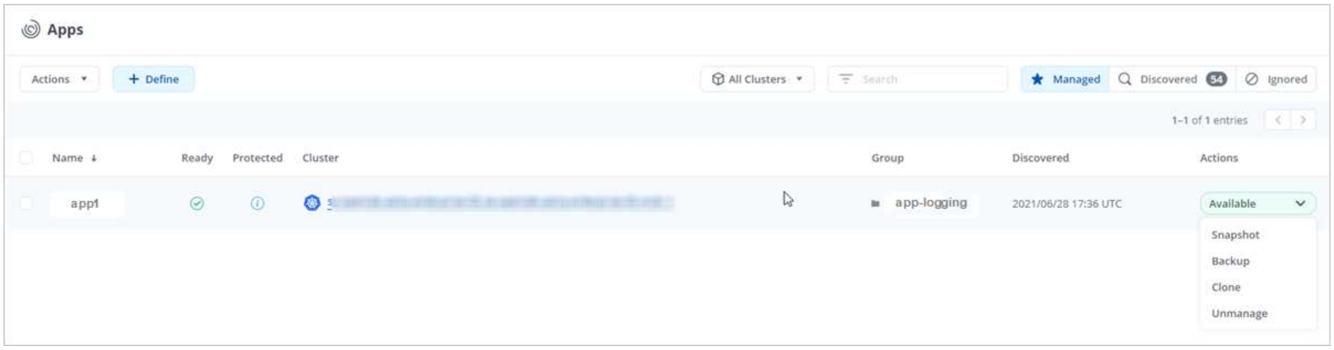
3. View the list of discovered namespaces. Expand the namespace to view the apps and associated resources.

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

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

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

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



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

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

Result

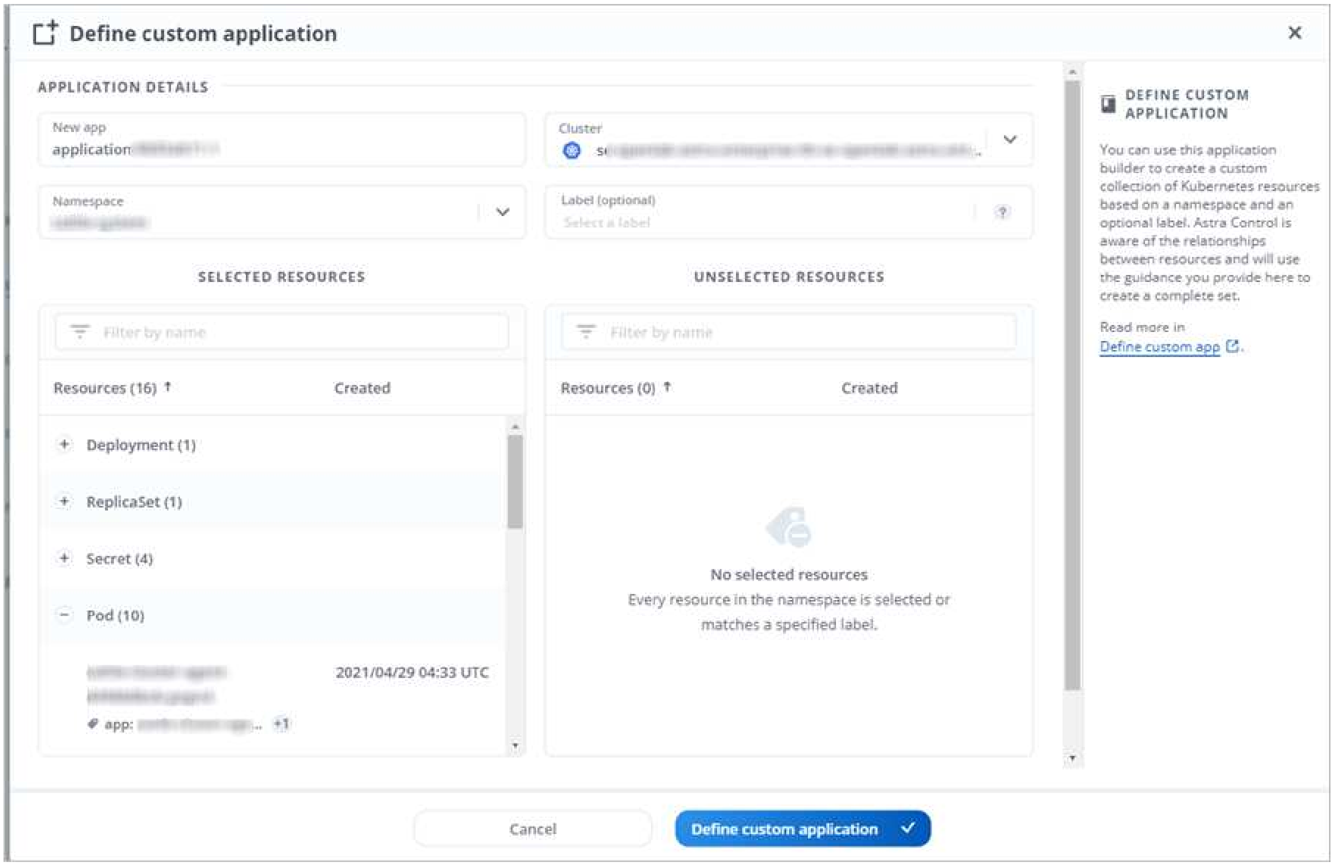
Apps that you chose to manage are now available from the **Managed** tab. Any ignored apps will move to the **Ignored** tab. Ideally, the Discovered tab will show zero apps, so that as new apps are installed, they are easier to find and manage.

Manage apps by Kubernetes label

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

Steps

1. From the left navigation bar, select **Apps**.
2. Select **Define**.



Define custom application

APPLICATION DETAILS

New app
application[redacted]

Cluster
s[redacted]

Namespace
[redacted]

Label (optional)
Select a label

SELECTED RESOURCES

Filter by name

Resources (16) ↑	Created
+ Deployment (1)	
+ ReplicaSet (1)	
+ Secret (4)	
− Pod (10)	
[redacted] app: [redacted] +1	2021/04/29 04:33 UTC

UNSELECTED RESOURCES

Filter by name

Resources (0) ↑	Created

No selected resources
Every resource in the namespace is selected or matches a specified label.

DEFINE CUSTOM APPLICATION

You can use this application builder to create a custom collection of Kubernetes resources based on a namespace and an optional label. Astra Control is aware of the relationships between resources and will use the guidance you provide here to create a complete set.

Read more in Define custom app.

Cancel

Define custom application ✓

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

        After you choose a label, it displays in the **Label** field. Astra Control also updates the **Unselected Resources** section to show the resources that don't match the selected label.
   f. **Unselected Resources**: Verify the app resources that you don't want to protect.
4. Click **Define custom application**.

Result

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

Ignore apps

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

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

Steps

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

Unmanage apps

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

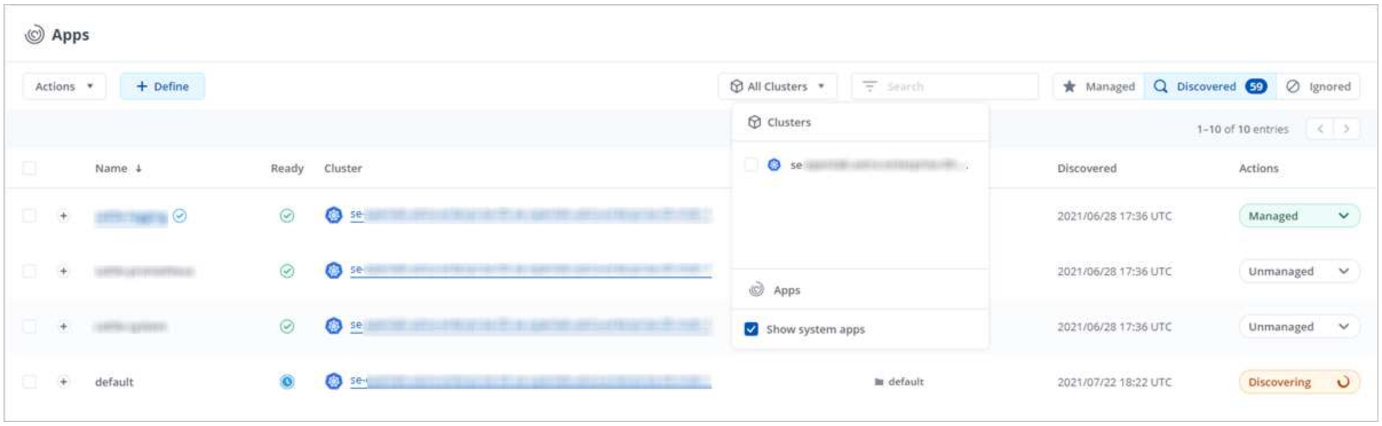
> If you unmanage an app, any backups or snapshots that were created earlier will be lost.

Steps

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

What about system apps?

Astra Control also discovers the system apps running on a Kubernetes cluster. You can display system apps by selecting the **Show system apps** checkbox under the Cluster filter in the toolbar.



Name	Ready	Cluster		Discovered	Actions
[redacted] ✓	✓	se[redacted]		2021/06/28 17:36 UTC	Managed
[redacted]	✓	se[redacted]		2021/06/28 17:36 UTC	Unmanaged
[redacted]	✓	se[redacted]		2021/06/28 17:36 UTC	Unmanaged
default		se[redacted]	default	2021/07/22 18:22 UTC	Discovering

We don’t show you these system apps by default because it’s rare that you’d need to back them up.

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

Find more information

- Use the Astra API

Define a custom app example

Creating a custom app lets you group elements of your Kubernetes cluster into a single app.

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

- Clone
- Snapshot
- Backup
- Protection Policy

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

To create a custom app, go to the Apps page and click **+ Define**.

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

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

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

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

Example: Separate Protection Policy for canary release

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

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

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

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

Steps

1. After the team adds the cluster to Astra Control, the next step is to define a custom app. To do this, the team clicks the **+ Define** button on the Apps page.
2. In the pop-up window which appears, the team sets `devops-canary-deployment` as the app name. The team chooses the cluster in the **Cluster** drop-down, then the app's namespace from the **Namespace** drop-down.
3. The team can either type `deployment=canary` in the **Labels** field, or select that label from the resources listed below.
4. After defining the custom app for the canary deployment, the team repeats the process for the stable deployment.

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

Protect apps

Protect apps with snapshots and backups

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

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

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

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

Snapshots and backups

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. Local snapshots are used 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.

A *backup* is stored in the external object store. A backup can be slower to take compared to local snapshots. You can migrate an app by restoring its backup to a different cluster. You can also choose a longer retention period for backups.

! *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.

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.

Steps

1. Click **Apps** and then click the name of an app.
2. Click **Data Protection**.
3. Click **Configure Protection Policy**.
4. Define a protection schedule by choosing the number of snapshots and backups to keep hourly, daily, weekly, and monthly.

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

   The following example sets four protection schedules: hourly, daily, weekly, and monthly for snapshots and backups.
5. Click **Review**.
6. Click **Set Protection Policy.**

Result

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

Create a snapshot

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

Steps

1. Click **Apps**.
2. Click the drop-down list in the **Actions** column for the desired app.
3. Click **Snapshot**.
4. Customize the name of the snapshot and then click **Review**.
5. Review the snapshot summary and click **Snapshot**.

Result

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

Create a backup

You can also back up an app at any time.

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

Steps

1. Click **Apps**.
2. Click the drop-down list in the **Actions** column for the desired app.
3. Click **Backup**.
4. Customize the name of the backup.
5. 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.
6. Choose a destination for the backup by selecting from the list of storage buckets.
7. Click **Review**.
8. Review the backup summary and click **Backup**.

Result

Astra Control Center creates a backup of the app.

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

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

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

View snapshots and backups

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

Steps

1. Click **Apps** and then click the name of an app.
2. Click **Data Protection**.

   The snapshots display by default.

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

Delete snapshots

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

Steps

1. Click **Apps** and then click the name of an app.
2. Click **Data Protection**.
3. Click the drop-down list in the **Actions** column for the desired snapshot.
4. Click **Delete snapshot**.

5. Type the word "delete" to confirm deletion and then click **Yes, Delete snapshot**.

Result

Astra Control Center deletes the snapshot.

Delete backups

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

> ⓘ There is no way to stop a running backup. If you need to delete the backup, wait until it has completed and then use these instructions. To delete a failed backup, use the Astra API.

1. Click **Apps** and then click the name of an app.
2. Click **Data Protection**.
3. Click **Backups**.
4. Click the drop-down list in the **Actions** column for the desired backup.
5. Click **Delete backup**.
6. Type the word "delete" to confirm deletion and then click **Yes, Delete backup**.

Result

Astra Control Center deletes the backup.

Restore apps

Astra Control Center can restore your application from a snapshot or backup. Persistent storage backups and snapshots are transferred from your object store, so restoring from an existing snapshot to the same cluster will be faster than other methods. You can use the Astra UI or the Astra API to restore 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.

> ⓘ 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.

> ⓘ 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. Click **Apps** and then click the name of an app.
2. Click **Data protection**.
3. If you want to restore from a snapshot, keep the **Snapshots** icon selected. Otherwise, click the **Backups** icon to restore from a backup.
4. Click the drop-down list in the **Actions** column for the snapshot or backup from which you want to restore.
5. Click **Restore application**.
6. **Restore details**: Specify details for the restore:
   - Enter a name and namespace for the app.

     > If you are restoring an app that has been deleted, choose a different name and namespace for the app than the original name. If the name for the restored app is the same as the deleted app, the restore operation will fail.

   - Choose the destination cluster for the app.
   - Click **Review**.
7. **Restore Summary**: Review details about the restore action and click **Restore**.

Result

Astra Control Center restores the app based on the information that you provided.

> 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.

Clone and migrate apps

Clone an existing app to create a duplicate app on the same Kubernetes cluster or on another cluster. Cloning can help if you need to move applications and storage from one Kubernetes cluster to another. For example, you might want to move workloads through a CI/CD pipeline and across Kubernetes namespaces. You can use the Astra UI or the Astra API to clone and migrate apps.

> If you clone an app between clusters, the source and destination clusters must be the same distribution of OpenShift. For example, if you clone an app from an OpenShift 4.7 cluster, use a destination cluster that is also OpenShift 4.7.

When Astra Control Center clones an app, it creates a clone of your application configuration and persistent storage.

> 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.

ⓘ 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
```

What you'll need

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

Steps

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

Result

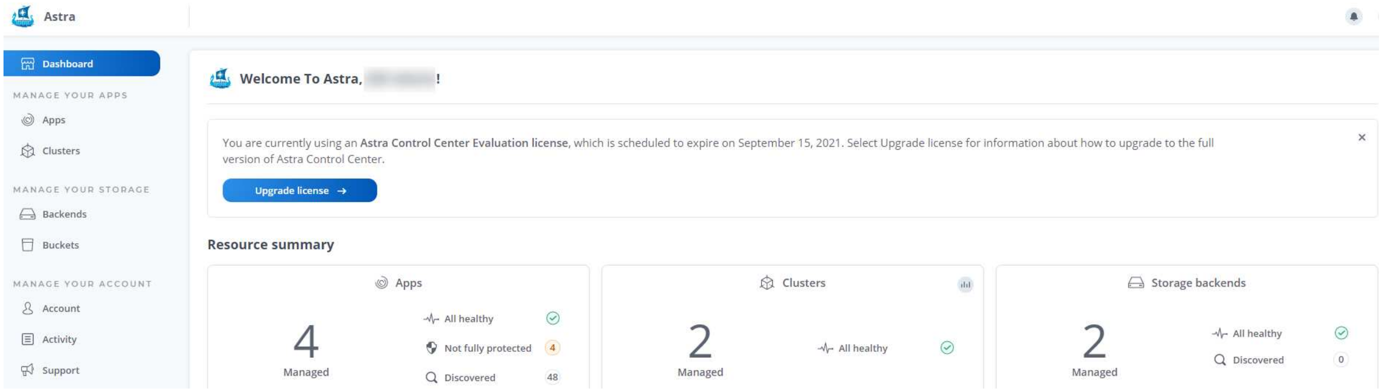
Astra Control Center clones that app based on the information that you provided. The clone operation is successful when the new app clone is in the `Available` state on the **Apps** page.

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

View app and cluster health

View a summary of app and cluster health

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



Welcome To Astra, [redacted]!

You are currently using an **Astra Control Center Evaluation license**, which is scheduled to expire on September 15, 2021. Select Upgrade license for information about how to upgrade to the full version of Astra Control Center.

Resource summary

These aren’t just static numbers or statuses—you can drill down from each of these. 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.

Apps tile

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

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

  Ideally, this number would be zero because you would either manage or ignore apps after they’re discovered. And then you would monitor the number of discovered apps on the Dashboard to identify when developers add new apps to a cluster.

Clusters tile

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

Storage backends tile

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

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

View the health and details of clusters

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

Steps

1. In the Astra Control Center UI, select **Clusters**.
2. On the **Clusters** page, select the cluster whose details you want to view.
3. View the information on the **Overview**, **Storage**, and **Activity** tabs to find the information that you’re looking for.
   - **Overview**: Details about the worker nodes, including their state.
   - **Storage**: The persistent volumes associated with the compute, including the storage class and state.
   - **Activity**: Shows the activities related to the cluster.

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

View the health and details of an app

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

Steps

1. In the Astra Control Center UI, select **Apps** and then select the name of an app.
2. Click around to find the information that you’re looking for:

   App Status

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

   App Protection Status

   Provides a status of how well the app is protected:

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

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

   Overview

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

   Data protection

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

Storage

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

Resources

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

Activity

Shows the activities related to the app.

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

Manage your account

Manage users

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

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, click **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, but cannot unmanage apps or clusters, or delete snapshots or 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. Click **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. Click the User icon at the top right of the screen.
2. Select **Profile**.
3. Click the **Actions** drop-down list, and select **Change Password**.
4. Enter a password that conforms to the password requirements.
5. Enter the password again to confirm.
6. Click **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, click **Account**.
2. In the **Users** tab, select the drop-down list in the **State** column for the user.
3. Select **Reset Password**.
4. Enter a temporary password that conforms to the password requirements.
5. Enter the password again to confirm.

   ⓘ Next time the user logs in, the user will be prompted to change the password.

6. Click **Reset password**.

Change a user’s role

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

Steps

1. In the **Manage Your Account** navigation area, click **Account**.
2. In the **Users** tab, select the drop-down list in the **Role** column for the user.
3. Select a new role and then click **Change Role** when prompted.

Result

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

Remove users

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

Steps

1. In the **Manage Your Account** navigation area, click **Account**.

2. In the **Users** tab, select the checkbox in the row of each user that you want to remove.
3. Click **Actions** and select **Remove user/s**.
4. When you’re prompted, confirm deletion by typing the word "remove" and then click **Yes, Remove User**.

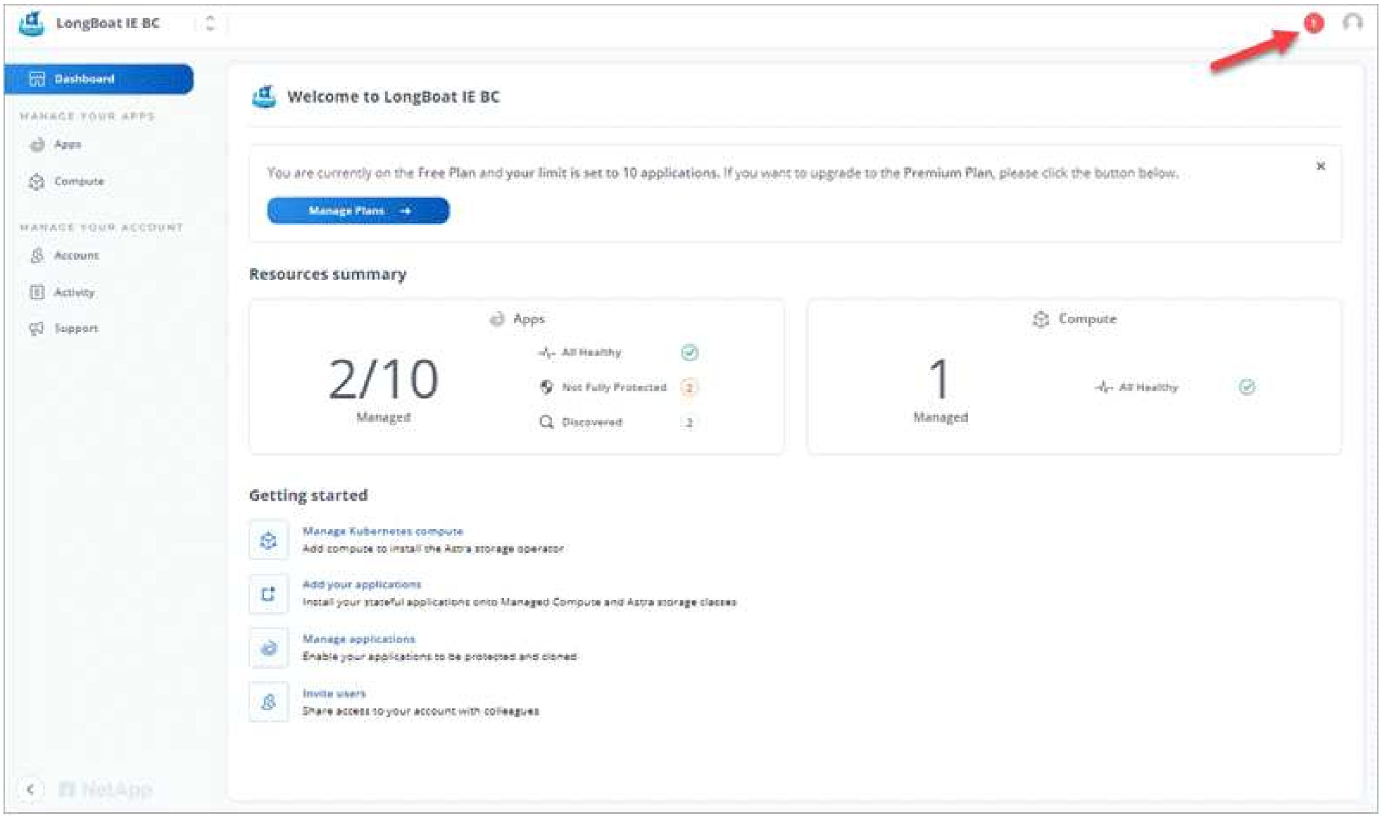
Result

Astra Control Center removes the user from the account.

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.

The number of unread notifications is available in the top right of the interface:



You are currently on the Free Plan and your limit is set to 10 applications. If you want to upgrade to the Premium Plan, please click the button below.

Add compute to install the Astra storage operator

Install your stateful applications onto Managed Compute and Astra storage classes

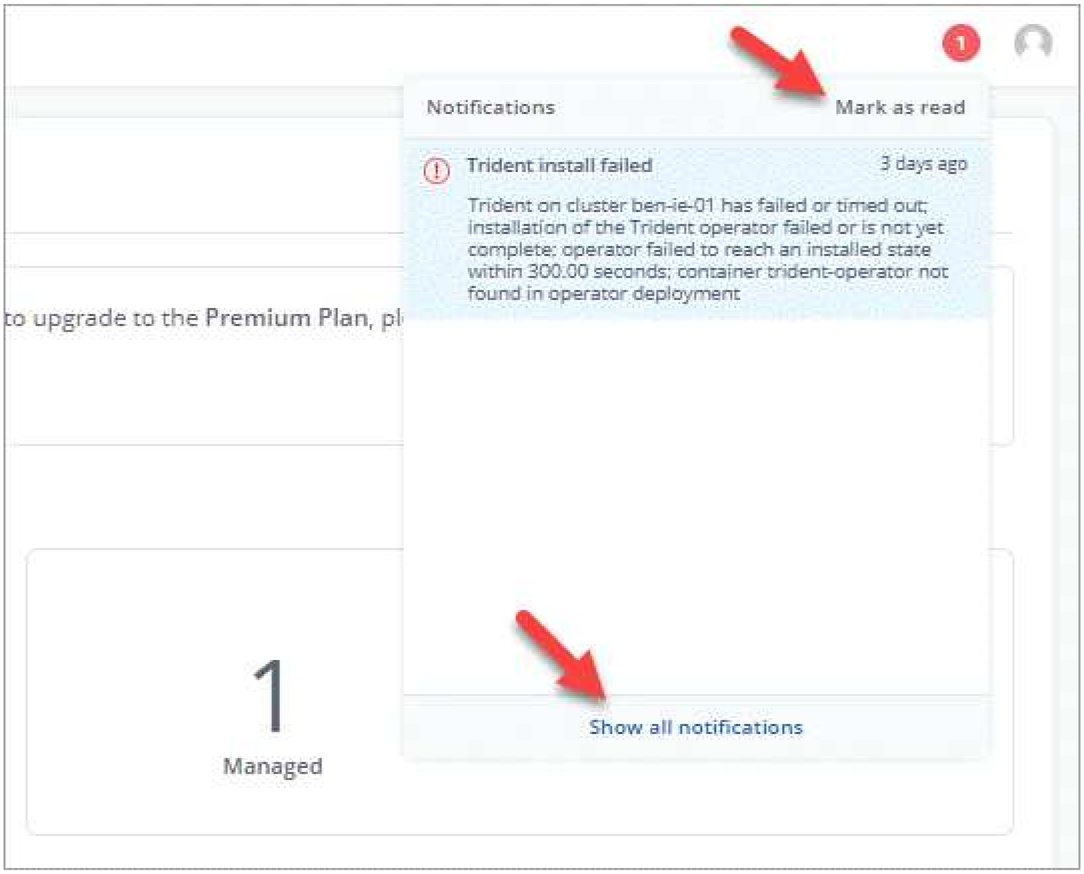
Enable your applications to be protected and cloned

Share access to your account with colleagues

You can view these notifications and mark them as read (this can come in handy if you like to clear unread notifications like we do).

Steps

1. Click the number of unread notifications in the top right.



Trident install failed

Trident on cluster ben-ie-01 has failed or timed out; installation of the Trident operator failed or is not yet complete: operator failed to reach an installed state within 300.00 seconds; container trident-operator not found in operator deployment

2. Review the notifications and then click **Mark as read** or **Show all notifications**.

   If you clicked **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, click **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.

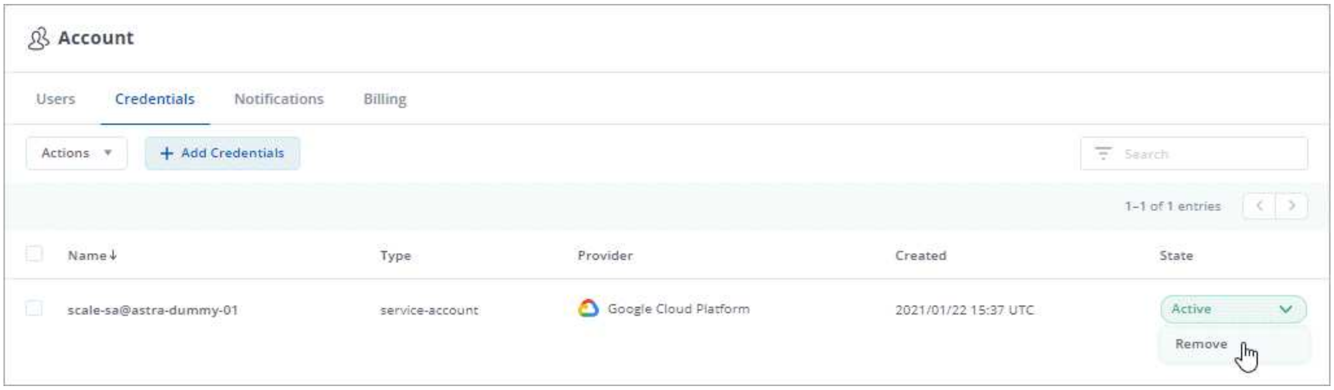
Remove credentials

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. Click **Account > Credentials**.
2. Click the drop-down list in the **State** column for the credentials that you want to remove.
3. Click **Remove**.



Name↓	Type	Provider	Created	State
scale-sa@astra-dummy-01	service-account	Google Cloud Platform	2021/01/22 15:37 UTC	Active

4. Type the word "remove" to confirm deletion and then click **Yes, Remove Credential**.

Result

Astra Control Center removes the credentials from the account.

Update an existing license

You can convert an evaluation license to a full license, or you can update an existing evaluation or full license with a new license. If you don’t have a full license, work with your NetApp sales contact to obtain a full license and serial number. You can use the Astra UI or the Astra 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, click 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.

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 any of the following bucket providers:

- NetApp ONTAP S3
- NetApp StorageGRID S3
- Generic S3

> 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.

You cannot delete a bucket; however, you can edit it.

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 API, see the Astra Automation and API information.

You can do these tasks related to managing buckets:

- Add a bucket
- Edit 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.

Remove credentials

Remove S3 credentials from an account at any time using the Astra Control API.

For details, see Use the Astra Control API.

> 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. It’s best not to remove these credentials.

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 type with credentials that are correct for that provider. For example, the UI accepts NetApp ONTAP S3 as the type with StorageGRID credentials; however, this will cause all future app backups and restores using this bucket to fail. See the Release Notes.

Steps

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

   You can't modify the bucket type.

4. Select **Update**.

Find more information

- Use the Astra 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 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

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

View storage backend details from the Backends option

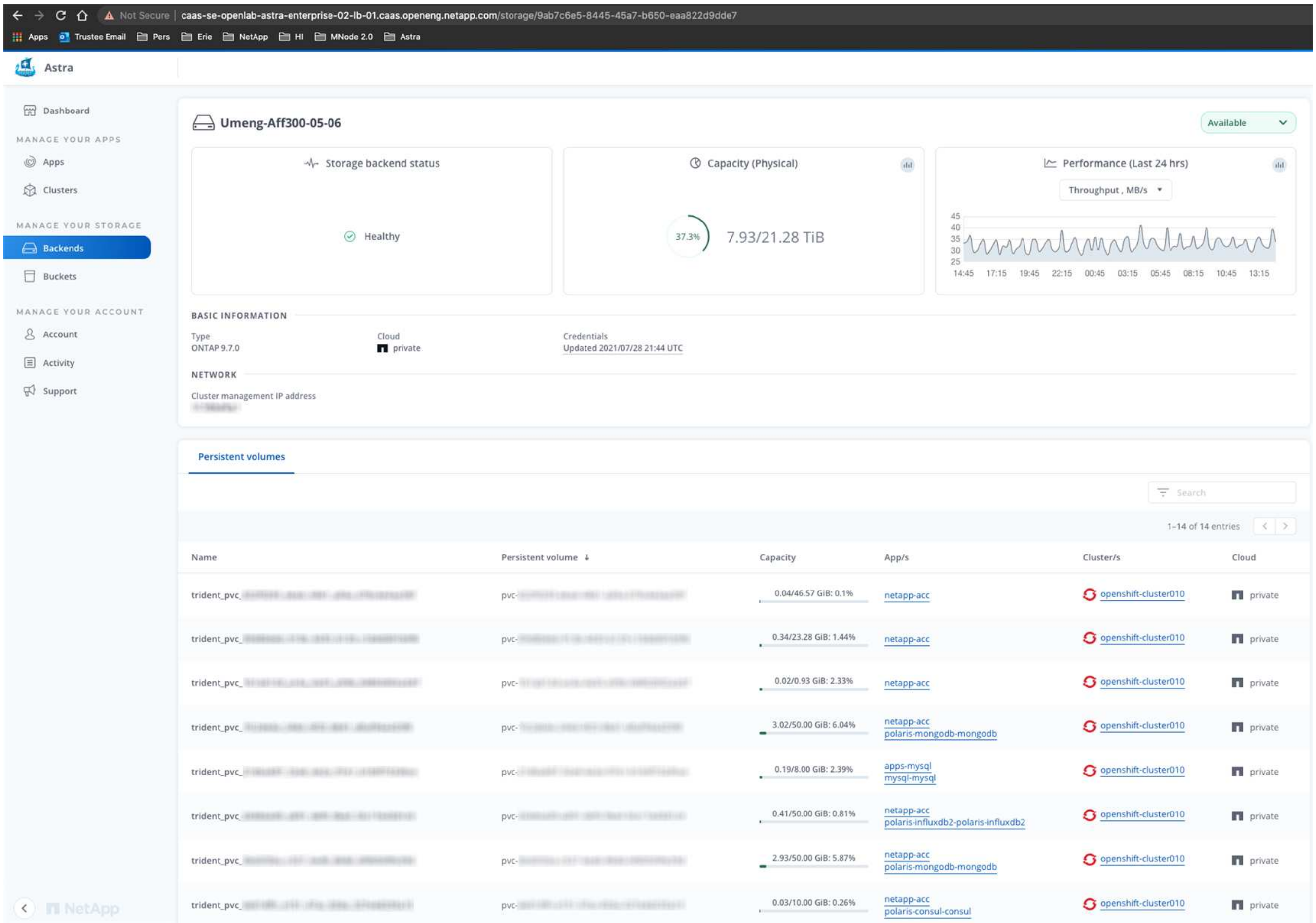
View information about the backend health, capacity, and performance (IOPS throughput and/or latency).

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

Steps

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

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



Umeng-Aff300-05-06

Storage backend status: Healthy

Capacity (Physical): 37.3% 7.93/21.28 TiB

BASIC INFORMATION

Type: ONTAP 9.7.0

Cloud: private

Credentials: Updated 2021/07/28 21:44 UTC

NETWORK

Cluster management IP address: [redacted]

1–14 of 14 entries

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

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

Unmanage a storage backend

You can unmanage the backend.

Steps

1. From left navigation, select **Backends**.
2. Select the backend storage.
3. From the Actions menu, select **Unmanage**.

4. Type "unmanage" to confirm the removal.
5. Select **Yes, remove storage backend**.

Find more information

- Use the Astra API

Monitor and protect infrastructure

You can configure several optional settings to enhance your Astra Control Center experience. If the network where you’re running Astra Control Center requires a proxy for connecting to the Internet (to upload support bundles to NetApp Support Site or establish a connection to Cloud Insights), you should configure a proxy server in Astra Control Center. To monitor and gain insight into your complete infrastructure, create a connection to NetApp Cloud Insights. To collect Kubernetes events from systems monitored by Astra Control Center, add a Fluentd connection.

> 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 here.

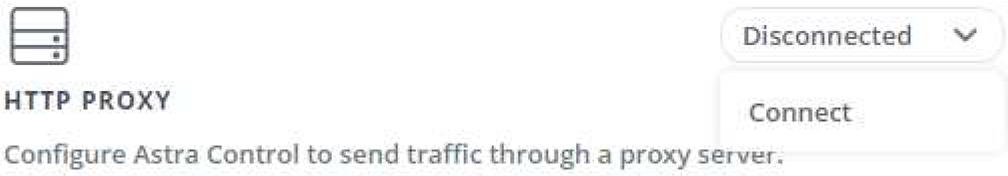
Add a proxy server

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

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

Steps

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

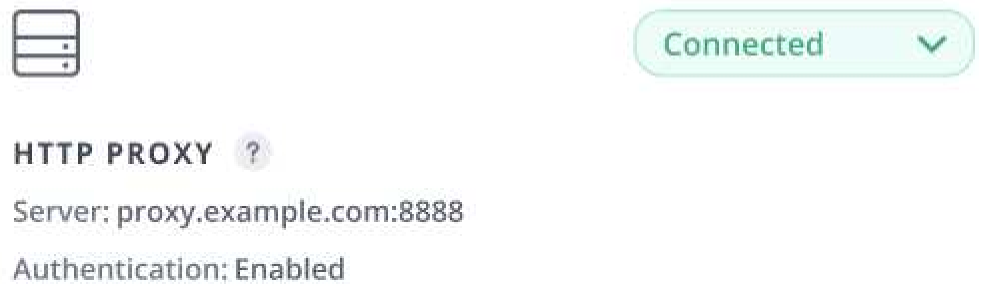


Configure Astra Control to send traffic through a proxy server.

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

Result

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



HTTP PROXY

Server: proxy.example.com:8888

Authentication: Enabled

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.

What you'll need

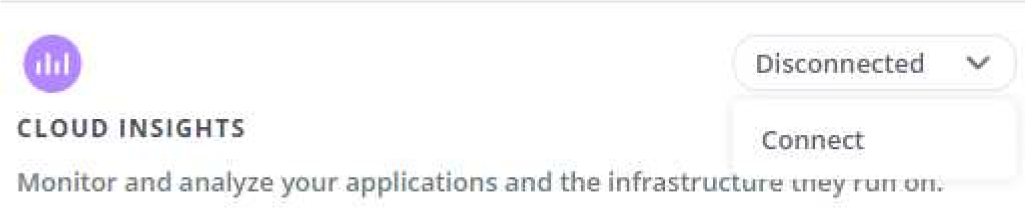
- An Astra Control Center account with **admin**/**owner** privileges.
- A valid Astra Control Center license.

- 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 here.

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.



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

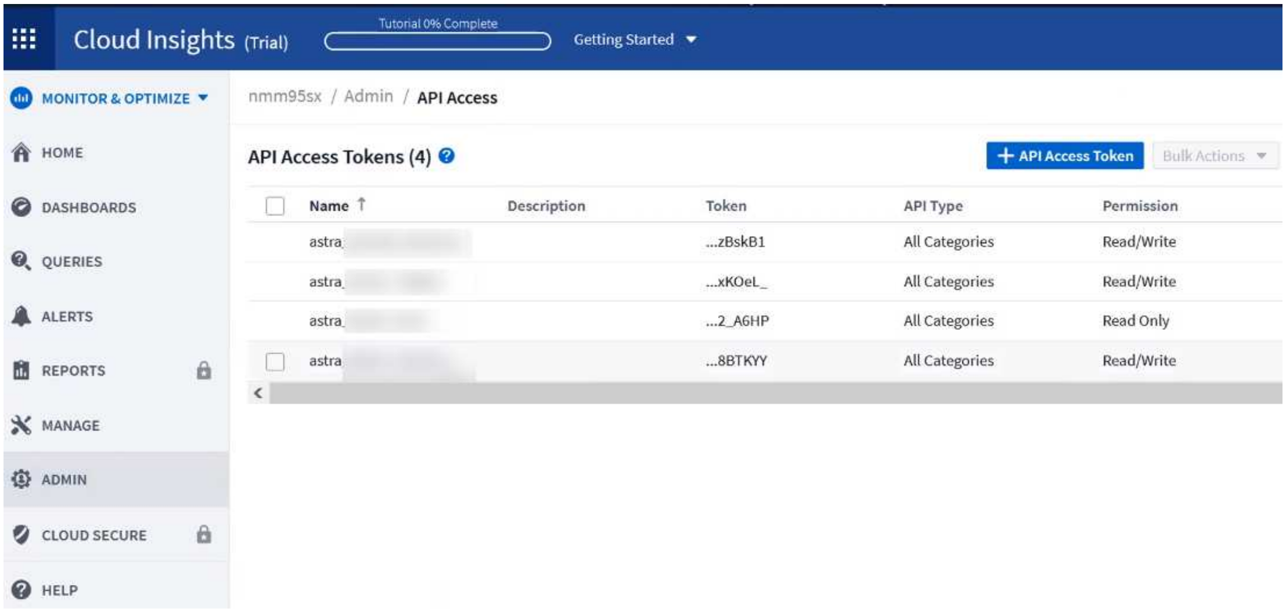
4. Enter the Cloud Insights API tokens and the tenant URL. The tenant URL has the following format, as an example:

```
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 a **Read only** type API token.



API Access Tokens (4)

Name	Description	Token	API Type	Permission
astra_[redacted]		...zBskB1	All Categories	Read/Write
astra_[redacted]		...xKOeL_	All Categories	Read/Write
astra_[redacted]		...2_A6HP	All Categories	Read Only
astra[redacted]		...8BTKYY	All Categories	Read/Write

   c. Copy the **Read only** key. You will need to paste it into the Astra Control Center window for enabling the Cloud Insights connection.

d. In Cloud Insights, generate a **Read/Write** type API token.

e. Copy the **Read/Write** key. You will need to paste it into the Astra Control Center **Connect Cloud Insights** window.

> 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.

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

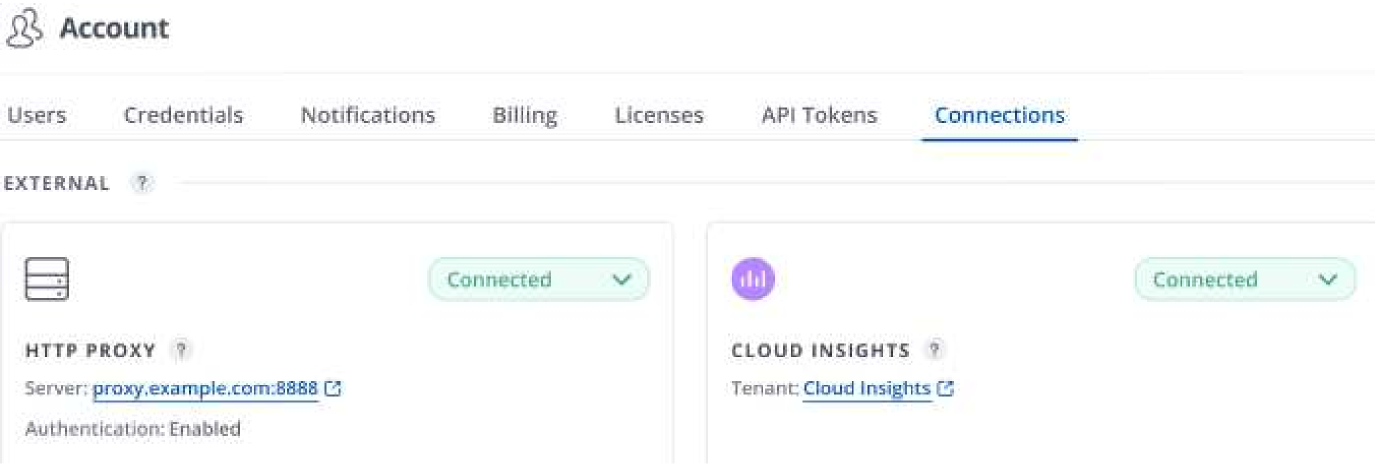
5. Select **Connect**.

> 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.



Connected

HTTP PROXY

Server: proxy.example.com:8888

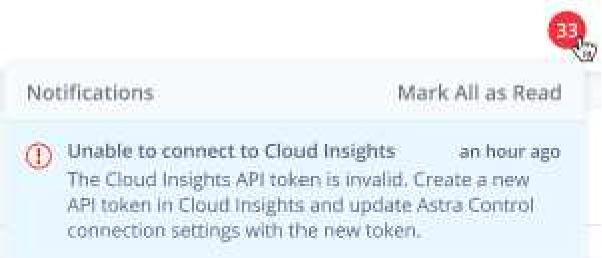
Authentication: Enabled

Connected

CLOUD INSIGHTS

Tenant: Cloud Insights

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.

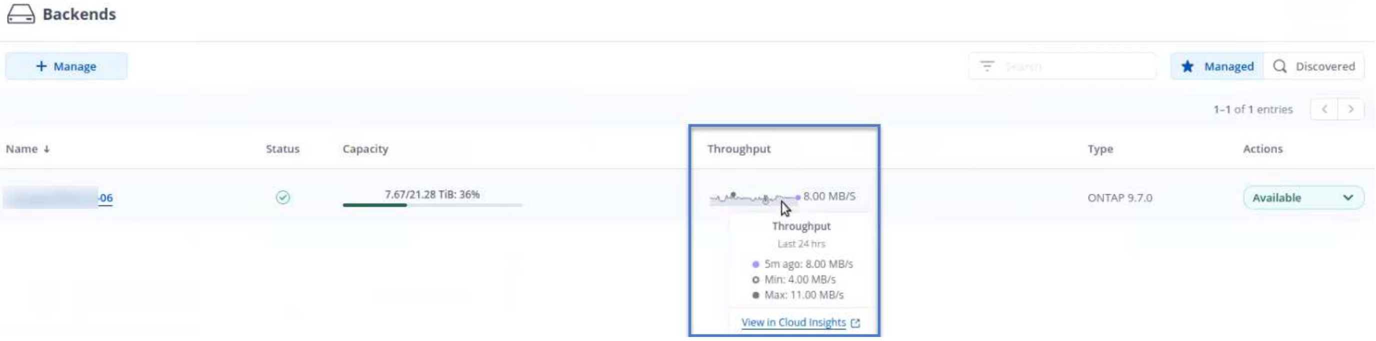


Unable to connect to Cloud Insights — an hour ago

The Cloud Insights API token is invalid. Create a new API token in Cloud Insights and update Astra Control connection settings with the new token.

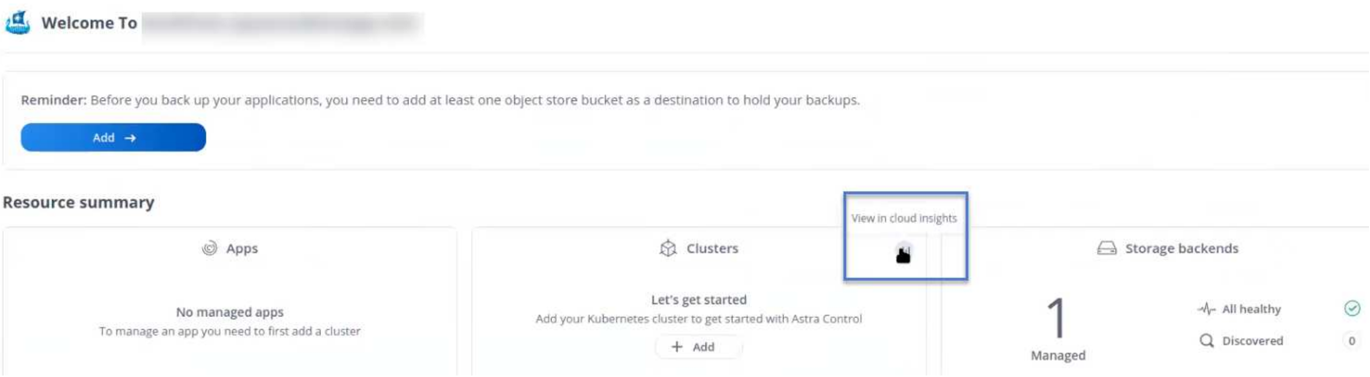
You can also find the same information under **Account** > **Notifications**.

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**.



Welcome To [redacted]

Reminder: Before you back up your applications, you need to add at least one object store bucket as a destination to hold your backups.

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

Edit Cloud Insights connection

You can edit the Cloud Insights connection.

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

Steps

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

Disable Cloud Insights connection

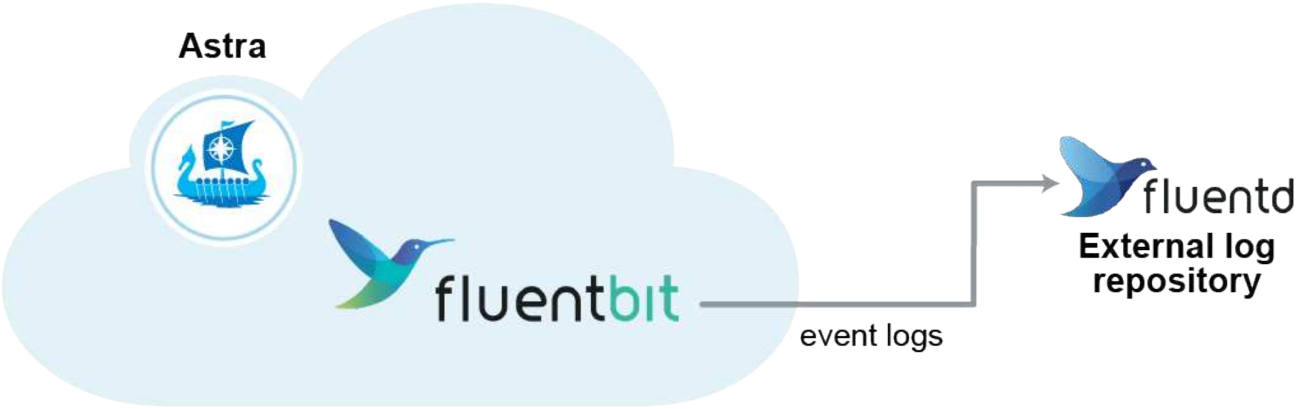
You can disable the Cloud Insights connection for a Kubernetes cluster managed by Astra Control Center. Disabling the Cloud Insights connection does not delete the telemetry data already uploaded to Cloud Insights.

Steps

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

Connect to Fluentd

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



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

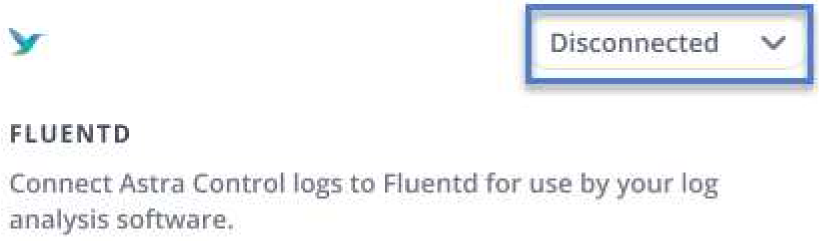
What you'll need

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

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

Steps

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



Disconnected

FLUENTD

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

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

Result

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

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

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

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

Edit the Fluentd connection

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

Steps

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

Disable the Fluentd connection

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

Steps

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

Update an existing license

You can convert an evaluation license to a full license, or you can update an existing evaluation or full license with a new license. If you don’t have a full license, work with your NetApp sales contact to obtain a full license and serial number. You can use the Astra UI or the Astra 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, click 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.

Unmanage apps and clusters

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

Unmanage an app

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

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

Steps

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

Result

Astra Control Center stops managing the app.

Unmanage a cluster

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

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

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

Steps

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

Result

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

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

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.

What you'll need

- Use Astra Control Center UI to unmanage all clusters.

Steps

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

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

   Result:

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

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

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

   Result:

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

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

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

   Result:

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

Find more information

- Known issues for uninstall

Automate with 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 UI or the Astra Control API.

To learn more, go to the Astra automation docs.

Deploy apps

Deploy Jenkins from a Helm chart

Learn how to deploy Jenkins from the Bitnami Helm chart. After you deploy Jenkins on your cluster, you can register the application with Astra Control.

Jenkins is a validated app for Astra Control.

- Learn the difference between a validated app and a standard app in Astra Control Center.

These instructions apply to both Astra Control Service and Astra Control Center.

> Applications deployed from Google Marketplace have not been validated. Some users report issues with discovery and/or backup with Google Marketplace deployments of Postgres, MariaDB, and MySQL.

Requirements

- A cluster that has been added to Astra Control.

  > For Astra Control Center, you can add the cluster to Astra Control Center first or add the app first.

- Updated versions of Helm (version 3.2+) and Kubectl installed on a local machine with the proper kubeconfig for the cluster

Astra Control does not currently support the Kubernetes plugin for Jenkins. You can run Jenkins in a Kubernetes cluster without the plugin. The plugin provides scalability to your Jenkins cluster.

Install Jenkins

Two important notes on this process:

- You must deploy your app after the cluster is added to Astra Control Service, not before. Astra Control Center will accept applications before or after the cluster is added to Astra Control Center.
- You must deploy the Helm chart in a namespace other than the default.

Steps

1. Add the Bitnami chart repo:

   ```
   helm repo add bitnami https://charts.bitnami.com/bitnami
   ```

2. Create the `jenkins` namespace and deploy Jenkins into it with the command:

   ```
   Helm install <name> --namespace <namespace> --create-namespace --set
   persistence.storageClass=<storage_class>
   ```

> ⓘ If the volume size is changed, use Kibibyte (Ki), Mebibyte (Mi) or Gibibyte (Gi) units.

You need to define the storage class only in these situations:

  - You are using Astra Control Service and you don’t want to use the default storage class.
  - You are using Astra Control Center and haven’t yet imported the cluster into Astra Control Center. Or, you have imported the cluster, but don’t want to use the default storage class.

Result

This does the following:

- Creates a namespace.
- Sets the correct storage class.

After the pods are online, you can manage the app with Astra Control. Astra Control enables you to manage an app at the namespace level or by using a helm label.

Deploy MariaDB from a Helm chart

Learn how to deploy MariaDB from the Bitnami Helm chart. After you deploy MariaDB on your cluster, you can manage the application with Astra Control.

MariaDB is a validated app for Astra.

- Learn the difference between a validated app and a standard app in Astra Control Center.

These instructions apply to both Astra Control Service and Astra Control Center.

> ⓘ Applications deployed from Google Marketplace have not been validated. Some users report issues with discovery and/or backup with Google Marketplace deployments of Postgres, MariaDB, and MySQL.

Requirements

- A cluster that has been added to Astra Control.

  > ⓘ For Astra Control Center, you can add the cluster to Astra Control Center first or add the app first.

- Updated versions of Helm (version 3.2+) and Kubectl installed on a local machine with the proper kubeconfig for the cluster

Install MariaDB

Two important notes on this process:

- You must deploy your app after the cluster is added to Astra Control Service, not before. Astra Control Center will accept applications before or after the cluster is added to Astra Control Center.
- You must deploy the Helm chart in a namespace other than the default.

Steps

1. Add the Bitnami chart repo:

```
helm repo add bitnami https://charts.bitnami.com/bitnami
```

2. Deploy MariaDB with the command:

```
Helm install <name> --namespace <namespace> --create-namespace --set
persistence.storageClass=<storage_class>
```

> If the volume size is changed, use Kibibyte (Ki), Mebibyte (Mi) or Gibibyte (Gi) units.

You need to define the storage class only in these situations:

- You are using Astra Control Service and you don’t want to use the default storage class.
- You are using Astra Control Center and haven’t yet imported the cluster into Astra Control Center. Or, you have imported the cluster, but don’t want to use the default storage class.

Result

This does the following:

- Creates a namespace.
- Deploys MariaDB on the namespace.
- Creates a database.

> This method of setting the password at deployment is insecure. We do not recommend this for a production environment.

After the pods are online, you can manage the app with Astra Control. Astra Control enables you to manage an app at the namespace level or by using a helm label.

Deploy MySQL from a Helm chart

Learn how to deploy MySQL from the Bitnami Helm chart. After you deploy MySQL on your Kubernetes cluster, you can manage the application with Astra Control.

MySQL is a validated app for Astra Control.

- Learn the difference between a validated app and a standard app in Astra Control Center.

These instructions apply to both Astra Control Service and Astra Control Center.

> Applications deployed from Google Marketplace have not been validated. Some users report issues with discovery and/or backup with Google Marketplace deployments of Postgres, MariaDB, and MySQL.

Requirements

- A cluster that has been added to Astra Control.

  > For Astra Control Center, you can add the cluster to Astra Control Center first or add the app first.

- Updated versions of Helm (version 3.2+) and Kubectl installed on a local machine with the proper kubeconfig for the cluster

Install MySQL

Two important notes on this process:

- You must deploy your app after the cluster is added to Astra Control Service, not before. Astra Control Center will accept applications before or after the cluster is added to Astra Control Center.
- We recommend that you deploy the Helm chart in a namespace other than the default.

Steps

1. Add the Bitnami chart repo:

   ```
   helm repo add bitnami https://charts.bitnami.com/bitnami
   ```

2. Deploy MySQL with the command:

   ```
   Helm install <name> --namespace <namespace> --create-namespace --set
   persistence.storageClass=<storage_class>
   ```

   > If the volume size is changed, use Kibibyte (Ki), Mebibyte (Mi) or Gibibyte (Gi) units.

   You need to define the storage class only in these situations:

   - You are using Astra Control Service and you don’t want to use the default storage class.
   - You are using Astra Control Center and haven’t yet imported the cluster into Astra Control Center. Or, you have imported the cluster, but don’t want to use the default storage class.

Result

This does the following:

- Creates a namespace.
- Deploys MySQL on the namespace.

After the pods are online, you can manage the app with Astra Control. Astra Control allows you to manage an app with its name, at the namespace level, or by using a helm label.

Deploy Postgres from a Helm chart

Learn how to deploy Postgres from the Bitnami Helm chart. After you deploy Postgres on your cluster, you can register the application with Astra Control.

Postgres is a validated app for Astra.

- Learn the difference between a validated app and a standard app in Astra Control Center.

These instructions apply to both Astra Control Service and Astra Control Center.

> Applications deployed from Google Marketplace have not been validated. Some users report issues with discovery and/or backup with Google Marketplace deployments of Postgres, MariaDB, and MySQL.

Requirements

- A cluster that has been added to Astra Control.

  > For Astra Control Center, you can add the cluster to Astra Control Center first or add the app first.

- Updated versions of Helm (version 3.2+) and Kubectl installed on a local machine with the proper kubeconfig for the cluster

Install Postgres

Two important notes on this process:

- You must deploy your app after the cluster is added to Astra Control Service, not before. Astra Control Center will accept applications before or after the cluster is added to Astra Control Center.
- You must deploy the Helm chart in a namespace other than the default.

Steps

1. Add the Bitnami chart repo:

   ```
   helm repo add bitnami https://charts.bitnami.com/bitnami
   ```

2. Deploy Postgres with the command:

   ```
   Helm install <name> --namespace <namespace> --create-namespace --set
   persistence.storageClass=<storage_class>
   ```

   > If the volume size is changed, use Kibibyte (Ki), Mebibyte (Mi) or Gibibyte (Gi) units.

   You need to define the storage class only in these situations:

   - You are using Astra Control Service and you don’t want to use the default storage class.

    - You are using Astra Control Center and haven’t yet imported the cluster into Astra Control Center. Or, you have imported the cluster, but don’t want to use the default storage class.

Result

This does the following:

- Creates a namespace.
- Deploys Postgres on the namespace.

After the pods are online, you can manage the app with Astra Control. Astra Control enables you to manage an app at the namespace level or by using a helm label.

Knowledge and support

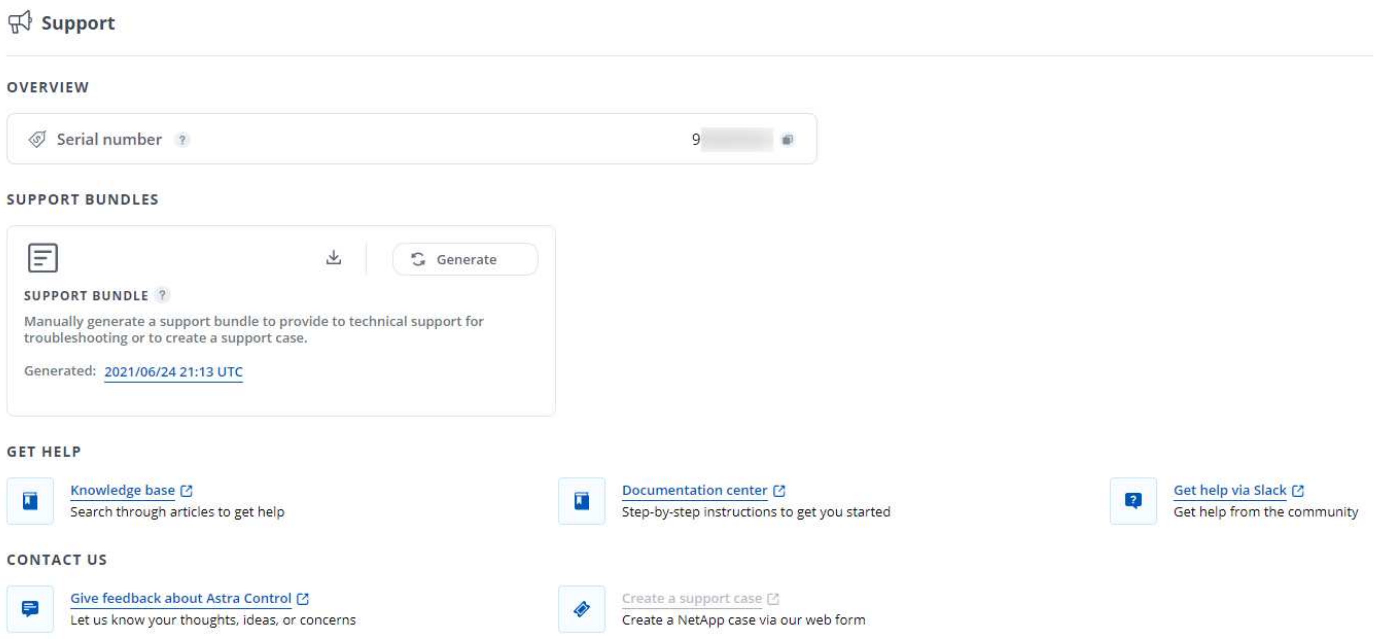
Get help

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 Slack 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 Slack 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.

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



Support

OVERVIEW

Serial number ? 9[redacted]

SUPPORT BUNDLES

Generate

SUPPORT BUNDLE ?

Manually generate a support bundle to provide to technical support for troubleshooting or to create a support case.

Generated: 2021/06/24 21:13 UTC

GET HELP

Knowledge base
Search through articles to get help

Documentation center
Step-by-step instructions to get you started

Get help via Slack
Get help from the community

CONTACT US

Give feedback about Astra Control
Let us know your thoughts, ideas, or concerns

Create a support case
Create a NetApp case via our web form

Self-support options

These options are available for free 24x7:

- Knowledge base (login required)

  Search for articles, FAQs, or Break Fix information related to Astra Control.

- Documentation

  This is the doc site that you’re currently viewing.

- Slack

Go to the containers channel in thePub workspace to connect with peers and experts.

- Generate support bundles to provide to NetApp Support for troubleshooting
- Feedback email

  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 Definition (CRD) file (`astra_control_center_min.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**.

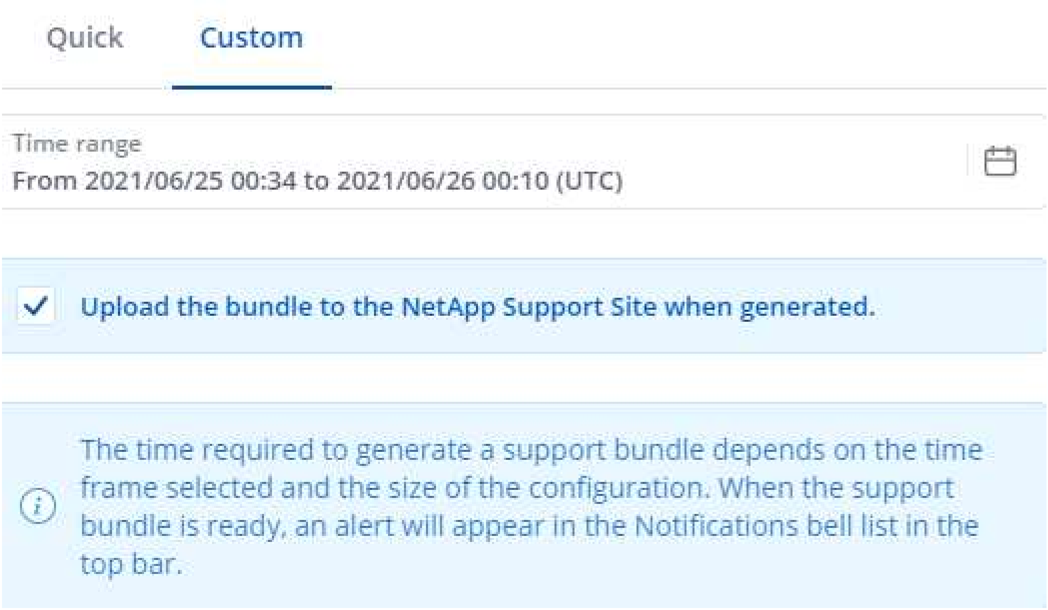
Steps

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. Check the **Upload the bundle to the NetApp Support Site when generated**.



Time range
From 2021/06/25 00:34 to 2021/06/26 00:10 (UTC)

☑ Upload the bundle to the NetApp Support Site when generated.

The time required to generate a support bundle depends on the time frame selected and the size of the configuration. When the support bundle is ready, an alert will appear in the Notifications bell list in the top bar.

e. Select **Generate Bundle**.

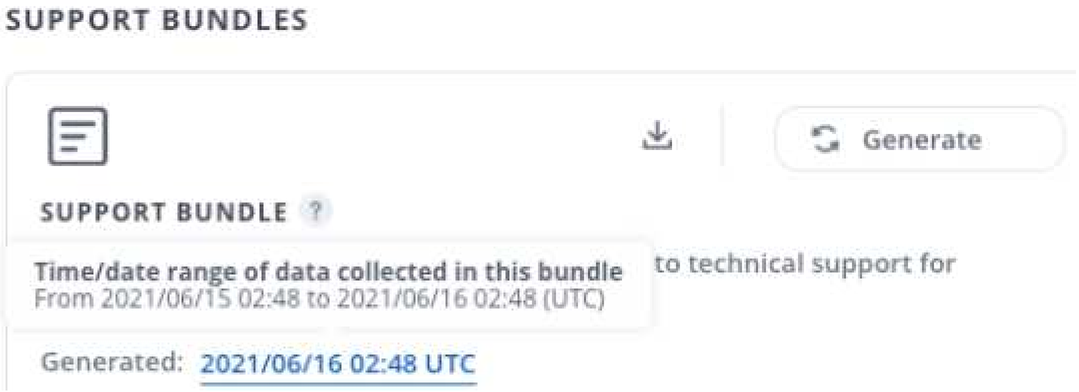
When the support bundle is ready, a notification appears on the **Accounts** > **Notification** page in the Alerts area, on the **Activity** page, and also in the notifications list (accessible by selecting the icon in the top-right side of the UI).

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, it appears next to **Generated** in the **Support Bundles** section of the **Support** page as shown:



Time/date range of data collected in this bundle
From 2021/06/15 02:48 to 2021/06/16 02:48 (UTC)

to technical support for

Generated: 2021/06/16 02:48 UTC

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 AIQ.

Find more information

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

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 21.08 release

Astra Control API license

https://docs.netapp.com/us-en/astra-automation-2108/media/astra-api-license.pdf

Copyright information

Trademark information