



# **Astra Control Service documentation**

## **Astra Control Service**

NetApp  
April 18, 2024

This PDF was generated from <https://docs.netapp.com/us-en/astra-control-service/index.html> on April 18, 2024. Always check [docs.netapp.com](https://docs.netapp.com) for the latest.

# Table of Contents

Astra Control Service documentation	1
Release notes	2
What's new with Astra Control Service	2
Known issues	11
Known limitations	12
Get started	15
Learn about Astra Control	15
Supported Kubernetes deployments	19
Quick start for Astra Control Service	19
Set up your cloud provider	20
Register for an Astra Control Service account	41
Add a cluster to Astra Control Service	42
What's next?	81
Astra Control Service videos	81
Concepts	83
Architecture and components	83
Data protection	84
Storage classes and performance for AWS clusters	85
Storage classes and PV size for AKS clusters	86
Service type, storage classes, and PV size for GKE clusters	87
App management	90
User roles and namespaces	92
Use Astra Control Service	93
Log in to Astra Control Service	93
Manage and protect apps	93
View app and compute health	121
Manage buckets	123
Monitor running tasks	126
Manage your account	126
Manage cloud instances	135
Enable Astra Control Provisioner	136
Unmanage apps and clusters	145
Deploy a self-managed instance of Astra Control	147
Use Astra Control Provisioner	148
Configure storage backend encryption	148
Recover volume data using a snapshot	155
Replicate volumes using SnapMirror	157
Automation using the Astra Control REST API	164
Knowledge and support	165
Register for support	165
Troubleshooting	167
Get help	167
Frequently asked questions	169

Overview .....	169
Access to Astra Control .....	169
Registering Kubernetes clusters .....	169
Registering Elastic Kubernetes Service (EKS) clusters .....	170
Registering Azure Kubernetes Service (AKS) clusters .....	170
Registering Google Kubernetes Engine (GKE) clusters .....	170
Removing clusters .....	170
Managing applications .....	171
Data management operations .....	171
Astra Control Provisioner .....	172
Legal notices .....	174
Copyright .....	174
Trademarks .....	174
Patents .....	174
Privacy policy .....	174
Open source .....	174
Astra Control API license .....	174

# Astra Control Service documentation

# Release notes

## What's new with Astra Control Service

NetApp periodically updates Astra Control Service to bring you new features, enhancements, and bug fixes.

### 14 March 2024

#### New features and support

This is a minor bug fix release.

### 7 November 2023

#### New features and support

- **Backup and restore capabilities for applications with `ontap-nas-economy` driver-backed storage backends:** Enable backup and restore operations for `ontap-nas-economy` with some [simple steps](#).
- **Astra Control Service support for on-premises Red Hat OpenShift Container Platform clusters**

[Add a cluster](#)

- **Immutable Backups:** Astra Control now supports [unalterable, read-only backups](#) as an additional security layer against malware and other threats.
- **Introducing Astra Control Provisioner**

With the 23.10 release, Astra Control introduces a new software component called Astra Control Provisioner that will be available to all licensed Astra Control users. Astra Control Provisioner provides access to a superset of advanced management and storage provisioning features beyond those that Astra Trident provides. These features are available to all Astra Control customers at no additional cost.

- **Get started with Astra Control Provisioner**

You can [enable Astra Control Provisioner](#) if you have installed and configured your environment to use Astra Trident 23.10.

- **Astra Control Provisioner functionality**

The following features are available with the Astra Control Provisioner 23.10 release:

- **Enhanced storage backend security with Kerberos 5 encryption:** You can improve storage security by [enabling encryption](#) for the traffic between your managed cluster and the storage backend. Astra Control Provisioner supports Kerberos 5 encryption over NFSv4.1 connections from Red Hat OpenShift clusters to Azure NetApp Files and on-premises ONTAP volumes.
  - **Recover data using a snapshot:** Astra Control Provisioner provides rapid, in-place volume restoration from a snapshot using the `TridentActionSnapshotRestore` (TASR) CR.
  - **Backup and restore capabilities for applications with `ontap-nas-economy` driver-backed storage backends:** As described [above](#).
- **Astra Control Service support for Red Hat OpenShift Service on AWS (ROSA) clusters**

[Add a cluster](#)

- **Support for managing applications that use NVMe/TCP storage**

Astra Control can now manage applications backed by persistent volumes that are connected using NVMe/TCP.

- **Execution hooks turned off by default:** Beginning with this release, execution hooks functionality can be [enabled](#) or disabled for additional security (it is disabled by default). If you have not yet created execution hooks for use with Astra Control, you need to [enable the execution hooks feature](#) to begin creating hooks. If you created execution hooks prior to this release, the execution hooks functionality stays enabled and you can use hooks as you would normally.

## 2 October 2023

### New features and support

This is a minor bug fix release.

## 27 July 2023

### New features and support

- Clone operations now support live clones only (current state of managed application). To clone from a snapshot or backup, use the restore workflow.

[Restore apps](#)

## 26 June 2023

### New features and support

- Azure Marketplace subscriptions are now billed per hour instead of per minute

[Set up billing](#)

## 30 May 2023

### New features and support

- Support for private Amazon EKS clusters

[Manage private clusters from Astra Control Service](#)

- Support for selecting the destination storage class during restore or clone operations

[Restore apps](#)

## 15 May 2023

### New features and support

This is a minor bug fix release.

## 25 April 2023

### New features and support

- Support for private Red Hat OpenShift clusters

[Manage private clusters from Astra Control Service](#)

- Support for including or excluding application resources during restore operations

[Restore apps](#)

- Support for managing data-only applications

[Start managing apps](#)

## 17 January 2023

### New features and support

- Enhanced execution hooks functionality with additional filtering options

[Manage app execution hooks](#)

- Support for NetApp Cloud Volumes ONTAP as a storage backend

[Learn about Astra Control](#)

## 22 November 2022

### New features and support

- Support for applications that span across multiple namespaces

[Define apps](#)

- Support for including cluster resources in an application definition

[Define apps](#)

- Enhanced progress reporting for your backup, restore, and clone operations

[Monitor running tasks](#)

- Support for managing clusters that already have a compatible version of Astra Trident installed

[Start managing Kubernetes clusters from Astra Control Service](#)

- Support for managing multiple cloud provider subscriptions in a single Astra Control Service account

[Manage cloud instances](#)

- Support for adding self-managed Kubernetes clusters that are hosted in public cloud environments to Astra Control Service

[Start managing Kubernetes clusters from Astra Control Service](#)

- Billing for Astra Control Service is now metered per namespace instead of per application

[Set up billing](#)

- Support for subscribing to Astra Control Service term-based offers through the AWS Marketplace

## [Set up billing](#)

### Known issues and limitations

- [Known issues for this release](#)
- [Known limitations for this release](#)

## 7 September 2022

This release includes stability and resiliency enhancements for the Astra Control Service infrastructure.

## 10 August 2022

This release includes the following new features and enhancements.

- Improved application management workflow  
Improved application management workflows provide increased flexibility when defining applications managed by Astra Control.

### [Manage apps](#)

- Support for Amazon Web Services clusters  
Astra Control Service can now manage apps that are running on clusters hosted in Amazon Elastic Kubernetes Service. You can configure the clusters to use Amazon Elastic Block Store or Amazon FSx for NetApp ONTAP as the storage backend.

### [Set up Amazon Web Services](#)

- Enhanced execution hooks  
In addition to pre- and post-snapshot execution hooks, you can now configure the following types of execution hooks:
  - Pre-backup
  - Post-backup
  - Post-restore

Among other improvements, Astra Control now supports using the same script for multiple execution hooks.

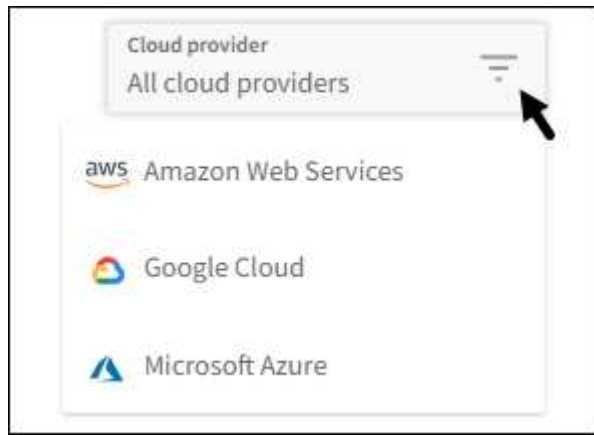


The NetApp-provided default pre- and post-snapshot execution hooks for specific applications have been removed in this release. If you do not provide your own execution hooks for snapshots, Astra Control Service will take crash-consistent snapshots only beginning August 4, 2022. Visit the [NetApp Verda GitHub repository](#) for sample execution hook scripts that you can modify to fit your environment.

### [Manage app execution hooks](#)

- Azure Marketplace support  
You can now sign up to Astra Control Service via Azure Marketplace.
- Cloud provider selection  
While reading the Astra Control Service documentation, you can now select your cloud provider at the top right of the page. You will see documentation relevant only to the cloud provider you select.





## 26 April 2022

This release includes the following new features and enhancements.

- Namespace role-based access control (RBAC)  
Astra Control Service now supports assigning namespace constraints to Member or Viewer users.

[Namespace role-based access control \(RBAC\)](#)

- Azure Active Directory support  
Astra Control Service supports AKS clusters that use Azure Active Directory for authentication and identity management.

[Start managing Kubernetes clusters from Astra Control Service](#)

- Support for private AKS clusters  
You can now manage AKS clusters that use private IP addresses.

[Start managing Kubernetes clusters from Astra Control Service](#)

- Bucket removal from Astra Control  
You can now remove a bucket from Astra Control Service.

[Remove a bucket](#)

## 14 December 2021

This release includes the following new features and enhancements.

- New storage backend options

Astra Control Service now supports Google Persistent Disk and Azure managed disks as storage backend options.

- [Set up Google Cloud](#)
- [Set up Microsoft Azure with Azure managed disks](#)

- In-place app restore  
You can now restore a snapshot, clone, or backup of an app in place by restoring to the same cluster and namespace.

### [Restore apps](#)

- Script events with execution hooks

Astra Control supports custom scripts that you can run before or after you take a snapshot of an application. This enables you to perform tasks like suspending database transactions so that the snapshot of your database app is consistent.

### [Manage app execution hooks](#)

- Operator-deployed apps

Astra Control supports some apps when they are deployed with operators.

### [Start managing apps](#)

- Service principals with resource group scope

Astra Control Service now supports service principals that use a resource group scope.

### [Create an Azure service principal](#)

## 5 August 2021

This release includes the following new features and enhancements.

- Astra Control Center

Astra Control is now available in a new deployment model. *Astra Control Center* is self-managed software that you install and operate in your data center so that you can manage Kubernetes application lifecycle management for on-premises Kubernetes clusters.

To learn more, [go to the Astra Control Center documentation](#).

- Bring your own bucket

You can now manage the buckets that Astra uses for backups and clones by adding additional buckets and by changing the default bucket for the Kubernetes clusters in your cloud provider.

### [Manage buckets](#)

## 2 June 2021

This release includes bug fixes and the following enhancements to Google Cloud support.

- Support for shared VPCs

You can now manage GKE clusters in GCP projects with a shared VPC network configuration.

- Persistent volume size for the CVS service type

Astra Control Service now creates persistent volumes with a minimum size of 300 GiB when using the CVS service type.

[Learn how Astra Control Service uses Cloud Volumes Service for Google Cloud as the storage backend for persistent volumes.](#)

- Support for Container-Optimized OS

Container-Optimized OS is now supported with GKE worker nodes. This is in addition to support for Ubuntu.

[Learn more about GKE cluster requirements.](#)

## 15 April 2021

This release includes the following new features and enhancements.

- Support for AKS clusters  
Astra Control Service can now manage apps that are running on a managed Kubernetes cluster in Azure Kubernetes Service (AKS).

[Learn how to get started.](#)

- REST API  
The Astra Control REST API is now available for use. The API is based on modern technologies and current best practices.

[Learn how to automate application data lifecycle management using the REST API.](#)

- Annual subscription  
Astra Control Service now offers a *Premium Subscription*.

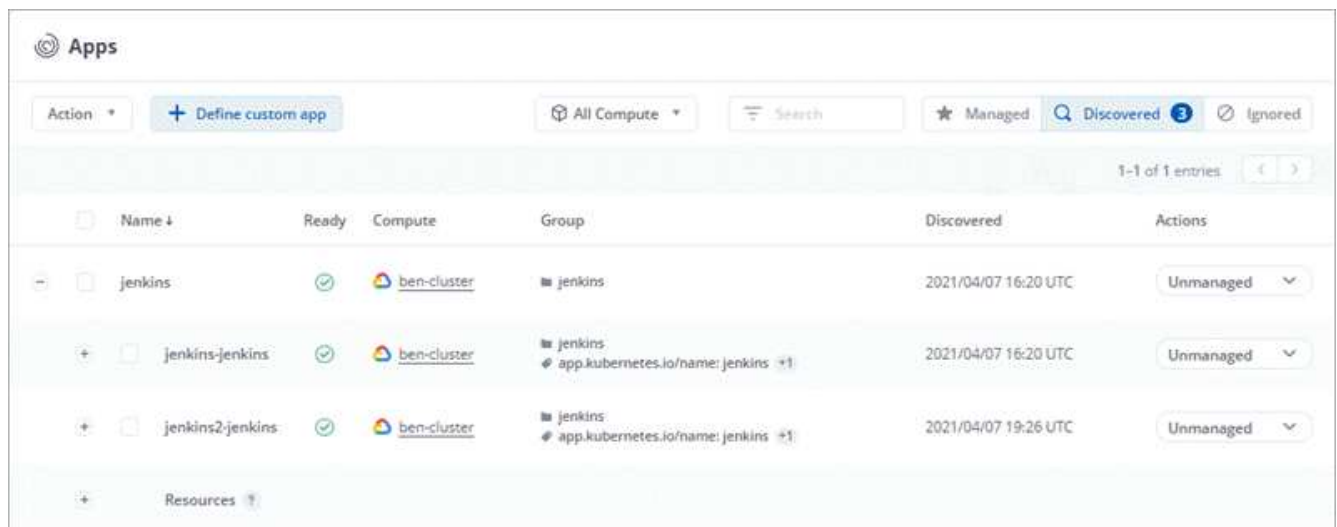
Pre-pay at a discounted rate with an annual subscription that enables you to manage up to 10 apps per *application pack*. Contact NetApp Sales to purchase as many packs as needed for your organization—for example, purchase 3 packs to manage 30 apps from Astra Control Service.

If you manage more apps than allowed by your annual subscription, then you'll be charged at the overage rate of \$0.005 per minute, per application (the same as Premium PayGo).

[Learn more about Astra Control Service pricing.](#)

- Namespace and app visualization  
We enhanced the Discovered Apps page to better show the hierarchy between namespaces and apps. Just expand a namespace to see the apps contained in that namespace.

[Learn more about managing apps.](#)



The screenshot shows the 'Apps' management interface. At the top, there's a header with 'Apps' and a search bar. Below the header, there's a table with columns: Name, Ready, Compute, Group, Discovered, and Actions. The table lists three entries under the 'jenkins' namespace: 'jenkins', 'jenkins-jenkins', and 'jenkins2-jenkins'. Each entry shows its status (Ready), the compute engine (ben-cluster), the group (jenkins), the discovery time (2021/04/07), and an 'Unmanaged' action button. The 'jenkins-jenkins' and 'jenkins2-jenkins' entries also show a link to the parent namespace.

Name	Ready	Compute	Group	Discovered	Actions
jenkins	✓	ben-cluster	jenkins	2021/04/07 16:20 UTC	Unmanaged
jenkins-jenkins	✓	ben-cluster	jenkins app.kubernetes.io/name: jenkins	2021/04/07 16:20 UTC	Unmanaged
jenkins2-jenkins	✓	ben-cluster	jenkins app.kubernetes.io/name: jenkins	2021/04/07 19:26 UTC	Unmanaged

- User interface enhancements  
Data protection wizards were enhanced for ease of use. For example, we refined the Protection Policy

wizard to more easily view the protection schedule as you define it.

**Configure Protection Policy** STEP 1/2: DETAILS

**PROTECTION SCHEDULE**

- Hourly**: Every hour on the 0th minute, keep the last 4 snapshots.
- Daily**: Daily at 02:00 (UTC), keep the last 15 snapshots.
- Weekly**: Weekly on Mondays at 02:00 (UTC), keep the last 26 snapshots.
- Monthly**: Every 1st of the month at 02:00 (UTC), keep the last 12 backups.

● Hourly ● Daily ● Weekly ● **Monthly**

Day(s) of Month: 1 x Time (UTC): 02:00 Snapshots to keep: 0 Backups to keep: 12

**OVERVIEW**

**Schedule and Retention**

Define a policy to continuously protect your application on a schedule and configure a retention count to get started.

For select stateful applications expect IO to pause for a short period of time during a backup or snapshot operation.

Read more in [Protection Policies](#).

Application: jenkins-jenkins  
Namespace: jenkins  
Labels: app.kubernetes.io/name: jenkins, app.kubernetes.io/instance: jenkins  
Compute: ben-cluster

Cancel Review Information →

- Activity enhancements

We've made it easier to view details about the activities in your Astra Control account.

- Filter the activity list by managed app, severity level, user, and time range.
- Download your Astra Control account activity to a CSV file.
- View activities directly from the Clusters page or the Apps page after selecting a cluster or an app.

[Learn more about viewing your account activity.](#)

## 1 March 2021

Astra Control Service now supports the [CVS service type](#) with Cloud Volumes Service for Google Cloud. This is in addition to already supporting the *CVS-Performance* service type. Just as a reminder, Astra Control Service uses Cloud Volumes Service for Google Cloud as the storage backend for your persistent volumes.

This enhancement means that Astra Control Service can now manage app data for Kubernetes clusters that are running in [any Google Cloud region where Cloud Volumes Service is supported](#).

If you have the flexibility to choose between Google Cloud regions, then you can pick either CVS or CVS-Performance, depending on your performance requirements. [Learn more about choosing a service type.](#)

## 25 January 2021

We're pleased to announce that Astra Control Service is now Generally Available. We incorporated a lot of the feedback that we received from the Beta release and made a few other notable enhancements.

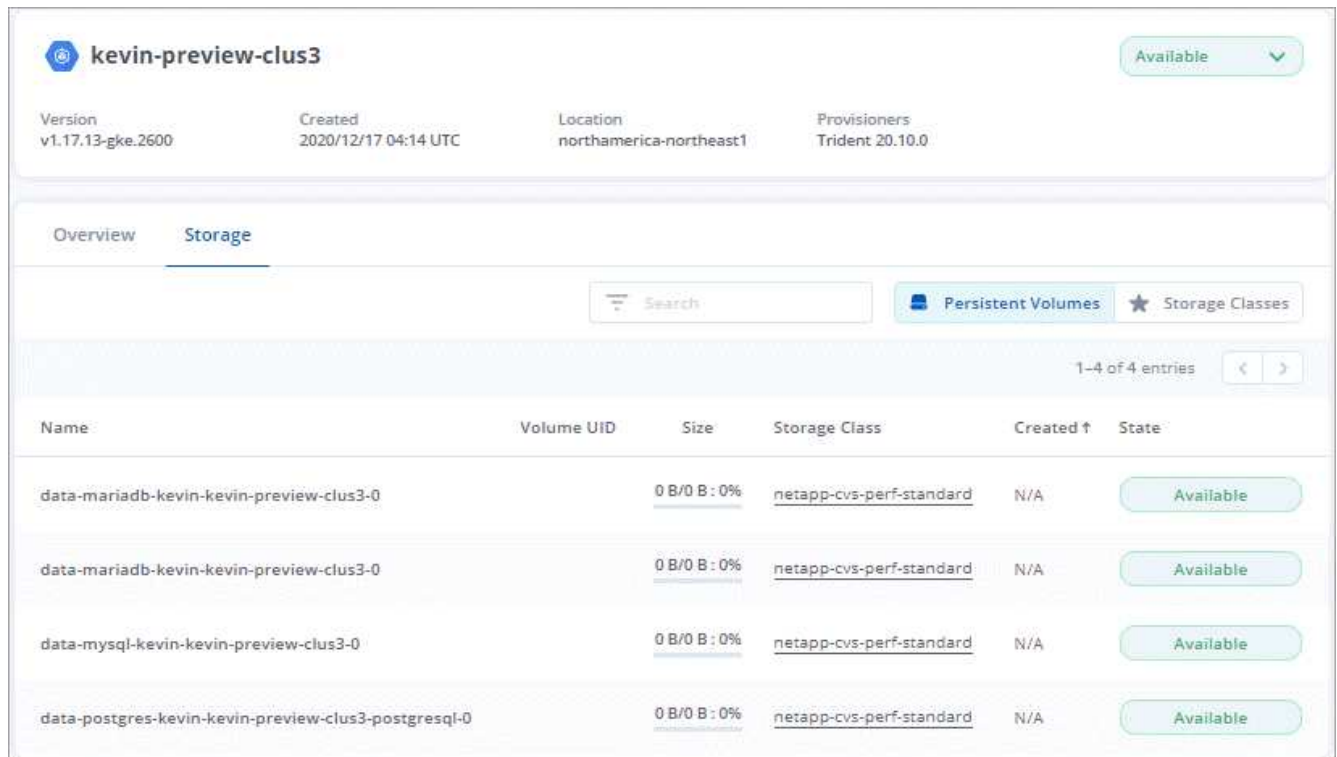
- Billing is now available, which enables you to move from the Free Plan to the Premium Plan. [Learn more about billing.](#)
- Astra Control Service now creates Persistent Volumes with a minimum size of 100 GiB when using the CVS-Performance service type.
- Astra Control Service can now discover apps faster.
- You can now create and delete accounts on your own.
- We've improved notifications when Astra Control Service can no longer access a Kubernetes cluster.

These notifications are important because Astra Control Service can't manage apps for disconnected clusters.

## 17 December 2020 (Beta update)

We primarily focused on bug fixes to improve your experience, but we made a few other notable enhancements:

- When you add your first Kubernetes compute to Astra Control Service, the object store is now created in the geography where the cluster resides.
- Details about persistent volumes is now available when you view storage details at the compute level.



kevin-preview-clus3 <span>Available</span>					
Version	Created	Location	Provisioners		
v1.17.13-gke.2600	2020/12/17 04:14 UTC	northamerica-northeast1	Trident 20.10.0		
Overview <b>Storage</b>					
<div>Search</div> <div>Persistent Volumes <span>Storage Classes</span></div>					
1-4 of 4 entries					
Name	Volume UID	Size	Storage Class	Created ↑	State
data-mariadb-kevin-kevin-preview-clus3-0		0 B/0 B : 0%	netapp-cvs-perf-standard	N/A	Available
data-mariadb-kevin-kevin-preview-clus3-0		0 B/0 B : 0%	netapp-cvs-perf-standard	N/A	Available
data-mysql-kevin-kevin-preview-clus3-0		0 B/0 B : 0%	netapp-cvs-perf-standard	N/A	Available
data-postgres-kevin-kevin-preview-clus3-postgresql-0		0 B/0 B : 0%	netapp-cvs-perf-standard	N/A	Available

- We added an option to restore an application from an existing snapshot or backup.

Overview

Data protection

Storage

Resources

Actions

Configure Protection Policy

Search

Snapshots

Backups

26-29 of 29 entries

<input type="checkbox"/>	Name	Ready	On-Schedule/On-Demand	Created ↑	Actions
<input type="checkbox"/>	ns-postgres-kevin-kevin-preview-clus3-snapshot-20201217103001		<a href="#">On-Schedule</a>	2020/12/17 10:30 UTC	<div>Available </div>
<input type="checkbox"/>	ns-postgres-kevin-kevin-preview-clus3-snapshot-20201217183636		<a href="#">On-Schedule</a>	2020/12/17 18:36 UTC	<div>Backup</div> <div>Restore application</div> <div>Delete snapshot</div> <div>Failed </div>
<input type="checkbox"/>	ns-postgres-kevin-kevin-preview-clus3-snapshot-20201217154314		<a href="#">On-Schedule</a>	2020/12/17 15:43 UTC	

- If you delete a Kubernetes cluster that Astra Control Service is managing, the cluster now shows up in a **Removed** state. You can then remove the cluster from Astra Control Service.
- Account owners can now modify the assigned roles for other users.
- We added a section for billing, which will be enabled when Astra Control Service is released for General Availability (GA).

## Known issues

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

The following known issues affect the current release:

### Apps

- [Unable to define an app on a namespace that has been deleted and recreated](#)

### Backup, restore, and clone

- [App clones fail using a specific version of PostgreSQL](#)
- [App backups and snapshots fail if the volumesnapshotclass is added after a cluster is managed](#)
- [In-place restore operations to ontap-nas-economy storage classes fail](#)
- [Restoring from a backup when using Kerberos in-flight encryption can fail](#)
- [Backup data remains in bucket after deletion for buckets with expired retention policy](#)

### Other issues

- [App data management operations fail with Internal Service Error \(500\) when Astra Trident is offline](#)

## Unable to define an app on a namespace that has been deleted and recreated

If you define an application with a namespace, delete the namespace, and then reinstall the app in the same namespace, the operation fails with a 409 error code. To define the app using the recreated namespace, delete the old application instance first.

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

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

## In-place restore operations to ontap-nas-economy storage classes fail

If you perform an in-place restore of an application (restore the app to its original namespace), and the app's storage class uses the `ontap-nas-economy` driver, the restore operation can fail if the snapshot directory is not hidden. Before restoring in-place, follow the instructions in [Enable backup and restore for ontap-nas-economy operations](#) to hide the snapshot directory.

## Restoring from a backup when using Kerberos in-flight encryption can fail

When you restore an application from a backup to a storage backend that is using Kerberos in-flight encryption, the restore operation can fail. This issue does not affect restoring from a snapshot or replicating the application data using NetApp SnapMirror.



When using Kerberos in-flight encryption with NFSv4 volumes, ensure that the NFSv4 volumes are using the correct settings. Refer to the NetApp NFSv4 Domain Configuration section (page 13) of the [NetApp NFSv4 Enhancements and Best Practices Guide](#).

## Backup data remains in bucket after deletion for buckets with expired retention policy

If you delete an app's immutable backup after the bucket's retention policy has expired, the backup is deleted from Astra Control but not from the bucket. This issue will be fixed in an upcoming release.

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

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

# Known limitations

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

## General limitations

The following limitations affect Astra Control Service's management of Kubernetes clusters in any supported Kubernetes deployment.

### 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 Service requires `psql` access to freeze and thaw the databases. If there is a pre-existing connection, the snapshot, backup, or clone will fail.

## The Activity page displays up to 100,000 events

The Astra Control Activity page can display up to 100,000 events. To view all logged events, retrieve the events using the [Astra Control REST API](#).

## Limitations for management of GKE clusters

The following limitations apply to the management of Kubernetes clusters in Google Kubernetes Engine (GKE).

### App management limitations

The following limitations affect Astra Control Service's management of applications.

#### Multiple applications that use the same namespace cannot be restored collectively to a different namespace

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

#### Astra Control does not automatically assign default buckets for cloud instances

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

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

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

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

After an application is deployed with a storage class explicitly set (for example, `helm install ...-set global.storageClass=netapp-cvs-perf-extreme`), subsequent attempts to clone the application require that the target cluster have the originally specified storage class.

Cloning an application with an explicitly set storage class to a cluster that does not have the same storage class will fail. There are no recovery steps in this scenario.

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

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

- [Apache K8ssandra](#)



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

- [Jenkins CI](#)



- [Percona XtraDB Cluster](#)

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



During clone operations, apps that need an IngressClass resource or webhooks to function properly must not have those resources already defined on the destination cluster.

## Role-based Access Control (RBAC) limitations

The following limitations apply to the way Astra Control limits user access to resources or capabilities.

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

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

### A Member user with namespace constraints cannot access cloned or restored apps until an Admin user adds the namespace to the constraint

Any `member` user with RBAC constraints by namespace name/ID can clone or restore an app to a new namespace on the same cluster or to any other cluster in their organization's account. However, the same user cannot access the cloned or restored app in the new namespace. After a clone or restore operation creates a new namespace, the account admin/owner can edit the `member` user account and update role constraints for the affected user to grant access to the new namespace.

### Snapshots might fail for Kubernetes 1.25 or later clusters with certain snapshot controller versions

Snapshots for Kubernetes clusters running version 1.25 or later can fail if version `v1beta1` of the snapshot controller APIs are installed on the cluster.

As a workaround, do the following when upgrading existing Kubernetes 1.25 or later installations:

1. Remove any existing Snapshot CRDs and any existing snapshot controller.
2. [Uninstall Astra Trident](#).
3. [Install the snapshot CRDs and the snapshot controller](#).
4. [Install the latest Astra Trident version](#).
5. [Create a VolumeSnapshotClass](#).

# Get started

## Learn about 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
- Replicate applications to a remote system using NetApp SnapMirror technology (Astra Control Center)
- Clone applications from staging to production
- Visualize application health and protection status
- Work with a web UI or an API to implement your backup and migration workflows

### 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 multiple cloud provider environments, as well as self-managed Kubernetes clusters.
- **Astra Control Center:** Self-managed software that provides application-aware data management of Kubernetes clusters running in your on-premises environment. Astra Control Center can also be installed on multiple cloud provider environments with a NetApp Cloud Volumes ONTAP storage backend.

	Astra Control Service	Astra Control Center
How is it offered?	As a fully managed cloud service from NetApp	As software that you can download, install, and manage
Where is it hosted?	On a public cloud of NetApp's choice	On your own Kubernetes cluster
How is it updated?	Managed by NetApp	You manage any updates

	Astra Control Service	Astra Control Center
What are the supported Kubernetes distributions?	<ul style="list-style-type: none"> <li>• <b>Cloud providers</b> <ul style="list-style-type: none"> <li>◦ Amazon Web Services <ul style="list-style-type: none"> <li>▪ Amazon Elastic Kubernetes Service (EKS)</li> </ul> </li> <li>◦ Google Cloud <ul style="list-style-type: none"> <li>▪ Google Kubernetes Engine (GKE)</li> </ul> </li> <li>◦ Microsoft Azure <ul style="list-style-type: none"> <li>▪ Azure Kubernetes Service (AKS)</li> </ul> </li> </ul> </li> <li>• <b>Self-managed clusters</b> <ul style="list-style-type: none"> <li>◦ Kubernetes (Upstream)</li> <li>◦ Rancher Kubernetes Engine (RKE)</li> <li>◦ Red Hat OpenShift Container Platform</li> </ul> </li> <li>• <b>On-premises clusters</b> <ul style="list-style-type: none"> <li>◦ Red Hat OpenShift Container Platform on-premises</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• Azure Kubernetes Service on Azure Stack HCI</li> <li>• Google Anthos</li> <li>• Kubernetes (Upstream)</li> <li>• Rancher Kubernetes Engine (RKE)</li> <li>• Red Hat OpenShift Container Platform</li> </ul>

	Astra Control Service	Astra Control Center
What are the supported storage backends?	<ul style="list-style-type: none"> <li>• <b>Cloud providers</b> <ul style="list-style-type: none"> <li>◦ Amazon Web Services <ul style="list-style-type: none"> <li>▪ Amazon EBS</li> <li>▪ Amazon FSx for NetApp ONTAP</li> <li>▪ <a href="#">Cloud Volumes ONTAP</a></li> </ul> </li> <li>◦ Google Cloud <ul style="list-style-type: none"> <li>▪ Google Persistent Disk</li> <li>▪ NetApp Cloud Volumes Service</li> <li>▪ <a href="#">Cloud Volumes ONTAP</a></li> </ul> </li> <li>◦ Microsoft Azure <ul style="list-style-type: none"> <li>▪ Azure Managed Disks</li> <li>▪ Azure NetApp Files</li> <li>▪ <a href="#">Cloud Volumes ONTAP</a></li> </ul> </li> </ul> </li> <li>• <b>Self-managed clusters</b> <ul style="list-style-type: none"> <li>◦ Amazon EBS</li> <li>◦ Azure Managed Disks</li> <li>◦ Google Persistent Disk</li> <li>◦ <a href="#">Cloud Volumes ONTAP</a></li> <li>◦ NetApp MetroCluster</li> <li>◦ <a href="#">Longhorn</a></li> </ul> </li> <li>• <b>On-premises clusters</b> <ul style="list-style-type: none"> <li>◦ NetApp MetroCluster</li> <li>◦ NetApp ONTAP AFF and FAS systems</li> <li>◦ NetApp ONTAP Select</li> <li>◦ <a href="#">Cloud Volumes ONTAP</a></li> <li>◦ <a href="#">Longhorn</a></li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• NetApp ONTAP AFF and FAS systems</li> <li>• NetApp ONTAP Select</li> <li>• <a href="#">Cloud Volumes ONTAP</a></li> <li>• <a href="#">Longhorn</a></li> </ul>

## 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](#) or Google Persistent Disks as the storage backend for your persistent volumes.
- For AKS clusters, Astra Control Service uses [Azure NetApp Files](#) or Azure managed disks as the storage backend for your persistent volumes.
- For Amazon EKS clusters, Astra Control Service uses [Amazon Elastic Block Store](#) or [Amazon FSx for NetApp ONTAP](#) as the storage backend 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 `link../concepts/architecture#astra-control-components[Astra Control Provisioner]` on the cluster and to create one or more storage classes.
  - If you use a NetApp cloud service storage offering as your storage backend, Astra Control Service uses Astra Control Provisioner to provision persistent volumes for your apps. If you use Amazon EBS or Azure managed disks as your storage backend, you need to install a provider-specific CSI driver. Installation instructions are provided in [Set up Amazon Web Services](#) and [Set up Microsoft Azure with Azure managed disks](#).
- At this point, you can define apps from your cluster. Persistent volumes will be provisioned on the storage backend through the new default storage class.
- You then use Astra Control Service to manage these apps, and start creating snapshots, backups, and clones.

Astra Control's Free Plan enables you to manage up to 10 namespaces in your account. If you want to manage more than 10 namespaces, 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.

Astra Control Center supports Kubernetes clusters with an Astra Control Provisioner-configured storage class with an ONTAP storage backend.

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 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 storage backend 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](#).

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

## For more information

- [NetApp Astra product family documentation](#)
- [Astra Control Center documentation](#)
- [Astra Control API documentation](#)
- [Astra Trident documentation](#)
- [ONTAP documentation](#)

## Supported Kubernetes deployments

Astra Control Service can manage apps that are running on a managed Kubernetes cluster in Google Kubernetes Engine (GKE), Amazon Elastic Kubernetes Service (EKS), and Azure Kubernetes Service (AKS). Astra Control Service can also manage clusters that you manage on your own.

- [Learn how to set up Amazon Web Services for Astra Control Service.](#)
- [Learn how to set up Google Cloud for Astra Control Service.](#)
- [Learn how to set up Microsoft Azure with Azure NetApp Files for Astra Control Service.](#)
- [Learn how to set up Microsoft Azure with Azure managed disks for Astra Control Service.](#)
- [Learn how to prepare self-managed clusters before adding them to Astra Control Service.](#)

## Quick start for Astra Control Service

This page provides a high-level overview of the steps that you need to complete to get started with Astra Control Service. The links within each step take you to a page that provides more details.

### [One] Set up your cloud provider

- a. Google Cloud:
  - Review Google Kubernetes Engine cluster requirements.
  - Purchase Cloud Volumes Service for Google Cloud from the Google Cloud Marketplace.

- Enable the required APIs.
- Create a service account and service account key.
- Set up network peering from your VPC to Cloud Volumes Service for Google Cloud.

[Learn more about Google Cloud requirements.](#)

b. Amazon Web Services:

- Review Amazon Web Services cluster requirements.
- Create an Amazon account.
- Install the Amazon Web Services CLI.
- Create an IAM user.
- Create and attach a permissions policy.
- Save the credentials for the IAM user.

[Learn more about Amazon Web Services requirements.](#)

c. Microsoft Azure:

- Review Azure Kubernetes Service cluster requirements for the storage backend you plan to use.

[Learn more about Microsoft Azure and Azure NetApp Files requirements.](#)

[Learn more about Microsoft Azure and Azure managed disk requirements.](#)

If you are managing your own cluster and it is not hosted by a cloud provider, review the requirements for self-managed clusters.

[Learn more about self-managed cluster requirements.](#)

## [Two] Complete the Astra Control registration

- Create a [NetApp BlueXP](#) account.
- Specify your NetApp BlueXP email ID when creating your Astra Control account [from the Astra Control product page](#).

[Learn more about the registration process.](#)

## [Three] Add clusters to Astra Control

After you log in, select **Add cluster** to start managing your cluster with Astra Control.

[Learn more about adding clusters.](#)

# Set up your cloud provider

## Set up Amazon Web Services

A few steps are required to prepare your Amazon Web Services project before you can manage Amazon Elastic Kubernetes Service (EKS) clusters with Astra Control Service.

## Quick start for setting up Amazon Web Services

Get started quickly by following these steps or scroll down to the remaining sections for full details.

### [One] Review Astra Control Service requirements for Amazon Web Services

Ensure that clusters are healthy and running a supported version of Kubernetes, that worker nodes are online and running Linux or Windows, and more. [Learn more about this step.](#)

### [Two] Create an Amazon account

If you don't already have an Amazon account, you need to create one so that you can use EKS. [Learn more about this step.](#)

### [Three] Install the Amazon Web Services CLI

Install the AWS CLI so that you can manage AWS from the command line. [Follow step-by-step instructions.](#)

### [Four] Optional: Create an IAM user

Create an Amazon Identity and Access Management (IAM) user. You can also skip this step and use an existing IAM user with Astra Control Service.

[Read step-by-step instructions.](#)

### [Five] Create and attach a permissions policy

Create a policy with the required permissions for Astra Control Service to interact with your AWS account.

[Read step-by-step instructions.](#)

### [Six] Save the credentials for the IAM user

Save the credentials for the IAM user so that you can import the credentials in to Astra Control Service.

[Read step-by-step instructions.](#)

## EKS cluster requirements

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

### Kubernetes version

A cluster must be running a Kubernetes version in the range of 1.25 to 1.28.

### Image type

The image type for each worker node must be Linux.

### Cluster state

Clusters must be running in a healthy state and have at least one online worker node with no worker nodes in a failed state.

### Astra Control Provisioner

Astra Control Provisioner and an external snapshot controller are required for operations with storage backends. To enable these operations, do the following:



1. [Install the snapshot CRDs and the snapshot controller.](#)
2. [Enable Astra Control Provisioner.](#)
3. [Create a VolumeSnapshotClass.](#)

### **CSI drivers for Amazon Elastic Block Store (EBS)**

If you use the Amazon EBS storage backend, you need to install the Container Storage Interface (CSI) driver for EBS (it is not installed automatically).

Refer to the steps for instructions for installing the CSI driver.

## Install an external snapshotter

If you haven't already done so, [install the snapshot CRDs and the snapshot controller](#).

## Install the CSI driver as an Amazon EKS add-on

1. Create the Amazon EBS CSI driver IAM role for service accounts. Follow the instructions [in the Amazon documentation](#), using the AWS CLI commands in the instructions.
2. Add the Amazon EBS CSI add-on using the following AWS CLI command, replacing information in brackets <> with values specific to your environment. Replace <DRIVER\_ROLE> with the name of the EBS CSI driver role that you created in the previous step:

```
aws eks create-addon \
  --cluster-name <CLUSTER_NAME> \
  --addon-name aws-ebs-csi-driver \
  --service-account-role-arn
arn:aws:iam::<ACCOUNT_ID>:role/<DRIVER_ROLE>
```

## Configure the EBS storage class

1. Clone the Amazon EBS CSI driver GitHub repository to your system.

```
git clone https://github.com/kubernetes-sigs/aws-ebs-csi-
driver.git
```

2. Navigate to the dynamic-provisioning example directory.

```
cd aws-ebs-csi-driver/examples/kubernetes/dynamic-provisioning/
```

3. Deploy the ebs-sc storage class and ebs-claim persistent volume claim from the manifests directory.

```
kubectl apply -f manifests/storageclass.yaml
kubectl apply -f manifests/claim.yaml
```

4. Describe the ebs-sc storage class.

```
kubectl describe storageclass ebs-sc
```

You should see output describing the storage class attributes.

## Create an Amazon account

If you don't already have an Amazon account, you need to create one to enable billing for Amazon EKS.

### Steps

1. Go to the [Amazon homepage](#) , select **Sign in** at the top right, and select **Start here**.
2. Follow the prompts to create an account.

## Install the Amazon Web Services CLI

Install the AWS CLI so that you can manage AWS resources from the command line.

### Step

1. Go to [Getting started with the AWS CLI](#) and follow the instructions to install the CLI.

## Optional: Create an IAM user

Create an IAM user so that you can use and manage AWS services and resources with increased security. You can also skip this step, and use an existing IAM user with Astra Control Service.

### Step

1. Go to [Creating IAM users](#) and follow the instructions to create an IAM user.

## Create and attach a permissions policy

Create a policy with the required permissions for Astra Control Service to interact with your AWS account.

### Steps

1. Create a new file called `policy.json`.
2. Copy the following JSON content into the file:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": [
        "cloudwatch:GetMetricData",
        "fsx:DescribeVolumes",
        "ec2:DescribeRegions",
        "s3:CreateBucket",
        "s3:ListBucket",
        "s3:PutObject",
        "s3:GetObject",
        "iam:SimulatePrincipalPolicy",
        "s3:ListAllMyBuckets",
        "eks:DescribeCluster",
        "eks:ListNodegroups",
        "eks:DescribeNodegroup",
        "eks:ListClusters",
        "iam:GetUser",
        "s3:DeleteObject",
        "s3:DeleteBucket",
        "autoscaling:DescribeAutoScalingGroups"
      ],
      "Resource": "*"
    }
  ]
}
```

### 3. Create the policy:

```
POLICY_ARN=$(aws iam create-policy --policy-name <policy-name> --policy
-document file://policy.json --query='Policy.Arn' --output=text)
```

### 4. Attach the policy to the IAM user. Replace <IAM-USER-NAME> with either the user name of the IAM user you created, or an existing IAM user:

```
aws iam attach-user-policy --user-name <IAM-USER-NAME> --policy-arn
=$POLICY_ARN
```

## Save the credentials for the IAM user

Save the credentials for the IAM user so that you can make Astra Control Service aware of the user.

### Steps

1. Download the credentials. Replace <IAM-USER-NAME> with the user name of the IAM user you want to use:

```
aws iam create-access-key --user-name <IAM-USER-NAME> --output json > credential.json
```

### Result

The `credential.json` file is created, and you can import the credentials in to Astra Control Service.

## Set up Google Cloud

A few steps are required to prepare your Google Cloud project before you can manage Google Kubernetes Engine clusters with Astra Control Service.



If you do not start out using Google Cloud Volumes Service for Google Cloud as a storage backend but plan to use it at a later date, you should complete the necessary steps to configure Google Cloud Volumes Service for Google Cloud now. Creating a service account later means that you might lose access to your existing storage buckets.

### Quick start for setting up Google Cloud

Get started quickly by following these steps or scroll down to the remaining sections for full details.

#### [One] Review Astra Control Service requirements for Google Kubernetes Engine

Ensure that clusters are healthy and running a supported Kubernetes version, that worker nodes are online and running a supported image type, and more. [Learn more about this step.](#)

#### [Two] (Optional): Purchase Cloud Volumes Service for Google Cloud

If you plan to use Cloud Volumes Service for Google Cloud as a storage backend, go to the NetApp Cloud Volumes Service page in the Google Cloud Marketplace and select Purchase. [Learn more about this step.](#)

#### [Three] Enable APIs in your Google Cloud project

Enable the following Google Cloud APIs:

- Google Kubernetes Engine
- Cloud Storage
- Cloud Storage JSON API
- Service Usage
- Cloud Resource Manager API
- NetApp Cloud Volumes Service

- Required for Cloud Volumes Service for Google Cloud
- Optional (but recommended) for Google Persistent Disk
- Service Consumer Management API
- Service Networking API
- Service Management API

[Follow step-by-step instructions.](#)

#### **[Four] Create a service account that has the required permissions**

Create a Google Cloud service account that has the following permissions:

- Kubernetes Engine Admin
- NetApp Cloud Volumes Admin
  - Required for Cloud Volumes Service for Google Cloud
  - Optional (but recommended) for Google Persistent Disk
- Storage Admin
- Service Usage Viewer
- Compute Network Viewer

[Read step-by-step instructions.](#)

#### **[Five] Create a service account key**

Create a key for the service account and save the key file in a secure location. [Follow step-by-step instructions.](#)

#### **[Six] (Optional): Set up network peering for your VPC**

If you plan to use Cloud Volumes Service for Google Cloud as a storage backend, set up network peering from your VPC to Cloud Volumes Service for Google Cloud. [Follow step-by-step instructions.](#)

### **GKE cluster requirements**

A Kubernetes cluster must meet the following requirements so you can discover and manage it from Astra Control Service. Note that some of these requirements only apply if you plan to use Cloud Volumes Service for Google Cloud as a storage backend.

#### **Kubernetes version**

A cluster must be running a Kubernetes version in the range of 1.26 to 1.28.

#### **Image type**

The image type for each worker node must be `COS_CONTAINERD`.

#### **Cluster state**

Clusters must be running in a healthy state and have at least one online worker node with no worker nodes in a failed state.

## Google Cloud region

If you plan to use Cloud Volumes Service for Google Cloud as a storage backend, clusters must be running in a [Google Cloud region where Cloud Volumes Service for Google Cloud is supported](#). Note that Astra Control Service supports both service types: CVS and CVS-Performance. As a best practice, you should choose a region that supports Cloud Volumes Service for Google Cloud, even if you do not use it as a storage backend. This makes it easier to use Cloud Volumes Service for Google Cloud as a storage backend in the future if your performance requirements change.

## Networking

If you plan to use Cloud Volumes Service for Google Cloud as a storage backend, the cluster must reside in a VPC that is peered with Cloud Volumes Service for Google Cloud. [This step is described below](#).

## Private clusters

If the cluster is private, the [authorized networks](#) must allow the Astra Control Service IP address:

52.188.218.166/32

## Mode of operation for a GKE cluster

You should use the Standard mode of operation. The Autopilot mode hasn't been tested at this time. [Learn more about modes of operation](#).

## Storage pools

If you use NetApp Cloud Volumes Service as a storage backend with the CVS service type, you need to configure storage pools before you can provision volumes. Refer to [Service type, storage classes, and PV size for GKE clusters](#) for more information.

## Optional: Purchase Cloud Volumes Service for Google Cloud

Astra Control Service can use Cloud Volumes Service for Google Cloud as the storage backend for your persistent volumes. If you plan to use this service, you need to purchase Cloud Volumes Service for Google Cloud from the Google Cloud Marketplace to enable billing for persistent volumes.

### Step

1. Go to the [NetApp Cloud Volumes Service page](#) in the Google Cloud Marketplace, select **Purchase**, and follow the prompts.

[Follow step-by-step instructions in the Google Cloud documentation to purchase and enable the service.](#)

## Enable APIs in your project

Your project needs permissions to access specific Google Cloud APIs. APIs are used to interact with Google Cloud resources, such as Google Kubernetes Engine (GKE) clusters and NetApp Cloud Volumes Service storage.

### Step

1. [Use the Google Cloud console or gcloud CLI to enable the following APIs](#):
  - Google Kubernetes Engine
  - Cloud Storage
  - Cloud Storage JSON API
  - Service Usage

- Cloud Resource Manager API
- NetApp Cloud Volumes Service (Required for Cloud Volumes Service for Google Cloud)
- Service Consumer Management API
- Service Networking API
- Service Management API

The following video shows how to enable the APIs from the Google Cloud console.

► <https://docs.netapp.com/us-en/astra-control-service/media/get-started/video-enable-gcp-apis.mp4> (video)

## Create a service account

Astra Control Service uses a Google Cloud service account to facilitate Kubernetes application data management on your behalf.

### Steps

1. Go to Google Cloud and [create a service account by using the console, gcloud command, or another preferred method](#).
2. Grant the service account the following roles:
  - **Kubernetes Engine Admin** - Used to list clusters and create admin access to manage apps.
  - **NetApp Cloud Volumes Admin** - Used to manage persistent storage for apps.
  - **Storage Admin** - Used to manage buckets and objects for backups of apps.
  - **Service Usage Viewer** - Used to check if the required Cloud Volumes Service for Google Cloud APIs are enabled.
  - **Compute Network Viewer** - Used to check if the Kubernetes VPC is allowed to reach Cloud Volumes Service for Google Cloud.

If you'd like to use gcloud, you can follow steps from within the Astra Control interface. Select **Account > Credentials > Add Credentials**, and then select **Instructions**.

If you'd like to use the Google Cloud console, the following video shows how to create the service account from the console.

► <https://docs.netapp.com/us-en/astra-control-service/media/get-started/video-create-gcp-service->



[account.mp4](#) (video)

### Configure the service account for a shared VPC

To manage GKE clusters that reside in one project, but use a VPC from a different project (a shared VPC), then you need to specify the Astra service account as a member of the host project with the **Compute Network Viewer** role.

#### Steps

1. From the Google Cloud console, go to **IAM & Admin** and select **Service Accounts**.
2. Find the Astra service account that has [the required permissions](#) and then copy the email address.
3. Go to your host project and then select **IAM & Admin > IAM**.
4. Select **Add** and add an entry for the service account.
  - a. **New members**: Enter the email address for the service account.
  - b. **Role**: Select **Compute Network Viewer**.
  - c. Select **Save**.

#### Result

Adding a GKE cluster using a shared VPC will fully work with Astra.

### Create a service account key

Instead of providing a user name and password to Astra Control Service, you'll provide a service account key when you add your first cluster. Astra Control Service uses the service account key to establish the identity of the service account that you just set up.

The service account key is plaintext stored in the JavaScript Object Notation (JSON) format. It contains information about the GCP resources that you have permission to access.

You can only view or download the JSON file when you create the key. However, you can create a new key at any time.

#### Steps

1. Go to Google Cloud and [create a service account key by using the console, gcloud command, or another preferred method](#).
2. When prompted, save the service account key file in a secure location.

The following video shows how to create the service account key from the Google Cloud console.

► <https://docs.netapp.com/us-en/astra-control-service/media/get-started/video-create-gcp-service-account->

## Optional: Set up network peering for your VPC

If you plan to use Cloud Volumes Service for Google Cloud as a storage backend service, the final step is to set up networking peering from your VPC to Cloud Volumes Service for Google Cloud.

The easiest way to set up network peering is by obtaining the gcloud commands directly from Cloud Volumes Service. The commands are available from Cloud Volumes Service when creating a new file system.

### Steps

1. [Go to NetApp BlueXP Global Regions Maps](#) and identify the service type that you'll be using in the Google Cloud region where your cluster resides.

Cloud Volumes Service provides two service types: CVS and CVS-Performance. [Learn more about these service types](#).

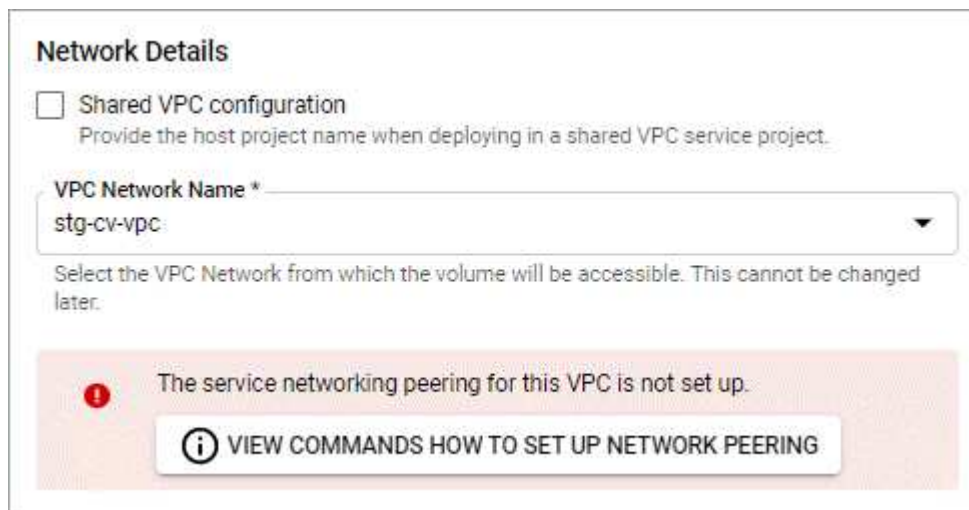
2. [Go to Cloud Volumes in Google Cloud Platform](#).
3. On the **Volumes** page, select **Create**.
4. Under **Service Type**, select either **CVS** or **CVS-Performance**.

You need to choose the correct service type for your Google Cloud region. This is the service type that you identified in step 1. After you select a service type, the list of regions on the page updates with the regions where that service type is supported.

After this step, you'll only need to enter your networking information to obtain the commands.

5. Under **Region**, select your region and zone.
6. Under **Network Details**, select your VPC.

If you haven't set up network peering, you'll see the following notification:



**Network Details**

☐ Shared VPC configuration  
Provide the host project name when deploying in a shared VPC service project.

VPC Network Name \*  
stg-cv-vpc

Select the VPC Network from which the volume will be accessible. This cannot be changed later.

**The service networking peering for this VPC is not set up.**

**VIEW COMMANDS HOW TO SET UP NETWORK PEERING**

7. Select the button to view the network peering set up commands.
8. Copy the commands and run them in Cloud Shell.

For more details about using these commands, refer to the [Quickstart for Cloud Volumes Service for GCP](#).

[Learn more about configuring private services access and setting up network peering.](#)

9. After you're done, you can select cancel on the **Create File System** page.

We started creating this volume only to get the commands for network peering.

## Set up Microsoft Azure with Azure NetApp Files

A few steps are required to prepare your Microsoft Azure subscription before you can manage Azure Kubernetes Service clusters with Astra Control Service. Follow these instructions if you plan to use Azure NetApp Files as a storage backend.

### Quick start for setting up Azure

Get started quickly by following these steps or scroll down to the remaining sections for full details.

#### [One] Review Astra Control Service requirements for Azure Kubernetes Service

Ensure that clusters are healthy and running a supported version of Kubernetes, that node pools are online and running Linux, and more. [Learn more about this step.](#)

#### [Two] Sign up for Microsoft Azure

Create a Microsoft Azure account. [Learn more about this step.](#)

#### [Three] Register for Azure NetApp Files

Register the NetApp Resource Provider. [Learn more about this step.](#)

#### [Four] Create a NetApp account

Go to Azure NetApp Files in the Azure portal and create a NetApp account. [Learn more about this step.](#)

#### [Five] Set up capacity pools

Set up one or more capacity pools for your persistent volumes. [Learn more about this step.](#)

#### [Six] Delegate a subnet to Azure NetApp Files

Delegate a subnet to Azure NetApp Files so that Astra Control Service can create persistent volumes in that subnet. [Learn more about this step.](#)

#### [Seven] Create an Azure service principal

Create an Azure service principal that has the Contributor role. [Learn more about this step.](#)

#### [Eight] Optional: Configure redundancy for Azure backup buckets

By default, the buckets Astra Control Service uses to store Azure Kubernetes Service backups use the Locally Redundant Storage (LRS) redundancy option. As an optional step, you can configure a more durable level of redundancy for Azure buckets. [Learn more about this step.](#)

## Azure Kubernetes Service cluster requirements

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

### Kubernetes version

Clusters must be running Kubernetes version 1.26 to 1.28.

### Image type

The image type for all node pools must be Linux.

### Cluster state

Clusters must be running in a healthy state and have at least one online worker node with no worker nodes in a failed state.

### Azure region

Clusters must reside in a region where Azure NetApp Files is available. [View Azure products by region](#).

### Subscription

Clusters must reside in a subscription where Azure NetApp Files is enabled. You'll choose a subscription when you [register for Azure NetApp Files](#).

### VNet

Consider the following VNet requirements:

- Clusters must reside in a VNet that has direct access to an Azure NetApp Files delegated subnet. [Learn how to set up a delegated subnet](#).
- If your Kubernetes clusters are in a VNet that's peered to the Azure NetApp Files delegated subnet that's in another VNet, then both sides of the peering connection must be online.
- Be aware that the default limit for the number of IPs used in a VNet (including immediately peered VNets) with Azure NetApp Files is 1,000. [View Azure NetApp Files resource limits](#).

If you're close to the limit, you have two options:

- You can [submit a request for a limit increase](#). Contact your NetApp representative if you need help.
- When creating a new Amazon Kubernetes Service (AKS) cluster, specify a new network for the cluster. Once the new network is created, provision a new subnet and delegate the subnet to Azure NetApp Files.

## Sign up for Microsoft Azure

If you don't have a Microsoft Azure account, begin by signing up for Microsoft Azure.

### Steps

1. Go to the [Azure subscription page](#) to subscribe to the Azure service.
2. Select a plan and follow the instructions to complete the subscription.

## Register for Azure NetApp Files

Get access to Azure NetApp Files by registering the NetApp Resource Provider.

### Steps

1. Log in to the Azure portal.
2. [Follow Azure NetApp Files documentation to register the NetApp Resource Provider.](#)

## Create a NetApp account

Create a NetApp account in Azure NetApp Files.

### Step

1. [Follow Azure NetApp Files documentation to create a NetApp account from the Azure portal.](#)

## Set up a capacity pool

One or more capacity pools are required so that Astra Control Service can provision persistent volumes in a capacity pool. Astra Control Service doesn't create capacity pools for you.

Take the following into consideration as you set up capacity pools for your Kubernetes apps:

- The capacity pools need to be created in the same Azure region where the AKS clusters will be managed with Astra Control Service.
- A capacity pool can have an Ultra, Premium, or Standard service level. Each of these service levels are designed for different performance needs. Astra Control Service supports all three.

You need to set up a capacity pool for each service level that you want to use with your Kubernetes clusters.

[Learn more about service levels for Azure NetApp Files.](#)

- Before you create a capacity pool for the apps that you intend to protect with Astra Control Service, choose the required performance and capacity for those apps.

Provisioning the right amount of capacity ensures that users can create persistent volumes as they are needed. If capacity isn't available, then the persistent volumes can't be provisioned.

- An Azure NetApp Files capacity pool can use the manual or auto QoS type. Astra Control Service supports auto QoS capacity pools. Manual QoS capacity pools aren't supported.

### Step

1. [Follow Azure NetApp Files documentation to set up an auto QoS capacity pool.](#)

## Delegate a subnet to Azure NetApp Files

You need to delegate a subnet to Azure NetApp Files so that Astra Control Service can create persistent volumes in that subnet. Note that Azure NetApp Files enables you to have only one delegated subnet in a VNet.

If you're using peered VNets, then both sides of the peering connection must be online: the VNet where your Kubernetes clusters reside and the VNet that has the Azure NetApp Files delegated subnet.

### Step

1. [Follow the Azure NetApp Files documentation to delegate a subnet to Azure NetApp Files.](#)

## After you're done

Wait about 10 minutes before discovering the cluster running in the delegated subnet.

## Create an Azure service principal

Astra Control Service requires a Azure service principal that is assigned the Contributor role. Astra Control Service uses this service principal to facilitate Kubernetes application data management on your behalf.

A service principal is an identity created specifically for use with applications, services, and tools. Assigning a role to the service principal restricts access to specific Azure resources.

Follow the steps below to create a service principal using the Azure CLI. You'll need to save the output in a JSON file and provide it to Astra Control Service later on. [Refer to Azure documentation for more details about using the CLI.](#)

The following steps assume that you have permission to create a service principal and that you have the Microsoft Azure SDK (az command) installed on your machine.

### Requirements

- The service principal must use regular authentication. Certificates aren't supported.
- The service principal must be granted Contributor or Owner access to your Azure subscription.
- The subscription or resource group you choose for scope must contain the AKS clusters and your Azure NetApp Files account.

### Steps

1. Identify the subscription and tenant ID where your AKS clusters reside (these are the clusters that you want to manage in Astra Control Service).

```
az configure --list-defaults
az account list --output table
```

2. Do one of the following, depending on if you use an entire subscription or a resource group:

- Create the service principal, assign the Contributor role, and specify the scope to the entire subscription where the clusters reside.

```
az ad sp create-for-rbac --name service-principal-name --role
contributor --scopes /subscriptions/SUBSCRIPTION-ID
```

- Create the service principal, assign the Contributor role, and specify the resource group where the clusters reside.

```
az ad sp create-for-rbac --name service-principal-name --role
contributor --scopes /subscriptions/SUBSCRIPTION-
ID/resourceGroups/RESOURCE-GROUP-ID
```

3. Store the resulting Azure CLI output as a JSON file.

You'll need to provide this file so that Astra Control Service can discover your AKS clusters and manage Kubernetes data management operations. [Learn about managing credentials in Astra Control Service.](#)

- Optional: Add the subscription ID to the JSON file so that Astra Control Service automatically populates the ID when you select the file.

Otherwise, you'll need to enter the subscription ID in Astra Control Service when prompted.

### Example

```
{
  "appId": "0db3929a-bfb0-4c93-baee-aaf8",
  "displayName": "sp-example-dev-sandbox",
  "name": "http://sp-example-dev-sandbox",
  "password": "mypassword",
  "tenant": "011cdf6c-7512-4805-aaf8-7721afd8ca37",
  "subscriptionId": "99ce999a-8c99-99d9-a9d9-99cce99f99ad"
}
```

- Optional: Test your service principal. Choose from the following example commands depending on the scope your service principal uses.

### Subscription scope

```
az login --service-principal --username APP-ID-SERVICEPRINCIPAL
--password PASSWORD --tenant TENANT-ID
az group list --subscription SUBSCRIPTION-ID
az aks list --subscription SUBSCRIPTION-ID
az storage container list --account-name STORAGE-ACCOUNT-NAME
```

### Resource group scope

```
az login --service-principal --username APP-ID-SERVICEPRINCIPAL
--password PASSWORD --tenant TENANT-ID
az aks list --subscription SUBSCRIPTION-ID --resource-group RESOURCE-
GROUP-ID
```

## Optional: Configure redundancy for Azure backup buckets

You can configure a more durable redundancy level for Azure backup buckets. By default, the buckets Astra Control Service uses to store Azure Kubernetes Service backups use the Locally Redundant Storage (LRS) redundancy option. To use a more durable redundancy option for Azure buckets, you need to do the following:

### Steps

- Create an Azure storage account that uses the redundancy level you need using [these instructions](#).
- Create an Azure container in the new storage account using [these instructions](#).
- Add the container as a bucket to Astra Control Service. Refer to [Add an additional bucket](#).
- (Optional) To use the newly created bucket as the default bucket for Azure backups, set it as the default bucket for Azure. Refer to [Change the default bucket](#).

## Set up Microsoft Azure with Azure managed disks

A few steps are required to prepare your Microsoft Azure subscription before you can manage Azure Kubernetes Service clusters with Astra Control Service. Follow these instructions if you plan to use Azure managed disks as a storage backend.

### Quick start for setting up Azure

Get started quickly by following these steps or scroll down to the remaining sections for full details.

#### [One] Review Astra Control Service requirements for Azure Kubernetes Service

Ensure that clusters are healthy and running a supported version of Kubernetes, that node pools are online and running Linux, and more. [Learn more about this step.](#)

#### [Two] Sign up for Microsoft Azure

Create a Microsoft Azure account. [Learn more about this step.](#)

#### [Three] Create an Azure service principal

Create an Azure service principal that has the Contributor role. [Learn more about this step.](#)

#### [Four] Configure Container Storage Interface (CSI) driver details

You need to configure your Azure subscription and the cluster to work with the CSI drivers. [Learn more about this step.](#)

#### [Five] Optional: Configure redundancy for Azure backup buckets

By default, the buckets Astra Control Service uses to store Azure Kubernetes Service backups use the Locally Redundant Storage (LRS) redundancy option. As an optional step, you can configure a more durable level of redundancy for Azure buckets. [Learn more about this step.](#)

## Azure Kubernetes Service cluster requirements

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

### Kubernetes version

Clusters must be running Kubernetes version 1.26 to 1.28.

### Image type

The image type for all node pools must be Linux.

### Cluster state

Clusters must be running in a healthy state and have at least one online worker node with no worker nodes in a failed state.

### Azure region

As a best practice, you should choose a region that supports Azure NetApp Files, even if you do not use it as a storage backend. This makes it easier to use Azure NetApp Files as a storage backend in the future if your performance requirements change. [View Azure products by region.](#)



## CSI drivers

Clusters must have the appropriate CSI drivers installed.

## Sign up for Microsoft Azure

If you don't have a Microsoft Azure account, begin by signing up for Microsoft Azure.

### Steps

1. Go to the [Azure subscription page](#) to subscribe to the Azure service.
2. Select a plan and follow the instructions to complete the subscription.

## Create an Azure service principal

Astra Control Service requires a Azure service principal that is assigned the Contributor role. Astra Control Service uses this service principal to facilitate Kubernetes application data management on your behalf.

A service principal is an identity created specifically for use with applications, services, and tools. Assigning a role to the service principal restricts access to specific Azure resources.

Follow the steps below to create a service principal using the Azure CLI. You'll need to save the output in a JSON file and provide it to Astra Control Service later on. [Refer to Azure documentation for more details about using the CLI.](#)

The following steps assume that you have permission to create a service principal and that you have the Microsoft Azure SDK (az command) installed on your machine.

### Requirements

- The service principal must use regular authentication. Certificates aren't supported.
- The service principal must be granted Contributor or Owner access to your Azure subscription.
- The subscription or resource group you choose for scope must contain the AKS clusters and your Azure NetApp Files account.

### Steps

1. Identify the subscription and tenant ID where your AKS clusters reside (these are the clusters that you want to manage in Astra Control Service).

```
az configure --list-defaults
az account list --output table
```

2. Do one of the following, depending on if you use an entire subscription or a resource group:
  - Create the service principal, assign the Contributor role, and specify the scope to the entire subscription where the clusters reside.

```
az ad sp create-for-rbac --name service-principal-name --role
contributor --scopes /subscriptions/SUBSCRIPTION-ID
```

- Create the service principal, assign the Contributor role, and specify the resource group where the clusters reside.

```
az ad sp create-for-rbac --name service-principal-name --role contributor --scopes /subscriptions/SUBSCRIPTION-ID/resourceGroups/RESOURCE-GROUP-ID
```

3. Store the resulting Azure CLI output as a JSON file.

You'll need to provide this file so that Astra Control Service can discover your AKS clusters and manage Kubernetes data management operations. [Learn about managing credentials in Astra Control Service.](#)

4. Optional: Add the subscription ID to the JSON file so that Astra Control Service automatically populates the ID when you select the file.

Otherwise, you'll need to enter the subscription ID in Astra Control Service when prompted.

### Example

```
{
  "appId": "0db3929a-bfb0-4c93-baee-aaf8",
  "displayName": "sp-example-dev-sandbox",
  "name": "http://sp-example-dev-sandbox",
  "password": "mypassword",
  "tenant": "011cdf6c-7512-4805-aaf8-7721afd8ca37",
  "subscriptionId": "99ce999a-8c99-99d9-a9d9-99cce99f99ad"
}
```

5. Optional: Test your service principal. Choose from the following example commands depending on the scope your service principal uses.

### Subscription scope

```
az login --service-principal --username APP-ID-SERVICEPRINCIPAL
--password PASSWORD --tenant TENANT-ID
az group list --subscription SUBSCRIPTION-ID
az aks list --subscription SUBSCRIPTION-ID
az storage container list --account-name STORAGE-ACCOUNT-NAME
```

### Resource group scope

```
az login --service-principal --username APP-ID-SERVICEPRINCIPAL
--password PASSWORD --tenant TENANT-ID
az aks list --subscription SUBSCRIPTION-ID --resource-group RESOURCE-
GROUP-ID
```

## Configure Container Storage Interface (CSI) driver details

To use Azure managed disks with Astra Control Service, you'll need to install the required CSI drivers.

### Enable the CSI driver feature in your Azure subscription

Before you install the CSI drivers, you need to enable the CSI driver feature in your Azure subscription.

#### Steps

1. Open the Azure command line interface.
2. Run the following command to register the driver:

```
az feature register --namespace "Microsoft.ContainerService" --name "EnableAzureDiskFileCSIDriver"
```

3. Run the following command to ensure the change is propagated:

```
az provider register -n Microsoft.ContainerService
```

You should see output similar to the following:

```
{
  "id": "/subscriptions/b200155f-001a-43be-87be-3edde83acef4/providers/Microsoft.Features/providers/Microsoft.ContainerService/features/EnableAzureDiskFileCSIDriver",
  "name": "Microsoft.ContainerService/EnableAzureDiskFileCSIDriver",
  "properties": {
    "state": "Registering"
  },
  "type": "Microsoft.Features/providers/features"
}
```

### Install the Azure managed disk CSI drivers in your Azure Kubernetes Service cluster

You can install the Azure CSI drivers to complete your preparation.

#### Step

1. Go to [the Microsoft CSI driver documentation](#).
2. Follow the instructions to install the required CSI drivers.

### Optional: Configure redundancy for Azure backup buckets

You can configure a more durable redundancy level for Azure backup buckets. By default, the buckets Astra Control Service uses to store Azure Kubernetes Service backups use the Locally Redundant Storage (LRS) redundancy option. To use a more durable redundancy option for Azure buckets, you need to do the following:

## Steps

1. Create an Azure storage account that uses the redundancy level you need using [these instructions](#).
2. Create an Azure container in the new storage account using [these instructions](#).
3. Add the container as a bucket to Astra Control Service. Refer to [Add an additional bucket](#).
4. (Optional) To use the newly created bucket as the default bucket for Azure backups, set it as the default bucket for Azure. Refer to [Change the default bucket](#).

## Register for an Astra Control Service account

To use Astra Control Service, you need an Astra Control Service account that is associated with your NetApp BlueXP account. Complete the Astra Control Service registration process and then, if you don't already have a BlueXP account, sign up to BlueXP to access Astra Control Service.

### Register for an Astra Control account

Before you can log in to Astra Control Service, you need to complete a registration process to obtain an Astra Control Service account.

When you use Astra Control Service, you'll manage your apps from within an account. An account includes users who can view and manage the apps within the account, as well as your billing details.

## Steps

1. [Go to the Astra Control page on BlueXP](#).
2. Select **Signup for the free plan**.
3. Provide the required information in the form.

A few important things to note as you fill out the form:

- Your business name and address must be accurate because we verify them to meet the requirements of Global Trade Compliance.
  - The **Astra Account Name** is the name of your business's Astra Control Service account. You'll see this name in the Astra Control Service user interface. Note that you can create additional accounts (up to 5), if needed.
  - In the **Business Email Address** field, if you have a NetApp BlueXP account, enter the email you use for that account here. If you don't yet have a NetApp BlueXP account, use the email address you enter here when you sign up to BlueXP.
4. Select **Create Account**.

## Sign up to BlueXP

Astra Control Service is integrated within NetApp BlueXP's authentication service. You can log in to NetApp BlueXP using your BlueXP or NetApp Support Site credentials. If you don't already have a NetApp BlueXP or NetApp Support Site account, sign up to BlueXP so you can access Astra Control Service and NetApp's other cloud services. If you already have a BlueXP or NetApp Support Site account and have completed registration, you can access [Astra Control Service](#) directly using your BlueXP or NetApp Support Site credentials.



You can also use single sign-on to log in to BlueXP using credentials from your corporate directory (federated identity). To learn more, go to the [Help Center](#) and then select **Cloud Central sign-in options**.

## Steps

1. Go to [NetApp BlueXP](#).
2. In the top right, select **Get Started**.
3. Select **Sign up**.
4. Fill out the form.

Ensure that the phone number and email address you enter here are the same that you used in the preceding Astra Control registration form.

5. Select **Sign up**.



The email address that you enter in these forms is for your NetApp BlueXP user ID. Use this BlueXP user ID when you sign up for a new Astra Control account, or when an Astra Control admin invites you to an existing Astra Control account.

6. Wait for an email from NetApp BlueXP. The email comes from the address [saas.support@netapp.com](mailto:saas.support@netapp.com), and might take several minutes to arrive. Be sure to check your spam folder.
7. When the email arrives, select the link in the email to verify your email address.

## Result

You now have an active BlueXP user login.

Now that you're registered, you can access Astra Control directly using your BlueXP credentials from <https://astra.netapp.io>.

# Add a cluster to Astra Control Service

After you set up your environment, you're ready to create a Kubernetes cluster and then add it to Astra Control Service. This enables you to use Astra Control Service to protect your applications on the cluster.

Depending on the type of cluster you need to add to Astra Control Service, you need to use different steps to add the cluster.

- [Add a public provider-managed cluster to Astra Control Service](#): Use these steps to add a cluster that has a public IP address and is managed by a cloud provider. You will need the Service Principal account, service account, or user account for the cloud provider.
- [Add a private provider-managed cluster to Astra Control Service](#): Use these steps to add a cluster that has a private IP address and is managed by a cloud provider. You will need the Service Principal account, service account, or user account for the cloud provider.
- [Add a public self-managed cluster to Astra Control Service](#): Use these steps to add a cluster that has a public IP address and is managed by your organization. You will need to create a kubeconfig file for the cluster you want to add.
- [Add a private self-managed cluster to Astra Control Service](#): Use these steps to add a cluster that has a private IP address and is managed by your organization. You will need to create a kubeconfig file for the

cluster you want to add.

## Install Astra Connector for private clusters

Astra Control Service uses Astra Connector to enable communication between Astra Control Service and private clusters. You need to install Astra Connector on private clusters that you want to manage.

Astra Connector supports the following types of private clusters:

- Amazon Elastic Kubernetes Service (EKS)
- Azure Kubernetes Service (AKS)
- Google Kubernetes Engine (GKE)
- Red Hat OpenShift Service on AWS (ROSA)
- ROSA with AWS PrivateLink
- Red Hat OpenShift Container Platform on-premises

### Install Astra Connector

#### About this task

- When you perform these steps, execute these commands against the private cluster that you want to manage with Astra Control Service.
- If you are using a bastion host, issue these commands from the command line of the bastion host.

#### Before you begin

- You need access to the private cluster you want to manage with Astra Control Service.
- You need Kubernetes administrator permissions to install the Astra Connector operator on the cluster.

#### Steps

1. Install the Astra Connector operator on the private cluster you want to manage with Astra Control Service. When you run this command, the namespace `astra-connector-operator` is created and the configuration is applied to the namespace:

```
kubectl apply -f https://github.com/NetApp/astra-connector-operator/releases/download/23.07.0-202310251519/astraconnector_operator.yaml
```

2. Verify that the operator is installed and ready:

```
kubectl get all -n astra-connector-operator
```

3. Get an API token from Astra Control. Refer to the [Astra Automation documentation](#) for instructions.
4. Create the `astra-connector` namespace:

```
kubectl create ns astra-connector
```

5. Create the Astra Connector CR file and name it `astra-connector-cr.yaml`. Update the values in brackets `<>` to match your Astra Control environment and cluster configuration:

- **<ASTRA\_CONTROL\_SERVICE\_URL>**: The web UI URL of Astra Control Service. For example:

```
https://astra.netapp.io
```

- **<ASTRA\_CONTROL\_SERVICE\_API\_TOKEN>**: The Astra Control API token you obtained in the preceding step.
- **<PRIVATE\_AKS\_CLUSTER\_NAME>**: (AKS clusters only) - The cluster name of the private Azure Kubernetes Service cluster. Uncomment and populate this line only if you are adding a private AKS cluster.
- **<ASTRA\_CONTROL\_ACCOUNT\_ID>**: Obtained from the Astra Control web UI. Select the figure icon at the top right of the page and select **API access**.

```
apiVersion: netapp.astraconnector.com/v1
kind: AstraConnector
metadata:
  name: astra-connector
  namespace: astra-connector
spec:
  natssync-client:
    cloud-bridge-url: <ASTRA_CONTROL_SERVICE_URL>
  imageRegistry:
    name: theotw
    secret: ""
  astra:
    token: <ASTRA_CONTROL_SERVICE_API_TOKEN>
    #clusterName: <PRIVATE_AKS_CLUSTER_NAME>
    accountId: <ASTRA_CONTROL_ACCOUNT_ID>
    acceptEULA: yes
```

6. After you populate the `astra-connector-cr.yaml` file with the correct values, apply the CR:

```
kubectl apply -f astra-connector-cr.yaml
```

7. Verify that the Astra Connector is fully deployed:

```
kubectl get all -n astra-connector
```

## 8. Verify that the cluster is registered with Astra Control:

```
kubectl get astraconnector -n astra-connector
```

You should see output similar to the following:

NAME	REGISTERED	ASTRACONNECTORID
STATUS		
astra-connector	true	be475ae5-1511-4eaa-9b9e-712f09b0d065
Registered with Astra		



Make note of the ASTRACONNECTORID; you will need it when you add the cluster to Astra Control.

## What's next?

Now that you've installed Astra Connector, you're ready to add your private cluster to Astra Control Service.

- [Add a private provider-managed cluster to Astra Control Service](#): Use these steps to add a cluster that has a private IP address and is managed by a cloud provider. You will need the Service Principal account, service account, or user account for the cloud provider.
- [Add a private self-managed cluster to Astra Control Service](#): Use these steps to add a cluster that has a private IP address and is managed by your organization. You will need to create a kubeconfig file for the cluster you want to add.

## For more information

- [Add a cluster](#)

## Add a provider-managed cluster

### Add a public provider-managed cluster to Astra Control Service

After you set up your cloud environment, you're ready to create a Kubernetes cluster and then add it to Astra Control Service.

- [Create a Kubernetes cluster](#)
- [Add the cluster to Astra Control Service](#)
- [Change the default storage class](#)

### Create a Kubernetes cluster

If you don't have a cluster yet, you can create one that meets the requirements of one of the following providers:

- [Astra Control Service requirements for Azure Kubernetes Service \(AKS\) with Azure NetApp Files](#)
- [Astra Control Service requirements for Azure Kubernetes Service \(AKS\) with Azure managed disks](#)



- [Astra Control Service requirements for Google Kubernetes Engine \(GKE\)](#)
- [Astra Control Service requirements for Amazon Elastic Kubernetes Service \(EKS\)](#)



Astra Control Service supports AKS clusters that use Azure Active Directory (Azure AD) for authentication and identity management. When you create the cluster, follow the instructions in the [official documentation](#) to configure the cluster to use Azure AD. You'll need to make sure your clusters meet the requirements for AKS-managed Azure AD integration.

### Add the cluster to Astra Control Service

After you log in to Astra Control Service, your first step is to start managing your clusters. Before you add a cluster to Astra Control Service, you'll need to perform specific tasks and make sure the cluster meets certain requirements.

When you manage Azure Kubernetes Service and Google Kubernetes Engine clusters, note that you have two options for Astra Control Provisioner installation and lifecycle management:

- You can use Astra Control Service to automatically manage the lifecycle of Astra Control Provisioner. To do this, make sure that Astra Trident is not installed and Astra Control Provisioner is not enabled on the cluster that you want to manage with Astra Control Service. In this case, Astra Control Service automatically enables Astra Control Provisioner when you begin managing the cluster, and Astra Control Provisioner upgrades are handled automatically.
- You can manage the lifecycle of Astra Control Provisioner yourself. To do this, enable Astra Control Provisioner on the cluster before managing the cluster with Astra Control Service. In this case, Astra Control Service detects that Astra Control Provisioner is already enabled and does not reinstall it or manage Astra Control Provisioner upgrades. Refer to [Enable Astra Control Provisioner](#) for steps enable Astra Control Provisioner.

When you manage Amazon Web Services clusters with Astra Control Service, if you need storage backends that can only be used with Astra Control Provisioner, you need to enable Astra Control Provisioner manually on the cluster before you manage it with Astra Control Service. Refer to [Enable Astra Control Provisioner](#) for steps to enable Astra Control Provisioner.

## Before you begin

### Amazon Web Services

- You should have the JSON file containing the credentials of the IAM user that created the cluster. [Learn how to create an IAM user.](#)
- Astra Control Provisioner is required for Amazon FSx for NetApp ONTAP. If you plan to use Amazon FSx for NetApp ONTAP as a storage backend for your EKS cluster, refer to the Astra Control Provisioner information in the [EKS cluster requirements](#).
- (Optional) If you need to provide `kubectl` command access for a cluster to other IAM users that are not the cluster's creator, refer to the instructions in [How do I provide access to other IAM users and roles after cluster creation in Amazon EKS?](#)
- If you plan to use NetApp Cloud Volumes ONTAP as a storage backend, you need to configure Cloud Volumes ONTAP to work with Amazon Web Services. Refer to the Cloud Volumes ONTAP [setup documentation](#).

### Microsoft Azure

- You should have the JSON file that contains the output from the Azure CLI when you created the service principal. [Learn how to set up a service principal.](#)

You'll also need your Azure subscription ID, if you didn't add it to the JSON file.

- If you plan to use NetApp Cloud Volumes ONTAP as a storage backend, you need to configure Cloud Volumes ONTAP to work with Microsoft Azure. Refer to the Cloud Volumes ONTAP [setup documentation](#).

### Google Cloud

- You should have the service account key file for a service account that has the required permissions. [Learn how to set up a service account.](#)
- If you plan to use NetApp Cloud Volumes ONTAP as a storage backend, you need to configure Cloud Volumes ONTAP to work with Google Cloud. Refer to the Cloud Volumes ONTAP [setup documentation](#).

## Steps

1. (Optional) If you are adding an Amazon EKS cluster or want to manage the installation and upgrades of Astra Control Provisioner yourself, enable Astra Control Provisioner on the cluster. Refer to [Enable Astra Control Provisioner](#) for enablement steps.
2. Open the Astra Control Service web UI in a browser.
3. On the Dashboard, select **Manage Kubernetes cluster**.

Follow the prompts to add the cluster.

4. **Provider:** Select your cloud provider and then either provide the required credentials to create a new cloud instance, or select an existing cloud instance to use.
  - a. **Amazon Web Services:** Provide details about your Amazon Web Services IAM user account by uploading a JSON file or by pasting the contents of that JSON file from your clipboard.

The JSON file should contain the credentials of the IAM user that created the cluster.
  - b. **Microsoft Azure:** Provide details about your Azure service principal by uploading a JSON file or by pasting the contents of that JSON file from your clipboard.

The JSON file should contain the output from the Azure CLI when you created the service principal. It can also include your subscription ID so it's automatically added to Astra. Otherwise, you need to manually enter the ID after providing the JSON.

- c. **Google Cloud Platform:** Provide the service account key file either by uploading the file or by pasting the contents from your clipboard.

Astra Control Service uses the service account to discover clusters running in Google Kubernetes Engine.

- d. **Other:** This tab is for use with self-managed clusters only.

5. **Cloud instance name:** Provide a name for the new cloud instance that will be created when you add this cluster. Learn more about [cloud instances](#).

6. Select **Next**.

Astra Control Service displays a list of clusters that you can choose from.

7. **Cluster:** Select a cluster from the list to add to Astra Control Service.



When you are selecting from the list of clusters, pay careful attention to the **Eligibility** column. If a cluster is "Ineligible" or "Partially eligible", hover over the status to determine if there's an issue with the cluster. For example, it might identify that the cluster doesn't have a worker node.

8. Select **Next**.

9. (Optional) **Storage:** Optionally, select the storage class that you'd like Kubernetes applications deployed to this cluster to use by default.

- a. To select a new default storage class for the cluster, enable the **Assign a new default storage class** check box.
- b. Select a new default storage class from the list.



Each cloud provider storage service displays the following price, performance, and resilience information:

- Cloud Volumes Service for Google Cloud: Price, performance, and resilience information
- Google Persistent Disk: No price, performance, or resilience information available
- Azure NetApp Files: Performance and resilience information
- Azure Managed disks: No price, performance, or resilience information available
- Amazon Elastic Block Store: No price, performance, or resilience information available
- Amazon FSx for NetApp ONTAP: No price, performance, or resilience information available
- NetApp Cloud Volumes ONTAP: No price, performance, or resilience information available

Each storage class can utilize one of the following services:

- [Cloud Volumes Service for Google Cloud](#)

- [Google Persistent Disk](#)
- [Azure NetApp Files](#)
- [Azure managed disks](#)
- [Amazon Elastic Block Store](#)
- [Amazon FSx for NetApp ONTAP](#)
- [NetApp Cloud Volumes ONTAP](#)

Learn more about [storage classes for Amazon Web Services clusters](#), [storage classes for GKE clusters](#), and [storage classes for AKS clusters](#).

10. Select **Next**.
11. **Review & Approve**: Review the configuration details.
12. Select **Add** to add the cluster to Astra Control Service.

### Result

If this is the first cluster that you have added for this cloud provider, Astra Control Service creates an object store for the cloud provider for backups of applications running on eligible clusters. (When you add subsequent clusters for this cloud provider, no further object stores are created.) If you specified a default storage class, Astra Control Service sets the default storage class that you specified. For clusters managed in Amazon Web Services or Google Cloud Platform, Astra Control Service also creates an admin account on the cluster. These actions can take several minutes.

### Change the default storage class

You can change the default storage class for a cluster.

### Change the default storage class using Astra Control

You can change the default storage class for a cluster from within Astra Control. If your cluster uses a previously installed storage backend service, you might not be able to use this method to change the default storage class (the **Set as default** action is not selectable). In this case, you can [Change the default storage class using the command line](#).

### Steps

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

### Change the default storage class using the command line

You can change the default storage class for a cluster using Kubernetes commands. This method works regardless of your cluster's configuration.

### Steps

1. Log in to your Kubernetes cluster.

2. List the storage classes in your cluster:

```
kubectl get storageclass
```

3. Remove the default designation from the default storage class. Replace <SC\_NAME> with the name of the storage class:

```
kubectl patch storageclass <SC_NAME> -p '{"metadata":  
{"annotations":{"storageclass.kubernetes.io/is-default-  
class":"false"}}}'
```

4. Mark a different storage class as default. Replace <SC\_NAME> with the name of the storage class:

```
kubectl patch storageclass <SC_NAME> -p '{"metadata":  
{"annotations":{"storageclass.kubernetes.io/is-default-class":"true"}}}'
```

5. Confirm the new default storage class:

```
kubectl get storageclass
```

## Add a private provider-managed cluster to Astra Control Service

You can use Astra Control Service to manage the following types of private provider-managed clusters:

- Amazon Elastic Kubernetes Service (EKS)
- Azure Kubernetes Service (AKS)
- Google Kubernetes Engine (GKE)
- Red Hat OpenShift Service on AWS (ROSA)
- ROSA with AWS PrivateLink

These instructions assume that you have already created a private cluster and prepared a secure method to remotely access it; for more information about creating and accessing private clusters, refer to the following documentation:

- [Azure documentation for private AKS clusters](#)
- [Azure documentation for private OpenShift clusters](#)
- [Amazon EKS documentation](#)
- [Google Kubernetes Engine \(GKE\) documentation](#)
- [Red Hat OpenShift Service on AWS \(ROSA\) documentation](#)

You need to perform the following tasks to add your private cluster to Astra Control Service:

1. [Install Astra Connector](#)
2. [Set up persistent storage](#)
3. [Add the private provider-managed cluster to Astra Control Service](#)

### **Install Astra Connector**

Before you add a private cluster, you need to install Astra Connector on the cluster so that Astra Control can communicate with it. Refer to [Install Astra Connector for private clusters](#) for instructions.

### **Set up persistent storage**

Configure persistent storage for the cluster. Refer to the Get Started documentation for more information about configuring persistent storage:

- [Set up Microsoft Azure with Azure NetApp Files](#)
- [Set up Microsoft Azure with Azure managed disks](#)
- [Set up Amazon Web Services](#)
- [Set up Google Cloud](#)

### **Add the private provider-managed cluster to Astra Control Service**

You can now add the private cluster to Astra Control Service.

When you manage Azure Kubernetes Service and Google Kubernetes Engine clusters, note that you have two options for Astra Control Provisioner installation and lifecycle management:

- You can use Astra Control Service to automatically manage the lifecycle of Astra Control Provisioner. To do this, make sure that Astra Trident is not installed and Astra Control Provisioner is not enabled on the cluster that you want to manage with Astra Control Service. In this case, Astra Control Service automatically enables Astra Control Provisioner when you begin managing the cluster, and Astra Control Provisioner upgrades are handled automatically.
- You can manage the lifecycle of Astra Control Provisioner yourself. To do this, enable Astra Control Provisioner on the cluster before managing the cluster with Astra Control Service. In this case, Astra Control Service detects that Astra Control Provisioner is already enabled and does not reinstall it or manage Astra Control Provisioner upgrades. Refer to [Enable Astra Control Provisioner](#) for steps enable Astra Control Provisioner.

When you manage Amazon Web Services clusters with Astra Control Service, if you need storage backends that can only be used with Astra Control Provisioner, you need to enable Astra Control Provisioner manually on the cluster before you manage it with Astra Control Service. Refer to [Enable Astra Control Provisioner](#) for steps to enable Astra Control Provisioner.

## Before you begin

### Amazon Web Services

- You should have the JSON file containing the credentials of the IAM user that created the cluster. [Learn how to create an IAM user.](#)
- Astra Control Provisioner is required for Amazon FSx for NetApp ONTAP. If you plan to use Amazon FSx for NetApp ONTAP as a storage backend for your EKS cluster, refer to the Astra Control Provisioner information in the [EKS cluster requirements](#).
- (Optional) If you need to provide `kubectl` command access for a cluster to other IAM users that are not the cluster's creator, refer to the instructions in [How do I provide access to other IAM users and roles after cluster creation in Amazon EKS?](#)
- If you plan to use NetApp Cloud Volumes ONTAP as a storage backend, you need to configure Cloud Volumes ONTAP to work with Amazon Web Services. Refer to the Cloud Volumes ONTAP [setup documentation](#).

### Microsoft Azure

- You should have the JSON file that contains the output from the Azure CLI when you created the service principal. [Learn how to set up a service principal.](#)

You'll also need your Azure subscription ID, if you didn't add it to the JSON file.

- If you plan to use NetApp Cloud Volumes ONTAP as a storage backend, you need to configure Cloud Volumes ONTAP to work with Microsoft Azure. Refer to the Cloud Volumes ONTAP [setup documentation](#).

### Google Cloud

- You should have the service account key file for a service account that has the required permissions. [Learn how to set up a service account.](#)
- If the cluster is private, the [authorized networks](#) must allow the Astra Control Service IP address:

52.188.218.166/32

- If you plan to use NetApp Cloud Volumes ONTAP as a storage backend, you need to configure Cloud Volumes ONTAP to work with Google Cloud. Refer to the Cloud Volumes ONTAP [setup documentation](#).

## Steps

1. (Optional) If you are adding an Amazon EKS cluster or want to manage the installation and upgrades of Astra Control Provisioner yourself, enable Astra Control Provisioner on the cluster. Refer to [Enable Astra Control Provisioner](#) for enablement steps.
2. Open the Astra Control Service web UI in a browser.
3. On the Dashboard, select **Manage Kubernetes cluster**.

Follow the prompts to add the cluster.

4. **Provider:** Select your cloud provider and then either provide the required credentials to create a new cloud instance, or select an existing cloud instance to use.
  - a. **Amazon Web Services:** Provide details about your Amazon Web Services IAM user account by uploading a JSON file or by pasting the contents of that JSON file from your clipboard.

The JSON file should contain the credentials of the IAM user that created the cluster.

- b. **Microsoft Azure:** Provide details about your Azure service principal by uploading a JSON file or by pasting the contents of that JSON file from your clipboard.

The JSON file should contain the output from the Azure CLI when you created the service principal. It can also include your subscription ID so it's automatically added to Astra. Otherwise, you need to manually enter the ID after providing the JSON.

- c. **Google Cloud Platform:** Provide the service account key file either by uploading the file or by pasting the contents from your clipboard.

Astra Control Service uses the service account to discover clusters running in Google Kubernetes Engine.

- d. **Other:** This tab is for use with self-managed clusters only.

5. **Cloud instance name:** Provide a name for the new cloud instance that will be created when you add this cluster. Learn more about [cloud instances](#).
6. Select **Next**.

Astra Control Service displays a list of clusters that you can choose from.

7. **Cluster:** Select a cluster from the list to add to Astra Control Service.



When you are selecting from the list of clusters, pay careful attention to the **Eligibility** column. If a cluster is "Ineligible" or "Partially eligible", hover over the status to determine if there's an issue with the cluster. For example, it might identify that the cluster doesn't have a worker node.

1. Select **Next**.
2. (Optional) **Storage:** Optionally, select the storage class that you'd like Kubernetes applications deployed to this cluster to use by default.
  - a. To select a new default storage class for the cluster, enable the **Assign a new default storage class** check box.
  - b. Select a new default storage class from the list.



Each cloud provider storage service displays the following price, performance, and resilience information:



- Cloud Volumes Service for Google Cloud: Price, performance, and resilience information
- Google Persistent Disk: No price, performance, or resilience information available
- Azure NetApp Files: Performance and resilience information
- Azure Managed disks: No price, performance, or resilience information available
- Amazon Elastic Block Store: No price, performance, or resilience information available
- Amazon FSx for NetApp ONTAP: No price, performance, or resilience information available
- NetApp Cloud Volumes ONTAP: No price, performance, or resilience information available

Each storage class can utilize one of the following services:

- [Cloud Volumes Service for Google Cloud](#)
- [Google Persistent Disk](#)
- [Azure NetApp Files](#)
- [Azure managed disks](#)
- [Amazon Elastic Block Store](#)
- [Amazon FSx for NetApp ONTAP](#)
- [NetApp Cloud Volumes ONTAP](#)

Learn more about [storage classes for Amazon Web Services clusters](#), [storage classes for GKE clusters](#), and [storage classes for AKS clusters](#).

3. Select **Next**.
4. **Review & Approve**: Review the configuration details.
5. Select **Add** to add the cluster to Astra Control Service.

## Result

If this is the first cluster that you have added for this cloud provider, Astra Control Service creates an object store for the cloud provider for backups of applications running on eligible clusters. (When you add subsequent clusters for this cloud provider, no further object stores are created.) If you specified a default storage class, Astra Control Service sets the default storage class that you specified. For clusters managed in Amazon Web Services or Google Cloud Platform, Astra Control Service also creates an admin account on the cluster. These actions can take several minutes.

## Change the default storage class

You can change the default storage class for a cluster.

## Change the default storage class using Astra Control

You can change the default storage class for a cluster from within Astra Control. If your cluster uses a

previously installed storage backend service, you might not be able to use this method to change the default storage class (the **Set as default** action is not selectable). In this case, you can [Change the default storage class using the command line](#).

### Steps

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

### Change the default storage class using the command line

You can change the default storage class for a cluster using Kubernetes commands. This method works regardless of your cluster's configuration.

### Steps

1. Log in to your Kubernetes cluster.
2. List the storage classes in your cluster:

```
kubectl get storageclass
```

3. Remove the default designation from the default storage class. Replace <SC\_NAME> with the name of the storage class:

```
kubectl patch storageclass <SC_NAME> -p '{"metadata":  
{"annotations":{"storageclass.kubernetes.io/is-default-  
class":"false"}}}'
```

4. Mark a different storage class as default. Replace <SC\_NAME> with the name of the storage class:

```
kubectl patch storageclass <SC_NAME> -p '{"metadata":  
{"annotations":{"storageclass.kubernetes.io/is-default-class":"true"}}}'
```

5. Confirm the new default storage class:

```
kubectl get storageclass
```

## Add a self-managed cluster

## Add a public self-managed cluster to Astra Control Service

After you set up your environment, you're ready to create a Kubernetes cluster and then add it to Astra Control Service.

A self-managed cluster is a cluster that you directly provision and manage. Astra Control Service supports self-managed clusters that run in a public cloud environment. You can add a self-managed cluster to Astra Control Service by uploading a `kubeconfig.yaml` file. You'll need to ensure the cluster meets the requirements outlined here.

### Supported Kubernetes distributions

You can use Astra Control Service to manage the following types of public, self-managed clusters:

Kubernetes distribution	Supported versions
Kubernetes (Upstream)	1.27 to 1.29
Rancher Kubernetes Engine (RKE)	RKE 1: Versions 1.24.17, 1.25.13, 1.26.8 with Rancher Manager 2.7.9 RKE 2: Versions 1.23.16 and 1.24.13 with Rancher Manager 2.6.13 RKE 2: Versions 1.24.17, 1.25.14, 1.26.9 with Rancher Manager 2.7.9
Red Hat OpenShift Container Platform	4.12 through 4.14

These instructions assume that you have already created a self-managed cluster.

- [Add the cluster to Astra Control Service](#)
- [Change the default storage class](#)

### Add the cluster to Astra Control Service

After you log in to Astra Control Service, your first step is to start managing your clusters. Before you add a cluster to Astra Control Service, you'll need to perform specific tasks and make sure the cluster meets certain requirements.

## Before you begin

A self-managed cluster is a cluster that you directly provision and manage. Astra Control Service supports self-managed clusters that run in a public cloud environment. Your self-managed clusters can use Astra Control Provisioner to interface with NetApp storage services, or they can use Container Storage Interface (CSI) drivers to interface with Amazon Elastic Block Store (EBS), Azure Managed Disks, and Google Persistent Disk.

Astra Control Service supports self-managed clusters that use the following Kubernetes distributions:

- Red Hat OpenShift Container Platform
- Rancher Kubernetes Engine
- Upstream Kubernetes

Your self-managed cluster needs to meet the following requirements:

- The cluster must be accessible via the internet.
- If you are using or plan to use storage enabled with CSI drivers, the appropriate CSI drivers must be installed on the cluster. For more information on using CSI drivers to integrate storage, refer to the documentation for your storage service.
- You have access to the cluster kubeconfig file that includes only one context element. Follow [these instructions](#) to generate a kubeconfig file.
- If you are adding the cluster using a kubeconfig file that references a private Certificate Authority (CA), add the following line to the `cluster` section of the kubeconfig file. This enables Astra Control to add the cluster:

```
insecure-skip-tls-verify: true
```

- **Rancher only:** When managing application clusters in a Rancher environment, modify the application cluster's default context in the kubeconfig file provided by Rancher to use a control plane context instead of the Rancher API server context. This reduces load on the Rancher API server and improves performance.
- **Astra Control Provisioner requirements:** You should have a properly configured Astra Control Provisioner, including its Astra Trident components, to manage clusters.
  - **Review Astra Trident environment requirements:** Prior to installing or upgrading Astra Control Provisioner, review the [supported frontends, backends, and host configurations](#).
  - **Enable Astra Control Provisioner functionality:** It's highly recommended that you install Astra Trident 23.10 or later and enable [Astra Control Provisioner advanced storage functionality](#). In coming releases, Astra Control will not support Astra Trident if the Astra Control Provisioner is not also enabled.
  - **Configure a storage backend:** At least one storage backend must be [configured in Astra Trident](#) on the cluster.
  - **Configure a storage class:** At least one storage class must be [configured in Astra Trident](#) on the cluster. If a default storage class is configured, ensure that it is the **only** storage class that has the default annotation.
  - **Configure a volume snapshot controller and install a volume snapshot class:** [Install a volume snapshot controller](#) so that snapshots can be created in Astra Control. [Create](#) at least one

## Steps

1. On the Dashboard, select **Manage Kubernetes cluster**.

Follow the prompts to add the cluster.

2. **Provider:** Select the **Other** tab to add details about your self-managed cluster.
  - a. **Other:** Provide details about your self-managed cluster by uploading a `kubeconfig.yaml` file or by pasting the contents of the `kubeconfig.yaml` file from your clipboard.



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

3. **Credential name:** Provide a name for the self-managed cluster credential you are uploading to Astra Control. By default, the credential name is auto-populated as the name of the cluster.
4. **Private route identifier:** This field is for use with private clusters only.
5. Select **Next**.
6. (Optional) **Storage:** Optionally, select the storage class that you'd like Kubernetes applications deployed to this cluster to use by default.
  - a. To select a new default storage class for the cluster, enable the **Assign a new default storage class** check box.
  - b. Select a new default storage class from the list.



Each cloud provider storage service displays the following price, performance, and resilience information:

- Cloud Volumes Service for Google Cloud: Price, performance, and resilience information
- Google Persistent Disk: No price, performance, or resilience information available
- Azure NetApp Files: Performance and resilience information
- Azure Managed disks: No price, performance, or resilience information available
- Amazon Elastic Block Store: No price, performance, or resilience information available
- Amazon FSx for NetApp ONTAP: No price, performance, or resilience information available
- NetApp Cloud Volumes ONTAP: No price, performance, or resilience information available

Each storage class can utilize one of the following services:

- [Cloud Volumes Service for Google Cloud](#)
- [Google Persistent Disk](#)
- [Azure NetApp Files](#)
- [Azure managed disks](#)

- [Amazon Elastic Block Store](#)
- [Amazon FSx for NetApp ONTAP](#)
- [NetApp Cloud Volumes ONTAP](#)

Learn more about [storage classes for Amazon Web Services clusters](#), [storage classes for GKE clusters](#), and [storage classes for AKS clusters](#).

7. Select **Next**.
8. **Review & Approve**: Review the configuration details.
9. Select **Add** to add the cluster to Astra Control Service.

### Change the default storage class

You can change the default storage class for a cluster.

### Change the default storage class using Astra Control

You can change the default storage class for a cluster from within Astra Control. If your cluster uses a previously installed storage backend service, you might not be able to use this method to change the default storage class (the **Set as default** action is not selectable). In this case, you can [Change the default storage class using the command line](#).

#### Steps

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

### Change the default storage class using the command line

You can change the default storage class for a cluster using Kubernetes commands. This method works regardless of your cluster's configuration.

#### Steps

1. Log in to your Kubernetes cluster.
2. List the storage classes in your cluster:

```
kubectl get storageclass
```

3. Remove the default designation from the default storage class. Replace <SC\_NAME> with the name of the storage class:

```
kubectl patch storageclass <SC_NAME> -p '{"metadata":  
{"annotations":{"storageclass.kubernetes.io/is-default-  
class":"false"}}}'
```

4. Mark a different storage class as default. Replace <SC\_NAME> with the name of the storage class:

```
kubectl patch storageclass <SC_NAME> -p '{"metadata":  
{"annotations":{"storageclass.kubernetes.io/is-default-class":"true"}}}'
```

5. Confirm the new default storage class:

```
kubectl get storageclass
```

## Add a private self-managed cluster to Astra Control Service

After you set up your environment, you're ready to create a Kubernetes cluster and then add it to Astra Control Service.

A self-managed cluster is a cluster that you directly provision and manage. Astra Control Service supports self-managed clusters that run in a public cloud environment. You can add a self-managed cluster to Astra Control Service by uploading a `kubeconfig.yaml` file. You'll need to ensure the cluster meets the requirements outlined here.

### Supported Kubernetes distributions

You can use Astra Control Service to manage the following types of private, self-managed clusters:

Kubernetes distribution	Supported versions
Kubernetes (Upstream)	1.27 to 1.29
Rancher Kubernetes Engine (RKE)	RKE 1: Versions 1.24.17, 1.25.13, 1.26.8 with Rancher Manager 2.7.9 RKE 2: Versions 1.23.16 and 1.24.13 with Rancher Manager 2.6.13 RKE 2: Versions 1.24.17, 1.25.14, 1.26.9 with Rancher Manager 2.7.9
Red Hat OpenShift Container Platform	4.12 through 4.14

These instructions assume that you have already created a private cluster and prepared a secure method to remotely access it.

You need to perform the following tasks to add your private cluster to Astra Control Service:

1. [Install Astra Connector](#)
2. [Set up persistent storage](#)

### 3. [Add the private self-managed cluster to Astra Control Service](#)

#### **Install Astra Connector**

Before you add a private cluster, you need to install Astra Connector on the cluster so that Astra Control can communicate with it. Refer to [Install Astra Connector for private clusters](#) for instructions.

#### **Set up persistent storage**

Configure persistent storage for the cluster. Refer to the Get Started documentation for more information about configuring persistent storage:

- [Set up Microsoft Azure with Azure NetApp Files](#)
- [Set up Microsoft Azure with Azure managed disks](#)
- [Set up Amazon Web Services](#)
- [Set up Google Cloud](#)

#### **Add the private self-managed cluster to Astra Control Service**

You can now add the private cluster to Astra Control Service.



## Before you begin

A self-managed cluster is a cluster that you directly provision and manage. Astra Control Service supports self-managed clusters that run in a public cloud environment. Your self-managed clusters can use Astra Control Provisioner to interface with NetApp storage services, or they can use Container Storage Interface (CSI) drivers to interface with Amazon Elastic Block Store (EBS), Azure Managed Disks, and Google Persistent Disk.

Astra Control Service supports self-managed clusters that use the following Kubernetes distributions:

- Red Hat OpenShift Container Platform
- Rancher Kubernetes Engine
- Upstream Kubernetes

Your self-managed cluster needs to meet the following requirements:

- The cluster must be accessible via the internet.
- If you are using or plan to use storage enabled with CSI drivers, the appropriate CSI drivers must be installed on the cluster. For more information on using CSI drivers to integrate storage, refer to the documentation for your storage service.
- You have access to the cluster kubeconfig file that includes only one context element. Follow [these instructions](#) to generate a kubeconfig file.
- If you are adding the cluster using a kubeconfig file that references a private Certificate Authority (CA), add the following line to the `cluster` section of the kubeconfig file. This enables Astra Control to add the cluster:

```
insecure-skip-tls-verify: true
```

- **Rancher only:** When managing application clusters in a Rancher environment, modify the application cluster's default context in the kubeconfig file provided by Rancher to use a control plane context instead of the Rancher API server context. This reduces load on the Rancher API server and improves performance.
- **Astra Control Provisioner requirements:** You should have a properly configured Astra Control Provisioner, including its Astra Trident components, to manage clusters.
  - **Review Astra Trident environment requirements:** Prior to installing or upgrading Astra Control Provisioner, review the [supported frontends, backends, and host configurations](#).
  - **Enable Astra Control Provisioner functionality:** It's highly recommended that you install Astra Trident 23.10 or later and enable [Astra Control Provisioner advanced storage functionality](#). In coming releases, Astra Control will not support Astra Trident if the Astra Control Provisioner is not also enabled.
  - **Configure a storage backend:** At least one storage backend must be [configured in Astra Trident](#) on the cluster.
  - **Configure a storage class:** At least one storage class must be [configured in Astra Trident](#) on the cluster. If a default storage class is configured, ensure that it is the **only** storage class that has the default annotation.
  - **Configure a volume snapshot controller and install a volume snapshot class:** [Install a volume snapshot controller](#) so that snapshots can be created in Astra Control. [Create](#) at least one

## Steps

1. On the Dashboard, select **Manage Kubernetes cluster**.

Follow the prompts to add the cluster.

2. **Provider:** Select the **Other** tab to add details about your self-managed cluster.
3. **Other:** Provide details about your self-managed cluster by uploading a `kubeconfig.yaml` file or by pasting the contents of the `kubeconfig.yaml` file from your clipboard.



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

4. **Credential name:** Provide a name for the self-managed cluster credential you are uploading to Astra Control. By default, the credential name is auto-populated as the name of the cluster.
5. **Private route identifier:** Enter the private route identifier, which you can obtain from the Astra Connector. If you query the Astra Connector via the `kubectl get astraconnector -n astra-connector` command, the private route identifier is referred to as the `ASTRACONNECTORID`.



The private route identifier is the name associated with the Astra Connector that enables a private Kubernetes cluster to be managed by Astra. In this context, a private cluster is a Kubernetes cluster that does not expose its API server to the internet.

6. Select **Next**.
7. (Optional) **Storage:** Optionally, select the storage class that you'd like Kubernetes applications deployed to this cluster to use by default.
  - a. To select a new default storage class for the cluster, enable the **Assign a new default storage class** check box.
  - b. Select a new default storage class from the list.



Each cloud provider storage service displays the following price, performance, and resilience information:

- Cloud Volumes Service for Google Cloud: Price, performance, and resilience information
- Google Persistent Disk: No price, performance, or resilience information available
- Azure NetApp Files: Performance and resilience information
- Azure Managed disks: No price, performance, or resilience information available
- Amazon Elastic Block Store: No price, performance, or resilience information available
- Amazon FSx for NetApp ONTAP: No price, performance, or resilience information available
- NetApp Cloud Volumes ONTAP: No price, performance, or resilience information available

Each storage class can utilize one of the following services:

- [Cloud Volumes Service for Google Cloud](#)
- [Google Persistent Disk](#)
- [Azure NetApp Files](#)
- [Azure managed disks](#)
- [Amazon Elastic Block Store](#)
- [Amazon FSx for NetApp ONTAP](#)
- [NetApp Cloud Volumes ONTAP](#)

Learn more about [storage classes for Amazon Web Services clusters](#), [storage classes for GKE clusters](#), and [storage classes for AKS clusters](#).

8. Select **Next**.
9. **Review & Approve**: Review the configuration details.
10. Select **Add** to add the cluster to Astra Control Service.

### Change the default storage class

You can change the default storage class for a cluster.

### Change the default storage class using Astra Control

You can change the default storage class for a cluster from within Astra Control. If your cluster uses a previously installed storage backend service, you might not be able to use this method to change the default storage class (the **Set as default** action is not selectable). In this case, you can [Change the default storage class using the command line](#).

#### Steps

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

### Change the default storage class using the command line

You can change the default storage class for a cluster using Kubernetes commands. This method works regardless of your cluster's configuration.

#### Steps

1. Log in to your Kubernetes cluster.
2. List the storage classes in your cluster:

```
kubectl get storageclass
```

3. Remove the default designation from the default storage class. Replace <SC\_NAME> with the name of the storage class:

```
kubectl patch storageclass <SC_NAME> -p '{"metadata":  
{"annotations":{"storageclass.kubernetes.io/is-default-  
class":"false"}}}'
```

4. Mark a different storage class as default. Replace <SC\_NAME> with the name of the storage class:

```
kubectl patch storageclass <SC_NAME> -p '{"metadata":  
{"annotations":{"storageclass.kubernetes.io/is-default-class":"true"}}}'
```

5. Confirm the new default storage class:

```
kubectl get storageclass
```

### Check the Astra Trident version

To add a self-managed cluster that uses Astra Control Provisioner or Astra Trident for storage services, ensure that the installed version of Astra Trident is 23.10 or latest.

#### Steps

1. Determine the Astra Trident version you are running:

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

If Astra Trident is installed, you see output similar to the following:

NAME	VERSION
trident	24.02.0

If Astra Trident is not installed, you see output similar to the following:

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

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

- If you are running Astra Trident 23.10 or later, verify that Astra Control Provisioner has been [enabled](#). Astra Control Provisioner will not work with releases of Astra Control Center earlier than 23.10. [Upgrade your Astra Control Provisioner](#) so that it has the same version as the Astra Control Center you are upgrading to access the latest functionality.

3. Ensure that the pods are running:

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

4. Check if the storage classes are using the supported Astra Trident drivers. The provisioner name should be `csi.trident.netapp.io`. Refer to the following example:

```
kubectl get sc
```

Sample response:

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

## Create a kubeconfig file

You can add a cluster to Astra Control Service using a kubeconfig file. Depending on the type of cluster you want to add, you might need to manually create a kubeconfig file for your cluster using specific steps.

- [Create a kubeconfig file for Amazon EKS clusters](#)
- [Create a kubeconfig file for Red Hat OpenShift Service on AWS \(ROSA\) clusters](#)
- [Create a kubeconfig file for other types of clusters](#)

### Create a kubeconfig file for Amazon EKS clusters

Follow these instructions to create a kubeconfig file and permanent token secret for Amazon EKS clusters. A permanent token secret is required for clusters hosted in EKS.

#### Steps

1. Follow the instructions in the Amazon documentation to generate a kubeconfig file:

[Creating or updating a kubeconfig file for an Amazon EKS cluster](#)

2. Create a service account as follows:

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

Adjust the service account name as needed. The namespace `kube-system` is required for these steps. If you change the service account name here, you should apply the same changes in the

following steps.

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

```
apiVersion: v1
kind: ServiceAccount
metadata:
  name: astra-admin-account
  namespace: kube-system
```

3. Apply the service account:

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

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

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

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: astra-admin-binding
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: cluster-admin
subjects:
- kind: ServiceAccount
  name: astra-admin-account
  namespace: kube-system
```

5. Apply the cluster role binding:

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

6. Create a service account token secret file called astracontrol-secret.yaml.

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

```
apiVersion: v1
kind: Secret
metadata:
  annotations:
    kubernetes.io/service-account.name: astra-admin-account
  name: astra-admin-account
  namespace: kube-system
type: kubernetes.io/service-account-token
```

7. Apply the token secret:

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

8. Retrieve the token secret:

```
kubectl get secret astra-admin-account -n kube-system -o  
jsonpath='{.data.token}' | base64 -d
```

9. Replace the `user` section of the AWS EKS kubeconfig file with the token, as shown in the following example:

```
user: token: k8s-aws-  
v1.aHR0cHM6Ly9zdHMudXMtd2Vzdc0yLmFtYXpvbmF3cy5jb20vP0FjdGlvbj1HZXRDYWxsZ  
XJJZGVudGl0eSZWZXJzaW9uPTIwMTETMDYtMTUmWC1BbXotQWxnbn3JpdGhtPUFXUzQtSE1BQ  
y1TSEEyNTYmWC1BbXotQ3JlZGVudGlhbD1BS01BM1JEWddkU0haWU9LSEQ2SyUyRjIwMjMwN  
DAzJTJGdXMtd2Vzdc0yJTJGc3RzJTJGYXdzNF9yZXFlZlZlZW0JlgtQW16LURhdGU9MjAyMzA0M  
DNUMjA0MzQwWiZYLUFtei1FeHBpcmVzPTYwJlgtQW16LVNpZ25lZEhlYWRLcnM9aG9zdCUzQ  
ngtazhzLWF3cy1pZCZYLUFtei1TaWduYXRlcMU9YjU4ZWM0NzdiM2NkZGYxNGRhNmU4MGFI2Z  
WQ2zy2NzI2YWIwM2UyNThjMjRhNTJjNmVhNjc4MTRlNjJkOTg2Mg
```

## Create a kubeconfig file for Red Hat OpenShift Service on AWS (ROSA) clusters

Follow these instructions to create a kubeconfig file for Red Hat OpenShift Service on AWS (ROSA) clusters.

## Steps

1. Log in to the ROSA cluster.
2. Create a service account:

```
oc create sa astracontrol-service-account
```

3. Add a cluster role:

```
oc adm policy add-cluster-role-to-user cluster-admin -z astracontrol-  
service-account
```

4. Using the following example, create a service account secret configuration file:

```
<strong>secret-astra-sa.yaml</strong>
```

```
apiVersion: v1  
kind: Secret  
metadata:  
  name: secret-astracontrol-service-account  
  annotations:  
    kubernetes.io/service-account.name: "astracontrol-service-account"  
type: kubernetes.io/service-account-token
```

5. Create the secret:

```
oc create -f secret-astra-sa.yaml
```

6. Edit the service account that you created, and add the Astra Control service account secret name to the secrets section:

```
oc edit sa astracontrol-service-account
```

```
apiVersion: v1  
imagePullSecrets:  
- name: astracontrol-service-account-dockercfg-dvfcd  
kind: ServiceAccount  
metadata:  
  creationTimestamp: "2023-08-04T04:18:30Z"  
  name: astracontrol-service-account  
  namespace: default  
  resourceVersion: "169770"  
  uid: 965fa151-923f-4fbd-9289-30cad15998ac  
secrets:  
- name: astracontrol-service-account-dockercfg-dvfcd  
- name: secret-astracontrol-service-account ####ADD THIS ONLY####
```



7. 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-dvfcfcd"},  
  { "name": "secret-astracontrol-service-account"}  
]
```

The indices for each element in the `secrets` array begin with 0. In the above example, the index for `astracontrol-service-account-dockercfg-dvfcfcd` would be 0 and the index for `secret-astracontrol-service-account` would be 1. In your output, make note of the index number for the service account secret. You will need this index number in the next step.

8. Generate the kubeconfig as follows:

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

**create-kubeconfig.sh**

```
# Update these to match your environment.  
# Replace TOKEN_INDEX with the correct value  
# from the output in the previous step. If you  
# didn't change anything else above, don't change  
# anything else here.  
  
SERVICE_ACCOUNT_NAME=astracontrol-service-account  
NAMESPACE=default  
NEW_CONTEXT=astracontrol  
KUBECONFIG_FILE='kubeconfig-sa'  
  
CONTEXT=$(kubectl config current-context)  
  
SECRET_NAME=$(kubectl get serviceaccount ${SERVICE_ACCOUNT_NAME} \  
  --context ${CONTEXT} \  
  --namespace ${NAMESPACE} \  
  -o jsonpath='{.secrets[TOKEN_INDEX].name}')
```

```
TOKEN_DATA=$(kubectl get secret ${SECRET_NAME} \  
  --context ${CONTEXT} \  
  --namespace ${NAMESPACE} \  
  -o jsonpath='{.data}'
```

```

-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
```

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

```
mv kubeconfig-sa YOUR_CLUSTER_NAME_kubeconfig
```

### Create a kubeconfig file for other types of clusters

Follow these instructions to create a limited or expanded role kubeconfig file for Rancher, Upstream Kubernetes, and Red Hat OpenShift clusters.

For clusters that are managed using kubeconfig, you can optionally create a limited permission or expanded permission administrator role for Astra Control Service.

This procedure helps you to create a separate kubeconfig if either of the following scenarios applies to your environment:

- You want to limit Astra Control permissions on the clusters it manages
- You use multiple contexts and cannot use the default Astra Control kubeconfig configured during installation or a limited role with a single context won't work in your environment

### Before you begin

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

- A [supported version](#) of kubectl is installed.
- kubectl access to the cluster that you intend to add and manage with Astra Control Service



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

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

### Steps

1. Create a service account:

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

```
<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. Create one of the following cluster roles with sufficient permissions for a cluster to be managed by Astra Control:

### Limited cluster role

This role contains the minimum permissions necessary for a cluster to be managed by Astra Control:

1. Create a ClusterRole file called, for example, `astra-admin-account.yaml`.

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

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

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

# Delete Resources
# Necessary for in-place restore and AppMirror failover
- apiGroups:
  - ""
  - apps
  - autoscaling
  - batch
  - crd.projectcalico.org
  - extensions
  - networking.k8s.io
  - policy
  - rbac.authorization.k8s.io
  - snapshot.storage.k8s.io
  - trident.netapp.io
  resources:
  - configmaps
  - cronjobs
  - daemonsets
  - deployments
```

```

- horizontalpodautoscalers
- ingresses
- jobs
- namespaces
- networkpolicies
- persistentvolumeclaims
- poddisruptionbudgets
- pods
- podtemplates
- replicaset
- replicationcontrollers
- replicationcontrollers/scale
- rolebindings
- roles
- secrets
- serviceaccounts
- services
- statefulsets
- tridentmirrorrelationships
- tridentnapshotinfos
- volumesnapshots
- volumesnapshotcontents
verbs:
- delete

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

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

```

```
- imagestreamtags
- imagetags
verbs:
- update
```

2. (For OpenShift clusters only) Append the following at the end of the `astra-admin-account.yaml` file:

```
# OpenShift security
- apiGroups:
  - security.openshift.io
  resources:
  - securitycontextconstraints
  verbs:
  - use
  - update
```

3. Apply the cluster role:

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

### Expanded cluster role

This role contains expanded permissions for a cluster to be managed by Astra Control. You might use this role if you use multiple contexts and cannot use the default Astra Control kubeconfig configured during installation or a limited role with a single context won't work in your environment:



The following `ClusterRole` steps are a general Kubernetes example. Refer to the documentation for your Kubernetes distribution for instructions specific to your environment.

1. Create a `ClusterRole` file called, for example, `astra-admin-account.yaml`.

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

```

apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: astra-admin-account
rules:
- apiGroups:
  - '*'
  resources:
  - '*'
  verbs:
  - '*'
- nonResourceURLs:
  - '*'
  verbs:
  - '*'

```

2. Apply the cluster role:

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

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

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

```
<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: astra-admin-account
subjects:
- kind: ServiceAccount
  name: astracontrol-service-account
  namespace: default

```

b. Apply the cluster role binding:



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

4. Create and apply the token secret:

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

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

```
apiVersion: v1
kind: Secret
metadata:
  name: secret-astracontrol-service-account
  namespace: default
  annotations:
    kubernetes.io/service-account.name: "astracontrol-service-
account"
type: kubernetes.io/service-account-token
```

- b. Apply the token secret:

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

5. Add the token secret to the service account by adding its name to the `secrets` array (the last line in the following example):

```
kubectl edit sa astracontrol-service-account
```

```

apiVersion: v1
imagePullSecrets:
- name: astracontrol-service-account-dockercfg-48xhx
kind: ServiceAccount
metadata:
  annotations:
    kubectl.kubernetes.io/last-applied-configuration: |

{"apiVersion":"v1","kind":"ServiceAccount","metadata":{"annotations":{},"name":"astracontrol-service-account","namespace":"default"},"creationTimestamp":"2023-06-14T15:25:45Z","name":"astracontrol-service-account","namespace":"default","resourceVersion":"2767069","uid":"2ce068c4-810e-4a96-ada3-49cbf9ec3f89"}
secrets:
- name: astracontrol-service-account-dockercfg-48xhx
<strong>- name: secret-astracontrol-service-account</strong>

```

6. 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-48xhx"},
{ "name": "secret-astracontrol-service-account"}
]

```

The indices for each element in the `secrets` array begin with 0. In the above example, the index for `astracontrol-service-account-dockercfg-48xhx` would be 0 and the index for `secret-astracontrol-service-account` would be 1. In your output, make note of the index number for the service account secret. You'll need this index number in the next step.

7. Generate the kubeconfig as follows:

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

```

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

```

```

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

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

CONTEXT=$(kubectl config current-context)

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

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

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

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

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

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

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

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

```

```

set-context ${NEW_CONTEXT} --user ${CONTEXT}-${NAMESPACE}-token-
user

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

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

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

```

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

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

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

```
mv kubeconfig-sa YOUR_CLUSTER_NAME_kubeconfig
```

## What's next?

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

- [Start managing apps](#)
- [Protect apps](#)
- [Clone apps](#)
- [Set up billing](#)
- [Invite and manage users](#)
- [Manage cloud provider credentials](#)
- [Manage notifications](#)
- [Deploy a self-managed instance of Astra Control](#)

## Astra Control Service videos

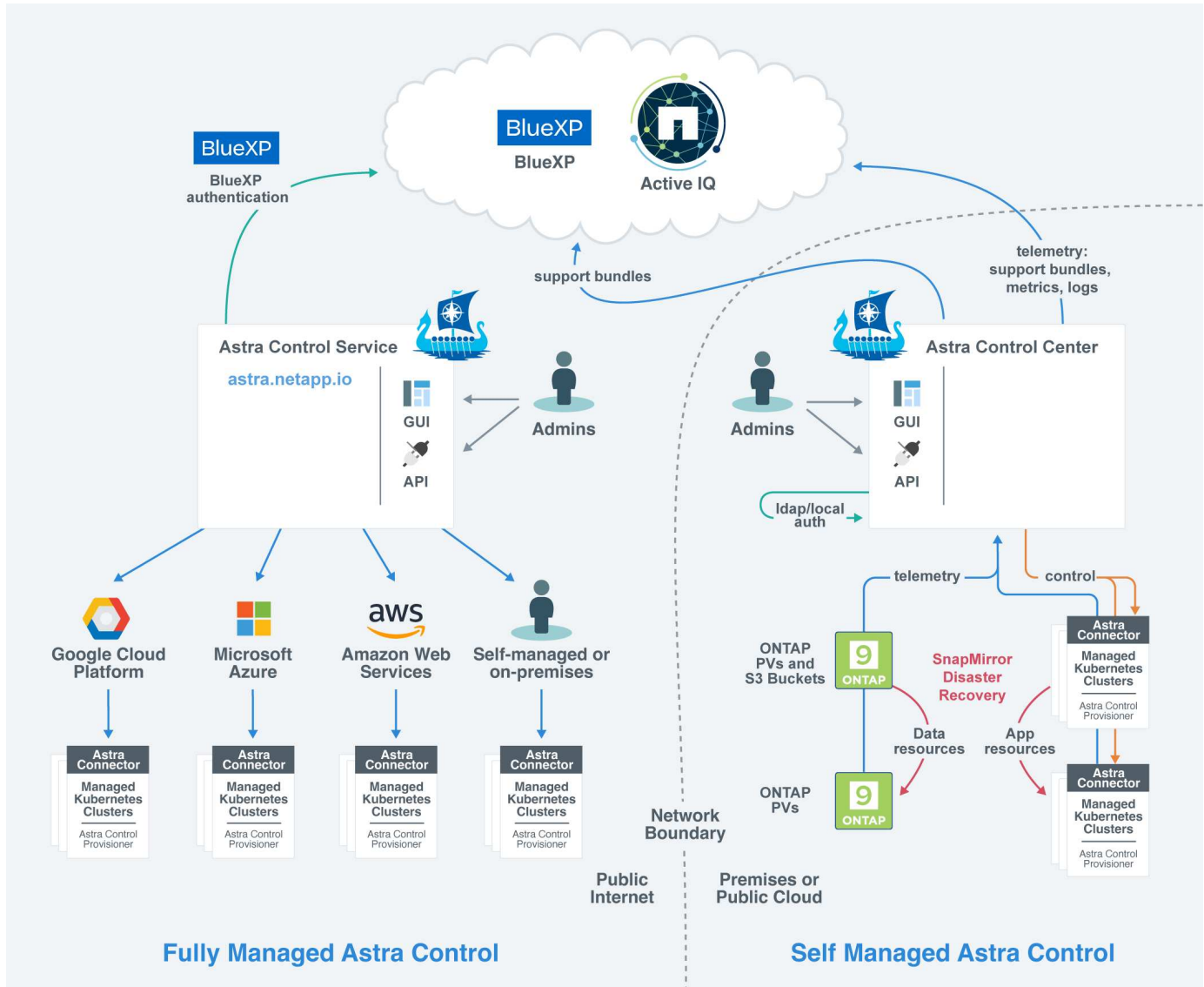
Check out NetApp TV for the latest video content featuring Astra Control Service. NetApp TV includes videos that demonstrate certain features of Astra Control Service or show you how to complete certain common tasks.



# Concepts

## Architecture and components

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



## Astra Control components

- **Kubernetes clusters:** Kubernetes is a portable, extensible, open-source platform for managing containerized workloads and services, that facilitates both declarative configuration and automation. Astra provides management services for applications hosted in a Kubernetes cluster.
- **Astra Control Provisioner:** As a fully supported storage provisioner and orchestrator maintained by NetApp that contains Astra Trident CSI functionality along with extended storage management capabilities, Astra Control Provisioner enables you to create storage volumes for containerized applications managed by Docker and Kubernetes. When deployed with Astra Control, Astra Control Provisioner includes a configured ONTAP storage backend.

## 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 Center documentation](#)
- [Astra Control API documentation](#)
- [Astra Trident documentation](#)
- [ONTAP documentation](#)

## Data protection

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

### Snapshots, backups, and protection policies

Both snapshots and backups protect the following types of data:

- The application itself
- Any persistent data volumes associated with the application
- Any resource artifacts belonging to the application

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

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

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



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



If you perform a snapshot or backup, but the operation fails with the error "The resource wasn't created because of an internal server issue", check to make sure the storage backend you are using has the correct drivers installed. Some storage backends need Container Storage Interface (CSI) drivers, while others need an external snapshot controller.

## Immutable backups

An immutable backup is a backup that cannot be changed or deleted during a specified period. When you create an immutable backup, Astra Control checks to ensure that the bucket you are using is a write once read many (WORM) bucket, and if so, ensures that the backup is immutable from within Astra Control.

Astra Control Service supports creating immutable backups with the following platforms and bucket types:

- Amazon Web Services using an Amazon S3 bucket with S3 Object Lock configured
- Microsoft Azure using an Azure bucket with a retention policy configured
- Google Kubernetes Engine (GKE) using a Google Cloud Storage bucket with a retention policy configured
- NetApp StorageGRID using an S3 bucket with S3 Object Lock configured

Note the following when working with immutable backups:

- If you back up to a WORM bucket in an unsupported platform or to an unsupported bucket type, you might get unpredictable results, such as backup deletion failing even if the retention time has elapsed.
- Astra Control does not support data lifecycle management policies or manual deletion of objects on the buckets you use with immutable backups. Make sure that your storage backend is not configured to manage the lifecycle of Astra Control snapshots or backed up data.

## Clones

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

## Storage classes and performance for AWS clusters

Astra Control Service can use Amazon Elastic Block Store (EBS), Amazon FSx for NetApp ONTAP, or NetApp Cloud Volumes ONTAP as the storage backend for Amazon Elastic Kubernetes Service (EKS) clusters.

### Amazon Elastic Block Store (EBS)

Your clusters can use Container Storage Interface (CSI) drivers to interface with EBS. When you use EBS as the storage backend for EKS clusters, you can configure some storage class parameters. For more information about what the parameters mean and how to configure them, refer to [the Kubernetes documentation](#).

You can use several different types of volumes with EBS:



- Solid state drives (SSD)
- Hard disk drives (HDD)
- Previous generation

For more information on each type of volume and their performance, refer to [the Amazon EBS documentation](#). For pricing information, refer to [Amazon EBS pricing](#).

## Amazon FSx for NetApp ONTAP

When you use FSx for NetApp ONTAP as the storage backend for AWS clusters, I/O performance depends on the configuration of the filesystem and the characteristics of your workloads. For specific information on FSx for NetApp ONTAP performance, refer to [Amazon FSx for NetApp ONTAP performance](#). For pricing information, refer to [Amazon FSx for NetApp ONTAP Pricing](#).

## NetApp Cloud Volumes ONTAP

For specific information on configuring NetApp Cloud Volumes ONTAP, including performance recommendations, visit the [NetApp Cloud Volumes ONTAP documentation](#).

## Storage classes and PV size for AKS clusters

Astra Control Service supports Azure NetApp Files, Azure managed disks, or NetApp Cloud Volumes ONTAP as the storage backend for Azure Kubernetes Service (AKS) clusters.

### Azure NetApp Files

Astra Control Service supports Azure NetApp Files as the storage backend for Azure Kubernetes Service (AKS) clusters. You should understand how choosing a storage class and persistent volume size can help you meet your performance objectives.

#### Service levels and storage classes

Azure NetApp Files supports three service levels: Ultra storage, Premium storage, and Standard storage. Each of these service levels are designed for different performance needs:

##### Ultra storage

Provides up to 128 MiB/s of throughput per 1 TiB.

##### Premium storage

Provides up to 64 MiB/s of throughput per 1 TiB.

##### Standard storage

Provides up to 16 Mib/s of throughput per 1 TiB.

These service levels are an attribute of a capacity pool. You need to set up a capacity pool for each service level that you want to use with your Kubernetes clusters. [Learn how to set up capacity pools](#).

Astra Control Service uses these service levels as storage classes for your persistent volumes. When you add Kubernetes clusters to Astra Control Service, you're prompted to choose either Ultra, Premium, or Standard as the default storage class. The names of the storage classes are *netapp-anf-perf-ultra*, *netapp-anf-perf-*

*premium*, and *netapp-anf-perf-standard*.

[Learn more about these service levels in the Azure NetApp Files docs.](#)

## Persistent volume size and performance

As described above, the throughput for each service level is per 1 TiB of provisioned capacity. That means larger volumes provide better performance. So you should take both capacity and performance needs into consideration when provisioning volumes.

## Minimum volume size

Astra Control Service provisions persistent volumes using a minimum volume size of 100 GiB, even if the PVC asks for a smaller volume size. For example, if the PVC in a Helm chart asks for 6 GiB, Astra Control Service automatically provisions a 100 GiB volume.

## Application backups

If you back up an application that resides on Azure NetApp Files storage, Astra Control Service automatically temporarily expands the capacity pool. After the backup is complete, Astra Control Service shrinks the capacity pool to its previous size. Depending on your Azure subscription, you might incur storage charges when this happens. You can see a history of capacity pool resize events in the **Activity** page event log.

If the capacity pool exceeds the maximum size allowed by the Azure subscription during the resize operation, the backup operation fails, and a warning is triggered from the Azure API.

## Azure managed disks

Astra Control Service can use Container Storage Interface (CSI) drivers to interface with Azure Managed Disks as a storage backend. This service provides block-level storage that is managed by Azure.

[Learn more about Azure managed disks.](#)

## NetApp Cloud Volumes ONTAP

For specific information on configuring NetApp Cloud Volumes ONTAP, including performance recommendations, visit the [NetApp Cloud Volumes ONTAP documentation](#).

# Service type, storage classes, and PV size for GKE clusters

Astra Control Service supports NetApp Cloud Volumes Service for Google Cloud, Google Persistent Disk, or NetApp Cloud Volumes ONTAP as the storage backend options for persistent volumes.

## Cloud Volumes Service for Google Cloud

Astra Control Service can use Cloud Volumes Service for Google Cloud as the storage backend for persistent volumes. You should understand how choosing a service type, storage class, and persistent volume size can help you meet your performance objectives.

## Overview

Cloud Volumes Service for Google Cloud provides two service types: *CVS* and *CVS-Performance*. These service types are supported in specific Google Cloud regions. [Go to NetApp BlueXP Global Regions Maps](#) to identify the service type that's supported in the Google Cloud region where your clusters reside.

If your Kubernetes clusters must reside in a specific region, then you'll be using the service type supported in that region.

But if you have the flexibility to choose between Google Cloud regions, then we recommend the following based on your performance requirements:

- For K8s applications that have medium-to-high performance storage needs, choose a Google Cloud region that supports CVS-Performance and use the Premium or Extreme storage class. Such workloads include AI/ML pipelines, CI/CD pipelines, media processing, and databases including relational, noSQL, time series, etc.
- For K8s applications that have low-to-medium storage performance needs (web apps, general purpose file storage, etc.), choose a Google Cloud region that supports either CVS or CVS-Performance, with the Standard storage class.



If you use the CVS service type with Astra Control Provisioner, you need to configure storage pools before you can provision volumes. If you provision volumes with no storage pools configured, volume provisioning will fail. Refer to the [Cloud Volumes Service documentation](#) for more information about creating volumes.

The following table provides a quick comparison of the information described on this page.

Service type	Use case	Supported regions	Storage classes	Min volume size
CVS-Performance	Apps with medium-to-high storage performance needs	<a href="#">View supported Google Cloud regions</a>	<ul style="list-style-type: none"><li>• netapp-cvs-perf-standard</li><li>• netapp-cvs-perf-premium</li><li>• netapp-cvs-perf-extreme</li></ul>	100 GiB
CVS	Apps with low-to-medium storage performance needs	<a href="#">View supported Google Cloud regions</a>	netapp-cvs-standard	300 GiB

## CVS-Performance service type

Learn more about the CVS-Performance service type before you choose a storage class and create persistent volumes.

### Storage classes

Three service levels are supported with the CVS-Performance service type: Standard, Premium, and Extreme. When you add a cluster to Astra Control Service, you're prompted to choose either Standard, Premium, or Extreme as the default storage class for persistent volumes. Each of these service levels are designed for different capacity and bandwidth needs.

The names of the storage classes are *netapp-cvs-perf-standard*, *netapp-cvs-perf-premium*, and *netapp-cvs-*

*perf-extreme*.

[Learn more about these service levels in the Cloud Volumes Service for Google Cloud docs.](#)

#### **Persistent volume size and performance**

[As the Google Cloud docs explain](#), the allowed bandwidth for each service level is per GiB of provisioned capacity. That means larger volumes will provide better performance.

Be sure to read through the Google Cloud page linked to above. It includes cost comparisons and examples that can help you better understand how to couple a service level with volume size to meet your performance objectives.

#### **Minimum volume size**

Astra Control Service provisions persistent volumes using a minimum volume size of 100 GiB with the CVS-Performance service type, even if the PVC requests a smaller volume size. For example, if the PVC in a Helm chart asks for 6 GiB, Astra Control Service automatically provisions a 100 GiB volume.

#### **CVS service type**

Learn more about the CVS service type before you choose a storage class and create persistent volumes.

#### **Storage class**

One service level is supported with the CVS service type: Standard. When you manage clusters in regions where the CVS service type is supported, Astra Control Service uses the Standard service level as the default storage class for persistent volumes. The storage class is named *netapp-cvs-standard*.

[Learn more about the Standard service level in the Cloud Volumes Service for Google Cloud docs.](#)

#### **Persistent volume size and performance**

The allowed bandwidth for the CVS service type is per GiB of provisioned capacity. That means larger volumes will provide better performance.

#### **Minimum volume size**

Astra Control Service provisions persistent volumes using a minimum volume size of 300 GiB with the CVS service type, even if the PVC asks for a smaller volume size. For example, if 20 GiB is requested, Astra Control Service automatically provisions a 300 GiB volume.

Due to a limitation, if a PVC requests a volume between 700-999 GiB, Astra Control Service automatically provisions a volume size of 1000 GiB.

## **Google Persistent Disk**

Astra Control Service can use Container Storage Interface (CSI) drivers to interface with Google Persistent Disk as a storage backend. This service provides block-level storage that is managed by Google.

[Learn more about Google Persistent Disk.](#)

[Learn more about different performance levels of Google Persistent Disks.](#)

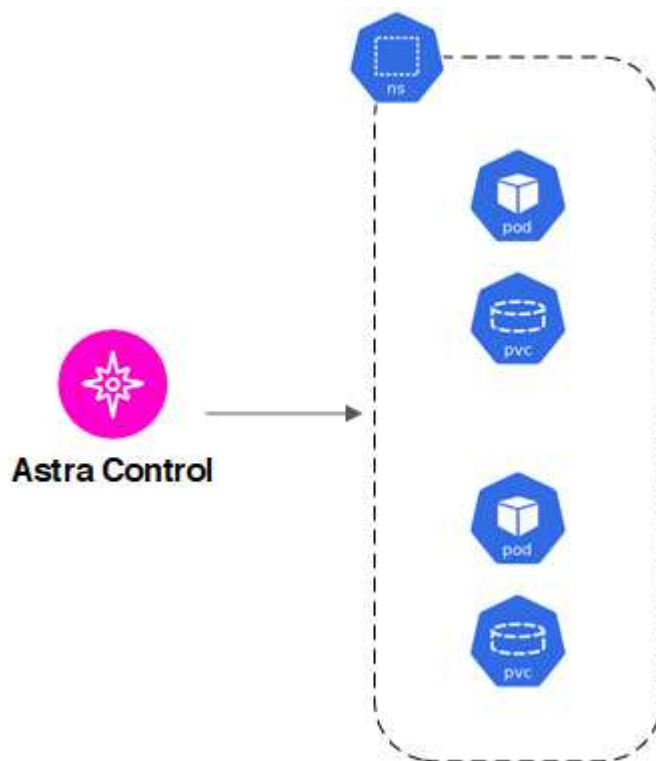
## NetApp Cloud Volumes ONTAP

For specific information on configuring NetApp Cloud Volumes ONTAP, including performance recommendations, visit the [NetApp Cloud Volumes ONTAP documentation](#).

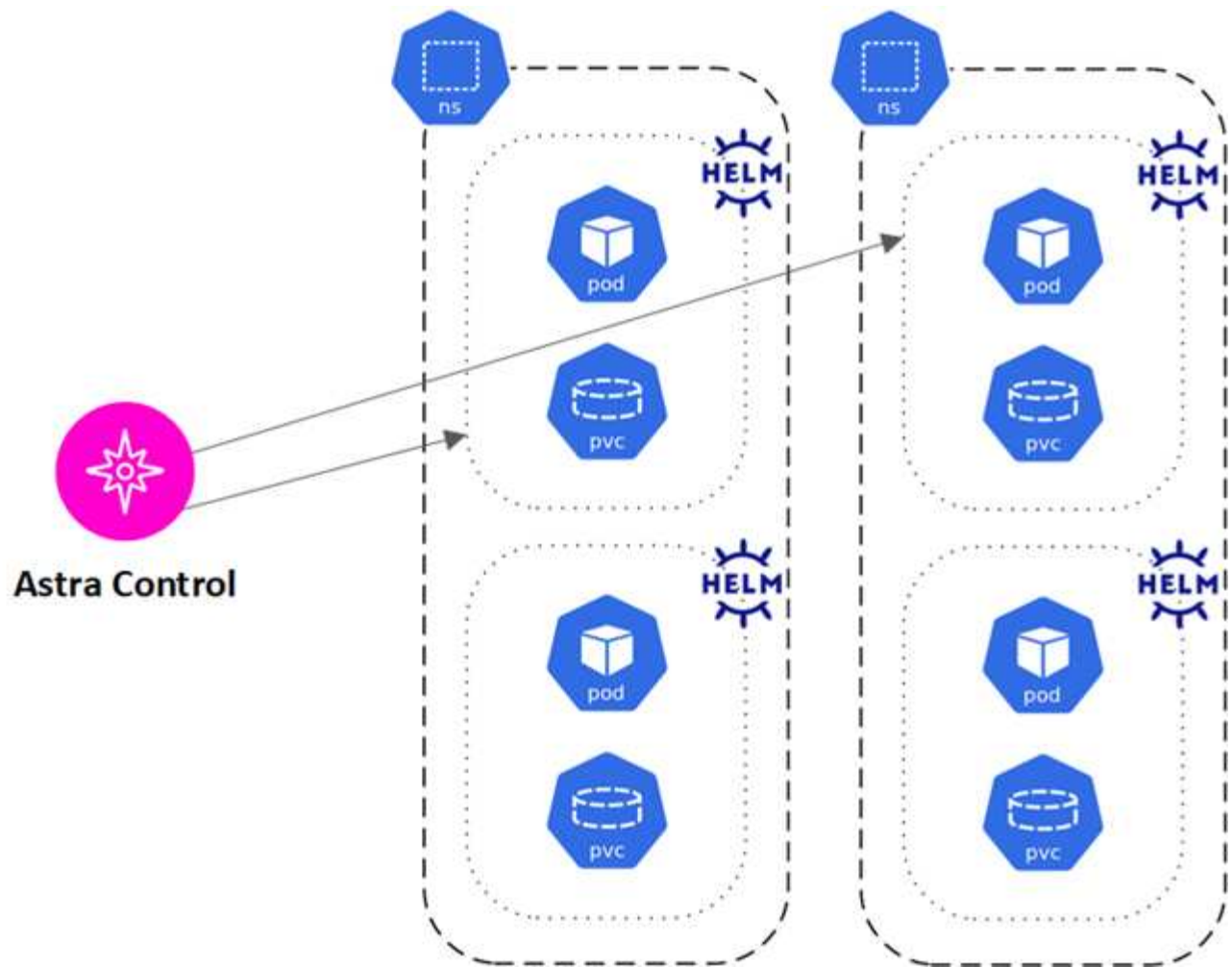
## App management

When Astra Control discovers your clusters, the apps on those clusters are unmanaged until you choose how you want to manage them. A managed application in Astra Control can be any of the following:

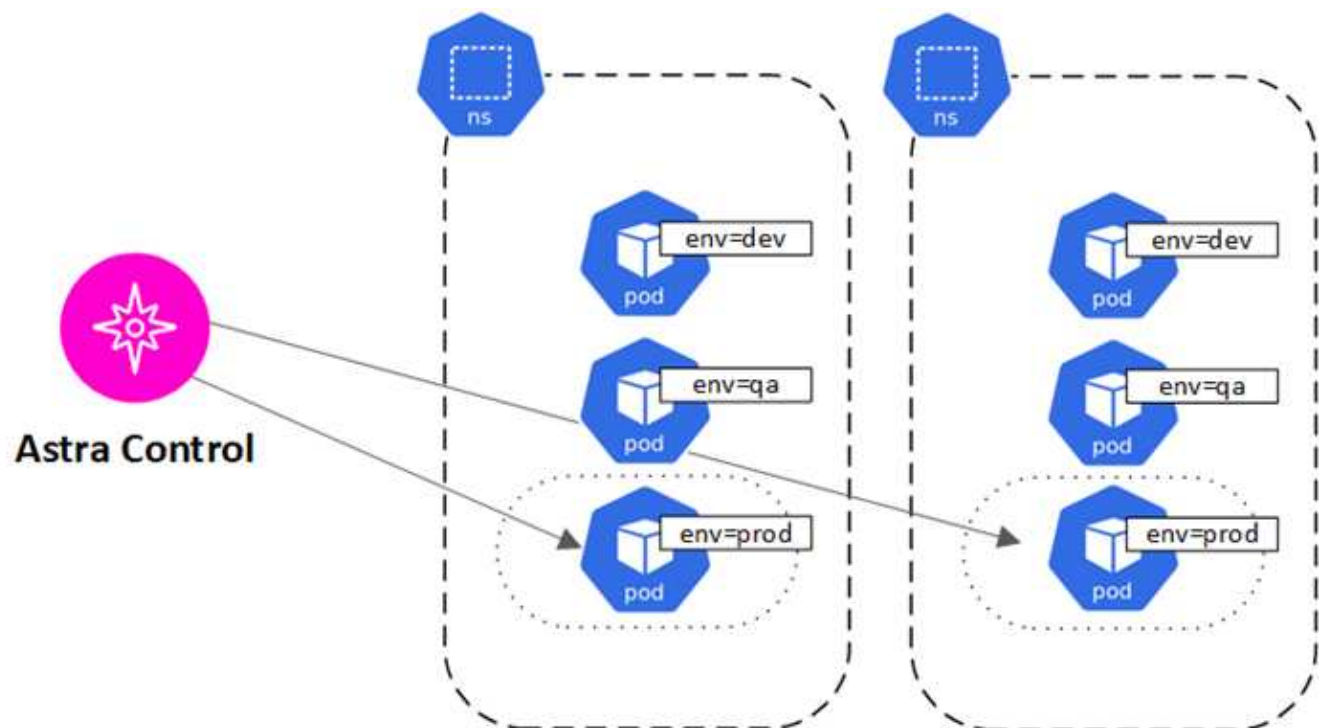
- A namespace, including all resources in that namespace



- An individual application deployed within one or more namespaces (Helm 3 is used in this example)



- A group of resources that are identified by a Kubernetes label within one or more namespaces



# User roles and namespaces

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

## User roles

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

- An **Owner** has Admin permissions and can delete accounts.
- An **Admin** has Member permissions and can invite other users.
- A **Member** can fully manage apps and clusters.
- A **Viewer** can view resources.

You can add constraints to a Member or Viewer user to restrict the user to one or more [Namespaces](#).

## Namespaces

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

## Find more information

- [Manage roles](#)



# Use Astra Control Service

## Log in to Astra Control Service

Astra Control Service is accessible through a SaaS-based user interface by going to <https://astra.netapp.io>.



You can use single sign-on to log in using credentials from your corporate directory (federated identity). To learn more, go to the [Help Center](#) and then select **Cloud Central sign-in options**.

### Before you begin

- A [BlueXP user ID](#).
- A [new Astra Control account](#) or [an invitation to an existing account](#).
- A supported web browser.

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

### Steps

1. Open a web browser and go to <https://astra.netapp.io>.
2. Log in using your NetApp BlueXP credentials.

## Manage and protect apps

### Start managing apps

After you [add a Kubernetes cluster to Astra Control](#), you can install apps on the cluster (outside of Astra Control), and then go to the Applications page in Astra Control to define the apps.

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

### App management requirements

Astra Control has the following app management requirements:

- **Licensing:** To manage more than 10 namespaces, you need an Astra Control subscription.
- **Namespaces:** Apps can be defined within one or more specified namespaces on a single cluster using Astra Control. An app can contain resources spanning multiple namespaces within the same cluster. Astra Control does not support the ability for apps to be defined across multiple clusters.
- **Storage class:** If you install an app with a storage class explicitly set and you need to clone the app, the target cluster for the clone operation must have the originally specified storage class. Cloning an application with an explicitly set storage class to a cluster that does not have the same storage class will fail.
- **Kubernetes resources:** Apps that use Kubernetes Resources not collected by Astra Control might not



have full app data management capabilities. Astra Control collects the following Kubernetes resources:

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

## Supported app installation methods

Astra Control supports the following application installation methods:

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

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

- **Helm 3:** If you use Helm to install apps, Astra Control requires Helm version 3. Managing and cloning apps installed with Helm 3 (or upgraded from Helm 2 to Helm 3) are fully supported. Managing apps installed with Helm 2 is not supported.
- **Operator-deployed apps:** Astra Control supports apps installed with namespace-scoped operators that are, in general, designed with a "pass-by-value" rather than "pass-by-reference" architecture. An operator and the app it installs must use the same namespace; you might need to modify the deployment .yaml file for the operator to ensure this is the case.

The following are some operator apps that follow these patterns:

- [Apache K8ssandra](#)



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

- [Jenkins CI](#)
- [Percona XtraDB Cluster](#)

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

## Install apps on your cluster

After you've [added your cluster](#) to Astra Control, you can install apps or manage existing apps on the cluster. Any app that is scoped to one or more namespaces can be managed.

Astra Control will manage stateful apps only if the storage is on a storage class supported by Astra Control. Astra Control Service supports any storage class that is supported by Astra Control Provisioner or a generic CSI driver.

- [Learn about storage classes for GKE clusters](#)
- [Learn about storage classes for AKS clusters](#)
- [Learn about storage classes for AWS clusters](#)

## Define apps

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

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



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

## Before you begin

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



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

## About this task

- Before you begin, you should also understand [managing standard and system namespaces](#).
- If you plan to use multiple namespaces with your apps in Astra Control, consider [modifying user roles with namespace constraints](#) before defining apps.
- For instructions on how to manage apps using the Astra Control API, refer to the [Astra Automation and API information](#).

## App management options

- [Define resources to manage as an app](#)

- [Define a namespace to manage as an app](#)

## Define resources to manage as an app

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

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



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

## Read more about adding cluster-scoped resources to your app namespaces.

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

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

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

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

## Steps

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



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

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



To learn more about Kubernetes labels, [refer to the official Kubernetes documentation](#).

6. (Optional) Add additional namespaces for the app by selecting **Add namespace** and choosing the namespace from the drop-down list.
7. (Optional) Enter single label or label selector criteria for any additional namespaces you add.
8. (Optional) To include cluster-scoped resources in addition to those that Astra Control automatically includes, check **Include additional cluster-scoped resources** and complete the following:
  - a. Select **Add include rule**.
  - b. **Group**: From the drop-down list, select the API group of resources.
  - c. **Kind**: From the drop-down list, select the name of the object schema.
  - d. **Version**: Enter the API version.
  - e. **Label selector**: Optionally, include a label to add to the rule. This label is used to retrieve only those resources matching this label. If you don't provide a label, Astra Control collects all instances of the resource kind specified for that cluster.
  - f. Review the rule that is created based on your entries.
  - g. Select **Add**.



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

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

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



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



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

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

### Define a namespace to manage as an app

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

### Steps

1. From the Clusters page, select a cluster.
2. Select the **Namespaces** tab.

3. Select the Actions menu for the namespace that contains the app resources you want to manage and select **Define as application**.



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



Select the **Show system namespaces** checkbox to reveal system namespaces that are usually not used in app management by default. ☐ [Show system namespaces](#) [Read more](#).

After the process completes, the applications that are associated with the namespace appear in the `Associated applications` column.

### What about system namespaces?

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

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

☐ [Show system namespaces](#)



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.

### Protect apps with snapshots and backups

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

Learn more about [data protection in Astra Control](#).

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

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

## Configure a protection policy

A protection policy protects an app by creating snapshots, backups, or both at a defined schedule. You can choose to create snapshots and backups hourly, daily, weekly, and monthly, and you can specify the number of copies to retain.

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



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

### Steps

1. Select **Applications** and then select the name of a managed app.
2. Select **Data Protection**.
3. Select **Configure Protection Policy**.
4. Define a protection schedule by choosing the number of snapshots and backups to keep for the hourly, daily, weekly, and monthly schedules.

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

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

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

**Configure protection policy**

STEP 1/2: DETAILS

×

**PROTECTION SCHEDULE**

**Hourly**

Every hour on the 0th minute, keep the last 4 snapshots

**Daily**

Daily at 05:00 (UTC), keep the last 7 snapshots

**Weekly**

Weekly on Mondays at 05:00 (UTC), keep the last 12 snapshots

**Monthly**

Every 1st of the month at 05:00 (UTC), keep the last 12 backups

Hourly
Daily
Weekly
**Monthly**

Day(s) of Month (optional)  
1 X

Time (UTC) (optional)  
05:00

−

Snapshots to keep

+

0

−

Backups to keep

+

12

**BACKUP DESTINATION**

Bucket:
ben-astra-bucket
Default

**OVERVIEW**

**Schedule and retention**

Define a policy to continuously protect your application on a schedule and configure a retention count to get started.

For select stateful applications, expect I/O to pause for a short time during a backup or snapshot operation.

Read more in [Protection policies](#)

Application  
maria

Namespace  
maria

Cluster  
david-ie-00

Cancel

Review →

5. Select **Review**.

6. Select **Set Protection Policy**.

## Result

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

## Create a snapshot

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

## About this task

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

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



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

## Steps

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

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

### Result

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

### Create a backup

You can also back up an app at any time.



Be aware of how storage space is handled when you back up an application hosted on Azure NetApp Files storage. Refer to [Application backups](#) for more information.



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

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

### About this task

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

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

### Steps

1. Select **Applications**.
2. From the Options menu in the **Actions** column for the desired app, select **Back up**.
3. Customize the name of the backup.
4. Choose whether to back up the app from an existing snapshot. If you select this option, you can choose from a list of existing snapshots.
5. Choose a destination bucket for the backup from the list of storage buckets.
6. Select **Next**.
7. Review the backup summary and select **Back up**.

### Result

Astra Control creates a backup of the app.





- If your network has an outage or is abnormally slow, a backup operation might time out. This causes the backup to fail.
- If you need to cancel a running backup, use the instructions in [Cancel backups](#). To delete the backup, wait until it has completed and then use the instructions in [Delete backups](#).
- After a data protection operation (clone, backup, restore) and subsequent persistent volume resize, there is up to a twenty-minute delay before the new volume size is shown in the UI. The data protection operation is successful within minutes, and you can use the management software for the storage backend to confirm the change in volume size.

## Enable backup and restore for ontap-nas-economy operations

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

### Before you begin

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

## Expand for configuration steps

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



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

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



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

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

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

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

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

Response:

```
snapshotDirectory: "true"
```

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

### Result

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

## Create an immutable backup

An immutable backup cannot be modified, deleted, or overwritten as long as the retention policy on the bucket that stores the backup forbids it. You can create immutable backups by backing up applications to buckets that

have a retention policy configured. Refer to [Data protection](#) for important information about working with immutable backups.

### Before you begin

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

- **Amazon Web Services:** [Enable S3 Object Lock when creating the bucket and set a default retention mode of "governance" with a default retention period.](#)
- **Google Cloud:** [Configure a bucket with a retention policy and specify a retention period.](#)
- **Microsoft Azure:** [Configure a blob storage bucket with a time-based retention policy on container-level scope.](#)
- **NetApp StorageGRID:** [Enable S3 Object Lock when creating the bucket and set a default retention mode of "compliance" with a default retention period.](#)



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



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

### Steps

1. Select **Applications**.
2. From the Options menu in the **Actions** column for the desired app, select **Back up**.
3. Customize the name of the backup.
4. Choose whether to back up the app from an existing snapshot. If you select this option, you can choose from a list of existing snapshots.
5. Choose a destination bucket for the backup from the list of storage buckets. A write once read many (WORM) bucket is indicated with a status of "Locked" next to the bucket name.



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

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

### Result

Astra Control creates an immutable backup of the app.



- If your network has an outage or is abnormally slow, a backup operation might time out. This causes the backup to fail.
- If you try to create two immutable backups of the same app to the same bucket at the same time, Astra Control prevents the second backup from starting. Wait until the first backup is complete before starting another.
- You cannot cancel a running immutable backup.
- After a data protection operation (clone, backup, restore) and subsequent persistent volume resize, there is up to a twenty-minute delay before the new volume size is shown in the UI. The data protection operation is successful within minutes, and you can use the management software for the storage backend to confirm the change in volume size.

## View snapshots and backups

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



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

### Steps

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

The snapshots display by default.

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

## Delete snapshots

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

### Steps

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

### Result

Astra Control deletes the snapshot.

## Cancel backups

You can cancel a backup that is in progress.



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



You cannot cancel a running immutable backup.

### Steps

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

## Delete backups

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



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



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

## Steps

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

## Result

Astra Control deletes the backup.

## [Tech preview] Protect an entire cluster

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

## How it works

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

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

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

## Example: Full cluster protection

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

### Example: Label-based protection

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

### Create a scheduled backup of all namespaces

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

#### Steps

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

### Create a scheduled backup of specific namespaces

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

#### Steps

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

## Restore apps

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



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

#### Before you begin

- **Protect your apps first:** It is strongly recommended that you take a snapshot or backup of your application before restoring it. This will enable you to clone from the snapshot or backup if the restore is unsuccessful.
- **Check destination volumes:** If you restore to a different storage class, ensure that the storage class uses the same persistent volume access mode (for example, ReadWriteMany). The restore operation will fail if the destination persistent volume access mode is different. For example, if your source persistent volume uses the RWX access mode, selecting a destination storage class that is not able to provide RWX, such as

Azure Managed Disks, AWS EBS, Google Persistent Disk or `ontap-san`, will cause the restore operation to fail. For more information about persistent volume access modes, refer to the [Kubernetes documentation](#).

- **Plan for space needs:** When you perform an in-place restore of an application that uses NetApp ONTAP storage, the space used by the restored app can double. After performing an in-place restore, remove any unwanted snapshots from the restored application to free up storage space.
- **Supported storage class drivers:** Astra Control supports restoring backups using storage classes backed by the following drivers:
  - `ontap-nas`
  - `ontap-nas-economy`
  - `ontap-san`
  - `ontap-san-economy`
- **(ontap-nas-economy driver only) Backups and restores:** Before backing up or restoring an app that uses a storage class backed by the `ontap-nas-economy` driver, verify that the [snapshot directory on the ONTAP storage backend is hidden](#). Failure to hide this directory might lead to loss of access to your application, especially if it is using NFSv3.



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

## Steps

1. Select **Applications** and then select the name of an app.
2. From the Options menu in the Actions column, select **Restore**.
3. Choose the restore type:
  - **Restore to original namespaces:** Use this procedure to restore the app in-place to the original cluster.
    - a. Select the snapshot or backup to use to restore the app in-place, which reverts the app to an earlier version of itself.
    - b. Select **Next**.



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

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



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

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



To learn about configuring include and exclude rules, see [Filter resources during an application restore](#).

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

## Result

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



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



Any member user with namespace constraints by namespace name/ID or by namespace labels can clone or restore an app to a new namespace on the same cluster or to any other cluster in their organization's account. However, the same user cannot access the cloned or restored app in the new namespace. After a clone or restore operation creates a new namespace, the account admin/owner can edit the member user account and update role constraints for the affected user to grant access to the new namespace.

## Filter resources during an application restore

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



## Read more about include and exclude scenarios

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

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

### Steps

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



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

2. Configure the filter rule:



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

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



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

- b. (Optional) Enter a resource name.
- c. (Optional) **Label selector:** Include a [label selector](#) to add to the rule. The label selector is used to filter only those resources matching the selected label.
- d. (Optional) Select **Use GVK (GroupVersionKind) set to filter resources** for additional filtering options.



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

- i. (Optional) **Group:** From the drop-down list, select the Kubernetes API group.
- ii. **Kind:** From the drop-down list, select the object schema for the Kubernetes resource type to use in the filter.

- iii. **Version:** Select the Kubernetes API version.
3. Review the rule that is created based on your entries.
4. Select **Add**.



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

## Clone and migrate apps

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

Cloning can help if you need to move applications and storage from one Kubernetes cluster to another. For example, you might want to move workloads through a CI/CD pipeline and across Kubernetes namespaces.



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

### Before you begin

- **Check destination volumes:** If you clone to a different storage class, ensure that the storage class uses the same persistent volume access mode (for example, ReadWriteMany). The clone operation will fail if the destination persistent volume access mode is different. For example, if your source persistent volume uses the RWX access mode, selecting a destination storage class that is not able to provide RWX, such as Azure Managed Disks, AWS EBS, Google Persistent Disk or `ontap-san`, will cause the clone operation to fail. For more information about persistent volume access modes, refer to the [Kubernetes](#) documentation.
- To clone apps to a different cluster, you need to make sure that you have assigned a default bucket for the cloud instance containing the source cluster. If the source cloud instance does not have a default bucket set, the cross-cluster clone operation will fail.
- During clone operations, apps that need an IngressClass resource or webhooks to function properly must not have those resources already defined on the destination cluster.

### Clone limitations

- **Explicit storage classes:** If you deploy an app with a storage class explicitly set and you need to clone the app, the target cluster must have the originally specified storage class. Cloning an application with an explicitly set storage class to a cluster that does not have the same storage class will fail.
- **ontap-nas-economy-backed applications:** You can't use clone operations if your application's storage class is backed by the `ontap-nas-economy` driver. You can, however, [enable backup and restore for ontap-nas-economy operations](#).
- **Clones and user constraints:** Any member user with namespace constraints by namespace name/ID or by namespace labels can clone or restore an app to a new namespace on the same cluster or to any other cluster in their organization's account. However, the same user cannot access the cloned or restored app in the new namespace. After a clone or restore operation creates a new namespace, the account admin/owner can edit the member user account and update role constraints for the affected user to grant access to the new namespace.
- **Clones use default buckets:**

- During an app backup or app restore, you can specify a bucket to use. You need to specify a default bucket when you clone across clusters, but specifying a bucket is optional when cloning within the same cluster.
- When you clone across clusters, the cloud instance containing the source cluster of the clone operation must have a default bucket set.
- There is no option to change buckets for a clone. If you want control over which bucket is used, you can either [change the bucket default](#) or do a [backup](#) followed by a [restore](#) separately.
- **With Jenkins CI:** If you clone an operator-deployed instance of Jenkins CI, you need to manually restore the persistent data. This is a limitation of the app's deployment model.

## Steps

1. Select **Applications**.
2. Do one of the following:
  - Select the Options menu in the **Actions** column for the desired app.
  - Select the name of the desired app, and select the status drop-down list at the top right of the page.
3. Select **Clone**.
4. Specify details for the clone:
  - Enter a name.
  - Choose a destination cluster for the clone.
  - Enter destination namespaces for the clone. Each source namespace associated with the app maps to a destination namespace.



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

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



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



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

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

## Result

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

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

## Manage app execution hooks

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

### Types of execution hooks

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

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

### Execution hook filters

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

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



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

### Important notes about custom execution hooks

Consider the following when planning execution hooks for your apps.



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

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

- The execution hooks feature is disabled by default for new Astra Control deployments.

- You need to enable the execution hooks feature before you can use execution hooks.
- Owner or Admin users can enable or disable the execution hooks feature for all users defined in the current Astra Control account. Refer to [Enable the execution hooks feature](#) and [Disable the execution hooks feature](#) for instructions.
- The feature enablement status is preserved during Astra Control upgrades.
- An execution hook must use a script to perform actions. Many execution hooks can reference the same script.
- Astra Control requires the scripts that execution hooks use to be written in the format of executable shell scripts.
- Script size is limited to 96KB.
- Astra Control uses execution hook settings and any matching criteria to determine which hooks are applicable to a snapshot, backup, or restore operation.
- All execution hook failures are soft failures; other hooks and the data protection operation are still attempted even if a hook fails. However, when a hook fails, a warning event is recorded in the **Activity** page event log.
- To create, edit, or delete execution hooks, you must be a user with Owner, Admin, or Member permissions.
- If an execution hook takes longer than 25 minutes to run, the hook will fail, creating an event log entry with a return code of "N/A". Any affected snapshot will time out and be marked as failed, with a resulting event log entry noting the timeout.
- For ad hoc data protection operations, all hook events are generated and saved in the **Activity** page event log. However, for scheduled data protection operations, only hook failure events are recorded in the event log (events generated by the scheduled data protection operations themselves are still recorded).

#### Order of execution

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

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

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

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

You can see an example of this configuration in scenario number 2 from the table in [Determine whether a hook](#)

will run.



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

#### Determine whether a hook will run

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

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

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



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

- The pre-backup and post-backup hooks are always run.
- If the backup includes a new snapshot and the snapshot has started, the pre-snapshot and post-snapshot hooks are run.
- If the backup is cancelled prior to the snapshot starting, the pre-snapshot and post-snapshot hooks are not run.

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

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

## Execution hook examples

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

## Enable the execution hooks feature

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

### Steps

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

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

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

## Disable the execution hooks feature

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

### Steps

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

## View existing execution hooks

You can view existing custom execution hooks for an app.

### Steps

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

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

### View existing scripts

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

#### Steps

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

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

### Add a script

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

#### Steps

1. Ensure that the execution hooks feature is [enabled](#).
2. Go to **Account**.
3. Select the **Scripts** tab.
4. Select **Add**.
5. Do one of the following:
  - Upload a custom script.
    - a. Select the **Upload file** option.
    - b. Browse to a file and upload it.
    - c. Give the script a unique name.
    - d. (Optional) Enter any notes other administrators should know about the script.
    - e. Select **Save script**.
  - Paste in a custom script from the clipboard.
    - a. Select the **Paste or type** option.
    - b. Select the text field and paste the script text into the field.
    - c. Give the script a unique name.
    - d. (Optional) Enter any notes other administrators should know about the script.
6. Select **Save script**.



## Result

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

## Delete a script

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

### Steps

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



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

## Create a custom execution hook

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



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

### Steps

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



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

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



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

7. When done, select **Next**.

8. In the **Script** area, do one of the following:

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

This instructs the execution hook to use this script.

9. Select **Next**.

10. Review the execution hook configuration.

11. Select **Add**.

## Check the state of an execution hook

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

### Steps

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

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

## View script usage

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

### Steps

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

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

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

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

## Edit an execution hook

You can edit an execution hook if you want to change its attributes, filters, or the script that it uses. You need to have Owner, Admin, or Member permissions to edit execution hooks.

### Steps

1. Select **Applications** and then select the name of a managed app.
2. Select the **Execution hooks** tab.
3. Select the Options menu in the **Actions** column for a hook that you wish to edit.
4. Select **Edit**.
5. Make any needed changes, selecting **Next** after you complete each section.
6. Select **Save**.

## Disable an execution hook

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

### Steps

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

## Delete an execution hook

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

## Steps

1. Select **Applications** and then select the name of a managed app.
2. Select the **Execution hooks** tab.
3. Select the Options menu in the **Actions** column for a hook that you wish to delete.
4. Select **Delete**.
5. In the resulting dialog, type "delete" to confirm.
6. Select **Yes, delete execution hook**.

## For more information

- [NetApp Verda GitHub project](#)

# View app and compute health

## View a summary of app and cluster health

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

The Apps tile helps you identify the following:

- How many apps you're currently managing.
- Whether those managed apps are healthy.
- Whether the apps are fully protected (they're protected if recent backups are available).

Note that these aren't just 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.

The Clusters tile provides similar details about the health of the cluster and you can drill down to get more details just like you can with an app.

## View the health and details of clusters

After you add Kubernetes clusters to Astra Control, 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 Service UI, select **Clusters**.
2. On the **Clusters** page, select the cluster whose details you want to view.



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

3. View the information on the **Overview**, **Storage**, and **Activity** tabs to find the information that you're looking for.

- **Overview:** Details about the worker nodes, including their state.
- **Storage:** The persistent volumes associated with the compute, including the storage class and state.
- **Activity:** The activities related to the cluster.



You can also view cluster information starting from the Astra Control Service **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 Control provides details about the app that enables you to identify its communication status (whether Astra Control can communicate with the app), its protection status (whether it's fully protected in case of failure), the pods, persistent storage, and more.

### Steps

1. Select **Applications** and then select the name of an app.
2. Find the information that you're looking for:

#### App Status

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

#### App Protection Status

Provides a status of how well the app is protected:

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

*You can't be fully protected until you have a recent backup.* This is important because backups are stored in an object store away from the persistent volumes. If a failure or accident wipes out the cluster and its 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

The Astra Control activities related to the app.

# Manage buckets

You can manage the buckets that Astra uses for backups and clones. You can add additional buckets, remove existing buckets, and change the default bucket for the Kubernetes clusters in a cloud instance.

Only Owners and Admins can manage buckets.

## How Astra Control uses buckets

When you start managing your first Kubernetes cluster for a cloud instance, Astra Control Service creates the initial bucket for that [cloud instance](#).

You can manually designate a bucket as the default bucket for a cloud instance. If you do so, Astra Control Service uses this bucket by default for the backups and clones that you create on any managed cluster in that cloud instance (you can select a different bucket for backups). If you perform a live clone of an application from any of the managed clusters in a cloud instance to another cluster, Astra Control Service uses the default bucket for the source cloud instance to perform the clone operation.

You can set the same bucket as the default bucket for multiple cloud instances.

You can select from any buckets when you create a protection policy or start an ad-hoc backup.



Astra Control Service checks whether a destination bucket is accessible prior to starting a backup or a clone.

## View existing buckets

View the list of buckets that are available to Astra Control Service to determine their status and to identify the default bucket (if defined) for your cloud instance.

A bucket can have any of the following states:

### Pending

After you add a bucket, it starts in the pending state while Astra Control discovers it.

### Available

The bucket is available for use by Astra Control.

### Removed

The bucket isn't operational at the moment. Hover your mouse over the status icon to identify what the problem is.

If a bucket is in the Removed state, you can still set it as the default bucket and assign it to a protection schedule. But if the bucket isn't in the Available state by the time a data protection operation starts, then that operation will fail.

## Step

1. Go to **Buckets**.

The list of buckets available to Astra Control Service is displayed.

## Add an additional bucket

You can add additional buckets at any time. This enables you to choose between buckets when creating a protection policy or starting an ad-hoc backup, and enables you to change the default bucket that a cloud instance uses.

You can add the following types of buckets:

- Amazon Web Services
- Generic S3
- Google Cloud Platform
- Microsoft Azure
- NetApp ONTAP S3
- NetApp StorageGRID S3

### Before you begin

- Ensure you know the name of an existing bucket.
- Ensure you have credentials for the bucket that provide Astra Control with the permissions that it needs to manage the bucket.
- If your bucket is in Microsoft Azure:
  - The bucket must belong to the resource group named *astra-backup-rg*.
  - If the Azure storage account instance performance setting is set to "Premium", the "Premium account type" setting must be set to "Block blobs".

### Steps

1. Go to **Buckets**.
2. Select **Add** and follow the prompts to add the bucket.
  - **Type**: Choose your cloud provider.
  - **Existing bucket name**: Enter the name of the bucket.
  - **Description**: Optionally enter a description of the bucket.
  - **Storage account** (Azure only): Enter the name of your Azure storage account. This bucket must belong to the resource group named *astra-backup-rg*.
  - **S3 server name or IP address** (AWS and S3 bucket types only): Enter the fully qualified domain name of the S3 endpoint that corresponds with your region, without `https://`. Refer to [the Amazon documentation](#) for more information.
  - **Select credentials**: Enter the credentials that provide Astra Control Service with the permissions that it needs to manage the bucket. The information you need to provide varies depending on the bucket type.
3. Select **Add** to add the bucket.

### Result

Astra Control Service adds the bucket. You can now choose this bucket when creating a protection policy or performing an ad-hoc backup. You can also set this bucket as the default bucket for a cloud instance.

## Change the default bucket

You can change the default bucket for a cloud instance. Astra Control Service will use this bucket by default for backups and clones. Each cloud instance has its own default bucket.



Astra Control does not automatically assign a default bucket for any cloud instance. You need to manually set a default bucket for a cloud instance before performing app clone operations between two clusters.

### Steps

1. Go to **Cloud instances**.
2. Select the configuration menu in the **Actions** column for the cloud instance that you want to edit.
3. Select **Edit**.
4. In the list of buckets, select the bucket you want to make the default bucket for this cloud instance.
5. Select **Update**.

## Remove a bucket

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



- You cannot remove a default bucket. If you want to remove that bucket, first select another bucket as the default.
- You cannot remove a write once read many (WORM) bucket before the bucket's cloud provider retention period has expired. WORM buckets are denoted with "Locked" next to the bucket name.

### Before you begin

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

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

### Steps

1. Go to **Buckets**.
2. From the **Actions** menu, select **Remove**.



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

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



## Find more information

- [Use the Astra Control API](#)

## Monitor running tasks

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

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



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

To refer to more information about failed tasks, refer to [Monitor account activity](#).

### Steps

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

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



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

## Manage your account

### Set up billing

You can use more than one method to manage your Astra Control Service account billing. If you are using Azure or Amazon AWS, you can subscribe to an Astra Control Service plan through the Microsoft Azure Marketplace or the AWS Marketplace. When you do this, you can manage your billing details through the Marketplace. Or, you can subscribe directly with NetApp. If you subscribe directly with NetApp, you can manage your billing details through Astra Control Service. If you use Astra Control Service without a subscription, you are automatically subscribed to the Free Plan.

The Astra Control Service Free Plan enables you to manage up to 10 namespaces in your account. If you want to manage more than 10 namespaces, then you'll need to set up billing by upgrading from the Free Plan to the Premium Plan, or subscribe through the Azure Marketplace or AWS Marketplace.

## Billing overview

There are two types of costs associated with using Astra Control Service: charges from NetApp for the Astra Control Service and charges from your cloud provider for persistent volumes and object storage.

### Astra Control Service billing

Astra Control Service offers three plans:

#### Free Plan

Manage up to 10 namespaces for free.

#### Premium PayGo

Manage an unlimited amount of namespaces at a specific rate, per namespace.

#### Premium Subscription

Pre-pay at a discounted rate with an annual subscription that enables you to manage up to 20 namespaces per *namespace pack*. Contact NetApp Sales to purchase as many packs as needed for your organization. For example, purchase 3 packs to manage 60 namespaces from Astra Control Service. If you manage more namespaces than allowed by your annual subscription, then you'll be charged at the subscription-dependent overage rate per extra namespace. If you don't have an Astra Control account yet, purchasing the Premium Subscription automatically creates an Astra Control account for you. If you have an existing Free Plan, then you're automatically converted to the Premium Subscription.

When you create an Astra Control account, you're automatically subscribed to the Free Plan. Astra Control's Dashboard shows you how many namespaces you're currently managing out of the 10 free namespaces that you're allowed. Billing starts for a namespace when the first app containing the namespace is managed, and stops for that namespace when the last app containing the namespace is unmanaged.

If you try to manage an 11th namespace, Astra Control notifies you that you've reached the limit of the Free Plan. It then prompts you to upgrade from the Free Plan to a Premium Plan. You'll be charged at the subscription-dependent overage rate per extra namespace.

You can upgrade to a Premium Plan at any time. After you upgrade, Astra Control starts charging you for *all* namespaces in the account. The first 10 namespaces don't stay in the Free Plan.

### Google Cloud billing

Persistent volumes are backed by NetApp Cloud Volumes Service and backups of your apps are stored in a Google Cloud Storage bucket.

- [View pricing details for Cloud Volumes Service.](#)

Note that Astra Control Service supports all service types and service levels. The service type that you use depends on your [Google Cloud region](#).

- [View pricing details for Google Cloud storage buckets.](#)

### Microsoft Azure billing

Persistent volumes are backed by Azure NetApp Files and backups of your apps are stored in an Azure Blob container.

- [View pricing details for Azure NetApp Files.](#)

- [View pricing details for Microsoft Azure Blob storage.](#)
- [View Astra Control Service plans and pricing in Azure Marketplace](#)



The Azure billing rate for Astra Control Service is per hour, and a new billing hour starts after 29 minutes of the usage hour has elapsed.

### Amazon Web Services billing

Persistent volumes are backed by EBS or FSx for NetApp ONTAP and backups of your apps are stored in an AWS bucket.

- [View pricing details for Amazon Web Services.](#)

### Subscribe to Astra Control Service in the Azure Marketplace

You can subscribe to Astra Control Service using the Azure Marketplace. Your account and billing details are managed through the Marketplace.



To see a video walkthrough of the Azure Marketplace subscription process, visit [NetApp TV](#).

#### Steps

1. Go to the [Azure Marketplace](#).
2. Select **Get It Now**.
3. Follow the instructions to subscribe to a plan.

### Subscribe to Astra Control Service in the AWS Marketplace

You can subscribe to Astra Control Service using the AWS Marketplace. Your account and billing details are managed through the Marketplace.

#### Steps

1. Go to the [AWS Marketplace](#).
2. Select **View purchase options**.
3. If prompted to do so, log in to your AWS account, or create a new account.
4. Follow the instructions to subscribe to a plan.

### Subscribe to Astra Control Service directly with NetApp

You can subscribe to Astra Control Service from within the Astra Control Service UI or by contacting NetApp Sales.

### Upgrade from the Free Plan to the Premium PayGo Plan

Upgrade your billing plan at any time to start managing more than 10 namespaces from Astra Control by paying as you go. All you need is a valid credit card.

#### Steps

1. Select **Account** and then select **Billing**.
2. Under **Plans**, go to **Premium PayGo** and select **Upgrade Now**.

3. Provide payment details for a valid credit card and select **Upgrade to Premium Plan**.



Astra Control will email you if the credit card is nearing expiration.

### Result

You can now manage more than 10 namespaces. Astra Control starts charging you for *all* namespaces that you're currently managing.

### Upgrade from the Free Plan to the Premium Subscription

Contact NetApp Sales to pre-pay at a discounted rate with an annual subscription.

### Steps

1. Select **Account** and then select **Billing**.
2. Under **Plans**, go to **Premium Subscription** and select **Contact Sales**.
3. Provide details to the sales team to start the process.

### Result

A NetApp Sales representative will contact you to process your purchase order. After the order is complete, Astra Control will reflect your current plan on the **Billing** tab.

### View your current costs and billing history

Astra Control shows you your current monthly costs, as well as a detailed billing history by namespace. If you subscribed to a plan through a Marketplace, the billing history is not visible (but you can view it by logging in to the Marketplace.)

### Steps

1. Select **Account** and then select **Billing**.

Your current costs appear under the billing overview.

2. To view the billing history by namespace, select **Billing history**.

Astra Control shows you the usage minutes and cost for each namespace. A usage minute is how many minutes Astra Control managed your namespace during a billing period.

3. Select the drop-down list to select a previous month.

### Change the credit card for Premium PayGo

If needed, you can change the credit card that Astra Control has on file for billing.

### Steps

1. Select **Account > Billing > Payment method**.
2. Select the configure icon.
3. Modify the credit card.

## Important notes

- Your billing plan is per Astra Control account.

If you have multiple accounts, then each has its own billing plan.

- Your Astra Control bill includes charges for managing your namespaces. You're charged separately by your cloud provider for the storage backend for persistent volumes.

[Learn more about Astra Control pricing.](#)

- Each billing period ends on the last day of the month.
- You can't downgrade from a Premium Plan to the Free Plan.

## Invite and remove users

Invite users to join your Astra Control account and remove users that should no longer have access to the account.

### Invite users

Account Owners and Admins can invite other users to join the Astra Control account.

#### Steps

1. Make sure that the user has a [BlueXP login](#).
2. Select **Account**.
3. In the **Users** tab, select **Invite**.
4. Enter the user's name, email address, and their role.

Note the following:

- The email address must match the email address that the user used to sign up to BlueXP.
- Each role provides the following permissions:
  - An **Owner** has Admin permissions and can delete accounts.
  - An **Admin** has Member permissions and can invite other users.
  - A **Member** can fully manage apps and clusters.
  - A **Viewer** can view resources.
- 5. To add constraints to a user with a Member or Viewer role, enable the **Restrict role to constraints** check box.

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

6. To invite another user, select **Add another user** and enter information for the new user.

You can invite up to 10 users at a time. You can navigate between the users you are inviting on the left side of the **Invite users** dialog.

7. Select **Invite users**.

### Result

The user or users will receive an email that invites them to join your account.

## Change a user's role

An Account Owner can change the role of all users, while an Account Admin can change the role of users who have the Admin, Member, or Viewer role.

### Steps

1. Select **Account**.
2. In the **Users** tab, select the menu in the **Actions** column for the user.
3. Select **Edit role**.
4. Select a new role.
5. To add constraints to a user with a Member or Viewer role, enable the **Restrict role to constraints** check box.

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

6. Select **Confirm**.

### Result

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

## Remove users

A user with the Owner role can remove other users from the account at any time.

### Steps

1. Select **Account**.
2. In the **Users** tab, select the users that you want to remove.
3. Select the menu in the **Actions** column and select **Remove user**.
4. When you're prompted, confirm deletion by typing "remove" and then select **Yes, Remove User**.

### Result

Astra Control removes the user from the account.

## Manage roles

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

### Add a namespace constraint to a role

An Admin or Owner user can add namespace constraints.

### Steps

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

3. In the **Actions** column, select the menu button for a user with the Member or Viewer role.
4. Select **Edit role**.
5. Enable the **Restrict role to constraints** check box.

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

6. Select **Add constraint**.

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

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

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

10. Select **Confirm**.

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



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

## Remove a namespace constraint from a role

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

### Steps

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

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

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

## For more information

- [User roles and namespaces](#)

## Add and remove credentials

Add and remove cloud provider credentials from your account at any time. Astra Control uses these credentials to discover a Kubernetes cluster, the apps on the cluster, and to provision resources on your behalf.

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

### Add credentials

The most common way to add credentials to Astra Control is when you manage clusters, but you can also add credentials from the Account page. The credentials will then be available to choose when you manage additional Kubernetes clusters.

#### Before you begin

- For Amazon Web Services, you should have the JSON output of the credentials for the IAM account used to create the cluster. [Learn how to set up an IAM user.](#)
- For GKE, you should have the service account key file for a service account that has the required permissions. [Learn how to set up a service account.](#)
- For AKS, you should have the JSON file that contains the output from the Azure CLI when you created the service principal. [Learn how to set up a service principal.](#)

You'll also need your Azure subscription ID, if you didn't add it to the JSON file.

#### Steps

1. Select **Account > Credentials**.
2. Select **Add Credentials**.
3. Select **Microsoft Azure**.
4. Select **Google Cloud Platform**.
5. Select **Amazon Web Services**.
6. Enter a name for the credentials that distinguishes them from other credentials in Astra Control.
7. Provide the required credentials.
  - a. **Microsoft Azure:** Provide Astra Control with details about your Azure service principal by uploading a JSON file or by pasting the contents of that JSON file from your clipboard.

The JSON file should contain the output from the Azure CLI when you created the service principal. It can also include your subscription ID so it's automatically added to Astra Control. Otherwise, you need to manually enter the ID after providing the JSON.
  - b. **Google Cloud Platform:** Provide the Google Cloud service account key file either by uploading the file or by pasting the contents from your clipboard.
  - c. **Amazon Web Services:** Provide the Amazon Web Services IAM user credentials either by uploading the file or by pasting the contents from your clipboard.
8. Select **Add Credentials**.

#### Result

The credentials are now available to select when you add a cluster to Astra Control.



## Remove credentials

Remove credentials from an account at any time. You should only remove credentials after [unmanaging all clusters](#), unless you are rotating credentials (refer to [Rotate credentials](#)).



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

### Steps

1. Select **Account > Credentials**.
2. Select the drop-down list in the **State** column for the credentials that you want to remove.
3. Select **Remove**.
4. Type the name of the credentials to confirm deletion and then select **Yes, Remove Credentials**.

### Result

Astra Control removes the credentials from the account.

## Rotate credentials

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

### Steps

1. Remove the existing credentials by following the steps in [Remove credentials](#).
2. Add the new credentials by following the steps in [Add credentials](#).
3. Update all buckets to use the new credentials:
  - a. From the left navigation, select **Buckets**.
  - b. Select the drop-down list in the **Actions** column for the bucket that you want to edit.
  - c. Select **Edit**.
  - d. In the **Select credentials** section, choose the new credentials that you added to Astra Control.
  - e. Select **Update**.
  - f. Repeat steps **b** through **e** for any remaining buckets on your system.

### Result

Astra Control begins using the new cloud provider credentials.

## Monitor account activity

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

### View all account activity in Astra Control

1. Select **Activity**.
2. Use the filters to narrow down the list of activities or use the search box to find exactly what you're looking for.

3. Select **Export to CSV** to download your account activity to a CSV file.

#### View account activity for a specific app

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

#### View account activity for clusters

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

## View and manage notifications

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

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

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

## Close your account

If you no longer need your Astra Control account, you can close it at any time.



Buckets that Astra Control automatically created will be automatically deleted when you close your account.

#### Steps

1. [Unmanage all apps and clusters](#).
2. [Remove credentials from Astra Control](#).
3. Select **Account > Billing > Payment method**.
4. Select **Close Account**.
5. Enter your account name and confirm to close the account.

## Manage cloud instances

A cloud instance is a unique domain within a cloud provider. You can create multiple cloud instances for each cloud provider, and each cloud instance has its own name, credentials, and associated clusters.

You create a cloud instance when you add a new cluster to Astra Control. You can edit a cloud instance to change its name or default bucket using the Astra Control UI, and perform other actions with the cloud instance using the Astra Control API.

## Add a cloud instance

You can add a new cloud instance when you add a new cluster to Astra Control. Refer to [Start managing Kubernetes clusters from Astra Control Service](#) for more information.

## Edit a cloud instance

You can modify an existing cloud instance for a cloud provider.

### Steps

1. Go to **Cloud instances**.
2. In the list of cloud instances, select the **Actions** menu for the cloud instance you want to edit.
3. Select **Edit**.

On this page, you can update the name and default bucket for the cloud instance.



Each cloud instance in Astra Control must have a unique name.

## Rotate the credentials for a cloud instance

You can use the Astra Control API to rotate the credentials for a cloud instance. To learn more, [go to the Astra automation docs](#).

## Remove a cloud instance

You can use the Astra Control API to remove a cloud instance from a cloud provider. To learn more, [go to the Astra automation docs](#).

# Enable Astra Control Provisioner

Astra Trident versions 23.10 and later include the option to use Astra Control Provisioner, which enables licensed Astra Control users to access advanced storage provisioning functionality. Astra Control Provisioner provides this extended functionality in addition to standard Astra Trident CSI-based functionality. You can use this procedure to enable and install Astra Control Provisioner.

Your Astra Control Service subscription automatically includes the license for Astra Control Provisioner use.

In coming Astra Control updates, Astra Control Provisioner will replace Astra Trident as storage provisioner and orchestrator and be mandatory for Astra Control use. Because of this, it's strongly recommended that Astra Control users enable Astra Control Provisioner. Astra Trident will continue to remain open source and be released, maintained, supported, and updated with new CSI and other features from NetApp.

### How do I know if I need to enable Astra Control Provisioner?

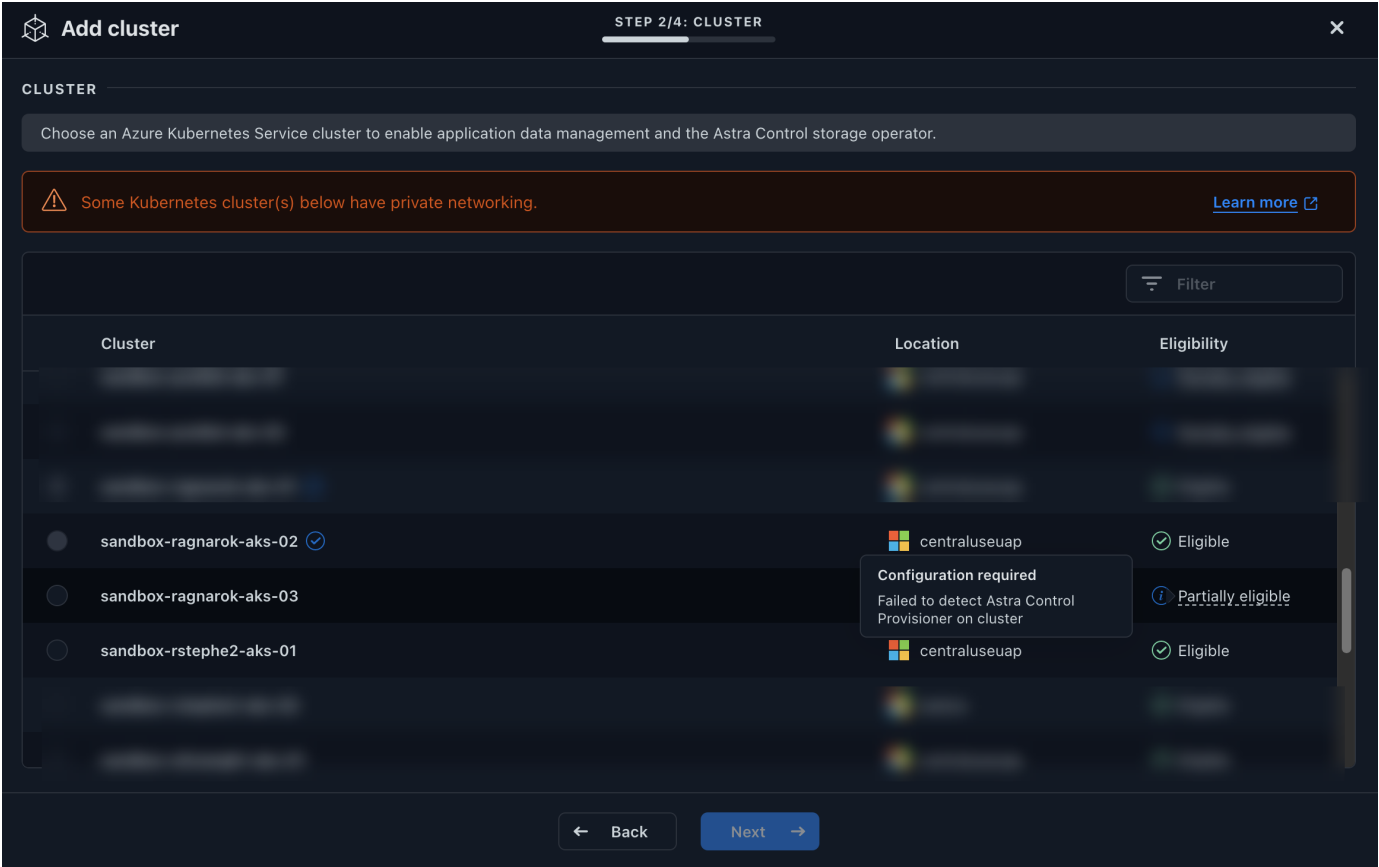
If you add a cluster to Astra Control Service that does not have Astra Trident previously installed, the cluster

will be marked as Eligible. After you [add the cluster to Astra Control](#), Astra Control Provisioner will be automatically enabled.

If your cluster is not marked Eligible, it will be marked Partially eligible because of one of the following:

- It's using an older version of Astra Trident
- It's using an Astra Trident 23.10 that does not yet have the provisioner option enabled
- It's a cluster type that does not allow automatic enablement

For Partially eligible cases, use these instructions to manually enable Astra Control Provisioner for your cluster.



Before you enable Astra Control Provisioner

If you have an existing Astra Trident without Astra Control Provisioner and want to enable Astra Control Provisioner, do the following first:

- **If you have Astra Trident installed, confirm that its version is within a four-release window:** You can perform a direct upgrade to Astra Trident 24.02 with Astra Control Provisioner if your Astra Trident is within a four-release window of version 24.02. For example, you can directly upgrade from Astra Trident 23.04 to 24.02.
- **Confirm that your cluster has an AMD64 system architecture:** The Astra Control Provisioner image is provided in both AMD64 and ARM64 CPU architectures, but only AMD64 is supported by Astra Control.

Steps

1. Access the NetApp Astra Control image registry:

- a. Log on to the Astra Control Service UI and record your Astra Control account ID.
  - i. Select the figure icon at the top right of the page.
  - ii. Select **API access**.
  - iii. Write down your account ID.
- b. From the same page, select **Generate API token** and copy the API token string to the clipboard and save it in your editor.
- c. Log into the Astra Control registry using your preferred method:

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

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

2. (Custom registries only) Follow these steps to move the image to your custom registry. If you aren't using a registry, follow the Trident operator steps in the [next section](#).



You can use Podman instead of Docker for the following commands. If you are using a Windows environment, PowerShell is recommended.

## Docker

1. Pull the Astra Control Provisioner image from the registry:



The image pulled will not support multiple platforms and will only support the same platform as the host that pulled the image, such as Linux AMD64.

```
docker pull cr.astra.netapp.io/astra/trident-acp:24.02.0
--platform <cluster platform>
```

Example:

```
docker pull cr.astra.netapp.io/astra/trident-acp:24.02.0
--platform linux/amd64
```

2. Tag the image:

```
docker tag cr.astra.netapp.io/astra/trident-acp:24.02.0
<my_custom_registry>/trident-acp:24.02.0
```

3. Push the image to your custom registry:

```
docker push <my_custom_registry>/trident-acp:24.02.0
```

## Crane

1. Copy the Astra Control Provisioner manifest to your custom registry:

```
crane copy cr.astra.netapp.io/astra/trident-acp:24.02.0
<my_custom_registry>/trident-acp:24.02.0
```

3. Determine if the original Astra Trident installation method used an [operator \(either manually or with Helm\)](#) or `tridentctl`.
4. Enable Astra Control Provisioner in Astra Trident using the installation method you used originally:

## Astra Trident operator

1. [Download the Astra Trident installer and extract it.](#)
2. Complete these steps if you have not yet installed Astra Trident or if you removed the operator from your original Astra Trident deployment:

- a. Create the CRD:

```
kubectl create -f
deploy/crds/trident.netapp.io_tridentorchestrators_crd_post1.1
6.yaml
```

- b. Create the trident namespace (`kubectl create namespace trident`) or confirm that the trident namespace still exists (`kubectl get all -n trident`). If the namespace has been removed, create it again.

3. Update Astra Trident to 24.02.0:



For clusters running Kubernetes 1.24 or earlier, use `bundle_pre_1_25.yaml`.  
For clusters running Kubernetes 1.25 or later, use `bundle_post_1_25.yaml`.

```
kubectl -n trident apply -f trident-installer/deploy/<bundle-
name.yaml>
```

4. Verify Astra Trident is running:

```
kubectl get torc -n trident
```

Response:

NAME	AGE
trident	21m

5. If you have a registry that uses secrets, create a secret to use to pull the Astra Control Provisioner image:

```
kubectl create secret docker-registry <secret_name> -n trident
--docker-server=<my_custom_registry> --docker-username=<username>
--docker-password=<token>
```

6. Edit the TridentOrchestrator CR and make the following edits:

```
kubectl edit torc trident -n trident
```

- a. Set a custom registry location for the Astra Trident image or pull it from the Astra Control registry (tridentImage: <my\_custom\_registry>/trident:24.02.0 or tridentImage: netapp/trident:24.02.0).
- b. Enable Astra Control Provisioner (enableACP: true).
- c. Set the custom registry location for the Astra Control Provisioner image or pull it from the Astra Control registry (acpImage: <my\_custom\_registry>/trident-acp:24.02.0 or acpImage: cr.astra.netapp.io/astra/trident-acp:24.02.0).
- d. If you established [image pull secrets](#) earlier in this procedure, you can set them here (imagePullSecrets: - <secret\_name>). Use the same name secret name you established in the previous steps.

```
apiVersion: trident.netapp.io/v1
kind: TridentOrchestrator
metadata:
  name: trident
spec:
  debug: true
  namespace: trident
  tridentImage: <registry>/trident:24.02.0
  enableACP: true
  acpImage: <registry>/trident-acp:24.02.0
  imagePullSecrets:
    - <secret_name>
```

7. Save and exit the file. The deployment process will begin automatically.
8. Verify the operator, deployment, and replicaset are created.

```
kubectl get all -n trident
```



There should only be **one instance** of the operator in a Kubernetes cluster. Do not create multiple deployments of the Astra Trident operator.

9. Verify the trident-acp container is running and that acpVersion is 24.02.0 with a status of Installed:

```
kubectl get torc -o yaml
```

Response:



```
status:
  acpVersion: 24.02.0
  currentInstallationParams:
    ...
    acpImage: <registry>/trident-acp:24.02.0
    enableACP: "true"
    ...
  ...
status: Installed
```

## tridentctl

1. [Download the Astra Trident installer and extract it.](#)
2. [If you have an existing Astra Trident, uninstall it from the cluster that hosts it.](#)
3. Install Astra Trident with Astra Control Provisioner enabled (`--enable-acp=true`):

```
./tridentctl -n trident install --enable-acp=true --acp
-image=mycustomregistry/trident-acp:24.02
```

4. Confirm that Astra Control Provisioner has been enabled:

```
./tridentctl -n trident version
```

## Response:

```
+-----+-----+-----+ | SERVER
VERSION | CLIENT VERSION | ACP VERSION | +-----+
+-----+-----+-----+ | 24.02.0 | 24.02.0 | 24.02.0. |
+-----+-----+-----+
```

## Helm

1. If you have Astra Trident 23.07.1 or earlier installed, [uninstall](#) the operator and other components.
2. If your Kubernetes cluster is running 1.24 or earlier, delete psp:

```
kubectl delete psp tridentoperatorpod
```

3. Add the Astra Trident Helm repository:

```
helm repo add netapp-trident https://netapp.github.io/trident-helm-chart
```

#### 4. Update the Helm chart:

```
helm repo update netapp-trident
```

Response:

```
Hang tight while we grab the latest from your chart
repositories...
...Successfully got an update from the "netapp-trident" chart
repository
Update Complete. ☐Happy Helming!☐
```

#### 5. List the images:

```
./tridentctl images -n trident
```

Response:

```
| v1.28.0          | netapp/trident:24.02.0|
|                  | docker.io/netapp/trident-
autosupport:24.02|
|                  | registry.k8s.io/sig-storage/csi-
provisioner:v4.0.0|
|                  | registry.k8s.io/sig-storage/csi-
attacher:v4.5.0|
|                  | registry.k8s.io/sig-storage/csi-
resizer:v1.9.3|
|                  | registry.k8s.io/sig-storage/csi-
snapshotter:v6.3.3|
|                  | registry.k8s.io/sig-storage/csi-node-
driver-registrar:v2.10.0 |
|                  | netapp/trident-operator:24.02.0 (optional)
```

#### 6. Ensure that trident-operator 24.02.0 is available:

```
helm search repo netapp-trident/trident-operator --versions
```

Response:

NAME	CHART VERSION	APP VERSION	
DESCRIPTION			
netapp-trident/trident-operator	100.2402.0	24.02.0	A

7. Use `helm install` and run one of the following options that include these settings:
- A name for your deployment location
  - The Astra Trident version
  - The name of the Astra Control Provisioner image
  - The flag to enable the provisioner
  - (Optional) A local registry path. If you are using a local registry, your [Trident images](#) can be located in one registry or different registries, but all CSI images must be located in the same registry.
  - The Trident namespace

### Options

- Images without a registry

```
helm install trident netapp-trident/trident-operator --version
100.2402.0 --set acpImage=cr.astra.netapp.io/astra/trident-
acp:24.02.0 --set enableACP=true --set operatorImage=netapp/trident-
operator:24.02.0 --set
tridentAutosupportImage=docker.io/netapp/trident-autosupport:24.02
--set tridentImage=netapp/trident:24.02.0 --namespace trident
```

- Images in one or more registries

```
helm install trident netapp-trident/trident-operator --version
100.2402.0 --set acpImage=<your-registry>:<acp image> --set
enableACP=true --set imageRegistry=<your-registry>/sig-storage --set
operatorImage=netapp/trident-operator:24.02.0 --set
tridentAutosupportImage=docker.io/netapp/trident-autosupport:24.02
--set tridentImage=netapp/trident:24.02.0 --namespace trident
```

You can use `helm list` to review installation details such as name, namespace, chart, status, app version, and revision number.



If you have any issues deploying Trident using Helm, run this command to fully uninstall Astra Trident:

```
./tridentctl uninstall -n trident
```

**Do not** [completely remove Astra Trident CRDs](#) as part of your uninstall before attempting to enable Astra Control Provisioner again.

## Result

Astra Control Provisioner functionality is enabled and you can use any features available for the version you are running.

After Astra Control Provisioner is installed, the cluster hosting the provisioner in the Astra Control UI will show an `ACP version` rather than `Trident version` field and current installed version number.

CLUSTER STATUS

Available

Version v1.24.9+rke2r2	Managed 2024/03/15 17:32 UTC	Kube-system namespace UID <div></div>	ACP Version <div></div>
Private route identifier <div>...</div>	Cloud instance private	Default bucket astra-bucket1 (inherited)	

Overview

Namespaces

Storage

Activity

## For more information

- [Astra Trident upgrades documentation](#)

# Unmanage apps and clusters

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

## Stop managing an app

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

When you unmanage an app:

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

## Steps

- From the left navigation bar, select **Applications**.
- Select the app.

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

### Result

Astra Control stops managing the app.

## Stop managing a cluster

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



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

As a best practice, we recommend that you remove the cluster from Astra Control before you delete it through GCP.

When you unmanage a cluster:

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

### Steps

1. Select **Clusters**.
2. Select the checkbox for the cluster that you no longer want to manage.
3. From the options menu in the **Actions** column, select **Unmanage**.
4. Confirm that you want to unmanage the cluster and then select **Yes, unmanage**.

### Result

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

## Deleting clusters from your cloud provider

Before you delete a Kubernetes cluster that has persistent volumes (PV) residing on NetApp storage classes, you need to first delete the persistent volume claims (PVC) following one of the methods below. Deleting the PVC and PV before deleting the cluster ensures that you don't receive unexpected bills from your cloud provider.

- **Method #1:** Delete the application workload namespaces from the cluster. Do *not* delete the Trident namespace.
- **Method #2:** Delete the PVCs and the pods, or the deployment where the PVs are mounted.

When you manage a Kubernetes cluster from Astra Control, applications on that cluster use your cloud provider as the storage backend for persistent volumes. If you delete the cluster from your cloud provider without first removing the PVs, the backend volumes are *not* deleted along with the cluster.

Using one of the above methods will delete the corresponding PVs from your cluster. Make sure that there are no PVs residing on NetApp storage classes on the cluster before you delete it.

If you didn't delete the persistent volumes before you deleted the cluster, then you'll need to manually delete the backend volumes from your cloud provider.

## Deploy a self-managed instance of Astra Control

If you want a self-managed instance of Astra Control that resides inside your network, you can deploy Astra Control Center directly from Astra Control Service.

### Steps

1. In the Getting Started area of the Dashboard, select **Deploy a self-managed instance of Astra Control**.
2. Do one of the following:
  - Generate a new API token by selecting **Generate**.
  - Paste in an existing Astra Control REST API token. Refer to the [Astra Automation documentation](#) for guidance on generating an API token.
3. Follow the instructions in the **Deploy Astra Control Center** window.

# Use Astra Control Provisioner

## Configure storage backend encryption

Using Astra Control Provisioner, you can improve data access security by enabling encryption for the traffic between your managed cluster and the storage backend.

Astra Control Provisioner supports Kerberos encryption for two types of storage backends:

- **On-premises ONTAP** - Astra Control Provisioner supports Kerberos encryption over NFSv3 and NFSv4 connections from Red Hat OpenShift and upstream Kubernetes clusters to on-premises ONTAP volumes.
- **Azure NetApp Files** - Astra Control Provisioner supports Kerberos encryption over NFSv4.1 connections from upstream Kubernetes clusters to Azure NetApp Files volumes.

You can create, delete, resize, snapshot, clone, read-only clone, and import volumes that use NFS encryption.

## Configure in-flight Kerberos encryption with on-premises ONTAP volumes

You can enable Kerberos encryption on the storage traffic between your managed cluster and an on-premises ONTAP storage backend.



Kerberos encryption for NFS traffic with on-premises ONTAP storage backends is only supported using the `ontap-nas` storage driver.

### Before you begin

- Ensure that you have [enabled Astra Control Provisioner](#) on the managed cluster.
- Ensure that you have access to the `tridentctl` utility.
- Ensure you have administrator access to the ONTAP storage backend.
- Ensure you know the name of the volume or volumes you will be sharing from the ONTAP storage backend.
- Ensure that you have prepared the ONTAP storage VM to support Kerberos encryption for NFS volumes. Refer to [Enable Kerberos on a data LIF](#) for instructions.
- Ensure that any NFSv4 volumes you use with Kerberos encryption are configured correctly. Refer to the NetApp NFSv4 Domain Configuration section (page 13) of the [NetApp NFSv4 Enhancements and Best Practices Guide](#).

### Add or modify ONTAP export policies

You need to add rules to existing ONTAP export policies or create new export policies that support Kerberos encryption for the ONTAP storage VM root volume as well as any ONTAP volumes shared with the upstream Kubernetes cluster. The export policy rules you add, or new export policies you create, need to support the following access protocols and access permissions:

#### Access protocols

Configure the export policy with NFS, NFSv3, and NFSv4 access protocols.

#### Access details

You can configure one of three different versions of Kerberos encryption, depending on your needs for the

volume:

- **Kerberos 5** - (authentication and encryption)
- **Kerberos 5i** - (authentication and encryption with identity protection)
- **Kerberos 5p** - (authentication and encryption with identity and privacy protection)

Configure the ONTAP export policy rule with the appropriate access permissions. For example, if clusters will be mounting the NFS volumes with a mixture of Kerberos 5i and Kerberos 5p encryption, use the following access settings:

Type	Read-only access	Read/Write access	Superuser access
UNIX	Enabled	Enabled	Enabled
Kerberos 5i	Enabled	Enabled	Enabled
Kerberos 5p	Enabled	Enabled	Enabled

Refer to the following documentation for how to create ONTAP export policies and export policy rules:

- [Create an export policy](#)
- [Add a rule to an export policy](#)

## Create a storage backend

You can create an Astra Control Provisioner storage backend configuration that includes Kerberos encryption capability.

### About this task

When you create a storage backend configuration file that configures Kerberos encryption, you can specify one of three different versions of Kerberos encryption using the `spec.nfsMountOptions` parameter:

- `spec.nfsMountOptions: sec=krb5` (authentication and encryption)
- `spec.nfsMountOptions: sec=krb5i` (authentication and encryption with identity protection)
- `spec.nfsMountOptions: sec=krb5p` (authentication and encryption with identity and privacy protection)

Specify only one Kerberos level. If you specify more than one Kerberos encryption level in the parameter list, only the first option is used.

### Steps

1. On the managed cluster, create a storage backend configuration file using the following example. Replace values in brackets <> with information from your environment:



```

apiVersion: v1
kind: Secret
metadata:
  name: backend-ontap-nas-secret
type: Opaque
stringData:
  clientID: <CLIENT_ID>
  clientSecret: <CLIENT_SECRET>
---
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-ontap-nas
spec:
  version: 1
  storageDriverName: "ontap-nas"
  managementLIF: <STORAGE_VM_MGMT_LIF_IP_ADDRESS>
  dataLIF: <PROTOCOL_LIF_FQDN_OR_IP_ADDRESS>
  svm: <STORAGE_VM_NAME>
  username: <STORAGE_VM_USERNAME_CREDENTIAL>
  password: <STORAGE_VM_PASSWORD_CREDENTIAL>
  nasType: nfs
  nfsMountOptions: ["sec=krb5i"] #can be krb5, krb5i, or krb5p
  qtreesPerFlexvol:
  credentials:
    name: backend-ontap-nas-secret

```

2. Use the configuration file you created in the previous step to create the backend:

```
tridentctl create backend -f <backend-configuration-file>
```

If the backend creation fails, something is wrong with the backend configuration. You can view the logs to determine the cause by running the following command:

```
tridentctl logs
```

After you identify and correct the problem with the configuration file, you can run the create command again.

## Create a storage class

You can create a storage class to provision volumes with Kerberos encryption.

## About this task

When you create a storage class object, you can specify one of three different versions of Kerberos encryption using the `mountOptions` parameter:

- `mountOptions: sec=krb5` (authentication and encryption)
- `mountOptions: sec=krb5i` (authentication and encryption with identity protection)
- `mountOptions: sec=krb5p` (authentication and encryption with identity and privacy protection)

Specify only one Kerberos level. If you specify more than one Kerberos encryption level in the parameter list, only the first option is used. If the level of encryption you specified in the storage backend configuration is different than the level you specify in the storage class object, the storage class object takes precedence.

## Steps

1. Create a StorageClass Kubernetes object, using the following example:

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontap-nas-sc
provisioner: csi.trident.netapp.io
mountOptions: ["sec=krb5i"] #can be krb5, krb5i, or krb5p
parameters:
  backendType: "ontap-nas"
  storagePools: "ontapnas_pool"
  trident.netapp.io/nasType: "nfs"
allowVolumeExpansion: True
```

2. Create the storage class:

```
kubectl create -f sample-input/storage-class-ontap-nas-sc.yaml
```

3. Make sure that the storage class has been created:

```
kubectl get sc ontap-nas-sc
```

You should see output similar to the following:

NAME	PROVISIONER	AGE
ontap-nas-sc	csi.trident.netapp.io	15h

## Provision volumes

After you create a storage backend and a storage class, you can now provision a volume. Refer to these instructions for [provisioning a volume](#).

## Configure in-flight Kerberos encryption with Azure NetApp Files volumes

You can enable Kerberos encryption on the storage traffic between your managed cluster and a single Azure NetApp Files storage backend or a virtual pool of Azure NetApp Files storage backends.

### Before you begin

- Ensure that you have enabled Astra Control Provisioner on the managed Red Hat OpenShift cluster. Refer to [Enable Astra Control Provisioner](#) for instructions.
- Ensure that you have access to the `tridentctl` utility.
- Ensure that you have prepared the Azure NetApp Files storage backend for Kerberos encryption by noting the requirements and following the instructions in [Azure NetApp Files documentation](#).
- Ensure that any NFSv4 volumes you use with Kerberos encryption are configured correctly. Refer to the NetApp NFSv4 Domain Configuration section (page 13) of the [NetApp NFSv4 Enhancements and Best Practices Guide](#).

### Create a storage backend

You can create an Azure NetApp Files storage backend configuration that includes Kerberos encryption capability.

### About this task

When you create a storage backend configuration file that configures Kerberos encryption, you can define it so that it should be applied at one of two possible levels:

- The **storage backend level** using the `spec.kerberos` field
- The **virtual pool level** using the `spec.storage.kerberos` field

When you define the configuration at the virtual pool level, the pool is selected using the label in the storage class.

At either level, you can specify one of three different versions of Kerberos encryption:

- `kerberos: sec=krb5` (authentication and encryption)
- `kerberos: sec=krb5i` (authentication and encryption with identity protection)
- `kerberos: sec=krb5p` (authentication and encryption with identity and privacy protection)

### Steps

1. On the managed cluster, create a storage backend configuration file using one of the following examples, depending on where you need to define the storage backend (storage backend level or virtual pool level). Replace values in brackets <> with information from your environment:

### Storage backend level example

```
apiVersion: v1
kind: Secret
metadata:
  name: backend-tbc-anf-secret
type: Opaque
stringData:
  clientID: <CLIENT_ID>
  clientSecret: <CLIENT_SECRET>
---
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-anf
spec:
  version: 1
  storageDriverName: azure-netapp-files
  subscriptionID: <SUBSCRIPTION_ID>
  tenantID: <TENANT_ID>
  location: <AZURE_REGION_LOCATION>
  serviceLevel: Standard
  networkFeatures: Standard
  capacityPools: <CAPACITY_POOL>
  resourceGroups: <RESOURCE_GROUP>
  netappAccounts: <NETAPP_ACCOUNT>
  virtualNetwork: <VIRTUAL_NETWORK>
  subnet: <SUBNET>
  nasType: nfs
  kerberos: sec=krb5i #can be krb5, krb5i, or krb5p
  credentials:
    name: backend-tbc-anf-secret
```

### Virtual pool level example

```

apiVersion: v1
kind: Secret
metadata:
  name: backend-tbc-anf-secret
type: Opaque
stringData:
  clientID: <CLIENT_ID>
  clientSecret: <CLIENT_SECRET>
---
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-anf
spec:
  version: 1
  storageDriverName: azure-netapp-files
  subscriptionID: <SUBSCRIPTION_ID>
  tenantID: <TENANT_ID>
  location: <AZURE_REGION_LOCATION>
  serviceLevel: Standard
  networkFeatures: Standard
  capacityPools: <CAPACITY_POOL>
  resourceGroups: <RESOURCE_GROUP>
  netappAccounts: <NETAPP_ACCOUNT>
  virtualNetwork: <VIRTUAL_NETWORK>
  subnet: <SUBNET>
  nasType: nfs
  storage:
    - labels:
        type: encryption
        kerberos: sec=krb5i #can be krb5, krb5i, or krb5p
  credentials:
    name: backend-tbc-anf-secret

```

2. Use the configuration file you created in the previous step to create the backend:

```
tridentctl create backend -f <backend-configuration-file>
```

If the backend creation fails, something is wrong with the backend configuration. You can view the logs to determine the cause by running the following command:

```
tridentctl logs
```

After you identify and correct the problem with the configuration file, you can run the create command again.

## Create a storage class

You can create a storage class to provision volumes with Kerberos encryption.

### Steps

1. Create a StorageClass Kubernetes object, using the following example:

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: anf-sc-nfs
provisioner: csi.trident.netapp.io
parameters:
  backendType: "azure-netapp-files"
  trident.netapp.io/nasType: "nfs"
  selector: "type=encryption"
```

2. Create the storage class:

```
kubectl create -f sample-input/storage-class-anf-sc-nfs.yaml
```

3. Make sure that the storage class has been created:

```
kubectl get sc anf-sc-nfs
```

You should see output similar to the following:

NAME	PROVISIONER	AGE
anf-sc-nfs	csi.trident.netapp.io	15h

## Provision volumes

After you create a storage backend and a storage class, you can now provision a volume. Refer to these instructions for [provisioning a volume](#).

## Recover volume data using a snapshot

Astra Control Provisioner provides rapid, in-place volume restoration from a snapshot using the `TridentActionSnapshotRestore` (TASR) CR. This CR functions as an imperative Kubernetes action and does not persist after the operation completes.

Astra Control Provisioner supports snapshot restore on the `ontap-san`, `ontap-san-economy`, `ontap-nas`, `ontap-nas-flexgroup`, `azure-netapp-files`, `gcp-cvs`, and `solidfire-san` drivers.

### Before you begin

You must have a bound PVC and available volume snapshot.

- Verify the PVC status is bound.

```
kubectl get pvc
```

- Verify the volume snapshot is ready to use.

```
kubectl get vs
```

### Steps

1. Create the TASR CR. This example creates a CR for PVC `pvc1` and volume snapshot `pvc1-snapshot`.

```
cat tasr-pvc1-snapshot.yaml

apiVersion: v1
kind: TridentActionSnapshotRestore
metadata:
  name: this-doesnt-matter
  namespace: trident
spec:
  pvcName: pvc1
  volumeSnapshotName: pvc1-snapshot
```

2. Apply the CR to restore from the snapshot. This example restores from snapshot `pvc1`.

```
kubectl create -f tasr-pvc1-snapshot.yaml

tridentactionsnapshotrestore.trident.netapp.io/this-doesnt-matter
created
```

### Results

Astra Control Provisioner restores the data from the snapshot. You can verify the snapshot restore status.

```
kubectl get tasr -o yaml

apiVersion: v1
items:
- apiVersion: trident.netapp.io/v1
  kind: TridentActionSnapshotRestore
  metadata:
    creationTimestamp: "2023-04-14T00:20:33Z"
    generation: 3
    name: this-doesnt-matter
    namespace: trident
    resourceVersion: "3453847"
    uid: <uid>
  spec:
    pvcName: pvc1
    volumeSnapshotName: pvc1-snapshot
  status:
    startTime: "2023-04-14T00:20:34Z"
    completionTime: "2023-04-14T00:20:37Z"
    state: Succeeded
kind: List
metadata:
  resourceVersion: ""
```



- In most cases, Astra Control Provisioner will not automatically retry the operation in case of failure. You will need to perform the operation again.
- Kubernetes users without admin access might have to be granted permission by the admin to create a TASR CR in their application namespace.

## Replicate volumes using SnapMirror

Using Astra Control Provisioner, you can create mirror relationships between a source volume on one cluster and the destination volume on the peered cluster for replicating data for disaster recovery. You can use a namespaced Custom Resource Definition (CRD) to perform the following operations:

- Create mirror relationships between volumes (PVCs)
- Remove mirror relationships between volumes
- Break the mirror relationships
- Promote the secondary volume during disaster conditions (failovers)
- Perform lossless transition of applications from cluster to cluster (during planned failovers or migrations)



## Replication prerequisites

Ensure that the following prerequisites are met before you begin:

### ONTAP clusters

- **Astra Control Provisioner:** Astra Control Provisioner version 23.10 or later must exist on both the source and destination Kubernetes clusters that utilize ONTAP as a backend.
- **Licenses:** ONTAP SnapMirror asynchronous licenses using the Data Protection bundle must be enabled on both the source and destination ONTAP clusters. Refer to [SnapMirror licensing overview in ONTAP](#) for more information.

### Peering

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



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

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

### Supported drivers

- Volume replication is supported for the `ontap-nas` and `ontap-san` drivers.

## Create a mirrored PVC

Follow these steps and use the CRD examples to create mirror relationship between primary and secondary volumes.

### Steps

1. Perform the following steps on the primary Kubernetes cluster:
  - a. Create a StorageClass object with the `trident.netapp.io/replication: true` parameter.

#### Example

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: csi-nas
provisioner: csi.trident.netapp.io
parameters:
  backendType: "ontap-nas"
  fsType: "nfs"
  trident.netapp.io/replication: "true"
```

- b. Create a PVC with previously created StorageClass.

### Example

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: csi-nas
spec:
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 1Gi
  storageClassName: csi-nas
```

- c. Create a MirrorRelationship CR with local information.

### Example

```
kind: TridentMirrorRelationship
apiVersion: trident.netapp.io/v1
metadata:
  name: csi-nas
spec:
  state: promoted
  volumeMappings:
    - localPVCName: csi-nas
```

Astra Control Provisioner fetches the internal information for the volume and the volume's current data protection (DP) state, then populates the status field of the MirrorRelationship.

- d. Get the TridentMirrorRelationship CR to obtain the internal name and SVM of the PVC.

```
kubect1 get tmr csi-nas
```

```

kind: TridentMirrorRelationship
apiVersion: trident.netapp.io/v1
metadata:
  name: csi-nas
  generation: 1
spec:
  state: promoted
  volumeMappings:
    - localPVCName: csi-nas
status:
  conditions:
    - state: promoted
    localVolumeHandle:
"datavserver:trident_pvc_3bedd23c_46a8_4384_b12b_3c38b313c1e1"
    localPVCName: csi-nas
    observedGeneration: 1

```

2. Perform the following steps on the secondary Kubernetes cluster:

- a. Create a StorageClass with the trident.netapp.io/replication: true parameter.

**Example**

```

apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: csi-nas
provisioner: csi.trident.netapp.io
parameters:
  trident.netapp.io/replication: true

```

- b. Create a MirrorRelationship CR with destination and source information.

**Example**

```

kind: TridentMirrorRelationship
apiVersion: trident.netapp.io/v1
metadata:
  name: csi-nas
spec:
  state: established
  volumeMappings:
    - localPVCName: csi-nas
      remoteVolumeHandle:
"datavserver:trident_pvc_3bedd23c_46a8_4384_b12b_3c38b313c1e1"

```

Astra Control Provisioner will create a SnapMirror relationship with the configured relationship policy name (or default for ONTAP) and initialize it.

- c. Create a PVC with previously created StorageClass to act as the secondary (SnapMirror destination).

#### Example

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: csi-nas
  annotations:
    trident.netapp.io/mirrorRelationship: csi-nas
spec:
  accessModes:
    - ReadWriteMany
resources:
  requests:
    storage: 1Gi
storageClassName: csi-nas
```

Astra Control Provisioner will check for the TridentMirrorRelationship CRD and fail to create the volume if the relationship does not exist. If the relationship exists, Astra Control Provisioner will ensure the new FlexVol volume is placed onto an SVM that is peered with the remote SVM defined in the MirrorRelationship.

## Volume Replication States

A Trident Mirror Relationship (TMR) is a CRD that represents one end of a replication relationship between PVCs. The destination TMR has a state, which tells Astra Control Provisioner what the desired state is. The destination TMR has the following states:

- **Established:** the local PVC is the destination volume of a mirror relationship, and this is a new relationship.
- **Promoted:** the local PVC is ReadWrite and mountable, with no mirror relationship currently in effect.
- **Reestablished:** the local PVC is the destination volume of a mirror relationship and was also previously in that mirror relationship.
  - The reestablished state must be used if the destination volume was ever in a relationship with the source volume because it overwrites the destination volume contents.
  - The reestablished state will fail if the volume was not previously in a relationship with the source.

## Promote secondary PVC during an unplanned failover

Perform the following step on the secondary Kubernetes cluster:

- Update the `spec.state` field of TridentMirrorRelationship to `promoted`.

## Promote secondary PVC during a planned failover

During a planned failover (migration), perform the following steps to promote the secondary PVC:

### Steps

1. On the primary Kubernetes cluster, create a snapshot of the PVC and wait until the snapshot is created.
2. On the primary Kubernetes cluster, create the SnapshotInfo CR to obtain internal details.

### Example

```
kind: SnapshotInfo
apiVersion: trident.netapp.io/v1
metadata:
  name: csi-nas
spec:
  snapshot-name: csi-nas-snapshot
```

3. On secondary Kubernetes cluster, update the *spec.state* field of the *TridentMirrorRelationship* CR to *promoted* and *spec.promotedSnapshotHandle* to be the internalName of the snapshot.
4. On secondary Kubernetes cluster, confirm the status (status.state field) of *TridentMirrorRelationship* to promoted.

## Restore a mirror relationship after a failover

Before restoring a mirror relationship, choose the side that you want to make as the new primary.

### Steps

1. On the secondary Kubernetes cluster, ensure that the values for the *spec.remoteVolumeHandle* field on the *TridentMirrorRelationship* is updated.
2. On secondary Kubernetes cluster, update the *spec.mirror* field of *TridentMirrorRelationship* to *reestablished*.

## Additional operations

Astra Control Provisioner supports the following operations on the primary and secondary volumes:

### Replicate primary PVC to a new secondary PVC

Ensure that you already have a primary PVC and a secondary PVC.

### Steps

1. Delete the *PersistentVolumeClaim* and *TridentMirrorRelationship* CRDs from the established secondary (destination) cluster.
2. Delete the *TridentMirrorRelationship* CRD from the primary (source) cluster.
3. Create a new *TridentMirrorRelationship* CRD on the primary (source) cluster for the new secondary (destination) PVC you want to establish.

## Resize a mirrored, primary or secondary PVC

The PVC can be resized as normal, ONTAP will automatically expand any destination flexvols if the amount of data exceeds the current size.

## Remove replication from a PVC

To remove replication, perform one of the following operations on the current secondary volume:

- Delete the MirrorRelationship on the secondary PVC. This breaks the replication relationship.
- Or, update the spec.state field to *promoted*.

## Delete a PVC (that was previously mirrored)

Astra Control Provisioner checks for replicated PVCs, and releases the replication relationship before attempting to delete the volume.

## Delete a TMR

Deleting a TMR on one side of a mirrored relationship causes the remaining TMR to transition to *promoted* state before Astra Control Provisioner completes the deletion. If the TMR selected for deletion is already in *promoted* state, there is no existing mirror relationship and the TMR will be removed and Astra Control Provisioner will promote the local PVC to *ReadWrite*. This deletion releases SnapMirror metadata for the local volume in ONTAP. If this volume is used in a mirror relationship in the future, it must use a new TMR with an *established* volume replication state when creating the new mirror relationship.

## Update mirror relationships when ONTAP is online

Mirror relationships can be updated any time after they are established. You can use the `state: promoted` or `state: reestablished` fields to update the relationships.

When promoting a destination volume to a regular ReadWrite volume, you can use *promotedSnapshotHandle* to specify a specific snapshot to restore the current volume to.

## Update mirror relationships when ONTAP is offline

You can use a CRD to perform a SnapMirror update without Astra Control having direct connectivity to the ONTAP cluster. Refer to the following example format of the TridentActionMirrorUpdate:

### Example

```
apiVersion: trident.netapp.io/v1
kind: TridentActionMirrorUpdate
metadata:
  name: update-mirror-b
spec:
  snapshotHandle: "pvc-1234/snapshot-1234"
  tridentMirrorRelationshipName: mirror-b
```

`status.state` reflects the state of the TridentActionMirrorUpdate CRD. It can take a value from *Succeeded*, *In Progress*, or *Failed*.

# 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 learn more, [go to the Astra automation docs](#).

# Knowledge and support

## Register for support

Astra Control attempts to automatically register your account for support when you set up your account. If it can't, then you can manually register for support yourself. Support registration is required to obtain help from NetApp technical support.

### Verify your support registration

Astra Control includes a Support Status field that enables you to confirm your support registration.

#### Steps

1. Select **Support**.
2. Take a look at the Support Status field.

The Support Status starts off as "Not Registered" but then moves to "In-Progress" and finally to "Registered" once complete.

This support registration status is polled every 15 minutes. New NetApp customers could take up to the next business day to complete onboarding and support registration. If the serial number doesn't show "Registered" within 48 hours, you can reach out to NetApp using [astra.feedback@netapp.com](mailto:astra.feedback@netapp.com) or register manually from <https://register.netapp.com>.

### Obtain your serial number

When you register for an account, Astra Control uses the information that you provided about your company to generate a 20-digit NetApp serial number that starts with "941."

The NetApp serial number represents your Astra Control account. You'll need to use this serial number when opening a web ticket.

You can find your serial number in the Astra Control interface from the **Support** page.

### Activate support entitlement

If Astra Control was unable to automatically register your account for support, then you must register the NetApp serial number associated with Astra Control to activate support entitlement. We offer 2 options for support registration:

1. Current NetApp customer with existing NetApp Support Site (NSS) SSO account
2. New NetApp customer with no existing NetApp Support Site (NSS) SSO account

#### Option 1: Current NetApp customer with an existing NetApp Support Site (NSS) account

##### Steps

1. Navigate to the [Cloud Data Services Support Registration](#) page.
2. Select **I am already registered as a NetApp customer**.
3. Enter your NetApp Support Site credentials to log in.



The Existing Customer Registration page appears.

4. Complete the required information on the form:
  - a. Enter your name, company, and email address.
  - b. Select **Astra Control Service** as the product line.
  - c. Select a billing provider.
  - d. Enter your serial number.
  - e. Select **Submit**.

### Result

You should be redirected to a "Registration Submitted Successfully" page. The email address associated with your registration will receive an email within a couple minutes stating that "your product is now eligible for support."

This is a one-time support registration for the applicable serial number.

## Option 2: New NetApp customer with no existing NetApp Support Site (NSS) account

### Steps

1. Navigate to the [Cloud Data Services Support Registration](#) page.
2. Select **I am not a registered NetApp Customer**.

The New Customer Registration page appears.

3. Complete the required information on the form:
  - a. Enter your name, company information, and contact details.
  - b. Select **Astra Control Service** as the Product Line.
  - c. Select a billing provider.
  - d. Enter your serial number.
  - e. Enter the captcha value.
  - f. Select the check box to confirm that you have read the NetApp Privacy Policy.
  - g. Select **Submit**.

You will receive a confirmation email from your submitted registration. If no errors occur, you will be redirected to a "Registration Submitted Successfully" page. You will also receive an email within an hour stating that "your product is now eligible for support".

This is a one-time support registration for the applicable serial number.

4. As a new NetApp customer, you also need to create a NetApp Support Site (NSS) user account for future support activations and for access to the support portal for technical support chat and web ticketing.

Go to the [NetApp Support Registration site](#) to perform this task. You can provide your newly registered Astra Control serial number to expedite the process.

# Troubleshooting

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

<https://kb.netapp.com/Cloud/Astra/Control>

## For more information

- [Troubleshooting](#)

## Get help

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

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 UI by selecting the **Support** tab from the main menu.

## Self support

These options are available for free 24x7:

- [Knowledge base](#)

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

- Documentation

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

- [Get help via Discord](#)

Go to Astra in The Pub category to connect with peers and experts.

- Feedback email

Send an email to [astra.feedback@netapp.com](mailto:astra.feedback@netapp.com) to let us know your thoughts, ideas, or concerns.

## Subscription support

In addition to the self-support options above, you can work with a NetApp Support Engineer to resolve any issues after you [activate support for your NetApp serial number](#).

Once your Astra Control serial number is activated, you can access NetApp technical support resources by creating a [Support ticket](#).

Select **Cloud Data Services > Astra**.

Learn more about your serial number.

## Create Case

1 Select System    2 Problem Details    3 Contact Info

SERIAL NUMBER	SYSTEM NAME	MODEL	PRODUCT SERIES
9419999999999999999		SREG-ASTRA-SAAS	CLOUD

PRIORITY ?

- ☐ P4 - General Technical questions or request for information
- ☒ P3 - Occasional disruption or problem
- ☐ P2 - Serious or repetitive disruption/very poor performance
- ☐ P1 - System not serving data

PROBLEM CATEGORY

?

Cloud Services > Project Astra

### PROBLEM DESCRIPTION

--

Please briefly describe your problem here (2000 characters maximum), you will have the opportunity to fully define and add more details to your problem later in the case creation process

# Frequently asked questions

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

## Overview

Astra Control aims to simplify your application data lifecycle management operations for Kubernetes-native applications. Astra Control Service supports Kubernetes clusters running on multiple cloud provider environments.

The following sections provide answers to some additional questions that you might come across as you use Astra Control. For any additional clarifications, please reach out to [astra.feedback@netapp.com](mailto:astra.feedback@netapp.com)

## Access to Astra Control

### **Why do I need to provide so many details when registering for Astra Control?**

Astra Control requires accurate customer information when registering. This information is required to go through a Global Trade Compliance (GTC) check.

### **Why am I getting a "Registration Failed" error when registering for Astra Control?**

Astra Control requires you to provide accurate customer information in the onboarding section. You will get a "Registration Failed" error if you provided incorrect information. Other accounts that you are a member of also get locked.

### **What's the Astra Control Service URL?**

You can access Astra Control Service at <https://astra.netapp.io>.

### **I sent an email invitation to a colleague, but they haven't received it. What should I do?**

Ask them to check their spam folder for an email from [do-not-reply@netapp.com](mailto:do-not-reply@netapp.com), or search their inbox for "invitation." You can also remove the user and attempt to re-add them.

### **I upgraded to the Premium PayGO Plan from the Free Plan. Will I get charged for the first 10 namespaces?**

Yes. After upgrading to the Premium Plan, Astra Control starts charging you for all managed namespaces in your account.

### **I upgraded to the Premium PayGO Plan in the middle of a month. Will I get charged for the entire month?**

No. Billing starts from the time that you upgraded to the Premium Plan.

### **I am using the Free Plan, will I get charged for the Persistent Volume Claims?**

Yes, you will be charged for the persistent volumes used by the clusters from your cloud provider.

## Registering Kubernetes clusters

### **Do I need to install CSI drivers on my cluster before adding it to Astra Control Service?**

No. When your cluster is added to Astra Control, the service will automatically install the Astra Trident Container Storage Interface (CSI) driver on the Kubernetes cluster. This CSI driver is used to provision persistent volumes for clusters backed by your cloud provider.

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

New worker nodes can be added to existing pools, or new pools can be created as long as they are the

COS\_CONTAINERD image type. 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.

## Registering Elastic Kubernetes Service (EKS) clusters

### Can I add a private EKS cluster to Astra Control Service?

Yes, you can add private EKS clusters to Astra Control Service. To add a private EKS cluster, refer to [Start managing Kubernetes clusters from Astra Control Service](#).

## Registering Azure Kubernetes Service (AKS) clusters

### Can I add a private AKS cluster to Astra Control Service?

Yes, you can add private AKS clusters to Astra Control Service. To add a private AKS cluster, refer to [Start managing Kubernetes clusters from Astra Control Service](#).

### Can I use Active Directory to manage authentication for my AKS clusters?

Yes, you can configure your AKS clusters to use Azure Active Directory (Azure AD) for authentication and identity management. When you create the cluster, follow the instructions in the [official documentation](#) to configure the cluster to use Azure AD. You'll need to make sure your clusters meet the requirements for AKS-managed Azure AD integration.

## Registering Google Kubernetes Engine (GKE) clusters

### Can I add a private GKE cluster to Astra Control Service?

Yes, you can add private GKE clusters to Astra Control Service. To add a private GKE cluster, refer to [Start managing Kubernetes clusters from Astra Control Service](#).

Private GKE clusters must have the [authorized networks](#) set to allow the Astra Control IP address:

52.188.218.166/32

### Can my GKE cluster reside on a shared VPC?

Yes. Astra Control can manage clusters that reside in a shared VPC. [Learn how to set up the Astra service account for a shared VPC configuration](#).

### Where can I find my service account credentials on GCP?

After you log in to the [Google Cloud Console](#), your service account details will be in the **IAM and Admin** section. For more details, refer to [how to set up Google Cloud for Astra Control](#).

### I would like to add different GKE clusters from different GCP projects. Is this supported in Astra Control?

No, this isn't a supported configuration. Only a single GCP project is supported.

## Removing clusters

### How do I properly unregister, bring down a cluster, and delete the associated volumes?

1. [Unmanage the applications from Astra Control](#).
2. [Unregister the cluster from Astra Control](#).
3. [Delete the persistent volume claims](#).

4. Delete the cluster.

### What happens to my applications and data after removing the 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. Volume snapshot data stored within the storage backend will not be removed. Persistent Storage backups created by Astra Control will remain within your cloud provider's object store, but they are unavailable for restore.



Always remove a cluster from Astra Control before you delete it through GCP. Deleting a cluster from GCP while it's still being managed by Astra Control can cause problems for your Astra Control account.

### Is Astra Control Provisioner automatically uninstalled from a cluster when I unmanage it?

When you unmanage a cluster from Astra Control Center, Astra Control Provisioner or Astra Trident isn't automatically uninstalled from the cluster. To uninstall Astra Control Provisioner and its components or Astra Trident, you'll need to [follow these steps to uninstall the Astra Trident instance that contains the Astra Control Provisioner service](#).

## Managing applications

### Can Astra Control deploy an application?

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

### I don't see any of my application's PVCs bound to GCP CVS. What's wrong?

The Astra Trident operator sets the default storage class to `netapp-cvs-perf-premium` after it's successfully added to Astra Control. When an application's PVCs are not bound to Cloud Volumes Service for Google Cloud, there are a few steps that you can take:

- Run `kubectl get sc` and check the default storage class.
- Check the yaml file or Helm chart that was used to deploy the application and see if a different storage class is defined.
- GKE version 1.24 and later does not support Docker-based node images. Check to make sure that the worker node image type in GKE is `COS_CONTAINERD` and that the NFS mount succeeded.

### 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.

## Data management operations

### Where does Astra Control create the object store bucket?

The geography of the first managed cluster determines the location of the object store. For example, if the first cluster that you add is in a European zone, then the bucket is created in that same geography. If needed, you can [add additional buckets](#).

### 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 performing another process. If these snapshots are more than a few minutes old, you can safely delete them.

**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 my cloud provider?**

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

# Astra Control Provisioner

**How are Astra Control Provisioner’s storage provisioning features different from those in Astra Trident?**

Astra Control Provisioner, as part of Astra Control, supports a superset of storage provisioning features that are unavailable in open-source Astra Trident. These features are in addition to all features that are available to the open-source Trident.

**Is Astra Control Provisioner replacing Astra Trident?**

Astra Control Provisioner has replaced Astra Trident as storage provisioner and orchestrator in the Astra Control architecture. Astra Control users should [enable Astra Control Provisioner](#) to use Astra Control. Astra Trident will continue to be supported in this release but will not be supported in future releases. Astra Trident will remain open source and be released, maintained, supported, and updated with new CSI and other features from NetApp. Only Astra Control Provisioner, however, that contains Astra Trident CSI functionality along with extended storage management capabilities can be used with coming Astra Control releases.

**Do I have to pay for Astra Trident?**

No. Astra Trident will continue to be open source and free to download. Astra Control Provisioner functionality use now requires an Astra Control license.

**Can I use the storage management and provisioning features in Astra Control without installing and using all of Astra Control?**

Yes, you can upgrade to Astra Control Provisioner and use its functionality even if you do not want to consume the complete feature set of Astra Control data management functionality.

**How do I know if Astra Control Provisioner has replaced Astra Trident on my cluster?**

After Astra Control Provisioner is installed, the host cluster in the Astra Control UI will show an `ACP version` rather than `Trident version` field and current installed version number.

⚡ CLUSTER STATUS

✔ Available

Version v1.24.9+rke2r2	Managed 2024/03/15 17:32 UTC	Kube-system namespace UID <div></div>	ACP Version <div></div>
Private route identifier <div>... ⓘ</div>	Cloud instance private ⓘ	Default bucket astra-bucket1 (inherited) ⓘ	

Overview

Namespaces

Storage

Activity

If you don’t have access to the UI, you can confirm successful installation using the following methods:

## Astra Trident operator

Verify the `trident-acp` container is running and that `acpVersion` is `23.10.0` or later with a status of `Installed`:

```
kubectl get torc -o yaml
```

Response:

```
status:
  acpVersion: 23.10.0
  currentInstallationParams:
    ...
    acpImage: <my_custom_registry>/trident-acp:v23.10.0
    enableACP: "true"
    ...
  ...
  status: Installed
```

## tridentctl

Confirm that Astra Control Provisioner has been enabled:

```
./tridentctl -n trident version
```

Response:

```
+-----+-----+-----+ | SERVER VERSION |
CLIENT VERSION | ACP VERSION | +-----+-----+
+-----+ | 23.10.0 | 23.10.0 | 23.10.0. | +-----+
+-----+-----+-----+
```



# 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](#)

## Astra Control API license

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

## Copyright information

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

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

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

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

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

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

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

## Trademark information

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