



Get started

Astra Control Center

NetApp
February 12, 2024

Table of Contents

- Get started 1
 - Astra Control Center requirements 1
 - Quick start for Astra Control Center 5
 - Installation overview 7
 - Set up Astra Control Center 59
 - Frequently asked questions for Astra Control Center 72

Get started

Astra Control Center requirements

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

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

Operational environment requirements

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

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

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

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

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



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

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

Supported storage backends

Astra Control Center supports the following storage backends.

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

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

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

[SnapMirror license information.](#)

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

To check whether your ONTAP system has the required licenses, refer to [Manage ONTAP licenses](#).

Access to the internet

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

License

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

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

To set up the license, refer to [use a 90-day evaluation license](#).

To learn more about how licenses work, see [Licensing](#).

For details about licenses needed for ONTAP storage backends, refer to [Supported storage backends](#).

Ingress for on-premises Kubernetes clusters

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



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



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

For more information, see [Set up ingress for load balancing](#).

Networking requirements

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



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

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

Supported web browsers

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

Additional requirements for application clusters

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

- **Application cluster requirements:** [Cluster management requirements](#)
 - **Managed application requirements:** [Application management requirements](#)
 - **Additional requirements for app replication:** [Replication prerequisites](#)

Google Anthos cluster requirements

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

VMware Tanzu Kubernetes Grid cluster requirements

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

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



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

What's next

View the [quick start](#) overview.

Quick start for Astra Control Center

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



Review Kubernetes cluster requirements

Ensure that your environment meets these requirements.

Kubernetes cluster

- ☐ [Ensure your environment meets operational environment requirements](#)
- ☐ [Configure ingress for load balancing on-premises Kubernetes clusters](#)

Storage integration

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

- ☐ [Create a volume snapshot class](#)

ONTAP credentials

- ☐ [Configure ONTAP credentials](#)

2

Download and install Astra Control Center

Complete these installation tasks.

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

3

Complete some initial setup tasks

Complete some basic tasks to get started.

- ☐ [Add a license](#)
- ☐ [Prepare your environment for cluster management](#)
- ☐ [Add a cluster](#)
- ☐ [Add a storage backend](#)
- ☐ [Add a bucket](#)

4

Use Astra Control Center

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

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

For more information

- [Astra Control API](#)
- [Upgrade Astra Control Center](#)
- [Get help with Astra Control](#)

Installation overview

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

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

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

- [Configure Astra Control Center after installation](#)

Install Astra Control Center using the standard process

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

Other installation procedures

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

For a demonstration of the Astra Control Center installation process, see [this video](#).

What you'll need

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

```
kubectl get apiservices
```

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

About this task

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

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



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

Steps

To install Astra Control Center, do the following steps:

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

Download and extract Astra Control Center

1. Go to the [Astra Control Center Evaluation downloads page](#) on the NetApp Support Site.
2. Download the bundle containing Astra Control Center (`astra-control-center-[version].tar.gz`).
3. (Recommended but optional) Download the certificates and signatures bundle for Astra Control Center (`astra-control-center-certs-[version].tar.gz`) to verify the signature of the bundle:

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

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

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

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

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

Install the NetApp Astra kubectl plugin

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

What you'll need

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

Steps

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



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

```
ls kubectl-astra/
```

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

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

Add the images to your local registry

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

Docker

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

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

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

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

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

Podman

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

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

- b. Log in to your registry:

```
podman login <YOUR_REGISTRY>
```

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

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

```

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

```

Podman 3

```

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

```



The image path the script creates should resemble the following, depending on your registry configuration: <https://netappdownloads.jfrog.io/docker-astra-control-prod/netapp/astra/acc/22.11.0-82/image:version>

Set up namespace and secret for registries with auth requirements

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

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



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

2. If you use a registry that requires authentication, you need to do the following:

a. Create the `netapp-acc-operator` namespace:

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

Response:

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

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



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

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

Sample response:

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



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

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

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

Sample response:

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

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

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

Response

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

Install the Astra Control Center operator

1. Change the directory:

```
cd manifests
```

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

```
vim astra_control_center_operator_deploy.yaml
```



An annotated sample YAML follows these steps.

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

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

- b. Change `[your_registry_path]` for the `kube-rbac-proxy` image to the registry path where you pushed the images in a [previous step](#).
- c. Change `[your_registry_path]` for the `acc-operator-controller-manager` image to the registry path where you pushed the images in a [previous step](#).

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

```
apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    control-plane: controller-manager
```

```

name: acc-operator-controller-manager
namespace: netapp-acc-operator
spec:
  replicas: 1
  selector:
    matchLabels:
      control-plane: controller-manager
  strategy:
    type: Recreate
  template:
    metadata:
      labels:
        control-plane: controller-manager
    spec:
      containers:
        - args:
            - --secure-listen-address=0.0.0.0:8443
            - --upstream=http://127.0.0.1:8080/
            - --logtostderr=true
            - --v=10
            image: [your_registry_path]/kube-rbac-proxy:v4.8.0
          name: kube-rbac-proxy
          ports:
            - containerPort: 8443
              name: https
        - args:
            - --health-probe-bind-address=:8081
            - --metrics-bind-address=127.0.0.1:8080
            - --leader-elect
          env:
            - name: ACCOP_LOG_LEVEL
              value: "2"
            - name: ACCOP_HELM_INSTALLTIMEOUT
              value: 5m
            image: [your_registry_path]/acc-operator:[version x.y.z]
          imagePullPolicy: IfNotPresent
          livenessProbe:
            httpGet:
              path: /healthz
              port: 8081
              initialDelaySeconds: 15
              periodSeconds: 20
          name: manager
          readinessProbe:
            httpGet:
              path: /readyz

```



```

        port: 8081
        initialDelaySeconds: 5
        periodSeconds: 10
    resources:
        limits:
            cpu: 300m
            memory: 750Mi
        requests:
            cpu: 100m
            memory: 75Mi
    securityContext:
        allowPrivilegeEscalation: false
imagePullSecrets: []
    securityContext:
        runAsUser: 65532
    terminationGracePeriodSeconds: 10

```

3. Install the Astra Control Center operator:

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

Sample response:

```

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

```

4. Verify pods are running:

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

Configure Astra Control Center

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

```
vim astra_control_center.yaml
```



An annotated sample YAML follows these steps.

2. Modify or confirm the following settings:

`accountName`

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

`astraVersion`

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

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

autoSupport

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

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

email

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

firstName

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

lastName

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

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

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

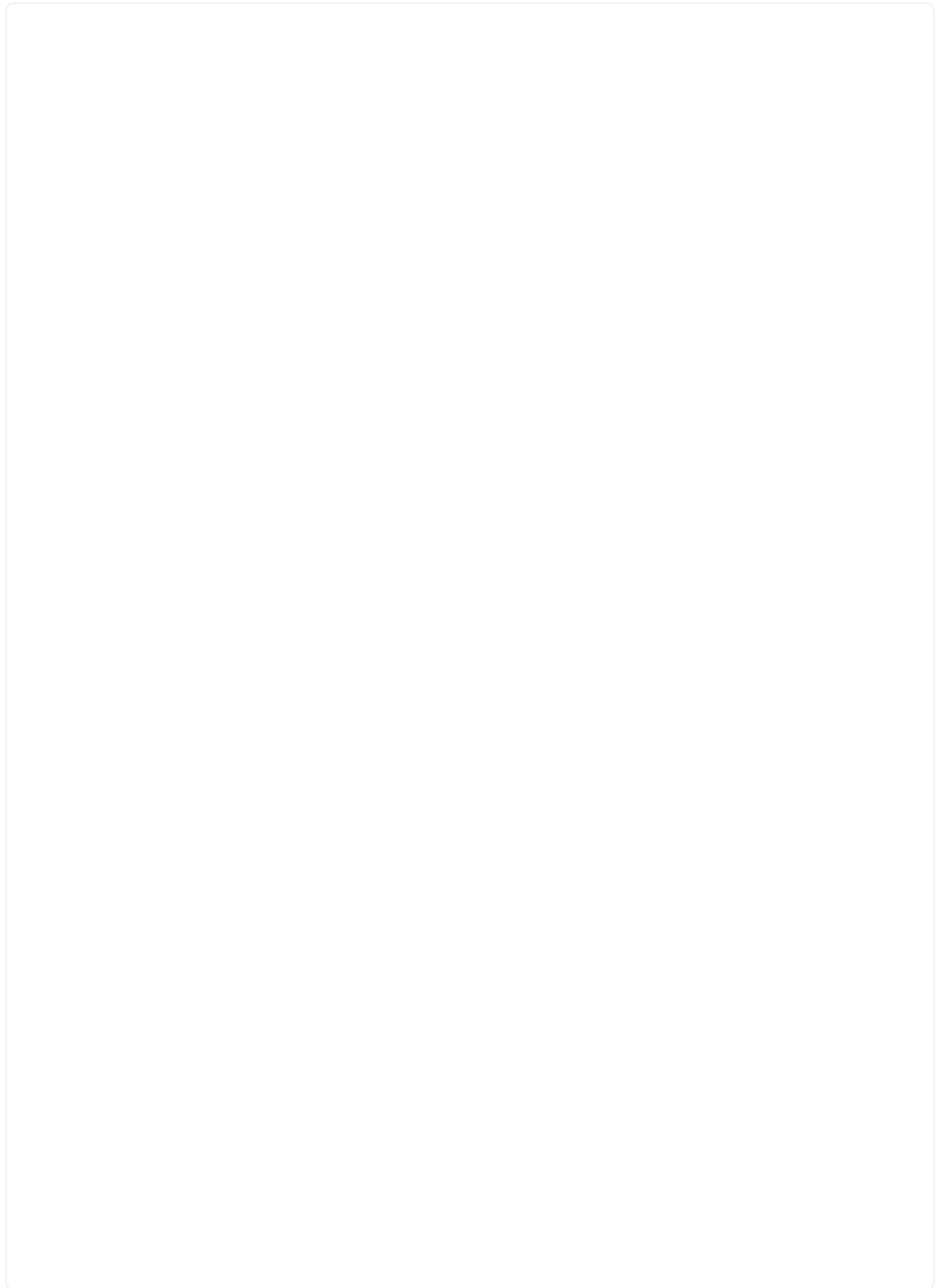
storageClass

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

volumeReclaimPolicy

Setting	Guidance	Type	Options
volumeReclaimPolicy	<p>This sets the reclaim policy for Astra's PVs. Setting this policy to <code>Retain</code> retains persistent volumes after Astra is deleted. Setting this policy to <code>Delete</code> deletes persistent volumes after astra is deleted. If this value is not set, the PVs are retained.</p>	string	<ul style="list-style-type: none">• Retain (This is the default value)• Delete

ingressType





Setting	Guidance	Type	Options
ingressType	<p>Use one of the following ingress types:</p> <p>Generic (ingressType: "Generic") (Default) Use this option when you have another ingress controller in use or would prefer to use your own ingress controller. After Astra Control Center is deployed, you will need to configure the ingress controller to expose Astra Control Center with a URL.</p> <p>AccTraefik (ingressType: "AccTraefik") Use this option when you would prefer not to configure an ingress controller. This deploys the Astra Control Center traefik gateway as a Kubernetes LoadBalancer type service.</p> <p>Astra Control Center uses a service of the type "LoadBalancer" (svc/traefik in the Astra Control Center namespace), and requires that it be assigned an accessible external IP address. If load balancers are permitted in your environment and you don't already have one configured, you can use MetalLB or another external service load balancer to assign an external IP address to the service. In the internal DNS server configuration, you should point the chosen</p>	string	<ul style="list-style-type: none"> • Generic (this is the default value) • AccTraefik

Setting	Guidance	Type	Options
<code>astraResourcesScaler</code>	<p>Scaling options for AstraControlCenter Resource limits. By default, Astra Control Center deploys with resource requests set for most of the components within Astra. This configuration allows the Astra Control Center software stack to perform better in environments under increased application load and scale.</p> <p>However, in scenarios using smaller development or test clusters, the CR field <code>astraResourcesScaler</code> may be set to <code>Off</code>. This disables resource requests and allows for deployment on smaller clusters.</p>	string	<ul style="list-style-type: none">• <code>Default</code> (This is the default value)• <code>Off</code>

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

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

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

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

Complete Astra Control Center and operator installation

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

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

Sample response:

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

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

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

Sample response:

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

Verify system status

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

Steps

1. Verify that all system components installed successfully.

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

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

Sample response

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

fluent-bit-ds-9rql1	1/1	Running	0
6m54s			
fluent-bit-ds-w5mp7	1/1	Running	0
6m54s			
graphql-server-7c7cc49776-jz2kn	1/1	Running	0
2m29s			
identity-87c59c975-9jpnf	1/1	Running	0
9m6s			
influxdb2-0	1/1	Running	0
13m			
keycloak-operator-84ff6d59d4-qcnmc	1/1	Running	0
7m1s			
krakend-cbf6c7df9-mdtzv	1/1	Running	0
2m30s			
license-5b888b78bf-plj6j	1/1	Running	0
9m32s			
login-ui-846b4664dd-fz8hv	1/1	Running	0
2m24s			
loki-0	1/1	Running	0
13m			
metrics-facade-779cc9774-n26rw	1/1	Running	0
9m18s			
monitoring-operator-974db78f-pkspq	2/2	Running	0
6m58s			
nats-0	1/1	Running	0
13m			
nats-1	1/1	Running	0
13m			
nats-2	1/1	Running	0
13m			
nautilus-7bdc7ddc54-49tfn	1/1	Running	0
7m50s			
nautilus-7bdc7ddc54-cwc79	1/1	Running	0
9m36s			
openapi-5584ff9f46-gbrdj	1/1	Running	0
9m17s			
openapi-5584ff9f46-z9mzk	1/1	Running	0
9m17s			
packages-bfc58cc98-lpxq9	1/1	Running	0
8m58s			
polaris-consul-consul-server-0	1/1	Running	0
13m			
polaris-consul-consul-server-1	1/1	Running	0
13m			
polaris-consul-consul-server-2	1/1	Running	0
13m			

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

traefik-7b87c4c474-jmcp2	1/1	Running	0
2m24s			
traefik-7b87c4c474-t9k8x	1/1	Running	0
2m24s			
trident-svc-c78f5b6bd-nwdsq	1/1	Running	0
9m22s			
vault-controller-55bbc96668-c6425	1/1	Running	0
11m			
vault-controller-55bbc96668-lq9n9	1/1	Running	0
11m			
vault-controller-55bbc96668-rfkqg	1/1	Running	0
11m			

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

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



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

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

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

Response:

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



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

Set up ingress for load balancing

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

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

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

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

- Istio ingress
- Nginx ingress controller
- OpenShift ingress controller

What you'll need

- The required [ingress controller](#) should already be deployed.
- The [ingress class](#) corresponding to the ingress controller should already be created.

Steps for Istio ingress

1. Configure Istio ingress.



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

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

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

Sample command:

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

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

Sample command:

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



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

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

```

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

```

5. Apply the changes:

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

6. Check the status of the ingress:

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

Response:

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

7. Finish Astra Control Center installation.

Steps for Nginx ingress controller

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

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

3. Apply the changes:

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



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

Steps for OpenShift ingress controller

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

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

Log in to the Astra Control Center UI

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

Steps

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



You can create a custom certificate after login.

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



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

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



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

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

Troubleshoot the installation

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

Steps

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

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

What's next

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

Install Astra Control Center using OpenShift OperatorHub

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

After you complete this procedure, you must return to the installation procedure to complete the [remaining steps](#) to verify installation success and log on.

What you'll need

- **Environmental prerequisites met:** [Before you begin installation, prepare your environment for Astra Control Center deployment](#).

- **Healthy cluster operators and API services:**

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

```
oc get clusteroperators
```

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

```
oc get apiservices
```

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

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

- b. [Complete setup](#) for your ingress controller type.

Steps

- [Download and extract Astra Control Center](#)
- [Install the NetApp Astra kubectl plugin](#)
- [Add the images to your local registry](#)

- [Find the operator install page](#)
- [Install the operator](#)
- [Install Astra Control Center](#)

Download and extract Astra Control Center

1. Go to the [Astra Control Center Evaluation downloads page](#) on the NetApp Support Site.
2. Download the bundle containing Astra Control Center (`astra-control-center-[version].tar.gz`).
3. (Recommended but optional) Download the certificates and signatures bundle for Astra Control Center (`astra-control-center-certs-[version].tar.gz`) to verify the signature of the bundle:

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

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

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

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

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

Install the NetApp Astra kubectl plugin

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

What you'll need

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

Steps

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



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

```
ls kubectl-astra/
```

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


```
cp kubect1-astra/<binary-name> /usr/local/bin/kubect1-astra
```

Add the images to your local registry

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

Docker

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

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

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

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

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

Podman

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

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

- b. Log in to your registry:

```
podman login <YOUR_REGISTRY>
```

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

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

```

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

```

Podman 3

```

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

```



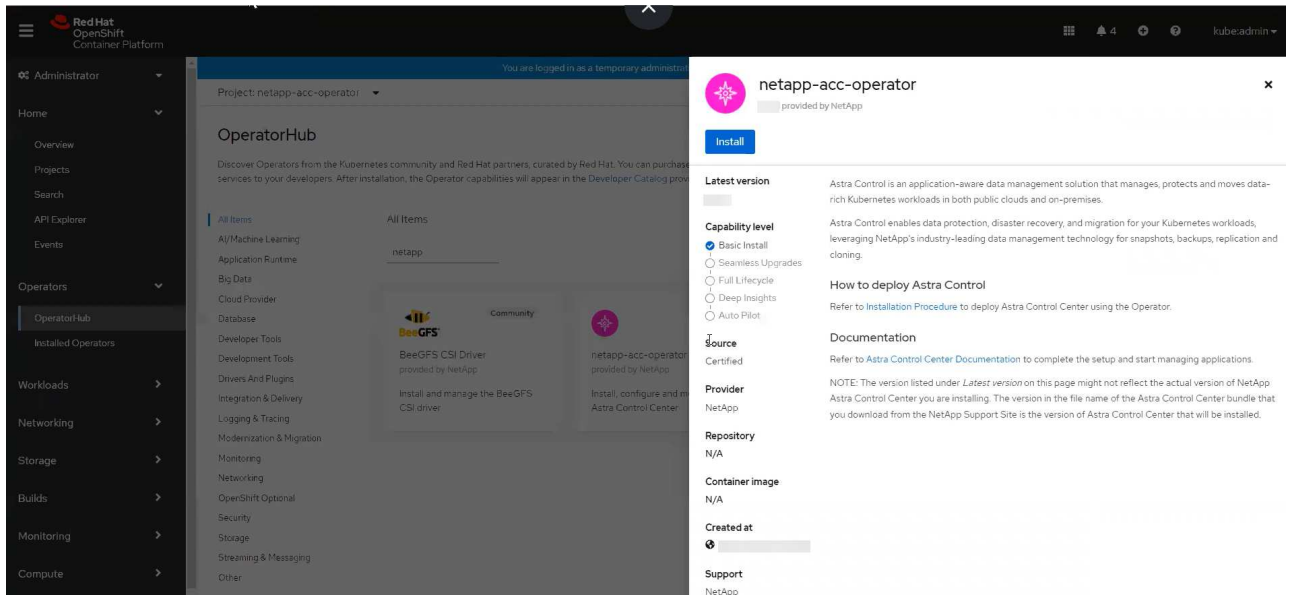
The image path the script creates should resemble the following, depending on your registry configuration: <https://netappdownloads.jfrog.io/docker-astra-control-prod/netapp/astra/acc/22.11.0-82/image:version>

Find the operator install page

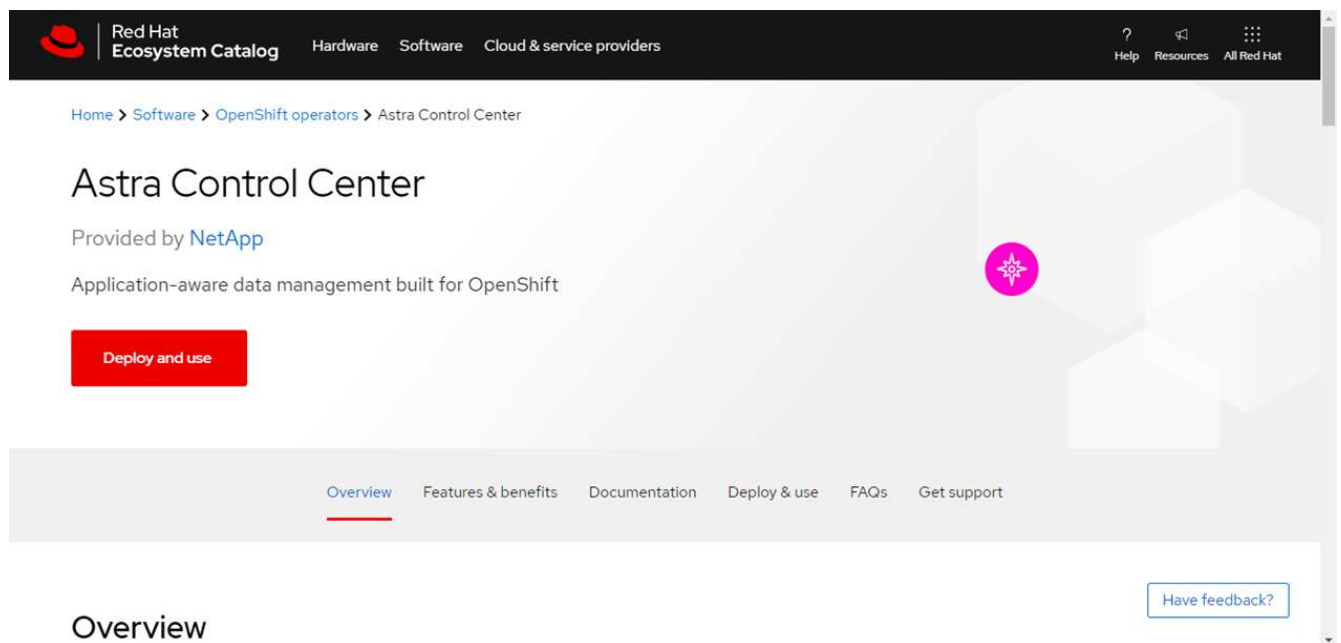
1. Complete one of the following procedures to access the operator install page:

- From Red Hat Openshift web console:
 - a. Log in to the OpenShift Container Platform UI.
 - b. From the side menu, select **Operators > OperatorHub**.

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



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



Overview

Install the operator

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



The operator will be available in all cluster namespaces.

- a. Select the operator namespace or `netapp-acc-operator` namespace will be created automatically as part of the operator installation.
- b. Select a manual or automatic approval strategy.



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

c. Select **Install**.

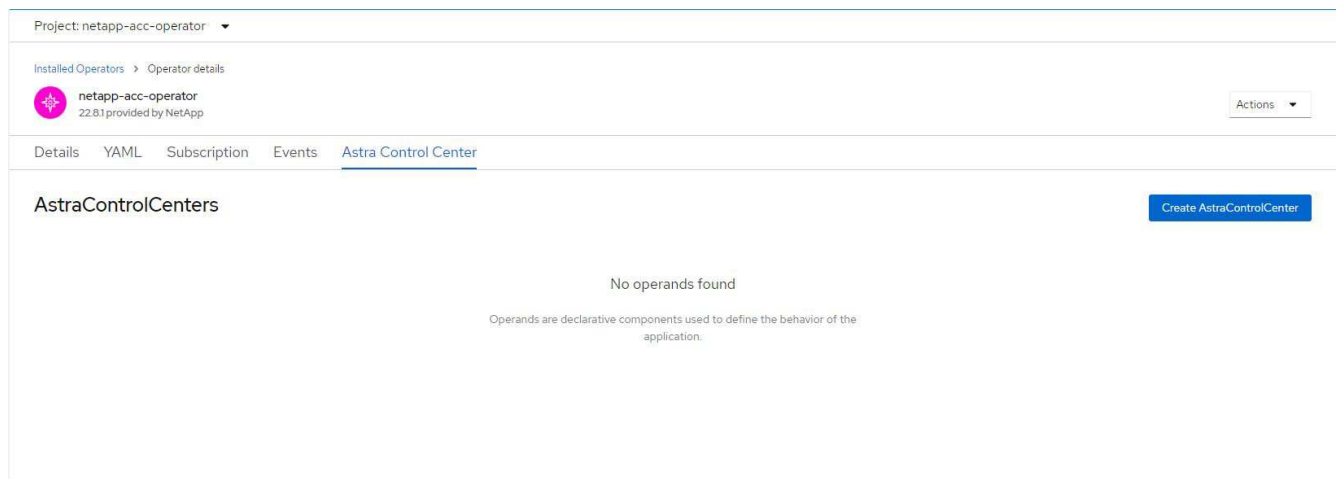


If you selected a manual approval strategy, you will be prompted to approve the manual install plan for this operator.

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

Install Astra Control Center

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



2. Complete the `Create AstraControlCenter` form field:

- Keep or adjust the Astra Control Center name.
- Add labels for the Astra Control Center.
- Enable or disable Auto Support. Retaining Auto Support functionality is recommended.
- Enter the Astra Control Center FQDN or IP address. Do not enter `http://` or `https://` in the address field.
- Enter the Astra Control Center version; for example, 22.04.1.
- Enter an account name, email address, and admin last name.
- Choose a volume reclaim policy of Retain, Recycle, or Delete. The default value is Retain.
- Select the ingress type:

- **Generic** (`ingressType: "Generic"`) (Default)

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

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

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

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



For details about the service type of "LoadBalancer" and ingress, see [Requirements](#).

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

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



If you use a registry that requires authentication, [create a secret on the cluster](#).

- k. Enter the admin first name.
- l. Configure resources scaling.
- m. Provide the default storage class.



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

- n. Define CRD handling preferences.
3. Select the YAML view to review the settings you have selected.
 4. Select **Create**.

Create a registry secret

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

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

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

2. Create a secret in this namespace:

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



Astra Control supports Docker registry secrets only.

3. Complete the remaining fields in [the Create AstraControlCenter form field](#).

What's next

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

Install Astra Control Center with a Cloud Volumes ONTAP storage backend

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

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

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

- [Deploy Astra Control Center in Amazon Web Services](#)
- [Deploy Astra Control Center in Google Cloud Platform](#)
- [Deploy Astra Control Center in Microsoft Azure](#)

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

Deploy Astra Control Center in Amazon Web Services

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

What you'll need for AWS

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

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

Operational environment requirements for AWS

Astra Control Center requires the following operational environment for AWS:

- Red Hat OpenShift Container Platform 4.8



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

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

Component	Requirement
Backend NetApp Cloud Volumes ONTAP storage capacity	At least 300GB available
Worker nodes (AWS EC2 requirement)	At least 3 worker nodes total, with 4 vCPU cores and 12GB RAM each
Load balancer	Service type "LoadBalancer" available for ingress traffic to be sent to services in the operational environment cluster
FQDN	A method for pointing the FQDN of Astra Control Center to the load balanced IP address
Astra Trident (installed as part of the Kubernetes cluster discovery in NetApp BlueXP, formerly Cloud Manager)	Astra Trident 21.04 or newer installed and configured and NetApp ONTAP version 9.5 or newer as a storage backend
Image registry	<p>You must have an existing private registry, such as AWS Elastic Container Registry, to which you can push Astra Control Center build images. You need to provide the URL of the image registry where you will upload the images.</p> <div> <p>The Astra Control Center hosted cluster and the managed cluster must have access to the same image registry to be able to back up and restore apps using the Restic-based image.</p> </div>
Astra Trident / ONTAP configuration	<p>Astra Control Center requires that a storage class be created and set as the default storage class. Astra Control Center supports the following ONTAP Kubernetes storage classes that are created when you import your Kubernetes cluster into NetApp BlueXP (formerly Cloud Manager). These are provided by Astra Trident:</p> <ul style="list-style-type: none"> • <code>vsaworkingenvironment-<>-ha-nas</code> <code>csi.trident.netapp.io</code> • <code>vsaworkingenvironment-<>-ha-san</code> <code>csi.trident.netapp.io</code> • <code>vsaworkingenvironment-<>-single-nas</code> <code>csi.trident.netapp.io</code> • <code>vsaworkingenvironment-<>-single-san</code> <code>csi.trident.netapp.io</code>



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



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

Overview of deployment for AWS

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

Each of these steps is explained in more detail below.

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

Ensure that you have sufficient IAM permissions

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

See [Initial AWS credentials](#).

Install a RedHat OpenShift cluster on AWS

Install a RedHat OpenShift Container Platform cluster on AWS.

For installation instructions, see [Installing a cluster on AWS in OpenShift Container Platform](#).

Configure AWS

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

Follow the AWS documentation to complete the following steps. See [AWS installation documentation](#).

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



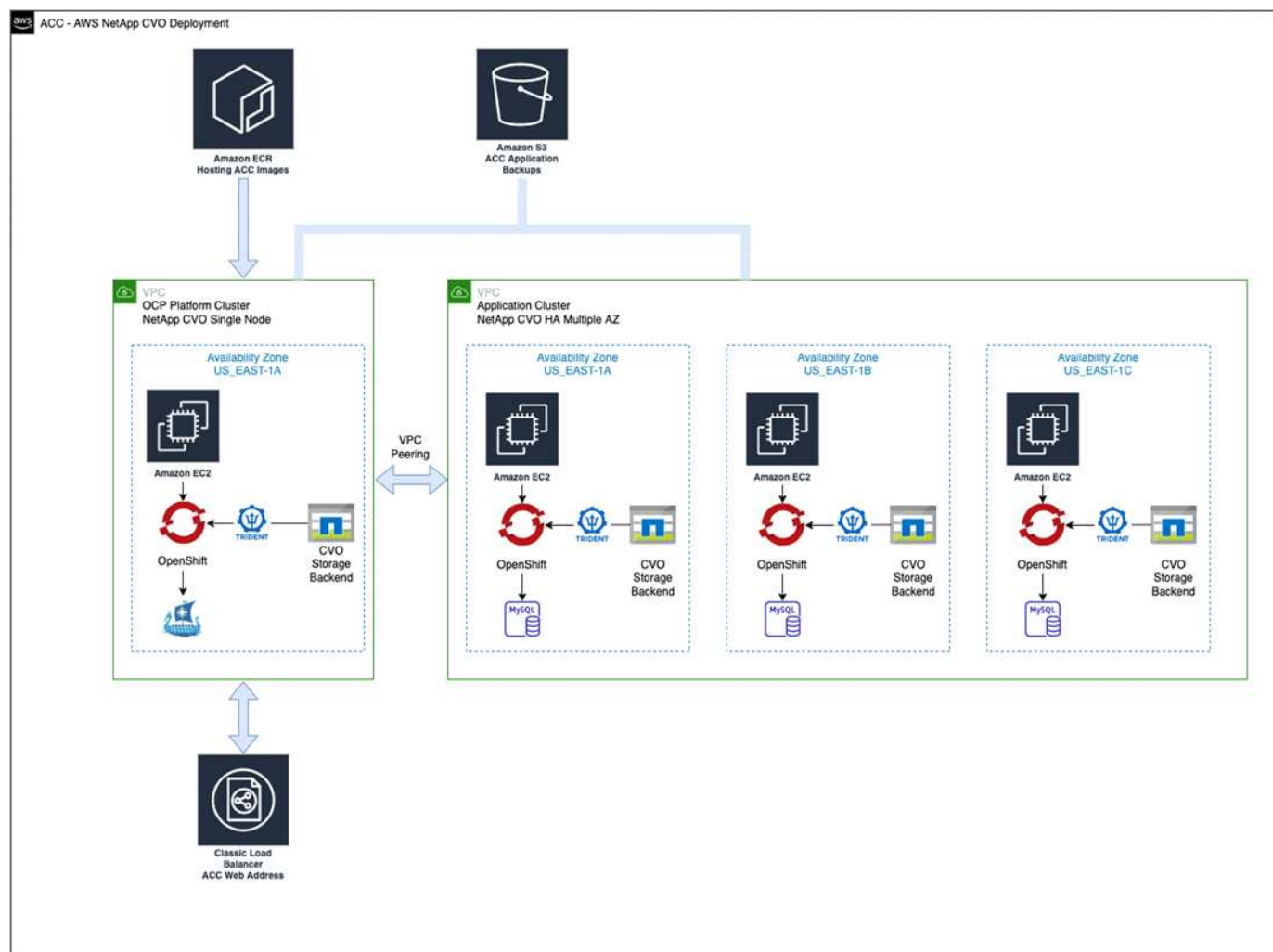
If you do not create the ECR, Astra Control Center cannot access monitoring data from a cluster containing Cloud Volumes ONTAP with an AWS backend. The issue is caused when the cluster you try to discover and manage using Astra Control Center does not have AWS ECR access.

6. Push the ACC images to your defined registry.



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

Here's an example of an AWS deployment:



Configure NetApp BlueXP for AWS

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

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

- [Getting started with Cloud Volumes ONTAP in AWS.](#)

- [Create a connector in AWS using BlueXP](#)

Steps

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

This imports your Red Hat OpenShift cluster and assigns it a default storage class. You select the storage class.

Trident is automatically installed as part of the import and discovery process.

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



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

Install Astra Control Center for AWS

Follow the standard [Astra Control Center installation instructions](#).



AWS uses the Generic S3 bucket type.

Deploy Astra Control Center in Google Cloud Platform

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

What you'll need for GCP

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


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

Operational environment requirements for GCP



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

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

Component	Requirement
Backend NetApp Cloud Volumes ONTAP storage capacity	At least 300GB available
Worker nodes (GCP compute requirement)	At least 3 worker nodes total, with 4 vCPU cores and 12GB RAM each
Load balancer	Service type "LoadBalancer" available for ingress traffic to be sent to services in the operational environment cluster
FQDN (GCP DNS zone)	A method for pointing the FQDN of Astra Control Center to the load balanced IP address
Astra Trident (installed as part of the Kubernetes cluster discovery in NetApp BlueXP, formerly Cloud Manager)	Astra Trident 21.04 or newer installed and configured and NetApp ONTAP version 9.5 or newer as a storage backend
Image registry	<p>You must have an existing private registry, such as Google Container Registry, to which you can push Astra Control Center build images. You need to provide the URL of the image registry where you will upload the images.</p> <div> You need to enable anonymous access to pull Restic images for backups.</div>
Astra Trident / ONTAP configuration	<p>Astra Control Center requires that a storage class be created and set as the default storage class. Astra Control Center supports the following ONTAP Kubernetes storage classes that are created when you import your Kubernetes cluster into NetApp BlueXP. These are provided by Astra Trident:</p> <ul style="list-style-type: none">• <code>vsaworkingenvironment-<>-ha-nas</code> <code>csi.trident.netapp.io</code>• <code>vsaworkingenvironment-<>-ha-san</code> <code>csi.trident.netapp.io</code>• <code>vsaworkingenvironment-<>-single-nas</code> <code>csi.trident.netapp.io</code>• <code>vsaworkingenvironment-<>-single-san</code> <code>csi.trident.netapp.io</code>



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

Overview of deployment for GCP

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

Each of these steps is explained in more detail below.

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

Install a RedHat OpenShift cluster on GCP

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

For installation instructions, see the following:

- [Installing an OpenShift cluster in GCP](#)
- [Creating a GCP Service Account](#)

Create a GCP Project and Virtual Private Cloud

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



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

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

Ensure that you have sufficient IAM permissions

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

See [Initial GCP credentials and permissions.](#)

Configure GCP

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

Follow the GCP documentation to complete the following steps. See [Installing OpenShift cluster in GCP.](#)

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

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

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

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

Example:

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

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

8. Set up DNS zones.

Configure NetApp BlueXP for GCP

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

Follow the BlueXP documentation to complete the following steps. See [Getting started with Cloud Volumes ONTAP in GCP](#).

What you'll need

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

Steps

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

This imports your Red Hat OpenShift cluster and assigns it a default storage class. You select the storage class.

Trident is automatically installed as part of the import and discovery process.

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



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

Install Astra Control Center for GCP

Follow the standard [Astra Control Center installation instructions](#).



GCP uses the Generic S3 bucket type.

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

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

Deploy Astra Control Center in Microsoft Azure

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

What you'll need for Azure

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

- Astra Control Center license. See [Astra Control Center licensing requirements](#).
- [Meet Astra Control Center requirements](#).


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

Operational environment requirements for Azure

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

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

See [Astra Control Center operational environment requirements](#).

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

Component	Requirement
Astra Trident / ONTAP configuration	<p>Astra Control Center requires that a storage class be created and set as the default storage class. Astra Control Center supports the following ONTAP Kubernetes storage classes that are created when you import your Kubernetes cluster into NetApp BlueXP. These are provided by Astra Trident:</p> <ul style="list-style-type: none"> • <code>vsaworkingenvironment-<>-ha-nas</code> <code>csi.trident.netapp.io</code> • <code>vsaworkingenvironment-<>-ha-san</code> <code>csi.trident.netapp.io</code> • <code>vsaworkingenvironment-<>-single-nas</code> <code>csi.trident.netapp.io</code> • <code>vsaworkingenvironment-<>-single-san</code> <code>csi.trident.netapp.io</code>



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

Overview of deployment for Azure

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

Each of these steps is explained in more detail below.

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

Install a RedHat OpenShift cluster on Azure

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

For installation instructions, see the following:

- [Installing OpenShift cluster on Azure.](#)
- [Installing an Azure account.](#)

Create Azure resource groups

Create at least one Azure resource group.



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

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

Ensure that you have sufficient IAM permissions

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

See [Azure credentials and permissions](#).

Configure Azure

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

Follow the Azure documentation to complete the following steps. See [Installing OpenShift cluster on Azure](#).

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

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

Example:

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

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

10. Set up DNS zones.

Configure NetApp BlueXP (formerly Cloud Manager) for Azure

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

Follow the BlueXP documentation to complete the following steps. See [Getting started with BlueXP in Azure](#).

What you'll need

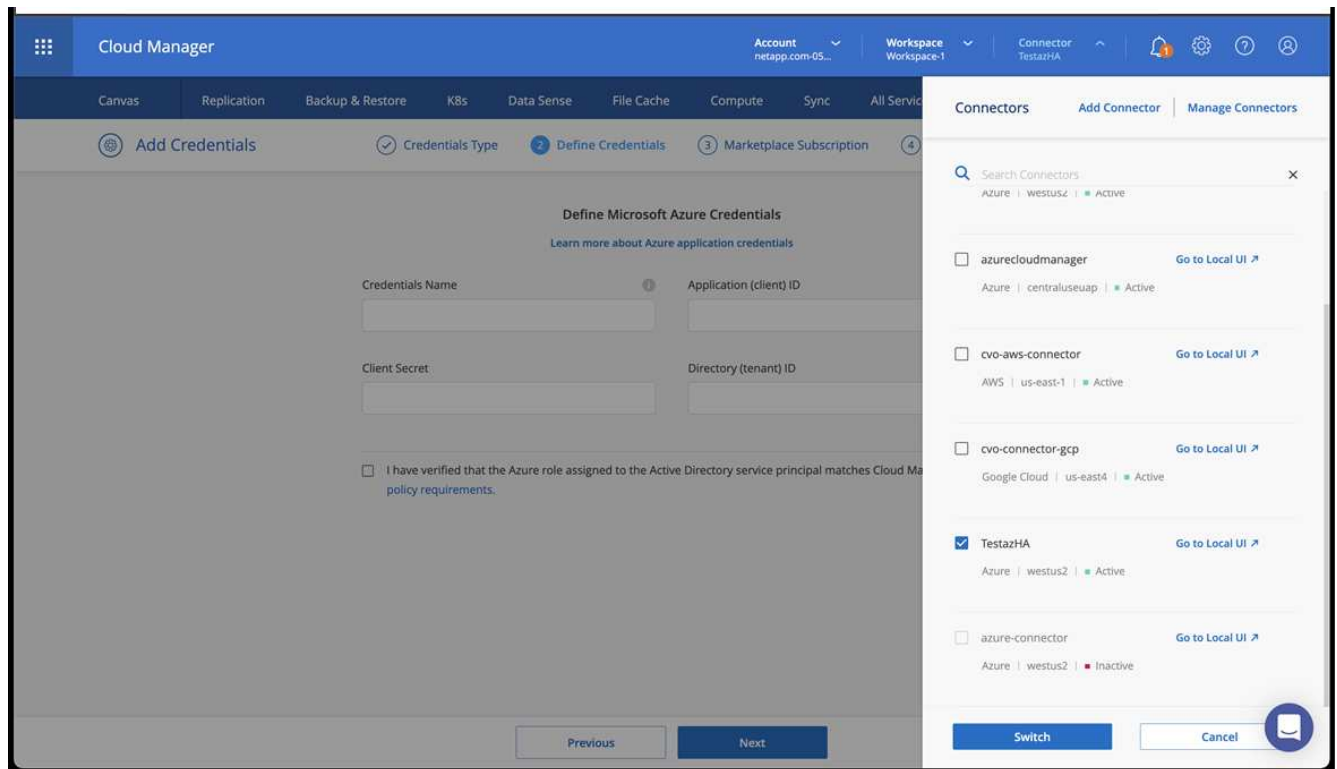
Access to the Azure account with the required IAM permissions and roles

Steps

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

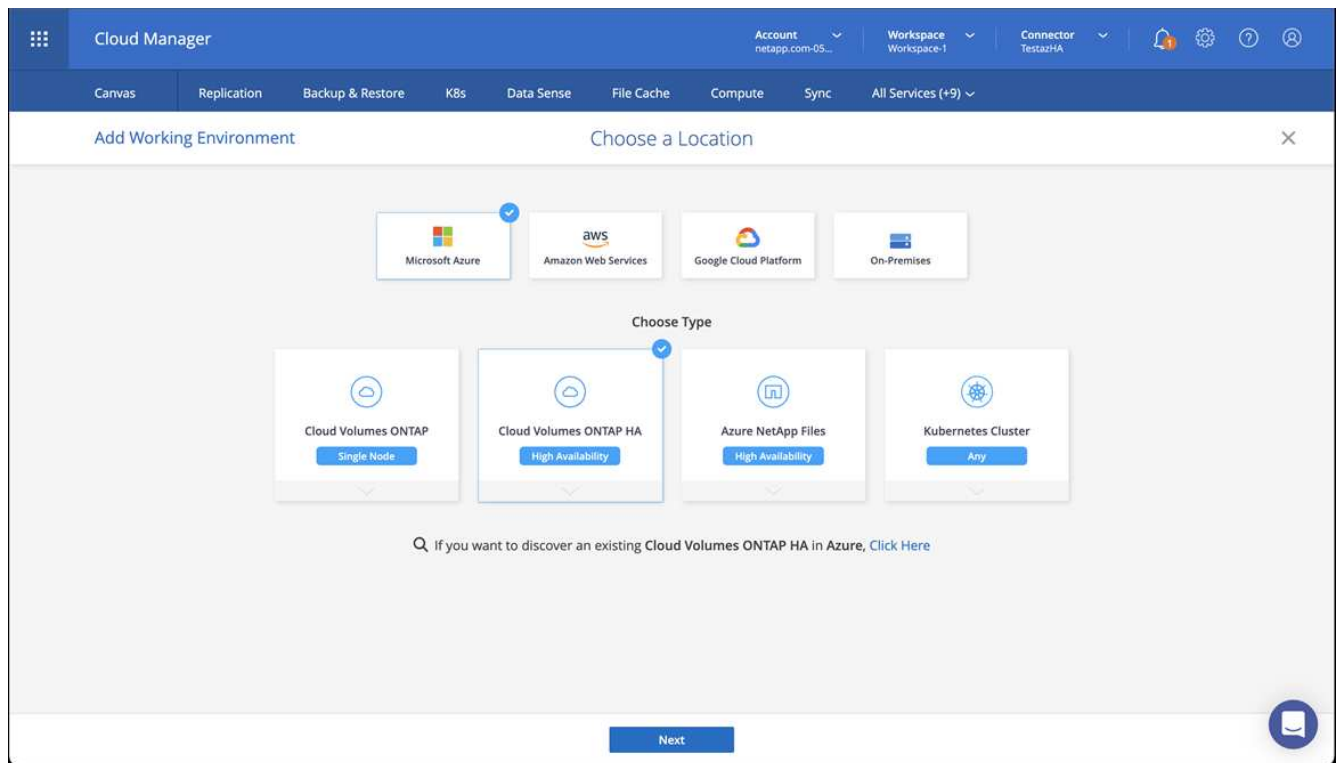
See [Creating a connector in Azure from BlueXP](#).

3. Ensure that the connector is running and switch to that connector.



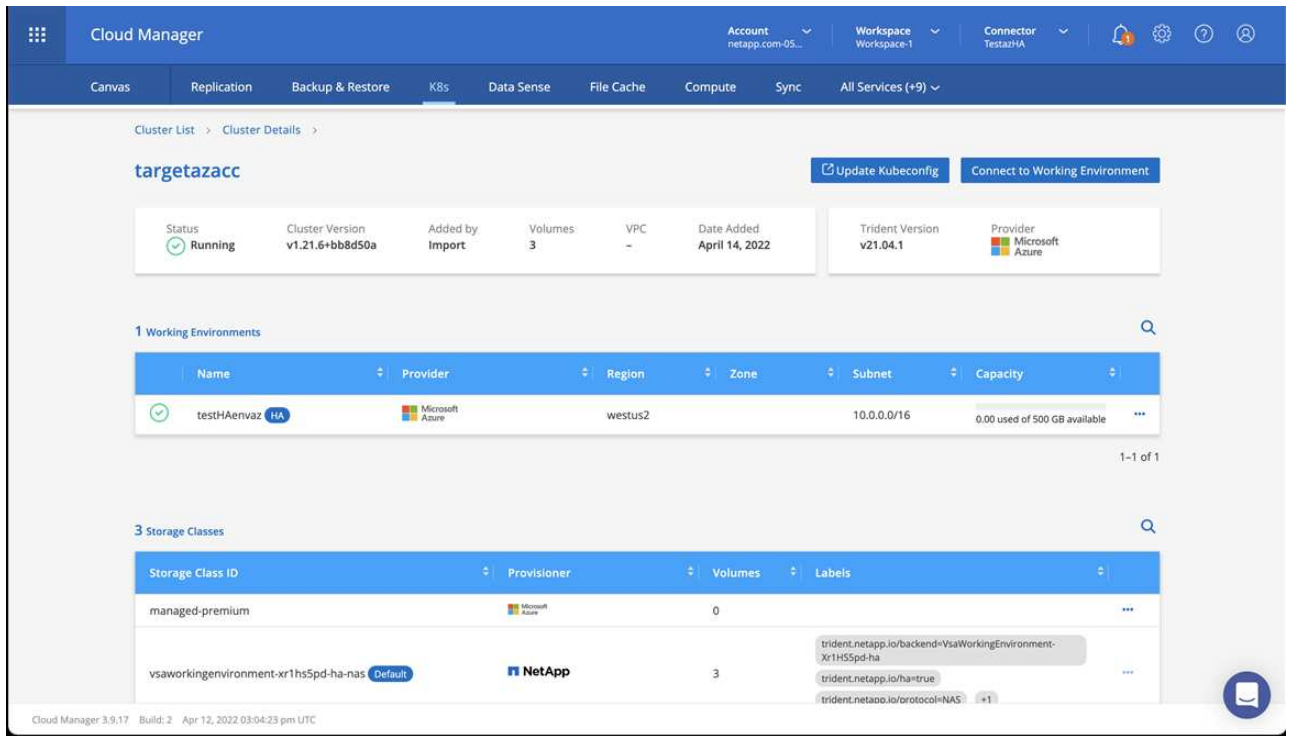
4. Create a working environment for your cloud environment.

- a. Location: "Microsoft Azure".
- b. Type: "Cloud Volumes ONTAP HA".



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

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



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

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

This imports your Red Hat OpenShift cluster and assigns a default storage class. You select the storage class.

Trident is automatically installed as part of the import and discovery process.

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

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

Install and configure Astra Control Center for Azure

Install Astra Control Center with the standard [installation instructions](#).

Using Astra Control Center, add an Azure bucket. See [Set up Astra Control Center and add buckets](#).

Set up Astra Control Center

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

Tasks

- [Add a license for Astra Control Center](#)
- [Prepare your environment for cluster management using Astra Control](#)
- [Add cluster](#)
- [Add a storage backend](#)
- [Add a bucket](#)

Add a license for Astra Control Center

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

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



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



To update an existing evaluation or full license, refer to [Update an existing license](#).

What you'll need

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

Steps

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

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



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

Prepare your environment for cluster management using Astra Control

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

What you'll need

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



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

- **Trident storage backend configured:** At least one Astra Trident storage backend must be [configured](#) on the cluster.
- **Trident storage classes configured:** At least one Astra Trident storage class must be [configured](#) on the cluster. If a default storage class is configured, ensure that it is the only storage class that has the default annotation.
- **Astra Trident volume snapshot controller and volume snapshot class installed and configured:** The volume snapshot controller must be [installed](#) so that snapshots can be created in Astra Control. At least one Astra Trident `VolumeSnapshotClass` has been [set up](#) by an administrator.
- **Kubeconfig accessible:** You have access to the [cluster kubeconfig](#) that includes only one context element.
- **ONTAP credentials:** You need ONTAP credentials and a superuser and user ID set on the backing ONTAP system to back up and restore apps with Astra Control Center.

Run the following commands in the ONTAP command line:

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

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

Run eligibility checks

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

Steps

1. Check the Trident version.

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

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

NAME	VERSION
trident	22.10.0

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

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



If Trident is not installed or the installed version is not the latest, you need to install the latest version of Trident before proceeding. Refer to the [Trident documentation](#) for instructions.

2. Ensure that the pods are running:

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

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

```
kubectl get sc
```

Sample response:

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

Create a limited cluster role kubeconfig

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

What you'll need

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

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



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

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

Steps

1. Create a service account:

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

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

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

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

- b. Apply the service account:

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

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

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

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

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

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

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

```

- get
- list
- create
- patch

# Delete Resources
# Necessary for in-place restore and AppMirror failover
- apiGroups:
  - ""
  - apps
  - autoscaling
  - batch
  - crd.projectcalico.org
  - extensions
  - networking.k8s.io
  - policy
  - rbac.authorization.k8s.io
  - snapshot.storage.k8s.io
  - trident.netapp.io
resources:
- configmaps
- cronjobs
- daemonsets
- deployments
- horizontalpodautoscalers
- ingresses
- jobs
- namespaces
- networkpolicies
- persistentvolumeclaims
- poddisruptionbudgets
- pods
- podtemplates
- podsecuritypolicies
- 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

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

```

b. Apply the cluster role:

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

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

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

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

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

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

b. Apply the cluster role binding:

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

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

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

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

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

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

5. Generate the kubeconfig as follows:

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

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.token}')
```

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

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

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

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

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

# Create token user
```

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

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

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

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

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

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

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

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

```
mv kubeconfig-sa YOUR_CLUSTER_NAME_kubeconfig
```

What's next?

Now that you've verified that the prerequisites are met, you're ready to [add a cluster](#).

Add cluster

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



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

What you'll need

- Before you add a cluster, review and perform the necessary [prerequisite tasks](#).

Steps

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



The `kubeconfig.yaml` file should include **only the cluster credential for one cluster**.



If you create your own `kubeconfig` file, you should define only **one** context element in it. Refer to [Kubernetes documentation](#) for information about creating `kubeconfig` files. If you created a `kubeconfig` for a limited cluster role using [the process above](#), be sure to upload or paste that `kubeconfig` in this step.

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



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

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

Result

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



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

Add a storage backend

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

Managing storage clusters in Astra Control as a storage backend enables you to get linkages between persistent volumes (PVs) and the storage backend as well as additional storage metrics.

Steps

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

managed cluster.

3. Enter the ONTAP cluster management IP address and admin credentials. The credentials must be cluster-wide credentials.



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

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

Result

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



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

Add a bucket

You can add a bucket using the Astra Control UI or [API](#). Adding object store bucket providers is essential if you want to back up your applications and persistent storage or if you want to clone applications across clusters. Astra Control stores those backups or clones in the object store buckets that you define.

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

What you'll need

- A bucket that is reachable from your clusters managed by Astra Control Center.
- Credentials for the bucket.
- A bucket of the following types:
 - NetApp ONTAP S3
 - NetApp StorageGRID S3
 - Microsoft Azure
 - Generic S3



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



Although Astra Control Center supports Amazon S3 as a Generic S3 bucket provider, Astra Control Center might not support all object store vendors that claim Amazon's S3 support.

Steps

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



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

4. Enter an existing bucket name and optional description.



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

5. Enter the name or IP address of the S3 endpoint.

6. Under **Select Credentials**, choose either the **Add** or **Use existing** tab.

- If you chose **Add**:
 - a. Enter a name for the credential that distinguishes it from other credentials in Astra Control.
 - b. Enter the access ID and secret key by pasting the contents from your clipboard.
- If you chose **Use existing**:
 - a. Select the existing credentials you want to use with the bucket.

7. Select **Add**.



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

What's next?

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

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

Find more information

- [Use the Astra Control API](#)
- [Known issues](#)

Frequently asked questions for Astra Control Center

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

Overview

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

Access to Astra Control Center

What's the Astra Control URL?

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

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

Licensing

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

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

Steps

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

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

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

Registering Kubernetes clusters

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

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

How do I properly unmanage a cluster?

1. [Unmanage the applications from Astra Control](#).
2. [Unmanage the cluster from Astra Control](#).

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

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

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



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

Is NetApp Trident automatically uninstalled from a cluster when I unmanage it?

When you unmanage a cluster from Astra Control Center, Trident isn't automatically uninstalled from the cluster. To uninstall Trident, you'll need to [follow these steps in the Trident documentation](#).

Managing applications

Can Astra Control deploy an application?

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

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

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

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

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

Should I manage Astra Control itself?

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

Do unhealthy pods affect app management?

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

Data management operations

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

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

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

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

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.