Installation overview

Astra Control Center

NetApp
March 22, 2022

Table of Contents

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 using the standard process

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

For Red Hat OpenShift environments, you can also use an alternative procedure to install Astra Control Center using OpenShift OperatorHub.

What you'll need

- Before you begin installation, prepare your environment for Astra Control Center deployment.
- Ensure all cluster operators are in a healthy state and available.

  OpenShift example:

  ```
  oc get clusteroperators
  ```

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

  OpenShift example:

  ```
  oc get apiservices
  ```

- You have created an FQDN address for Astra Control Center in your data center.

About this task

The Astra Control Center installation process does the following:

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

Podman commands can be used in place of Docker commands if you are using Red Hat's Podman instead of Docker Engine.

Do not execute the following command during the entirety of the installation process to avoid deleting all Astra Control Center pods: `kubectl delete -f astra_control_center_operator_deploy.yaml`

Steps

To install Astra Control Center, do the following steps:

- Download the Astra Control Center bundle
- Unpack the bundle and change directory
- 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
- Log in to the Astra Control Center UI

Complete the deployment by performing setup tasks.

Download the Astra Control Center bundle

1. Download the Astra Control Center bundle (`astra-control-center-[version].tar.gz`) from the NetApp Support Site.
2. Download the zip of Astra Control Center certificates and keys from NetApp Support Site.
3. (Optional) Use the following command to verify the signature of the bundle:

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

Unpack the bundle and change directory

1. Extract the images:

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

2. Change to the Astra directory.

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

Add the images to your local registry

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

   See sample scripts for the automatic loading of images below.

   a. Log in to your registry:

      Docker:

      ```
      docker login [your_registry_path]
      ```

      Podman:

      ```
      podman login [your_registry_path]
      ```

   b. Use the appropriate script to load the images, tag the images, and push the images to your local registry:

      Docker:

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

      Podman:

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

Set up namespace and secret for registries with auth requirements

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

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

   Response:

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

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

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

   Sample response:

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

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

   ```
   kubectl create ns [netapp-acc or custom 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. Edit the Astra Control Center operator deployment YAML (`astra_control_center_operator_deploy.yaml`) to refer to your local registry and secret.

```
vim astra_control_center_operator_deploy.yaml
```

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

```
imagePullSecrets:
- name: <name_of_secret_with_creds_to_local_registry>
```

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.

d. (For installations using Astra Data Store preview) See this known issue regarding storage class provisioners and additional changes you will need to make to the YAML.

```
apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    control-plane: controller-manager
  name: acc-operator-controller-manager
  namespace: netapp-acc-operator
spec:
  replicas: 1
  selector:
    matchLabels:
      control-plane: controller-manager
  template:
    metadata:
      labels:
        control-plane: controller-manager
    spec:
      containers:
      - args:
        - --secure-listen-address=0.0.0.0:8443
        - --upstream=http://127.0.0.1:8080/
        - --logtostderr=true
        - --v=10
        image: [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
        command:
        - /manager
        env:
        - name: ACCOP_LOG_LEVEL
          value: "2"
        image: [your_registry_path]/acc-operator:[version x.y.z]
        imagePullPolicy: IfNotPresent
      imagePullSecrets: []
```

2. 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
```

Configure Astra Control Center

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

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

   ```
   vim astra_control_center_min.yaml
   ```

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

   If you are using a registry that does not require authorization, you must delete the `secret` line within `imageRegistry` or the installation will fail.

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

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

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

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

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

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

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

h. (For installations using Astra Data Store preview) See this known issue for additional required changes to the YAML.

```
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"
```

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_min.yaml -n [netapp-acc or custom
namespace]
```

Sample response:

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

Verify system status

If you prefer to use OpenShift, you can use comparable oc commands for verification 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-5f75c5f564-bzqmt           1/1     Running   0
11m
activity-6b8f7cccb9-mlrn4                1/1     Running   0
9m2s
api-token-authentication-6hznt           1/1     Running   0
8m50s
api-token-authentication-qpfgb           1/1     Running   0
8m50s
api-token-authentication-sqnb7           1/1     Running   0
8m50s
asup-5578bbdd57-dxkbp                    1/1     Running   0
9m3s
authentication-56bff4f95d-mspmq          1/1     Running   0
7m31s
bucketservice-6f7968b95d-9rrrl           1/1     Running   0
8m36s
cert-manager-5f6cf4bc4b-82khn            1/1     Running   0
6m19s
cert-manager-cainjector-76cf976458-sdrbc 1/1     Running   0
6m19s
cert-manager-webhook-5b7896bfd8-2n45j    1/1     Running   0
```

```
6m19s
cloud-extension-749d9f684c-8bdhq           1/1     Running   0
9m6s
cloud-insights-service-7d58687d9-h5tzw     1/1     Running   2
8m56s
composite-compute-968c79cb5-nv7l4          1/1     Running   0
9m11s
composite-volume-7687569985-jg9gg          1/1     Running   0
8m33s
credentials-5c9b75f4d6-nx9cz               1/1     Running   0
8m42s
entitlement-6c96fd8b78-zt7f8               1/1     Running   0
8m28s
features-5f7bfc9f68-gsjnl                  1/1     Running   0
8m57s
fluent-bit-ds-h88p7                        1/1     Running   0
7m22s
fluent-bit-ds-krhnj                        1/1     Running   0
7m23s
fluent-bit-ds-l5bjj                        1/1     Running   0
7m22s
fluent-bit-ds-lrclb                        1/1     Running   0
7m23s
fluent-bit-ds-s5t4n                        1/1     Running   0
7m23s
fluent-bit-ds-zpr6v                        1/1     Running   0
7m22s
graphql-server-5f5976f4bd-vbb4z            1/1     Running   0
7m13s
identity-56f78b8f9f-8h9p9                  1/1     Running   0
8m29s
influxdb2-0                                1/1     Running   0
11m
krakend-6f8d995b4d-5khkl                   1/1     Running   0
7m7s
license-5b5db87c97-jmxzc                   1/1     Running   0
9m
login-ui-57b57c74b8-6xtv7                  1/1     Running   0
7m10s
loki-0                                     1/1     Running   0
11m
monitoring-operator-9dbc9c76d-8znck        2/2     Running   0
7m33s
nats-0                                     1/1     Running   0
11m
nats-1                                     1/1     Running   0
```

```
10m
nats-2                                      1/1     Running   0
10m
nautilus-6b9d88bc86-h8kfb                   1/1     Running   0
8m6s
nautilus-6b9d88bc86-vn68r                   1/1     Running   0
8m35s
openapi-b87d77dd8-5dz9h                     1/1     Running   0
9m7s
polaris-consul-consul-5ljfb                 1/1     Running   0
11m
polaris-consul-consul-s5d5z                 1/1     Running   0
11m
polaris-consul-consul-server-0              1/1     Running   0
11m
polaris-consul-consul-server-1              1/1     Running   0
11m
polaris-consul-consul-server-2              1/1     Running   0
11m
polaris-consul-consul-twmpq                 1/1     Running   0
11m
polaris-mongodb-0                           2/2     Running   0
11m
polaris-mongodb-1                           2/2     Running   0
10m
polaris-mongodb-2                           2/2     Running   0
10m
polaris-ui-84dc87847f-zrg8w                 1/1     Running   0
7m12s
polaris-vault-0                             1/1     Running   0
11m
polaris-vault-1                             1/1     Running   0
11m
polaris-vault-2                             1/1     Running   0
11m
public-metrics-657698b66f-67pgt             1/1     Running   0
8m47s
storage-backend-metrics-6848b9fd87-w7x8r    1/1     Running   0
8m39s
storage-provider-5ff5868cd5-r9hj7           1/1     Running   0
8m45s
telegraf-ds-dw4hg                           1/1     Running   0
7m23s
telegraf-ds-k92gn                           1/1     Running   0
7m23s
telegraf-ds-mmxjl                           1/1     Running   0
```

```
7m23s
telegraf-ds-nhs8s                              1/1     Running   0
7m23s
telegraf-ds-rj7lw                              1/1     Running   0
7m23s
telegraf-ds-tqrkb                              1/1     Running   0
7m23s
telegraf-rs-9mwgj                              1/1     Running   0
7m23s
telemetry-service-56c49d689b-ffrzx             1/1     Running   0
8m42s
tenancy-767c77fb9d-g9ctv                       1/1     Running   0
8m52s
traefik-5857d87f85-7pmx8                       1/1     Running   0
6m49s
traefik-5857d87f85-cpxgv                       1/1     Running   0
5m34s
traefik-5857d87f85-lvmlb                       1/1     Running   0
4m33s
traefik-5857d87f85-t2xlk                       1/1     Running   0
4m33s
traefik-5857d87f85-v9wpf                       1/1     Running   0
7m3s
trident-svc-595f84dd78-zb8l6                   1/1     Running   0
8m54s
vault-controller-86c94fbf4f-krttq              1/1     Running   0
9m24s
```

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
```

3. When all the pods are running, verify installation success by retrieving the AstraControlCenter instance installed by the Astra Control Center Operator.

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

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

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

```
name: astra
   namespace: netapp-acc
   resourceVersion: "104424560"
   selfLink: /apis/astra.netapp.io/v1/namespaces/netapp-
acc/astracontrolcenters/astra
   uid: 9aa5fdae-4214-4cb7-9976-5d8b4c0ce27f
 spec:
   accountName: Example
   astraAddress: astra.example.com
   astraVersion: 21.12.60
   autoSupport:
     enrolled: true
     url: https://support.netapp.com/asupprod/post/1.0/postAsup
   crds: {}
   email: admin@example.com
   firstName: SRE
   imageRegistry:
     name: registry_name/astra
     secret: astra-registry-cred
   lastName: Admin
 status:
   accConditionHistory:
     items:
     - astraVersion: 21.12.60
       condition:
         lastTransitionTime: "2021-11-23T02:23:59Z"
         message: Deploying is currently in progress.
         reason: InProgress
         status: "False"
         type: Ready
       generation: 2
       observedSpec:
         accountName: Example
         astraAddress: astra.example.com
         astraVersion: 21.12.60
         autoSupport:
           enrolled: true
           url: https://support.netapp.com/asupprod/post/1.0/postAsup
         crds: {}
         email: admin@example.com
         firstName: SRE
         imageRegistry:
           name: registry_name/astra
           secret: astra-registry-cred
         lastName: Admin
       timestamp: "2021-11-23T02:23:59Z"
```

```
  - astraVersion: 21.12.60
    condition:
      lastTransitionTime: "2021-11-23T02:23:59Z"
      message: Deploying is currently in progress.
      reason: InProgress
      status: "True"
      type: Deploying
    generation: 2
    observedSpec:
      accountName: Example
      astraAddress: astra.example.com
      astraVersion: 21.12.60
      autoSupport:
        enrolled: true
        url: https://support.netapp.com/asupprod/post/1.0/postAsup
      crds: {}
      email: admin@example.com
      firstName: SRE
      imageRegistry:
        name: registry_name/astra
        secret: astra-registry-cred
      lastName: Admin
    timestamp: "2021-11-23T02:23:59Z"
  - astraVersion: 21.12.60
    condition:
      lastTransitionTime: "2021-11-23T02:29:41Z"
      message: Post Install was successful
      observedGeneration: 2
      reason: Complete
      status: "True"
      type: PostInstallComplete
    generation: 2
    observedSpec:
      accountName: Example
      astraAddress: astra.example.com
      astraVersion: 21.12.60
      autoSupport:
        enrolled: true
        url: https://support.netapp.com/asupprod/post/1.0/postAsup
      crds: {}
      email: admin@example.com
      firstName: SRE
      imageRegistry:
        name: registry_name/astra
        secret: astra-registry-cred
      lastName: Admin
```

```
    timestamp: "2021-11-23T02:29:41Z"
  - astraVersion: 21.12.60
    condition:
      lastTransitionTime: "2021-11-23T02:29:41Z"
      message: Deploying succeeded.
      reason: Complete
      status: "False"
      type: Deploying
    generation: 2
    observedGeneration: 2
    observedSpec:
      accountName: Example
      astraAddress: astra.example.com
      astraVersion: 21.12.60
      autoSupport:
        enrolled: true
        url: https://support.netapp.com/asupprod/post/1.0/postAsup
      crds: {}
      email: admin@example.com
      firstName: SRE
      imageRegistry:
        name: registry_name/astra
        secret: astra-registry-cred
      lastName: Admin
    observedVersion: 21.12.60
    timestamp: "2021-11-23T02:29:41Z"
  - astraVersion: 21.12.60
    condition:
      lastTransitionTime: "2021-11-23T02:29:41Z"
      message: Astra is deployed
      reason: Complete
      status: "True"
      type: Deployed
    generation: 2
    observedGeneration: 2
    observedSpec:
      accountName: Example
      astraAddress: astra.example.com
      astraVersion: 21.12.60
      autoSupport:
        enrolled: true
        url: https://support.netapp.com/asupprod/post/1.0/postAsup
      crds: {}
      email: admin@example.com
      firstName: SRE
      imageRegistry:
```

```
          name: registry_name/astra
          secret: astra-registry-cred
        lastName: Admin
      observedVersion: 21.12.60
      timestamp: "2021-11-23T02:29:41Z"
    - astraVersion: 21.12.60
      condition:
        lastTransitionTime: "2021-11-23T02:29:41Z"
        message: Astra is deployed
        reason: Complete
        status: "True"
        type: Ready
      generation: 2
      observedGeneration: 2
      observedSpec:
        accountName: Example
        astraAddress: astra.example.com
        astraVersion: 21.12.60
        autoSupport:
          enrolled: true
          url: https://support.netapp.com/asupprod/post/1.0/postAsup
        crds: {}
        email: admin@example.com
        firstName: SRE
        imageRegistry:
          name: registry_name/astra
          secret: astra-registry-cred
        lastName: Admin
      observedVersion: 21.12.60
      timestamp: "2021-11-23T02:29:41Z"
  certManager: deploy
  cluster:
    type: OCP
    vendorVersion: 4.7.5
    version: v1.20.0+bafe72f
  conditions:
  - lastTransitionTime: "2021-12-08T16:19:55Z"
    message: Astra is deployed
    reason: Complete
    status: "True"
    type: Ready
  - lastTransitionTime: "2021-12-08T16:19:55Z"
    message: Deploying succeeded.
    reason: Complete
    status: "False"
    type: Deploying
```

```
    - lastTransitionTime: "2021-12-08T16:19:53Z"
      message: Post Install was successful
      observedGeneration: 2
      reason: Complete
      status: "True"
      type: PostInstallComplete
    - lastTransitionTime: "2021-12-08T16:19:55Z"
      message: Astra is deployed
      reason: Complete
      status: "True"
      type: Deployed
    deploymentState: Deployed
    observedGeneration: 2
    observedSpec:
      accountName: Example
      astraAddress: astra.example.com
      astraVersion: 21.12.60
      autoSupport:
        enrolled: true
        url: https://support.netapp.com/asupprod/post/1.0/postAsup
      crds: {}
      email: admin@example.com
      firstName: SRE
      imageRegistry:
        name: registry_name/astra
        secret: astra-registry-cred
      lastName: Admin
    observedVersion: 21.12.60
    postInstall: Complete
    uuid: 9aa5fdae-4214-4cb7-9976-5d8b4c0ce27f
kind: List
metadata:
 resourceVersion: ""
 selfLink: ""
```

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

Log in to the Astra Control Center UI

After installing 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 you used in the `astraAddress` in the `astra_control_center_min.yaml` CR when you installed Astra Control Center.

2. Accept the self-signed certificates when prompted.

   You can create a custom certificate after login.

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

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

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

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

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

Troubleshoot the installation

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

Steps

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

```
kubectl logs --follow -n netapp-acc-operator $(kubectl get pods -n
netapp-acc-operator -o name)  -c manager
```

What's next

Complete the deployment by performing setup tasks.

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

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

```
oc get clusteroperators
```

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

```
oc get apiservices
```

- You have created an FQDN address for Astra Control Center in your data center.
- You have the necessary permissions and access to the Red Hat OpenShift Container Platform to perform the installation steps described.

Steps

- Download the Astra Control Center bundle
- Unpack the bundle and change directory
- Add the images to your local registry
- Find the operator install page
- Install the operator
- Install Astra Control Center

Download the Astra Control Center bundle

1. Download the Astra Control Center bundle (`astra-control-center-[version].tar.gz`) from the NetApp Support Site.
2. Download the zip of Astra Control Center certificates and keys from NetApp Support Site.
3. (Optional) Use the following command to verify the signature of the bundle:

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

Unpack the bundle and change directory

1. Extract the images:

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

2. Change to the Astra directory.

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

Add the images to your local registry

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

   See sample scripts for the automatic loading of images below.

   a. Log in to your registry:

      Docker:

      ```
      docker login [your_registry_path]
      ```

      Podman:

      ```
      podman login [your_registry_path]
      ```

   b. Use the appropriate script to load the images, tag the images, and push the images to your local registry:

      Docker:

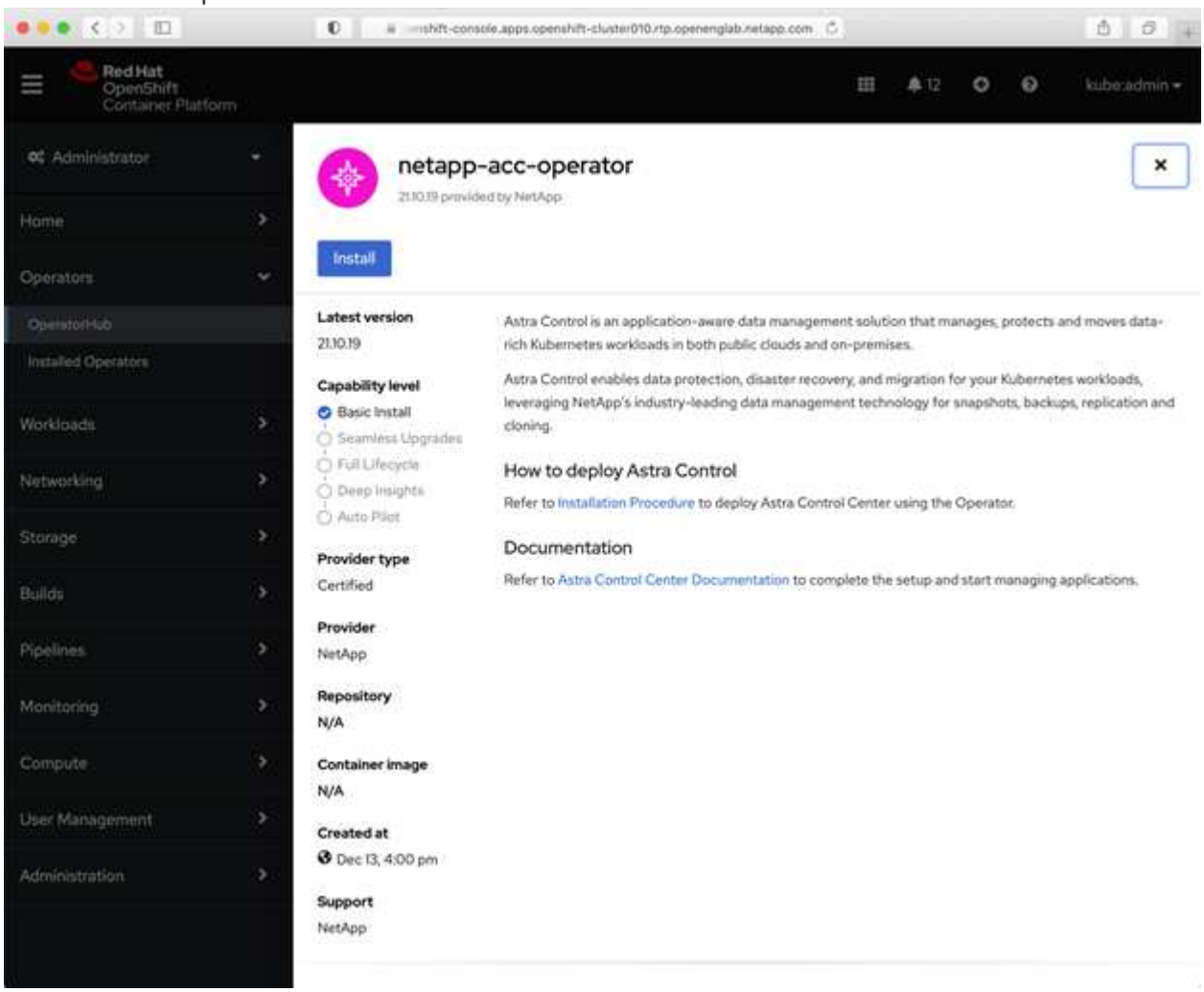
      ```
      export REGISTRY=[Docker_registry_path]
      for astraImageFile in $(ls images/*.tar) ; do
        # Load to local cache. And store the name of the loaded image
      trimming the 'Loaded images: '
        astraImage=$(docker load --input ${astraImageFile} | sed 's/Loaded
      image: //')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

      Podman:

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

Find the operator install page

1. Complete one of the following procedures to access the operator install page:
   - From Red Hat Openshift web console:



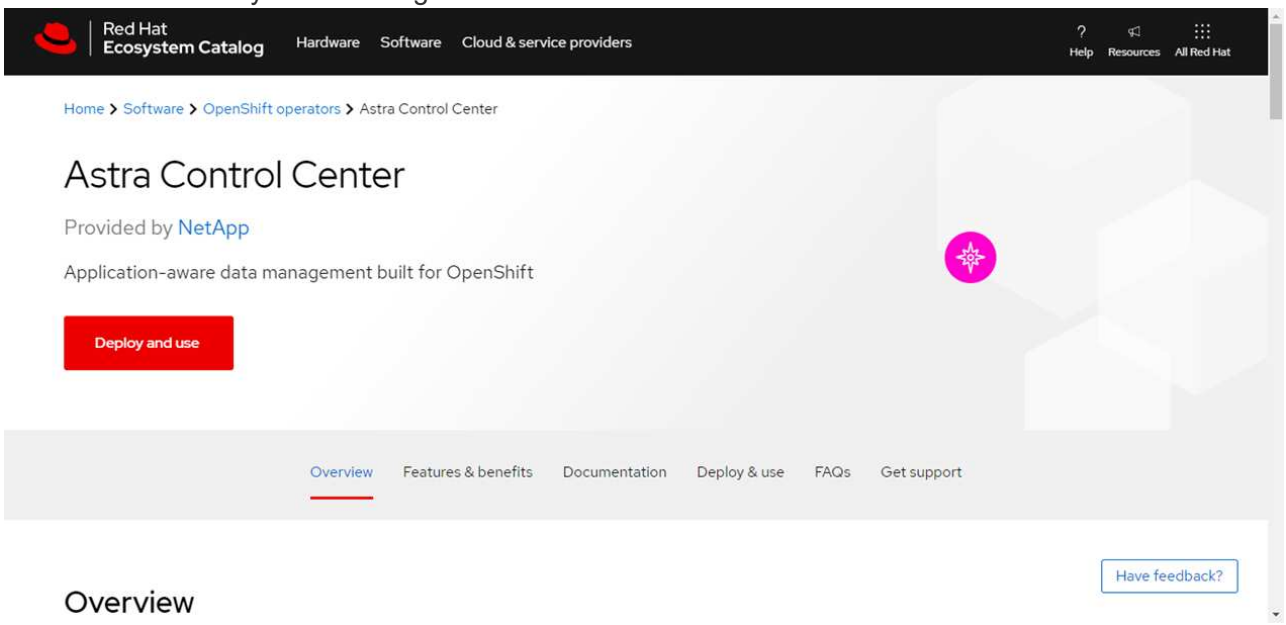
     a. Log in to the OpenShift Container Platform UI.

b. From the side menu, select **Operators > OperatorHub**.

c. Select the NetApp Astra Control Center operator.

d. Select **Install**.

- From Red Hat Ecosystem Catalog:



a. Select the NetApp Astra Control Center operator.

b. Select **Deploy and Use**.

Install the operator

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

   The operator will be available in all cluster namespaces.

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

   b. Select a manual or automatic approval strategy.

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

   c. Select **Install**.

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

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

Install Astra Control Center

1. From the console within the details view of the Astra Control Center operator, select `Create instance` in the Provided APIs section.

2. Complete the `Create AstraControlCenter` form field:
    a. Keep or adjust the Astra Control Center name.
    b. (Optional) Enable or disable Auto Support. Retaining Auto Support functionality is recommended.
    c. Enter the Astra Control Center address. Do not enter `http://` or `https://` in the address.
    d. Enter the Astra Control Center version; for example, 21.12.60.
    e. Enter an account name, email address, and admin last name.
    f. Retain the default volume reclaim policy.
    g. In **Image Registry**, enter your local container image registry path. Do not enter `http://` or `https://` in the address.
    h. If you use a registry that requires authentication, enter the secret.
    i. Enter the admin first name.
    j. Configure resources scaling.
    k. Retain the default storage class.
    l. Define CRD handling preferences.
3. Select `Create`.

What's next

Verify the successful installation of Astra Control Center and complete the remaining steps to log in. Additionally, you will complete the deployment by also performing setup tasks.