



Panoramica dell'installazione

Astra Control Center

NetApp

November 21, 2023

Sommario

- Panoramica dell'installazione 1
 - Installare Astra Control Center utilizzando il processo standard 1
 - Installare Astra Control Center utilizzando OpenShift OperatorHub 23
 - Installare il centro di controllo Astra con un backend di storage Cloud Volumes ONTAP 29

Panoramica dell'installazione

Scegliere e completare una delle seguenti procedure di installazione di Astra Control Center:

- ["Installare Astra Control Center utilizzando il processo standard"](#)
- ["\(Se utilizzi Red Hat OpenShift\) Installa Astra Control Center usando OpenShift OperatorHub"](#)
- ["Installare il centro di controllo Astra con un backend di storage Cloud Volumes ONTAP"](#)

Installare Astra Control Center utilizzando il processo standard

Per installare Astra Control Center, scaricare il pacchetto di installazione dal NetApp Support Site ed eseguire la procedura seguente per installare Astra Control Center Operator e Astra Control Center nel proprio ambiente. È possibile utilizzare questa procedura per installare Astra Control Center in ambienti connessi a Internet o con connessione ad aria.

Per gli ambienti Red Hat OpenShift, è possibile utilizzare anche un ["procedura alternativa"](#) Per installare Astra Control Center utilizzando OpenShift OperatorHub.

Di cosa hai bisogno

- ["Prima di iniziare l'installazione, preparare l'ambiente per l'implementazione di Astra Control Center"](#).
- Assicurarsi che tutti gli operatori del cluster siano in buono stato e disponibili.

Esempio di OpenShift:

```
oc get clusteroperators
```

- Assicurarsi che tutti i servizi API siano in buono stato e disponibili:

Esempio di OpenShift:

```
oc get apiservices
```

- L'FQDN Astra che si intende utilizzare deve essere instradabile a questo cluster. Ciò significa che si dispone di una voce DNS nel server DNS interno o si sta utilizzando un percorso URL principale già registrato.

A proposito di questa attività

Il processo di installazione di Astra Control Center esegue le seguenti operazioni:

- Installa i componenti Astra in `netapp-acc` namespace (o personalizzato).
- Crea un account predefinito.
- Stabilisce un indirizzo e-mail predefinito per l'utente amministrativo e una password monouso predefinita di `ACC-<UUID_of_installation>` Per questo caso di Astra Control Center. A questo utente viene assegnato il ruolo Owner (Proprietario) nel sistema ed è necessario per il primo accesso all'interfaccia utente.

- Consente di determinare se tutti i pod Astra Control Center sono in esecuzione.
- Installa l'interfaccia utente Astra.



(Valido solo per la release Astra Data Store Early Access Program (EAP)). Se si intende gestire Astra Data Store utilizzando Astra Control Center e abilitare i flussi di lavoro VMware, implementare Astra Control Center solo su `pcloud` namespace e non su `netapp-acc` namespace o uno spazio dei nomi personalizzato descritto nei passaggi di questa procedura.



Non eseguire il seguente comando durante l'intero processo di installazione per evitare di eliminare tutti i pod di Astra Control Center: `kubectl delete -f astra_control_center_operator_deploy.yaml`



Se si utilizza Podman di Red Hat invece di Docker Engine, è possibile utilizzare i comandi Podman al posto dei comandi Docker.

Fasi

Per installare Astra Control Center, procedere come segue:

- [Scarica e disimballa il bundle Astra Control Center](#)
- [Installare il plug-in NetApp Astra kubectl](#)
- [Aggiungere le immagini al registro locale](#)
- [Impostare namespace e secret per i registri con requisiti di autenticazione](#)
- [Installare l'operatore del centro di controllo Astra](#)
- [Configurare Astra Control Center](#)
- [Completare l'installazione dell'Astra Control Center e dell'operatore](#)
- [Verificare lo stato del sistema](#)
- [Impostare l'ingresso per il bilanciamento del carico](#)
- [Accedere all'interfaccia utente di Astra Control Center](#)

Scarica e disimballa il bundle Astra Control Center

1. Scarica il bundle Astra Control Center (`astra-control-center-[version].tar.gz`) da "[Sito di supporto NetApp](#)".
2. Scarica la zip dei certificati e delle chiavi di Astra Control Center dal "[Sito di supporto NetApp](#)".
3. (Facoltativo) utilizzare il seguente comando per verificare la firma del bundle:

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

4. Estrarre le immagini:

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

Installare il plug-in NetApp Astra kubectl

NetApp Astra `kubectl` Il plug-in della riga di comando consente di risparmiare tempo durante l'esecuzione di attività comuni associate all'implementazione e all'aggiornamento di Astra Control Center.

Di cosa hai bisogno

NetApp fornisce binari per il plug-in per diverse architetture CPU e sistemi operativi. Prima di eseguire questa attività, è necessario conoscere la CPU e il sistema operativo in uso. Sui sistemi operativi Linux e Mac, è possibile utilizzare `uname -a` per raccogliere queste informazioni.

Fasi

1. Elencare NetApp Astra disponibile `kubectl` Binari del plug-in e annotare il nome del file necessario per il sistema operativo e l'architettura della CPU:

```
ls kubectl-astra/
```

2. Copiare il file nella stessa posizione dello standard `kubectl` utility. In questo esempio, il `kubectl` l'utility si trova in `/usr/local/bin` directory. Sostituire `<binary-name>` con il nome del file desiderato:

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

Aggiungere le immagini al registro locale

1. Passare alla directory Astra:

```
cd acc
```

2. Aggiungere i file nella directory dell'immagine di Astra Control Center al registro locale.



Vedere gli script di esempio per il caricamento automatico delle immagini di seguito.

- a. Accedere al Registro di sistema:

Docker:

```
docker login [your_registry_path]
```

Podman:

```
podman login [your_registry_path]
```

- b. Utilizzare lo script appropriato per caricare le immagini, contrassegnare le immagini e inviare le immagini al registro locale:

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
```

Impostare namespace e secret per i registri con requisiti di autenticazione

1. Se si utilizza un registro che richiede l'autenticazione, è necessario effettuare le seguenti operazioni:
 - a. Creare il netapp-acc-operator spazio dei nomi:

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

Risposta:

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

- b. Creare un segreto per netapp-acc-operator namespace. Aggiungere informazioni su Docker ed

eseguire il seguente comando:

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

Esempio di risposta:

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

c. Creare il netapp-acc namespace (o personalizzato).

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

Esempio di risposta:

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

d. Creare un segreto per netapp-acc namespace (o personalizzato). Aggiungere informazioni su Docker ed eseguire il seguente comando:

```
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]
```

Risposta

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

a. (opzionale) se si desidera che il cluster venga gestito automaticamente da Astra Control Center dopo l'installazione, assicurarsi di fornire il kubeconfig come segreto all'interno dello spazio dei nomi di Astra Control Center in cui si intende eseguire la distribuzione utilizzando questo comando:

```
kubectl create secret generic [acc-kubeconfig-cred or custom secret name] --from-file=<path-to-your-kubeconfig> -n [netapp-acc or custom namespace]
```

Installare l'operatore del centro di controllo Astra

1. Modificare l'YAML di implementazione dell'operatore di Astra Control Center

(astra_control_center_operator_deploy.yaml) per fare riferimento al registro locale e al segreto.

```
vim astra_control_center_operator_deploy.yaml
```

- a. Se si utilizza un registro che richiede l'autenticazione, sostituire la riga predefinita di `imagePullSecrets: []` con i seguenti elementi:

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

- b. Cambiare `[your_registry_path]` per `kube-rbac-proxy` al percorso del registro in cui sono state inviate le immagini in a. [passaggio precedente](#).
- c. Cambiare `[your_registry_path]` per `acc-operator-controller-manager` al percorso del registro in cui sono state inviate le immagini in a. [passaggio precedente](#).
- d. (Per le installazioni che utilizzano l'anteprima di Astra Data Store) vedere questo problema noto relativo a. ["Provisioning delle classi di storage e modifiche aggiuntive da apportare al programma 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. Installare l'operatore del centro di controllo Astra:

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

Esempio di risposta:

```
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
```

Configurare Astra Control Center

1. Modificare il file delle risorse personalizzate (CR) di Astra Control Center (`astra_control_center_min.yaml`) Per creare account, AutoSupport, Registro di sistema e altre configurazioni necessarie:



Se sono necessarie personalizzazioni aggiuntive per il proprio ambiente, è possibile utilizzare `astra_control_center.yaml` Come CR alternativa. `astra_control_center_min.yaml` È il CR predefinito ed è adatto per la maggior parte delle installazioni.



Le proprietà configurate dal CR non possono essere modificate dopo l'implementazione iniziale di Astra Control Center.



Se si utilizza un registro che non richiede autorizzazione, è necessario eliminare `secret` linea entro `imageRegistry` in caso negativo, l'installazione non riesce.

- a. Cambiare `[your_registry_path]` al percorso del registro di sistema in cui sono state inviate le immagini nel passaggio precedente.
- b. Modificare il `accountName` stringa al nome che si desidera associare all'account.
- c. Modificare il `astraAddress` Stringa all'FQDN che si desidera utilizzare nel browser per accedere ad Astra. Non utilizzare `http://` oppure `https://` nell'indirizzo. Copiare questo FQDN per utilizzarlo in

un [passo successivo](#).

- d. Modificare il `email` stringa all'indirizzo iniziale predefinito dell'amministratore. Copiare questo indirizzo e-mail per utilizzarlo in [passo successivo](#).
- e. Cambiare `enrolled Per AutoSupport` a `false` per i siti senza connettività internet o senza `retain true` per i siti connessi.
- f. (Facoltativo) aggiungere un nome `firstName` e cognome `lastName` dell'utente associato all'account. È possibile eseguire questo passaggio ora o in un secondo momento all'interno dell'interfaccia utente.
- g. (Facoltativo) modificare `storageClass` Valore per un'altra risorsa Trident `storageClass`, se richiesto dall'installazione.
- h. (Facoltativo) se si desidera che il cluster venga gestito automaticamente da Astra Control Center dopo l'installazione e si è già provveduto [creato il segreto contenente il kubeconfig per questo cluster](#), Fornire il nome del segreto aggiungendo un nuovo campo a questo file YAML chiamato `astraKubeConfigSecret: "acc-kubeconfig-cred or custom secret name"`
- i. Completare una delle seguenti operazioni:

- **Other ingress controller (ingressType:Generic):** Questa è l'azione predefinita con Astra Control Center. Dopo l'implementazione di Astra Control Center, è necessario configurare il controller di ingresso per esporre Astra Control Center con un URL.

L'installazione predefinita di Astra Control Center imposta il gateway (`service/traefik`) per essere del tipo `ClusterIP`. Questa installazione predefinita richiede l'impostazione di Kubernetes IngressController/Ingress per instradare il traffico verso di essa. Se si desidera utilizzare un ingresso, vedere ["Impostare l'ingresso per il bilanciamento del carico"](#).

- **Service load balancer (ingressType:AccTraefik):** Se non si desidera installare un IngressController o creare una risorsa Ingress, impostare `ingressType` a `AccTraefik`.

In questo modo viene implementato l'Astra Control Center `traefik` Gateway come servizio di tipo Kubernetes LoadBalancer.

Astra Control Center utilizza un servizio del tipo "LoadBalancer" (`svc/traefik` Nello spazio dei nomi di Astra Control Center) e richiede l'assegnazione di un indirizzo IP esterno accessibile. Se nel proprio ambiente sono consentiti i bilanciatori di carico e non ne è già configurato uno, è possibile utilizzare MetalLB o un altro servizio di bilanciamento del carico esterno per assegnare un indirizzo IP esterno al servizio. Nella configurazione del server DNS interno, puntare il nome DNS scelto per Astra Control Center sull'indirizzo IP con bilanciamento del carico.



Per ulteriori informazioni sul tipo di servizio "LoadBalancer" e sull'ingresso, vedere ["Requisiti"](#).

```
apiVersion: astra.netapp.io/v1
kind: AstraControlCenter
metadata:
  name: astra
spec:
  accountName: "Example"
  astraVersion: "ASTRA_VERSION"
  astraAddress: "astra.example.com"
  astraKubeConfigSecret: "acc-kubeconfig-cred or custom secret name"
  ingressType: "Generic"
  autoSupport:
    enrolled: true
  email: "[admin@example.com]"
  firstName: "SRE"
  lastName: "Admin"
  imageRegistry:
    name: "[your_registry_path]"
    secret: "astra-registry-cred"
  storageClass: "ontap-gold"
```

Completare l'installazione dell'Astra Control Center e dell'operatore

1. Se non lo si è già fatto in un passaggio precedente, creare il `netapp-acc` namespace (o personalizzato):

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

Esempio di risposta:

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

2. Installare Astra Control Center in `netapp-acc` spazio dei nomi (o personalizzato):

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

Esempio di risposta:

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

Verificare lo stato del sistema



Se preferisci utilizzare OpenShift, puoi utilizzare comandi oc paragonabili per le fasi di verifica.

1. Verificare che tutti i componenti del sistema siano installati correttamente.

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

Ogni pod deve avere uno stato di Running. L'implementazione dei pod di sistema potrebbe richiedere alcuni minuti.

Esempio di risposta:

NAME	READY	STATUS	RESTARTS
AGE			
acc-helm-repo-5f75c5f564-bzqmt 11m	1/1	Running	0
activity-6b8f7ccb9-mlrn4 9m2s	1/1	Running	0
api-token-authentication-6hznt 8m50s	1/1	Running	0
api-token-authentication-qpfqb 8m50s	1/1	Running	0
api-token-authentication-sqnb7 8m50s	1/1	Running	0
asup-5578bbdd57-dxkbp 9m3s	1/1	Running	0
authentication-56bff4f95d-mspmq 7m31s	1/1	Running	0
bucket-service-6f7968b95d-9rrrl 8m36s	1/1	Running	0
cert-manager-5f6cf4bc4b-82khn 6m19s	1/1	Running	0
cert-manager-cainjector-76cf976458-sdrbc 6m19s	1/1	Running	0
cert-manager-webhook-5b7896bfd8-2n45j 6m19s	1/1	Running	0
cloud-extension-749d9f684c-8bdhq 9m6s	1/1	Running	0
cloud-insights-service-7d58687d9-h5tzw 8m56s	1/1	Running	2
composite-compute-968c79cb5-nv714 9m11s	1/1	Running	0
composite-volume-7687569985-jg9gg 8m33s	1/1	Running	0
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-15bjj	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-67pqt	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-t2x1k         1/1      Running  0
4m33s
traefik-5857d87f85-v9wvf         1/1      Running  0
7m3s
trident-svc-595f84dd78-zb816     1/1      Running  0
8m54s
vault-controller-86c94fbf4f-krttq 1/1      Running  0
9m24s

```

2. (Facoltativo) per assicurarsi che l'installazione sia completata, è possibile guardare `acc-operator` registra usando il seguente comando.

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



`accHost` la registrazione del cluster è una delle ultime operazioni e, in caso di errore, la distribuzione non avrà esito negativo. In caso di errore di registrazione del cluster indicato nei registri, è possibile tentare di nuovo la registrazione attraverso il flusso di lavoro `add cluster` "Nell'interfaccia utente" O API.

3. Una volta eseguiti tutti i pod, verificare che l'installazione sia riuscita recuperando `AstraControlCenter` Istanza installata dall'operatore di Astra Control Center.

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

4. Nell'YAML, selezionare `status.deploymentState` nella risposta per `Deployed` valore. Se l'implementazione non ha avuto esito positivo, viene visualizzato un messaggio di errore.
5. Per ottenere la password monouso da utilizzare quando si accede ad Astra Control Center, copiare il `status.uuid` valore. La password è `ACC- Seguito dal valore UUID (ACC- [UUID] oppure, in questo esempio, ACC-9aa5fdae-4214-4cb7-9976-5d8b4c0ce27f)`.

Dettagli YAML di esempio

```
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: ""

```

Impostare l'ingresso per il bilanciamento del carico

È possibile configurare un controller di ingresso Kubernetes che gestisce l'accesso esterno ai servizi, come il bilanciamento del carico in un cluster.

Questa procedura spiega come configurare un controller di ingresso (`ingressType:Generic`). Questa è l'azione predefinita con Astra Control Center. Dopo l'implementazione di Astra Control Center, è necessario configurare il controller di ingresso per esporre Astra Control Center con un URL.



Se non si desidera configurare un controller di ingresso, è possibile impostarlo (`ingressType:AccTraefik`). Astra Control Center utilizza un servizio del tipo "LoadBalancer" (`svc/traefik` Nello spazio dei nomi di Astra Control Center) e richiede l'assegnazione di un indirizzo IP esterno accessibile. Se nel proprio ambiente sono consentiti i bilanciatori di carico e non ne è già configurato uno, è possibile utilizzare MetalLB o un altro servizio di bilanciamento del carico esterno per assegnare un indirizzo IP esterno al servizio. Nella configurazione del server DNS interno, puntare il nome DNS scelto per Astra Control Center sull'indirizzo IP con bilanciamento del carico. Per ulteriori informazioni sul tipo di servizio "LoadBalancer" e sull'ingresso, vedere ["Requisiti"](#).

I passaggi variano a seconda del tipo di controller di ingresso utilizzato:

- Controller di ingresso nginx
- Controller di ingresso OpenShift

Di cosa hai bisogno

- Il necessario ["controller di ingresso"](#) dovrebbe essere già implementato.
- Il ["classe di ingresso"](#) corrispondente al controller di ingresso dovrebbe già essere creato.
- Si stanno utilizzando versioni di Kubernetes comprese tra v1.19 e v1.22.

Procedura per il controller di ingresso Nginx

1. Creare un segreto di tipo[`kubernetes.io/tls`] Per una chiave privata TLS e un certificato in `netapp-acc` (o con nome personalizzato) come descritto in ["Segreti TLS"](#).
2. Implementare una risorsa income in `netapp-acc` (o con nome personalizzato) namespace utilizzando `v1beta1` (Obsoleto in Kubernetes versione inferiore a o 1.22) o. `v1` tipo di risorsa per uno schema obsoleto o nuovo:
 - a. Per a. `v1beta1` schema obsoleto, seguire questo esempio:

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: ingress-acc
  namespace: [netapp-acc or custom namespace]
  annotations:
    kubernetes.io/ingress.class: [class name for nginx controller]
spec:
  tls:
    - hosts:
      - <ACC address>
      secretName: [tls secret name]
  rules:
    - host: [ACC address]
      http:
        paths:
          - backend:
              serviceName: traefik
              servicePort: 80
            pathType: ImplementationSpecific
```

b. Per v1 nuovo schema, seguire questo esempio:

```

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

```

Procedura per il controller di ingresso OpenShift

1. Procurarsi il certificato e ottenere la chiave, il certificato e i file CA pronti per l'uso con il percorso OpenShift.
2. Creare il percorso OpenShift:

```

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

```

Accedere all'interfaccia utente di Astra Control Center

Dopo aver installato Astra Control Center, si modifica la password dell'amministratore predefinito e si accede alla dashboard dell'interfaccia utente di Astra Control Center.

Fasi

1. In un browser, immettere l'FQDN utilizzato in `astraAddress` in `astra_control_center_min.yaml` CR quando [Astra Control Center è stato installato](#).
2. Accettare i certificati autofirmati quando richiesto.



È possibile creare un certificato personalizzato dopo l'accesso.

3. Nella pagina di accesso di Astra Control Center, inserire il valore utilizzato per `email` poll `astra_control_center_min.yaml` CR quando [Astra Control Center è stato installato](#), seguito dalla password monouso (`ACC-[UUID]`).



Se si immette una password errata per tre volte, l'account admin viene bloccato per 15 minuti.

4. Selezionare **Login**.
5. Modificare la password quando richiesto.



Se si tratta del primo accesso e si dimentica la password e non sono ancora stati creati altri account utente amministrativi, contattare il supporto NetApp per assistenza per il recupero della password.

6. (Facoltativo) rimuovere il certificato TLS autofirmato esistente e sostituirlo con un ["Certificato TLS personalizzato firmato da un'autorità di certificazione \(CA\)"](#).

Risolvere i problemi di installazione

Se uno dei servizi è in `Error` stato, è possibile esaminare i registri. Cercare i codici di risposta API nell'intervallo da 400 a 500. Questi indicano il luogo in cui si è verificato un guasto.

Fasi

1. Per esaminare i registri dell'operatore di Astra Control Center, immettere quanto segue:

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

Cosa succederà

Completare l'implementazione eseguendo ["attività di installazione"](#).

Installare Astra Control Center utilizzando OpenShift OperatorHub

Se utilizzi Red Hat OpenShift, puoi installare Astra Control Center usando l'operatore certificato Red Hat. Seguire questa procedura per installare Astra Control Center da ["Catalogo Red Hat Ecosystem"](#) Oppure utilizzando Red Hat OpenShift Container Platform.

Una volta completata questa procedura, tornare alla procedura di installazione per completare la ["fasi rimanenti"](#) per verificare che l'installazione sia riuscita e accedere.

Di cosa hai bisogno

- ["Prima di iniziare l'installazione, preparare l'ambiente per l'implementazione di Astra Control Center"](#).
- Dal tuo cluster OpenShift, assicurati che tutti gli operatori del cluster siano in buono stato (`available` è

true):

```
oc get clusteroperators
```

- Dal cluster OpenShift, assicurati che tutti i servizi API siano in buono stato (`available` è `true`):

```
oc get apiservices
```

- Hai creato un indirizzo FQDN per Astra Control Center nel tuo data center.
- Hai i permessi necessari e l'accesso alla piattaforma container Red Hat OpenShift per eseguire le fasi di installazione descritte.

Fasi

- [Scarica e disimballa il bundle Astra Control Center](#)
- [Installare il plug-in NetApp Astra kubect1](#)
- [Aggiungere le immagini al registro locale](#)
- [Individuare la pagina di installazione dell'operatore](#)
- [Installare l'operatore](#)
- [Installare Astra Control Center](#)

Scarica e disimballa il bundle Astra Control Center

1. Scarica il bundle Astra Control Center (`astra-control-center-[version].tar.gz`) da "[Sito di supporto NetApp](#)".
2. Scarica la zip dei certificati e delle chiavi di Astra Control Center dal "[Sito di supporto NetApp](#)".
3. (Facoltativo) utilizzare il seguente comando per verificare la firma del bundle:

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

4. Estrarre le immagini:

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

Installare il plug-in NetApp Astra kubect1

NetApp Astra `kubect1` Il plug-in della riga di comando consente di risparmiare tempo durante l'esecuzione di attività comuni associate all'implementazione e all'aggiornamento di Astra Control Center.

Di cosa hai bisogno

NetApp fornisce binari per il plug-in per diverse architetture CPU e sistemi operativi. Prima di eseguire questa

attività, è necessario conoscere la CPU e il sistema operativo in uso. Sui sistemi operativi Linux e Mac, è possibile utilizzare `uname -a` per raccogliere queste informazioni.

Fasi

1. Elencare NetApp Astra disponibile `kubectl` Binari del plug-in e annotare il nome del file necessario per il sistema operativo e l'architettura della CPU:

```
ls kubectl-astra/
```

2. Copiare il file nella stessa posizione dello standard `kubectl` utility. In questo esempio, il `kubectl` l'utility si trova in `/usr/local/bin` directory. Sostituire `<binary-name>` con il nome del file desiderato:

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

Aggiungere le immagini al registro locale

1. Passare alla directory Astra:

```
cd acc
```

2. Aggiungere i file nella directory dell'immagine di Astra Control Center al registro locale.



Vedere gli script di esempio per il caricamento automatico delle immagini di seguito.

- a. Accedere al Registro di sistema:

Docker:

```
docker login [your_registry_path]
```

Podman:

```
podman login [your_registry_path]
```

- b. Utilizzare lo script appropriato per caricare le immagini, contrassegnare le immagini e inviare le immagini al registro locale:

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:

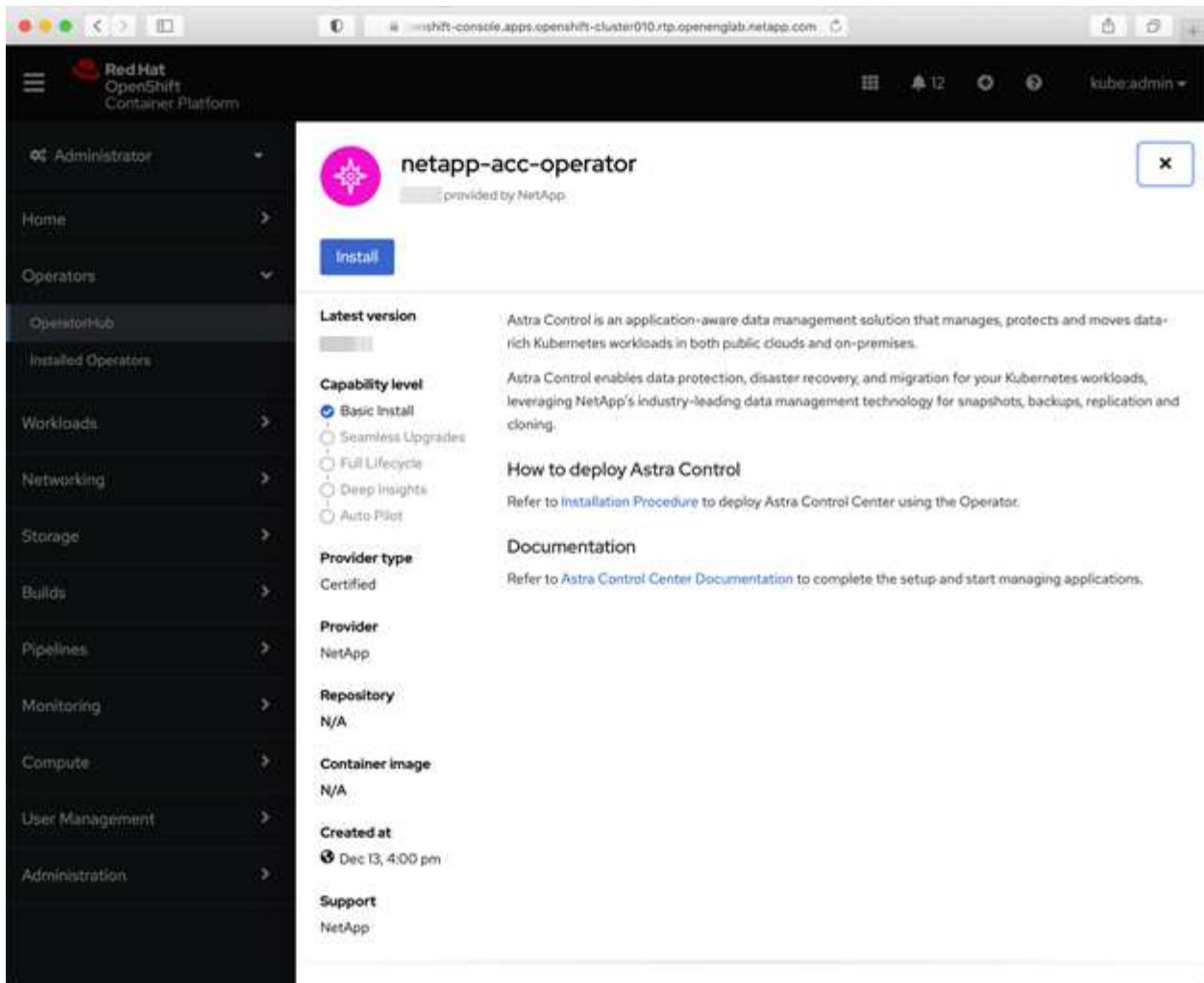
```

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

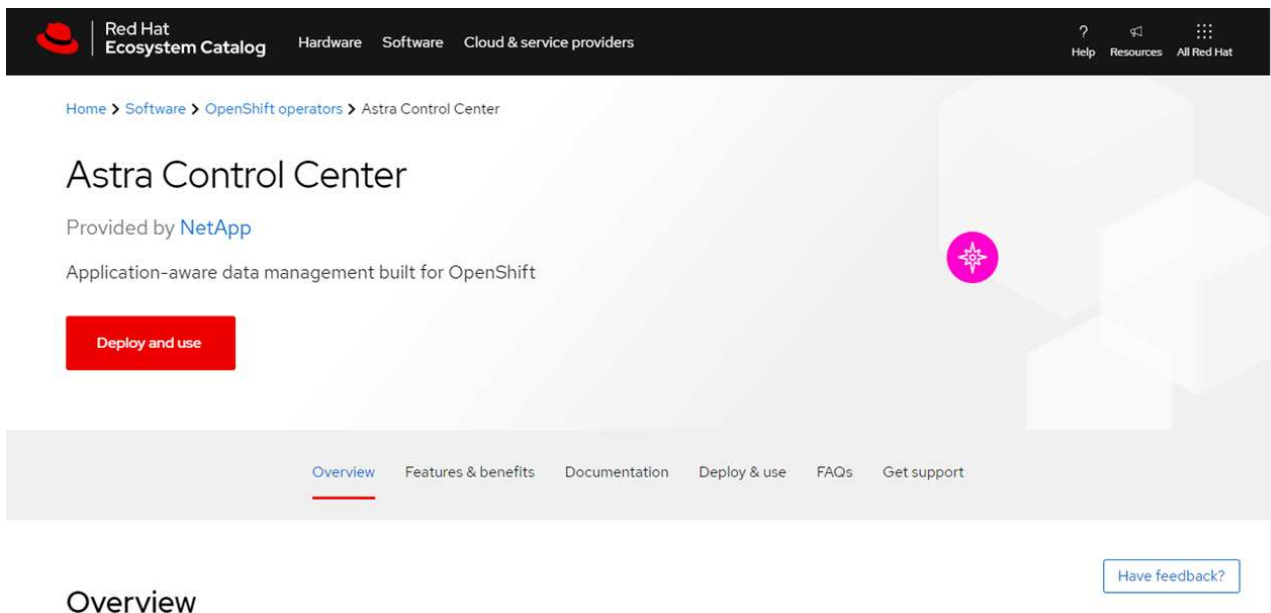
```

Individuare la pagina di installazione dell'operatore

1. Completare una delle seguenti procedure per accedere alla pagina di installazione dell'operatore:
 - Dalla console Web Red Hat OpenShift:



- i. Accedere all'interfaccia utente di OpenShift Container Platform.
 - ii. Dal menu laterale, selezionare **Operator (operatori) > OperatorHub**.
 - iii. Selezionare l'operatore di NetApp Astra Control Center.
 - iv. Selezionare **Installa**.
- o Dal Red Hat Ecosystem Catalog:



Overview

- i. Selezionare NetApp Astra Control Center "operator".
- ii. Selezionare **Deploy and Use** (implementazione e utilizzo).

Installare l'operatore

1. Completare la pagina **Install Operator** (Installazione operatore) e installare l'operatore:



L'operatore sarà disponibile in tutti gli spazi dei nomi dei cluster.

- a. Selezionare lo spazio dei nomi dell'operatore o `netapp-acc-operator` lo spazio dei nomi verrà creato automaticamente come parte dell'installazione dell'operatore.
- b. Selezionare una strategia di approvazione manuale o automatica.



Si consiglia l'approvazione manuale. Per ogni cluster dovrebbe essere in esecuzione una sola istanza dell'operatore.

- c. Selezionare **Installa**.



Se è stata selezionata una strategia di approvazione manuale, verrà richiesto di approvare il piano di installazione manuale per questo operatore.

2. Dalla console, accedere al menu OperatorHub e verificare che l'installazione dell'operatore sia stata eseguita correttamente.

Installare Astra Control Center

1. Dalla console nella vista dettagli dell'operatore Astra Control Center, selezionare `Create instance` Nella sezione API fornite.
2. Completare il `Create AstraControlCenter` campo del modulo:
 - a. Mantenere o regolare il nome di Astra Control Center.
 - b. (Facoltativo) attivare o disattivare il supporto automatico. Si consiglia di mantenere la funzionalità di

supporto automatico.

- c. Inserire l'indirizzo di Astra Control Center. Non entrare `http://` oppure `https://` nell'indirizzo.
 - d. Inserire la versione di Astra Control Center, ad esempio 21.12.60.
 - e. Immettere un nome account, un indirizzo e-mail e un cognome amministratore.
 - f. Mantenere la policy di recupero del volume predefinita.
 - g. In **Image Registry**, immettere il percorso locale del Registro di sistema dell'immagine container. Non entrare `http://` oppure `https://` nell'indirizzo.
 - h. Se si utilizza un registro che richiede l'autenticazione, immettere il segreto.
 - i. Inserire il nome admin.
 - j. Configurare la scalabilità delle risorse.
 - k. Mantenere la classe di storage predefinita.
 - l. Definire le preferenze di gestione CRD.
3. Selezionare `Create`.

Cosa succederà

Verificare che Astra Control Center sia stato installato correttamente e completare il "[fasi rimanenti](#)" per accedere. Inoltre, completerai l'implementazione eseguendo anche questa operazione "[attività di installazione](#)".

Installare il centro di controllo Astra con un backend di storage Cloud Volumes ONTAP

Con Astra Control Center, puoi gestire le tue app in un ambiente di cloud ibrido con cluster Kubernetes e istanze di Cloud Volumes ONTAP autogestiti. Puoi implementare Astra Control Center nei tuoi cluster Kubernetes on-premise o in uno dei cluster Kubernetes autogestiti nell'ambiente cloud.

Con una di queste implementazioni, è possibile eseguire operazioni di gestione dei dati delle applicazioni utilizzando Cloud Volumes ONTAP come back-end dello storage. È inoltre possibile configurare un bucket S3 come destinazione del backup.

Per installare Astra Control Center in Amazon Web Services (AWS) e Microsoft Azure con un backend di storage Cloud Volumes ONTAP, eseguire i seguenti passaggi a seconda dell'ambiente cloud in uso.

- [Implementare Astra Control Center in Amazon Web Services](#)
- [Implementare Astra Control Center in Microsoft Azure](#)

Implementare Astra Control Center in Amazon Web Services

Puoi implementare Astra Control Center su un cluster Kubernetes autogestito ospitato su un cloud pubblico Amazon Web Services (AWS).

Solo i cluster OpenShift Container Platform (OCP) autogestiti sono supportati per l'implementazione di Astra Control Center.

Ciò di cui hai bisogno per AWS

Prima di implementare Astra Control Center in AWS, sono necessari i seguenti elementi:

- Licenza Astra Control Center. Vedere "[Requisiti di licenza di Astra Control Center](#)".
- "[Soddisfare i requisiti di Astra Control Center](#)".
- Account NetApp Cloud Central
- Autorizzazioni Red Hat OpenShift Container Platform (OCP) (a livello di spazio dei nomi per la creazione di pod)
- Credenziali AWS, Access ID e Secret Key con autorizzazioni che consentono di creare bucket e connettori
- Accesso e login al Registro dei container elastici (ECR) dell'account AWS
- Per accedere all'interfaccia utente di Astra Control, è necessario immettere AWS Hosted zone e Route 53

Requisiti dell'ambiente operativo per AWS

Astra Control Center richiede il seguente ambiente operativo per AWS:

- Red Hat OpenShift Container Platform 4.8



Assicurarsi che l'ambiente operativo scelto per ospitare Astra Control Center soddisfi i requisiti di base delle risorse descritti nella documentazione ufficiale dell'ambiente.

Astra Control Center richiede le seguenti risorse oltre ai requisiti delle risorse dell'ambiente:

Componente	Requisito
Capacità di storage NetApp Cloud Volumes ONTAP di back-end	Almeno 300 GB disponibili
Nodi di lavoro (requisito AWS EC2)	Almeno 3 nodi di lavoro in totale, con 4 core vCPU e 12 GB di RAM ciascuno
Bilanciamento del carico	Tipo di servizio "LoadBalancer" disponibile per il traffico in ingresso da inviare ai servizi nel cluster dell'ambiente operativo
FQDN	Metodo per indirizzare l'FQDN di Astra Control Center all'indirizzo IP con bilanciamento del carico
Astra Trident (installato come parte del rilevamento dei cluster Kubernetes in NetApp Cloud Manager)	Astra Trident 21.04 o versione successiva installata e configurata e NetApp ONTAP versione 9.5 o successiva come backend di storage

Componente	Requisito
Registro delle immagini	<p>È necessario disporre di un registro privato esistente, ad esempio AWS Elastic Container Registry, in cui è possibile trasferire le immagini di build di Astra Control Center. È necessario fornire l'URL del registro delle immagini in cui verranno caricate le immagini.</p> <div style="border: 1px solid #ccc; padding: 10px; margin-top: 10px;">  <p>Il cluster ospitato da Astra Control Center e il cluster gestito devono avere accesso alla stessa immagine di registro per poter eseguire il backup e il ripristino delle applicazioni utilizzando l'immagine basata su Restic.</p> </div>
Configurazione di Astra Trident/ONTAP	<p>Astra Control Center richiede la creazione e l'impostazione di una classe di storage come classe di storage predefinita. Il centro di controllo Astra supporta le seguenti classi di storage Kubernetes create quando si importa il cluster Kubernetes in ONTAP. Questi sono forniti da 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>



Questi requisiti presuppongono che Astra Control Center sia l'unica applicazione in esecuzione nell'ambiente operativo. Se nell'ambiente sono in esecuzione applicazioni aggiuntive, modificare di conseguenza questi requisiti minimi.



Il token del Registro di sistema AWS scade tra 12 ore, dopodiché sarà necessario rinnovare il segreto del Registro di sistema dell'immagine Docker.

Panoramica dell'implementazione per AWS

Di seguito viene fornita una panoramica del processo di installazione di Astra Control Center per AWS con Cloud Volumes ONTAP come backend di storage.

Ciascuna di queste fasi viene illustrata più dettagliatamente di seguito.

1. [Assicurarsi di disporre di autorizzazioni IAM sufficienti.](#)
2. [Installare un cluster RedHat OpenShift su AWS.](#)
3. [Configurare AWS.](#)
4. [Configurare NetApp Cloud Manager.](#)
5. [Installare Astra Control Center.](#)

Assicurarsi di disporre di autorizzazioni IAM sufficienti

Assicurarsi di disporre di ruoli e autorizzazioni IAM sufficienti per installare un cluster RedHat OpenShift e un connettore NetApp Cloud Manager.

Vedere ["Credenziali AWS iniziali"](#).

Installare un cluster RedHat OpenShift su AWS

Installare un cluster RedHat OpenShift Container Platform su AWS.

Per istruzioni sull'installazione, vedere ["Installazione di un cluster su AWS in OpenShift Container Platform"](#).

Configurare AWS

Quindi, configurare AWS per creare una rete virtuale, configurare istanze di calcolo EC2, creare un bucket AWS S3, creare un Elastic Container Register (ECR) per ospitare le immagini di Astra Control Center e inviare le immagini a questo registro.

Seguire la documentazione di AWS per completare i seguenti passaggi. Vedere ["Documentazione di installazione di AWS"](#).

1. Creare una rete virtuale AWS.
2. Esaminare le istanze di calcolo EC2. Può trattarsi di un server bare metal o di macchine virtuali in AWS.
3. Se il tipo di istanza non corrisponde già ai requisiti minimi di risorsa Astra per i nodi master e worker, modificare il tipo di istanza in AWS per soddisfare i requisiti Astra. Vedere ["Requisiti di Astra Control Center"](#).
4. Creare almeno un bucket AWS S3 per memorizzare i backup.
5. Creare un AWS Elastic Container Registry (ECR) per ospitare tutte le immagini ACC.



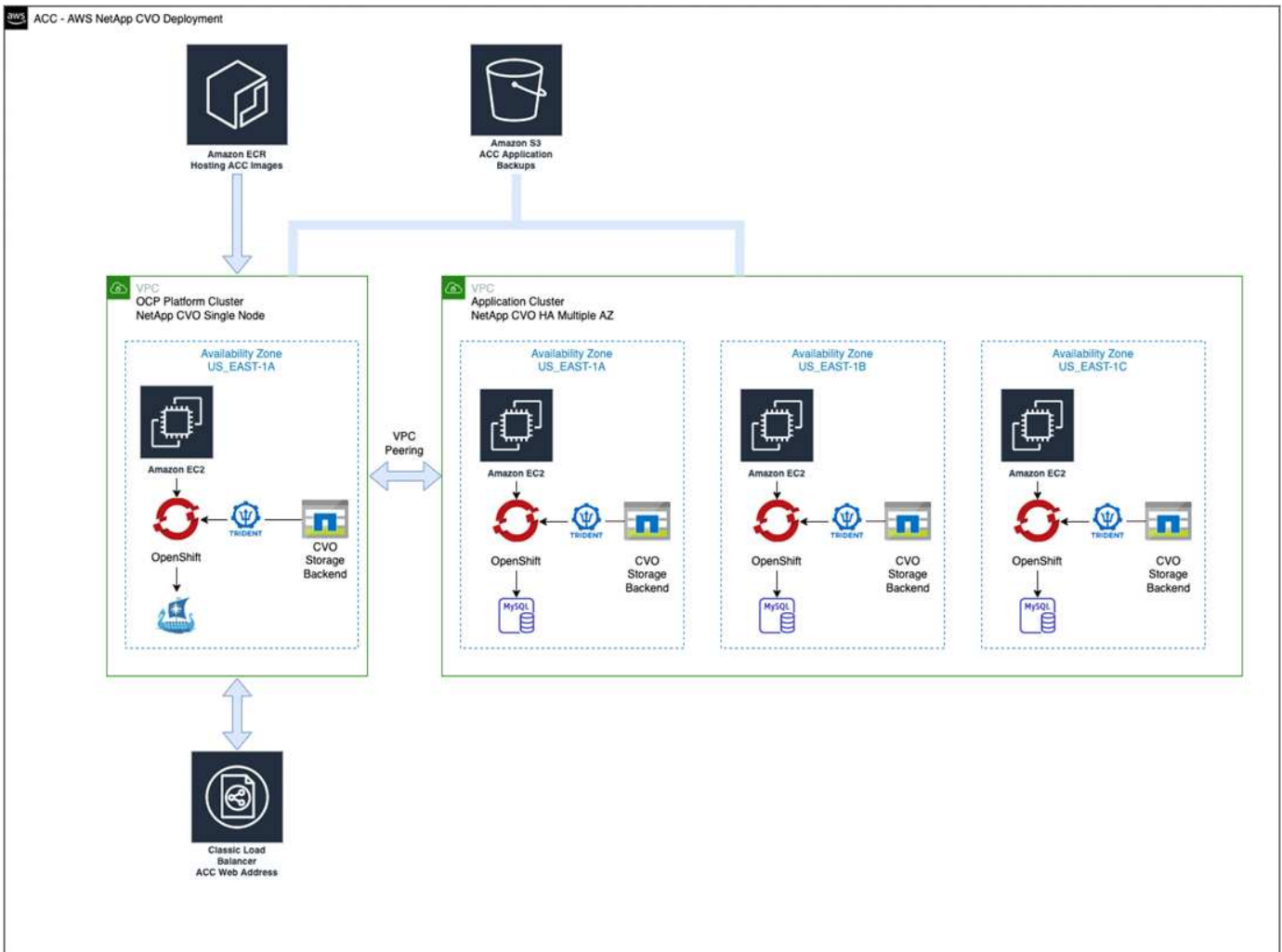
Se non si crea ECR, il centro di controllo Astra non può accedere ai dati di monitoraggio da un cluster contenente Cloud Volumes ONTAP con un backend AWS. Il problema si verifica quando il cluster che si tenta di rilevare e gestire utilizzando Astra Control Center non dispone dell'accesso ad AWS ECR.

6. Trasferire le immagini ACC nel registro definito.



Il token AWS Elastic Container Registry (ECR) scade dopo 12 ore e causa il fallimento delle operazioni di cloni tra cluster. Questo problema si verifica quando si gestisce un backend di storage da Cloud Volumes ONTAP configurato per AWS. Per correggere questo problema, autenticare nuovamente con ECR e generare un nuovo segreto per la ripresa delle operazioni di clonazione.

Ecco un esempio di implementazione di AWS:



Configurare NetApp Cloud Manager

Utilizzando Cloud Manager, creare un'area di lavoro, aggiungere un connettore ad AWS, creare un ambiente di lavoro e importare il cluster.

Seguire la documentazione di Cloud Manager per completare i seguenti passaggi. Vedere quanto segue:

- ["Introduzione a Cloud Volumes ONTAP in AWS"](#).
- ["Creare un connettore in AWS utilizzando Cloud Manager"](#)

Fasi

1. Aggiungere le tue credenziali a Cloud Manager.
2. Creare un'area di lavoro.
3. Aggiungere un connettore per AWS. Scegliere AWS come provider.
4. Creare un ambiente di lavoro per il tuo ambiente cloud.
 - a. Location: "Amazon Web Services (AWS)"
 - b. Tipo: "Cloud Volumes ONTAP ha"
5. Importare il cluster OpenShift. Il cluster si conatterà all'ambiente di lavoro appena creato.
 - a. Per visualizzare i dettagli del cluster NetApp, selezionare **K8s > elenco cluster > Dettagli cluster**.

- b. Nell'angolo in alto a destra, prendere nota della versione di Trident.
- c. Si noti che le classi di storage cluster Cloud Volumes ONTAP mostrano NetApp come provider.

In questo modo, il cluster Red Hat OpenShift viene importato e viene assegnata una classe di storage predefinita. Selezionare la classe di storage. Trident viene installato automaticamente come parte del processo di importazione e rilevamento.

6. Tenere presenti tutti i volumi e i volumi persistenti in questa implementazione di Cloud Volumes ONTAP.



Cloud Volumes ONTAP può funzionare come nodo singolo o in alta disponibilità. Se ha è attivato, annotare lo stato ha e lo stato di implementazione del nodo in esecuzione in AWS.

Installare Astra Control Center

Seguire lo standard ["Istruzioni di installazione di Astra Control Center"](#).

Implementare Astra Control Center in Microsoft Azure

Puoi implementare Astra Control Center su un cluster Kubernetes autogestito ospitato su un cloud pubblico Microsoft Azure.

Ciò di cui hai bisogno per Azure

Prima di implementare Astra Control Center in Azure, sono necessari i seguenti elementi:

- Licenza Astra Control Center. Vedere ["Requisiti di licenza di Astra Control Center"](#).
- ["Soddisfare i requisiti di Astra Control Center"](#).
- Account NetApp Cloud Central
- Red Hat OpenShift Container Platform (OCP) 4.8
- Autorizzazioni Red Hat OpenShift Container Platform (OCP) (a livello di spazio dei nomi per la creazione di pod)
- Credenziali Azure con autorizzazioni che consentono di creare bucket e connettori


Requisiti dell'ambiente operativo per Azure

Assicurarsi che l'ambiente operativo scelto per ospitare Astra Control Center soddisfi i requisiti di base delle risorse descritti nella documentazione ufficiale dell'ambiente.

Astra Control Center richiede le seguenti risorse oltre ai requisiti delle risorse dell'ambiente:

Vedere ["Requisiti dell'ambiente operativo di Astra Control Center"](#).

Componente	Requisito
Capacità di storage NetApp Cloud Volumes ONTAP di back-end	Almeno 300 GB disponibili
Nodi di lavoro (requisito di calcolo di Azure)	Almeno 3 nodi di lavoro in totale, con 4 core vCPU e 12 GB di RAM ciascuno

Componente	Requisito
Bilanciamento del carico	Tipo di servizio "LoadBalancer" disponibile per il traffico in ingresso da inviare ai servizi nel cluster dell'ambiente operativo
FQDN (Azure DNS zone)	Metodo per indirizzare l'FQDN di Astra Control Center all'indirizzo IP con bilanciamento del carico
Astra Trident (installato come parte del rilevamento dei cluster Kubernetes in NetApp Cloud Manager)	Astra Trident 21.04 o versione successiva installata e configurata e NetApp ONTAP versione 9.5 o successiva verrà utilizzato come backend di storage
Registro delle immagini	<p>È necessario disporre di un registro privato esistente, ad esempio Azure Container Registry (ACR), in cui è possibile trasferire le immagini di build di Astra Control Center. È necessario fornire l'URL del registro delle immagini in cui verranno caricate le immagini.</p> <div style="border-left: 1px solid #ccc; padding-left: 10px; margin-top: 10px;">  È necessario abilitare l'accesso anonimo per estrarre le immagini Restic per i backup. </div>
Configurazione di Astra Trident/ONTAP	<p>Astra Control Center richiede la creazione e l'impostazione di una classe di storage come classe di storage predefinita. Il centro di controllo Astra supporta le seguenti classi di storage Kubernetes create quando si importa il cluster Kubernetes in ONTAP. Questi sono forniti da 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>



Questi requisiti presuppongono che Astra Control Center sia l'unica applicazione in esecuzione nell'ambiente operativo. Se nell'ambiente sono in esecuzione applicazioni aggiuntive, modificare di conseguenza questi requisiti minimi.

Panoramica dell'implementazione di Azure

Ecco una panoramica del processo di installazione di Astra Control Center per Azure.

Ciascuna di queste fasi viene illustrata più dettagliatamente di seguito.

1. [Installare un cluster RedHat OpenShift su Azure.](#)
2. [Creare gruppi di risorse Azure.](#)

3. [Assicurarsi di disporre di autorizzazioni IAM sufficienti](#).
4. [Configurare Azure](#).
5. [Configurare NetApp Cloud Manager](#).
6. [Installare e configurare Astra Control Center](#).

Installare un cluster RedHat OpenShift su Azure

Il primo passo consiste nell'installare un cluster RedHat OpenShift su Azure.

Per istruzioni sull'installazione, consulta la documentazione di RedHat all'indirizzo ["Installazione del cluster OpenShift su Azure"](#) e ["Installazione di un account Azure"](#).

Creare gruppi di risorse Azure

Creare almeno un gruppo di risorse Azure.



OpenShift potrebbe creare i propri gruppi di risorse. Oltre a questi, è necessario definire anche i gruppi di risorse di Azure. Fare riferimento alla documentazione di OpenShift.

È possibile creare un gruppo di risorse del cluster di piattaforme e un gruppo di risorse del cluster OpenShift dell'applicazione di destinazione.

Assicurarsi di disporre di autorizzazioni IAM sufficienti

Assicurarsi di disporre di ruoli e autorizzazioni IAM sufficienti per installare un cluster RedHat OpenShift e un connettore NetApp Cloud Manager.

Vedere ["Credenziali e permessi di Azure"](#).

Configurare Azure

Quindi, configurare Azure per creare una rete virtuale, configurare istanze di calcolo, creare un container Azure Blob, creare un Azure Container Register (ACR) per ospitare le immagini di Astra Control Center e inviare le immagini a questo registro.

Seguire la documentazione di Azure per completare i seguenti passaggi. Vedere ["Installazione del cluster OpenShift su Azure"](#).

1. Creare una rete virtuale Azure.
2. Esaminare le istanze di calcolo. Si tratta di un server bare metal o di macchine virtuali in Azure.
3. Se il tipo di istanza non corrisponde già ai requisiti minimi di risorsa Astra per i nodi master e worker, modificare il tipo di istanza in Azure per soddisfare i requisiti Astra. Vedere ["Requisiti di Astra Control Center"](#).
4. Creare almeno un container Azure Blob per memorizzare i backup.
5. Creare un account storage. Per creare un container da utilizzare come bucket in Astra Control Center è necessario un account storage.
6. Creare un segreto, necessario per l'accesso al bucket.
7. Creare un Azure Container Registry (ACR) per ospitare tutte le immagini di Astra Control Center.
8. Impostare l'accesso ACR per il push/pull di tutte le immagini di Astra Control Center di Docker.

9. Inviare le immagini ACC a questo registro inserendo il seguente script:

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

Esempio:

```
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. Impostare le zone DNS.

Configurare NetApp Cloud Manager

Utilizzando Cloud Manager, creare un'area di lavoro, aggiungere un connettore ad Azure, creare un ambiente di lavoro e importare il cluster.

Seguire la documentazione di Cloud Manager per completare i seguenti passaggi. Vedere ["Introduzione a Cloud Manager in Azure"](#).

Di cosa hai bisogno

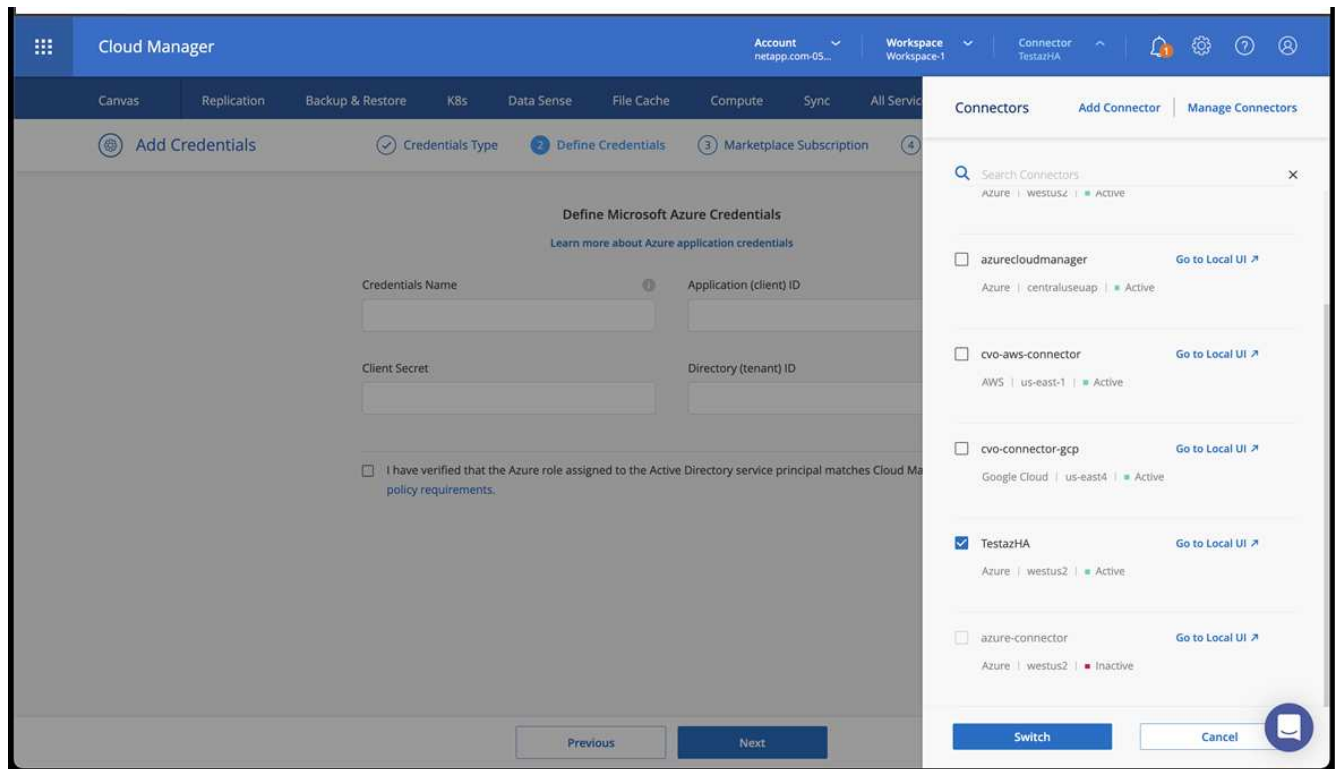
Accesso all'account Azure con le autorizzazioni e i ruoli IAM richiesti

Fasi

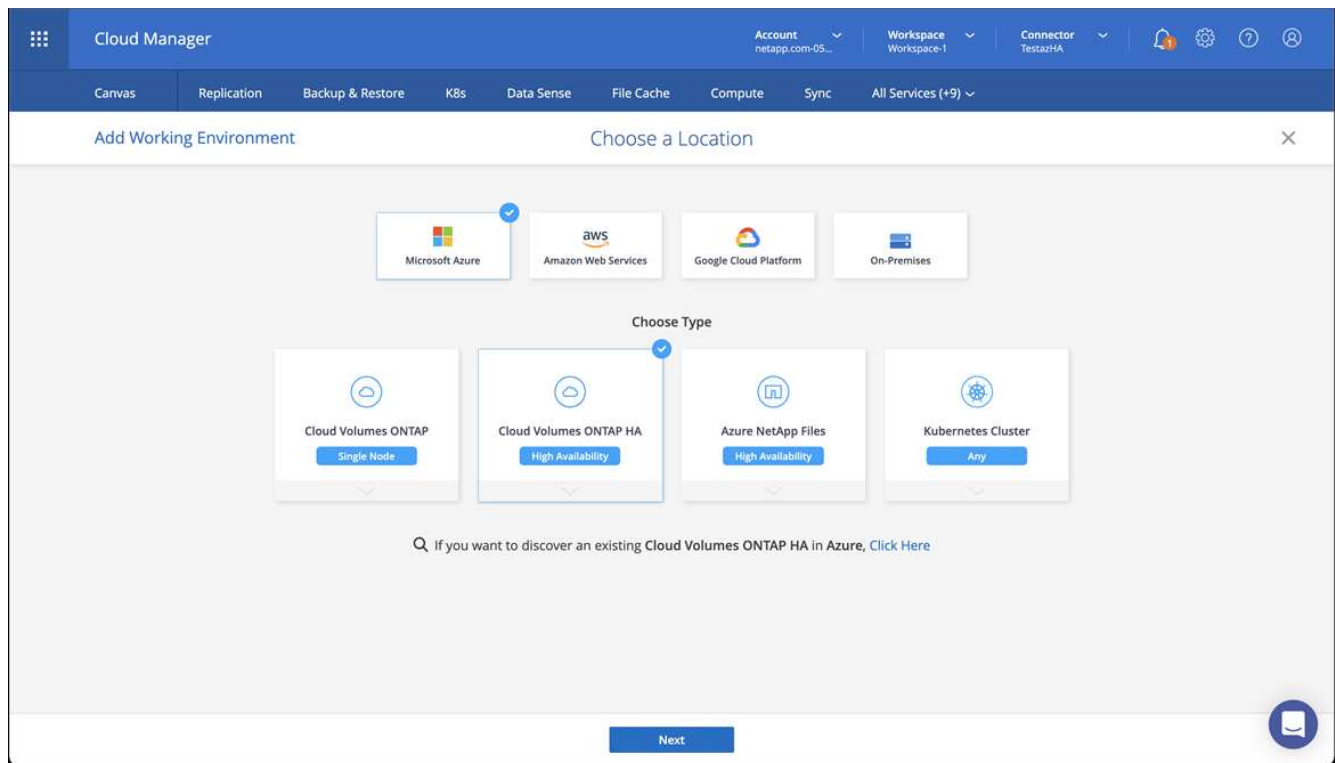
1. Aggiungi le tue credenziali a Cloud Manager.
2. Aggiungere un connettore per Azure. Vedere ["Policy di Cloud Manager"](#).
 - a. Scegliere **Azure** come provider.
 - b. Immettere le credenziali Azure, inclusi ID applicazione, segreto client e ID directory (tenant).

Vedere ["Creazione di un connettore in Azure da Cloud Manager"](#).

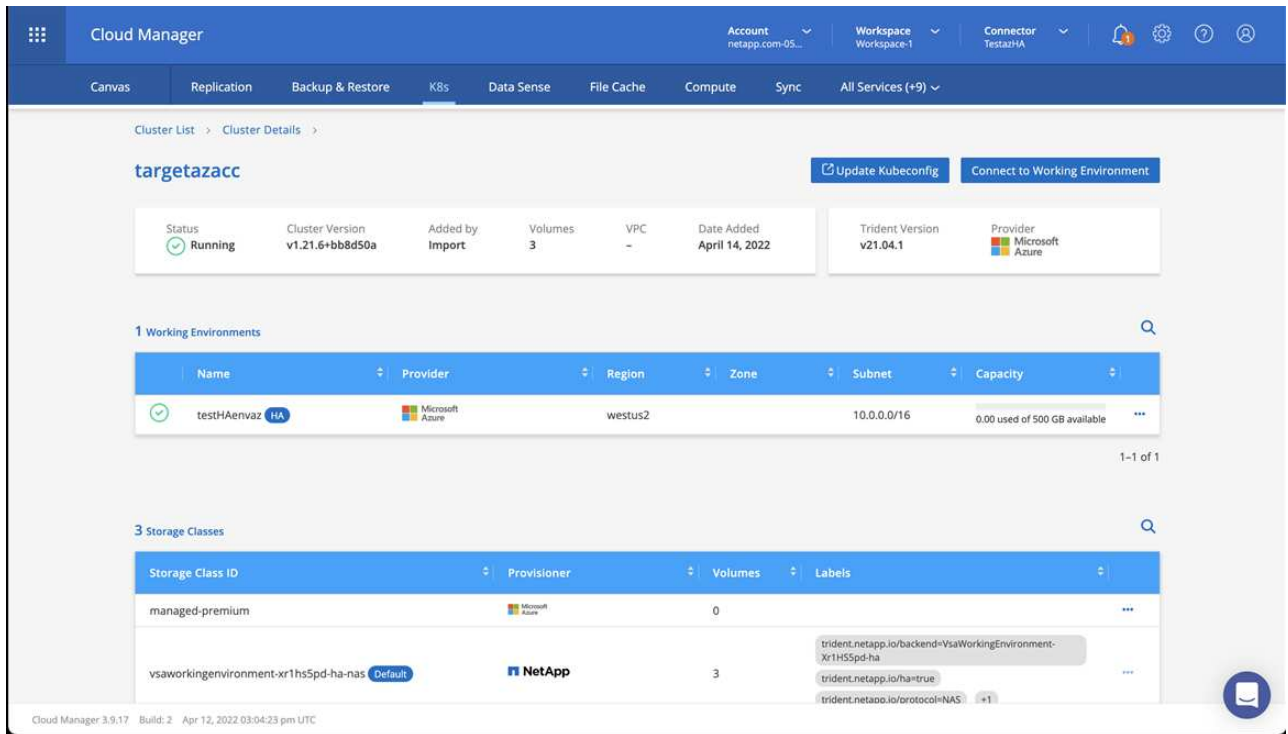
3. Assicurarsi che il connettore sia in funzione e passare a tale connettore.



4. Crea un ambiente di lavoro per il tuo ambiente cloud.
 - a. Percorso: "Microsoft Azure".
 - b. Tipo: "Cloud Volumes ONTAP ha".



5. Importare il cluster OpenShift. Il cluster si conetterà all'ambiente di lavoro appena creato.
 - a. Per visualizzare i dettagli del cluster NetApp, selezionare **K8s > elenco cluster > Dettagli cluster**.



b. Nell'angolo in alto a destra, prendere nota della versione di Trident.

c. Si noti che le classi di storage cluster Cloud Volumes ONTAP mostrano NetApp come provider.

In questo modo viene importato il cluster Red Hat OpenShift e viene assegnata una classe di storage predefinita. Selezionare la classe di storage. Trident viene installato automaticamente come parte del processo di importazione e rilevamento.

6. Tenere presenti tutti i volumi e i volumi persistenti in questa implementazione di Cloud Volumes ONTAP.

7. Cloud Volumes ONTAP può funzionare come nodo singolo o in alta disponibilità. Se ha è attivato, annotare lo stato ha e lo stato di implementazione del nodo in esecuzione in Azure.

Installare e configurare Astra Control Center

Installare Astra Control Center con lo standard ["istruzioni per l'installazione"](#).

Utilizzando Astra Control Center, aggiungere un bucket Azure. Vedere ["Configurare Astra Control Center e aggiungere i bucket"](#).

Informazioni sul copyright

Copyright © 2023 NetApp, Inc. Tutti i diritti riservati. Stampato negli Stati Uniti d'America. Nessuna porzione di questo documento soggetta a copyright può essere riprodotta in qualsiasi formato o mezzo (grafico, elettronico o meccanico, inclusi fotocopie, registrazione, nastri o storage in un sistema elettronico) senza previo consenso scritto da parte del detentore del copyright.

Il software derivato dal materiale sottoposto a copyright di NetApp è soggetto alla seguente licenza e dichiarazione di non responsabilità:

IL PRESENTE SOFTWARE VIENE FORNITO DA NETAPP "COSÌ COM'È" E SENZA QUALSIVOGLIA TIPO DI GARANZIA IMPLICITA O ESPRESSA FRA CUI, A TITOLO ESEMPLIFICATIVO E NON ESAUSTIVO, GARANZIE IMPLICITE DI COMMERCIALIZZABILITÀ E IDONEITÀ PER UNO SCOPO SPECIFICO, CHE VENGONO DECLINATE DAL PRESENTE DOCUMENTO. NETAPP NON VERRÀ CONSIDERATA RESPONSABILE IN ALCUN CASO PER QUALSIVOGLIA DANNO DIRETTO, INDIRETTO, ACCIDENTALE, SPECIALE, ESEMPLARE E CONSEGUENZIALE (COMPRESI, A TITOLO ESEMPLIFICATIVO E NON ESAUSTIVO, PROCUREMENT O SOSTITUZIONE DI MERCI O SERVIZI, IMPOSSIBILITÀ DI UTILIZZO O PERDITA DI DATI O PROFITTI OPPURE INTERRUZIONE DELL'ATTIVITÀ AZIENDALE) CAUSATO IN QUALSIVOGLIA MODO O IN RELAZIONE A QUALUNQUE TEORIA DI RESPONSABILITÀ, SIA ESSA CONTRATTUALE, RIGOROSA O DOVUTA A INSOLVENZA (COMPRESA LA NEGLIGENZA O ALTRO) INSORTA IN QUALSIASI MODO ATTRAVERSO L'UTILIZZO DEL PRESENTE SOFTWARE ANCHE IN PRESENZA DI UN PREAVVISO CIRCA L'EVENTUALITÀ DI QUESTO TIPO DI DANNI.

NetApp si riserva il diritto di modificare in qualsiasi momento qualunque prodotto descritto nel presente documento senza fornire alcun preavviso. NetApp non si assume alcuna responsabilità circa l'utilizzo dei prodotti o materiali descritti nel presente documento, con l'eccezione di quanto concordato espressamente e per iscritto da NetApp. L'utilizzo o l'acquisto del presente prodotto non comporta il rilascio di una licenza nell'ambito di un qualche diritto di brevetto, marchio commerciale o altro diritto di proprietà intellettuale di NetApp.

Il prodotto descritto in questa guida può essere protetto da uno o più brevetti degli Stati Uniti, esteri o in attesa di approvazione.

LEGENDA PER I DIRITTI SOTTOPOSTI A LIMITAZIONE: l'utilizzo, la duplicazione o la divulgazione da parte degli enti governativi sono soggetti alle limitazioni indicate nel sottoparagrafo (b)(3) della clausola Rights in Technical Data and Computer Software del DFARS 252.227-7013 (FEB 2014) e FAR 52.227-19 (DIC 2007).

I dati contenuti nel presente documento riguardano un articolo commerciale (secondo la definizione data in FAR 2.101) e sono di proprietà di NetApp, Inc. Tutti i dati tecnici e il software NetApp forniti secondo i termini del presente Contratto sono articoli aventi natura commerciale, sviluppati con finanziamenti esclusivamente privati. Il governo statunitense ha una licenza irrevocabile limitata, non esclusiva, non trasferibile, non cedibile, mondiale, per l'utilizzo dei Dati esclusivamente in connessione con e a supporto di un contratto governativo statunitense in base al quale i Dati sono distribuiti. Con la sola esclusione di quanto indicato nel presente documento, i Dati non possono essere utilizzati, divulgati, riprodotti, modificati, visualizzati o mostrati senza la previa approvazione scritta di NetApp, Inc. I diritti di licenza del governo degli Stati Uniti per il Dipartimento della Difesa sono limitati ai diritti identificati nella clausola DFARS 252.227-7015(b) (FEB 2014).

Informazioni sul marchio commerciale

NETAPP, il logo NETAPP e i marchi elencati alla pagina <http://www.netapp.com/TM> sono marchi di NetApp, Inc. Gli altri nomi di aziende e prodotti potrebbero essere marchi dei rispettivi proprietari.