Panoramica dell'installazione

Astra Control Center

NetApp
April 25, 2024

Sommario

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"

A seconda dell'ambiente in uso, potrebbe essere necessaria una configurazione aggiuntiva dopo l'installazione di Astra Control Center:

- "Configurare Astra Control Center dopo l'installazione"

Installare Astra Control Center utilizzando il processo standard

Per installare Astra Control Center, scaricare le immagini di installazione ed eseguire i seguenti passaggi. È possibile utilizzare questa procedura per installare Astra Control Center in ambienti connessi a Internet o con connessione ad aria.

Per una dimostrazione del processo di installazione di Astra Control Center, vedere "questo video".

Prima di iniziare

- **Soddisfare i requisiti ambientali**: "Prima di iniziare l'installazione, preparare l'ambiente per l'implementazione di Astra Control Center".

  Implementare Astra Control Center in un terzo dominio di errore o in un sito secondario. Questa opzione è consigliata per la replica delle applicazioni e il disaster recovery perfetto.

- **Garantire servizi integri**: Controllare che tutti i servizi API siano in buono stato e disponibili:

```
kubectl get apiservices
```

- **Assicurarsi che un FQDN instradabile**: Il FQDN Astra che si intende utilizzare può essere instradato al cluster. Ciò significa che si dispone di una voce DNS nel server DNS interno o si sta utilizzando un percorso URL principale già registrato.
- **Configure cert manager**: Se nel cluster esiste già un cert manager, è necessario eseguirne alcuni "fasi preliminari" In modo che Astra Control Center non tenti di installare il proprio cert manager. Per impostazione predefinita, Astra Control Center installa il proprio cert manager durante l'installazione.
- * (Solo driver SAN ONTAP) Abilita multipath*: Se stai utilizzando un driver SAN ONTAP, assicurati che multipath sia abilitato su tutti i tuoi cluster Kubernetes.

È inoltre necessario considerare quanto segue:

- **Ottenere l'accesso al Registro di sistema dell'immagine di controllo Astra di NetApp**:

  È possibile ottenere le immagini di installazione e i miglioramenti delle funzionalità per Astra Control, come

Astra Control provisioner, dal registro delle immagini di NetApp.

a. Registrare l'ID dell'account Astra Control necessario per accedere al Registro di sistema.

   Puoi visualizzare l'ID dell'account nell'interfaccia utente Web di Astra Control Service. Selezionare l'icona a forma di figura in alto a destra nella pagina, selezionare **accesso API** e annotare l'ID account.

b. Nella stessa pagina, selezionare **generate API token**, copiare la stringa del token API negli Appunti e salvarla nell'editor.

c. Accedere al registro Astra Control:

```
docker login cr.astra.netapp.io -u <account-id> -p <api-token>
```

- **Installare una mesh di servizio per comunicazioni sicure**: Si consiglia vivamente di proteggere i canali di comunicazione del cluster host Astra Control utilizzando un "mesh di servizio supportata".

> L'integrazione di Astra Control Center con una mesh di servizio può essere eseguita solo durante Astra Control Center "installazione" e non indipendente da questo processo. Il passaggio da un ambiente con mesh a un ambiente senza mesh non è supportato.

Per l'uso della mesh del servizio Istio, è necessario effettuare le seguenti operazioni:

  - Aggiungere un `istio-injection:enabled` etichetta Al namespace Astra prima di implementare Astra Control Center.
  - Utilizzare `Generic` impostazione ingresso e fornire un ingresso alternativo per bilanciamento del carico esterno.
  - Per i cluster Red Hat OpenShift, è necessario definire `NetworkAttachmentDefinition` Su tutti i namespace Astra Control Center associati (`netapp-acc-operator`, `netapp-acc`, `netapp-monitoring` per i cluster di applicazioni o qualsiasi namespace personalizzato che sia stato sostituito).

```
cat <<EOF | oc -n netapp-acc-operator create -f -
apiVersion: "k8s.cni.cncf.io/v1"
kind: NetworkAttachmentDefinition
metadata:
  name: istio-cni
EOF

cat <<EOF | oc -n netapp-acc create -f -
apiVersion: "k8s.cni.cncf.io/v1"
kind: NetworkAttachmentDefinition
metadata:
  name: istio-cni
EOF

cat <<EOF | oc -n netapp-monitoring create -f -
apiVersion: "k8s.cni.cncf.io/v1"
kind: NetworkAttachmentDefinition
metadata:
  name: istio-cni
EOF
```

Fasi

Per installare Astra Control Center, procedere come segue:

- Scarica ed estrai Astra Control Center
- Completare ulteriori passaggi se si utilizza un 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

> Non eliminare l'operatore di Astra Control Center (ad esempio, `kubectl delete -f astra_control_center_operator_deploy.yaml`) In qualsiasi momento durante l'installazione o il funzionamento di Astra Control Center per evitare di eliminare i pod.

Scarica ed estrai Astra Control Center

Scarica le immagini di Astra Control Center da una delle seguenti posizioni:

- **Registro di sistema dell'immagine del servizio di controllo Astra**: Utilizzare questa opzione se non si utilizza un registro locale con le immagini del centro di controllo Astra o se si preferisce questo metodo per

il download del pacchetto dal sito di supporto NetApp.

- **Sito di supporto NetApp**: Utilizzare questa opzione se si utilizza un registro locale con le immagini del Centro di controllo Astra.

Registro delle immagini di Astra Control

1. Effettua l'accesso ad Astra Control Service.
2. Nella Dashboard, selezionare **distribuire un'istanza autogestita di Astra Control**.
3. Seguire le istruzioni per accedere al registro delle immagini di Astra Control, estrarre l'immagine di installazione di Astra Control Center ed estrarre l'immagine.

Sito di supporto NetApp

1. Scarica il bundle contenente Astra Control Center (`astra-control-center-[version].tar.gz`) da "Pagina di download di Astra Control Center".
2. (Consigliato ma opzionale) Scarica il bundle di certificati e firme per Astra Control Center (`astra-control-center-certs-[version].tar.gz`) per verificare la firma del bundle.

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

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

   Viene visualizzato l'output `Verified OK` una volta completata la verifica.

3. Estrarre le immagini dal bundle Astra Control Center:

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

Completare ulteriori passaggi se si utilizza un registro locale

Se si intende inviare il pacchetto Astra Control Center al registro locale, è necessario utilizzare il plugin della riga di comando di NetApp Astra kubectl.

Installare il plug-in NetApp Astra kubectl

Completare questi passaggi per installare il più recente plugin della riga di comando di NetApp Astra kubectl.

Prima di iniziare

NetApp fornisce binari per 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.

Se il plug-in è già stato installato da un'installazione precedente, "assicurarsi di disporre della versione più recente" prima di completare questa procedura.

Fasi

1. Elencare i binari disponibili per il plugin NetApp Astra kubectl:

> La libreria di plugin kubectl fa parte del bundle tar e viene estratta nella cartella `kubectl-astra`.

```
ls kubectl-astra/
```

2. Spostare il file necessario per il sistema operativo e l'architettura della CPU nel percorso corrente e rinominarlo `kubectl-astra`:

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

Aggiungere le immagini al registro

1. Se si prevede di inviare il pacchetto Astra Control Center al registro locale, completare la sequenza di passaggi appropriata per il motore del contenitore:

Docker

a. Passare alla directory root del tarball. Viene visualizzata la `acc.manifest.bundle.yaml` file e queste directory:

```
acc/
kubectl-astra/
acc.manifest.bundle.yaml
```

b. Trasferire le immagini del pacchetto nella directory delle immagini di Astra Control Center nel registro locale. Eseguire le seguenti sostituzioni prima di eseguire `push-images` comando:

- Sostituire <BUNDLE_FILE> con il nome del file bundle di controllo Astra (`acc.manifest.bundle.yaml`).
- Sostituire <MY_FULL_REGISTRY_PATH> con l’URL del repository Docker; ad esempio, "<a href="https://<docker-registry>"" class="bare">https://<docker-registry>".
- Sostituire <MY_REGISTRY_USER> con il nome utente.
- Sostituire <MY_REGISTRY_TOKEN> con un token autorizzato per il registro.

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

Podman

a. Passare alla directory root del tarball. Vengono visualizzati il file e la directory seguenti:

```
acc/
kubectl-astra/
acc.manifest.bundle.yaml
```

b. Accedere al Registro di sistema:

```
podman login <YOUR_REGISTRY>
```

c. Preparare ed eseguire uno dei seguenti script personalizzato per la versione di Podman utilizzata. Sostituire <MY_FULL_REGISTRY_PATH> con l'URL del repository che include le sottodirectory.

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

```
export REGISTRY=<MY_FULL_REGISTRY_PATH>
export PACKAGENAME=acc
export PACKAGEVERSION=24.02.0-69
export DIRECTORYNAME=acc
for astraImageFile in $(ls ${DIRECTORYNAME}/images/*.tar) ; do
astraImage=$(podman load --input ${astraImageFile} | sed
's/Loaded image: //')
astraImageNoPath=$(echo ${astraImage} | sed 's:.*/::')
podman tag ${astraImageNoPath} ${REGISTRY}/netapp/astra/
${PACKAGENAME}/${PACKAGEVERSION}/${astraImageNoPath}
podman push ${REGISTRY}/netapp/astra/${PACKAGENAME}/
${PACKAGEVERSION}/${astraImageNoPath}
done
```

```
<strong>Podman 3</strong>
```

```
export REGISTRY=<MY_FULL_REGISTRY_PATH>
export PACKAGENAME=acc
export PACKAGEVERSION=24.02.0-69
export DIRECTORYNAME=acc
for astraImageFile in $(ls ${DIRECTORYNAME}/images/*.tar) ; do
astraImage=$(podman load --input ${astraImageFile} | sed
's/Loaded image: //')
astraImageNoPath=$(echo ${astraImage} | sed 's:.*/::')
podman tag ${astraImageNoPath} ${REGISTRY}/netapp/astra/
${PACKAGENAME}/${PACKAGEVERSION}/${astraImageNoPath}
podman push ${REGISTRY}/netapp/astra/${PACKAGENAME}/
${PACKAGEVERSION}/${astraImageNoPath}
done
```

Il percorso dell'immagine creato dallo script deve essere simile al seguente, a seconda della configurazione del Registro di sistema:

```
https://downloads.example.io/docker-astra-control-
prod/netapp/astra/acc/24.02.0-69/image:version
```

2. Modificare la directory:

```
cd manifests
```

Impostare namespace e secret per i registri con requisiti di autenticazione

1. Esportare il file kubeconfig per il cluster host Astra Control Center:

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

> Prima di completare l'installazione, assicurarsi che kubeconfig punti al cluster in cui si desidera installare Astra Control Center.

2. 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
```

   b. Creare un segreto per `netapp-acc-operator` namespace. Aggiungere informazioni su Docker ed eseguire il seguente comando:

> Il segnaposto `your_registry_path` deve corrispondere alla posizione delle immagini caricate in precedenza (ad esempio, `[Registry_URL]/netapp/astra/astracc/24.02.0-69`).

```
kubectl create secret docker-registry astra-registry-cred -n netapp-acc-
operator --docker-server=cr.astra.netapp.io --docker
-username=[astra_account_id] --docker-password=[astra_api_token]
```

+

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

+

> Se si elimina lo spazio dei nomi dopo la generazione del segreto, ricreare lo spazio dei nomi e rigenerare il segreto per lo spazio dei nomi.

   a. Creare il `netapp-acc` namespace (o personalizzato).

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

   b. Creare un segreto per `netapp-acc` namespace (o personalizzato). Aggiungere informazioni su Docker ed eseguire uno dei comandi appropriati in base alle preferenze del Registro di sistema:

```
kubectl create secret docker-registry astra-registry-cred -n [netapp-
acc or custom namespace] --docker-server=cr.astra.netapp.io --docker
-username=[astra_account_id] --docker-password=[astra_api_token]
```

```
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]
```

Installare l'operatore del centro di controllo Astra

1. (Solo registri locali) se si utilizza un registro locale, completare i seguenti passaggi:

   a. Aprire il programma YAML di distribuzione dell'operatore Astra Control Center:

   ```
   vim astra_control_center_operator_deploy.yaml
   ```

   Un YAML di esempio annotato segue questi passaggi.

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

   ```
   imagePullSecrets: [{name: astra-registry-cred}]
   ```

   c. Cambiare `ASTRA_IMAGE_REGISTRY` per `kube-rbac-proxy` al percorso del registro in cui sono state inviate le immagini in a. passaggio precedente.

   d. Cambiare `ASTRA_IMAGE_REGISTRY` per `acc-operator-controller-manager` al percorso del registro in cui sono state inviate le immagini in a. passaggio precedente.

```
apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    control-plane: controller-manager
  name: acc-operator-controller-manager
  namespace: netapp-acc-operator
spec:
  replicas: 1
  selector:
    matchLabels:
      control-plane: controller-manager
  strategy:
    type: Recreate
```

```
  template:
    metadata:
      labels:
        control-plane: controller-manager
    spec:
      containers:
      - args:
        - --secure-listen-address=0.0.0.0:8443
        - --upstream=http://127.0.0.1:8080/
        - --logtostderr=true
        - --v=10
        image: ASTRA_IMAGE_REGISTRY/kube-rbac-proxy:v4.8.0
        name: kube-rbac-proxy
        ports:
        - containerPort: 8443
          name: https
      - args:
        - --health-probe-bind-address=:8081
        - --metrics-bind-address=127.0.0.1:8080
        - --leader-elect
        env:
        - name: ACCOP_LOG_LEVEL
          value: "2"
        - name: ACCOP_HELM_INSTALLTIMEOUT
          value: 5m
        image: ASTRA_IMAGE_REGISTRY/acc-operator:24.02.68
        imagePullPolicy: IfNotPresent
        livenessProbe:
          httpGet:
            path: /healthz
            port: 8081
          initialDelaySeconds: 15
          periodSeconds: 20
        name: manager
        readinessProbe:
          httpGet:
            path: /readyz
            port: 8081
          initialDelaySeconds: 5
          periodSeconds: 10
        resources:
          limits:
            cpu: 300m
            memory: 750Mi
          requests:
            cpu: 100m
```

```
            memory: 75Mi
        securityContext:
          allowPrivilegeEscalation: false
      imagePullSecrets: []
      securityContext:
        runAsUser: 65532
      terminationGracePeriodSeconds: 10
```

2. Installare l'operatore del centro di controllo Astra:

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

Espandi per la risposta di esempio:

```
namespace/netapp-acc-operator created
customresourcedefinition.apiextensions.k8s.io/astracontrolcenters.as
tra.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
```

3. Verificare che i pod siano in esecuzione:

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

Configurare Astra Control Center

1. Modificare il file delle risorse personalizzate (CR) di Astra Control Center (`astra_control_center.yaml`) per creare account, supporto, registro e altre configurazioni necessarie:

```
vim astra_control_center.yaml
```

> Un YAML di esempio annotato segue questi passaggi.

2. Modificare o confermare le seguenti impostazioni:

Nome account

Impostazione	Guida	Tipo	Esempio
`accountName`	Modificare il `accountName` Stringa al nome che si desidera associare all'account Astra Control Center. Può essere presente un solo nome account.	stringa	`Example`

AstraVersion

Impostazione	Guida	Tipo	Esempio
`astraVersion`	La versione di Astra Control Center da implementare. Non è necessaria alcuna azione per questa impostazione, in quanto il valore verrà pre-compilato.	stringa	`24.02.0-69`

AstraAddress

Impostazione	Guida	Tipo	Esempio
`astraAddress`	Modificare il `astraAddress` Inserire l'FQDN (consigliato) o l'indirizzo IP che si desidera utilizzare nel browser per accedere ad Astra Control Center. Questo indirizzo definisce il modo in cui Astra Control Center verrà trovato nel data center e corrisponde allo stesso FQDN o indirizzo IP fornito dal bilanciamento del carico al termine dell'operazione "Requisiti di Astra Control Center". NOTA: Non utilizzare `http://` oppure `https://` nell'indirizzo. Copiare questo FQDN per utilizzarlo in un passo successivo.	stringa	`astra.example.com`

AutoSupport

Le selezioni effettuate in questa sezione determinano se parteciperai all'applicazione di supporto proattivo di NetApp, NetApp Active IQ, e dove vengono inviati i dati. È necessaria una connessione a Internet (porta 442) e tutti i dati di supporto sono resi anonimi.

Impostazione	Utilizzare	Guida	Tipo	Esempio
`autoSupport.enrolled`	Entrambi `enrolled` oppure `url` i campi devono essere selezionati	Cambiare `enrolled` Per AutoSupport a. `false` per i siti senza connettività internet o senza retain `true` per i siti connessi. Un'impostazione di `true` Consente l'invio di dati anonimi a NetApp a scopo di supporto. L'elezione predefinita è `false` E indica che non verranno inviati dati di supporto a NetApp.	Booleano	`false` (valore predefinito)
`autoSupport.url`	Entrambi `enrolled` oppure `url` i campi devono essere selezionati	Questo URL determina dove verranno inviati i dati anonimi.	stringa	`https://support.netapp.com/asupprod/post/1.0/postAsup`

e-mail

Impostazione	Guida	Tipo	Esempio
`email`	Modificare il `email` stringa all'indirizzo iniziale predefinito dell'amministratore. Copiare questo indirizzo e-mail per utilizzarlo in passo successivo. Questo indirizzo e-mail verrà utilizzato come nome utente per l'account iniziale per accedere all'interfaccia utente e verrà notificato degli eventi in Astra Control.	stringa	`admin@example.com`

Nome

Impostazione	Guida	Tipo	Esempio
`firstName`	Il nome dell'amministratore iniziale predefinito associato all'account Astra. Il nome utilizzato qui sarà visibile in un'intestazione dell'interfaccia utente dopo il primo accesso.	stringa	`SRE`

Cognome

Impostazione	Guida	Tipo	Esempio
`lastName`	Il cognome dell'amministratore iniziale predefinito associato all'account Astra. Il nome utilizzato qui sarà visibile in un'intestazione dell'interfaccia utente dopo il primo accesso.	stringa	`Admin`

ImageRegistry

Le selezioni effettuate in questa sezione definiscono il registro delle immagini container che ospita le immagini dell'applicazione Astra, Astra Control Center Operator e il repository Astra Control Center Helm.

Impostazione	Utilizzare	Guida	Tipo	Esempio
`imageRegistry.name`	Obbligatorio	Nome del registro delle immagini di Astra Control che ospita tutte le immagini richieste per distribuire Astra Control Center. Il valore viene precompilato e non è richiesta alcuna azione, a meno che non sia stato configurato un registro locale. Per un registro locale, sostituire questo valore esistente con il nome del registro delle immagini in cui sono state inserite le immagini in passaggio precedente. Non utilizzare `http://` oppure `https://` nel nome del registro di sistema.	stringa	`cr.astra.netapp.io` (impostazione predefinita) `example.registry.com/astra` (esempio di registro locale)

Impostazione	Utilizzare	Guida	Tipo	Esempio
`imageRegistry.secret`	Opzionale	Il nome del segreto Kubernetes utilizzato per l'autenticazione con il registro delle immagini. Il valore viene precompilato e non è richiesta alcuna azione, a meno che non sia stato configurato un registro locale e la stringa immessa per tale registro `imageRegistry.name` richiede un segreto. IMPORTANTE: Se si utilizza un registro locale che non richiede l'autorizzazione, è necessario eliminarlo `secret` linea entro `imageRegistry` in caso negativo, l'installazione non riesce.	stringa	`astra-registry-cred`

StorageClass

Impostazione	Guida	Tipo	Esempio
`storageClass`	Modificare il `storageClass` valore da `ontap-gold` A un'altra risorsa storageClass come richiesto dall'installazione. Eseguire il comando `kubectl get sc` per determinare le classi di storage configurate esistenti. Una delle classi di storage configurate per Astra Control provisioner deve essere inserita nel file manifest (`astra-control-center-<version>.manifest`) E verranno utilizzati per Astra PVS. Se non è impostata, viene utilizzata la classe di storage predefinita. NOTA: Se è configurata una classe di storage predefinita, assicurarsi che sia l'unica classe di storage con l'annotazione predefinita.	stringa	`ontap-gold`

VolumeReclaimPolicy

Impostazione	Guida	Tipo	Opzioni
`volumeReclaimPolicy`	In questo modo viene impostata la policy di recupero per il PVS di Astra. Impostare questo criterio su `Retain` Conserva i volumi persistenti dopo l'eliminazione di Astra. Impostare questo criterio su `Delete` elimina i volumi persistenti dopo l'eliminazione di astra. Se questo valore non viene impostato, il PVS viene mantenuto.	stringa	• `Retain` (Valore predefinito) • `Delete`

Tipo di ingessType

Impostazione	Guida	Tipo	Opzioni
`ingressType`	Utilizzare uno dei seguenti tipi di ingresso: **Generico** (`ingressType: "Generic"`) (Impostazione predefinita) Utilizzare questa opzione quando si utilizza un altro controller di ingresso o si preferisce utilizzare un controller di ingresso personalizzato. Dopo aver implementato Astra Control Center, è necessario configurare "controller di ingresso" Per esporre Astra Control Center con un URL. IMPORTANTE: Se si intende utilizzare una mesh di servizio con Astra Control Center, è necessario selezionare `Generic` come tipo di ingresso e configurare il proprio "controller di ingresso". **AccTraefik** (`ingressType: "AccTraefik"`) Utilizzare questa opzione quando si preferisce non configurare un controller di ingresso. 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	stringa	• `Generic` (valore predefinito) • `AccTraefik`

Dimensione scala

Impostazione	Guida	Tipo	Opzioni
`scaleSize`	Per impostazione predefinita, Astra utilizza High Availability (ha) `scaleSize` di `Medium`, Che implementa la maggior parte dei servizi in ha e implementa più repliche per la ridondanza. Con `scaleSize` come `Small`, Astra ridurrà il numero di repliche per tutti i servizi ad eccezione dei servizi essenziali per ridurre il consumo. SUGGERIMENTO: `Medium` le implementazioni sono costituite da circa 100 pod (non inclusi i carichi di lavoro transitori. 100 pod si basa su una configurazione a tre nodi master e tre nodi worker). Tenere a conoscenza dei limiti di rete per pod che potrebbero rappresentare un problema nell'ambiente, in particolare quando si prendono in considerazione scenari di disaster recovery.	stringa	• `Small` • `Medium` (Valore predefinito)

AstraResourcesScaler

Impostazione	Guida	Tipo	Opzioni
`astraResourcesScaler`	Opzioni di scalabilità per i limiti delle risorse di AstraControlCenter. Per impostazione predefinita, Astra Control Center implementa le richieste di risorse impostate per la maggior parte dei componenti all'interno di Astra. Questa configurazione consente allo stack software Astra Control Center di migliorare le prestazioni in ambienti con maggiore carico e scalabilità delle applicazioni. Tuttavia, negli scenari che utilizzano cluster di sviluppo o test più piccoli, il campo CR `astraResourcesScalar` può essere impostato su `Off`. In questo modo vengono disattivate le richieste di risorse e viene eseguita l'implementazione su cluster più piccoli.	stringa	• `Default` (Valore predefinito) • `Off`

AdditionalValues

Aggiungere i seguenti valori aggiuntivi ad Astra Control Center CR per evitare un problema noto durante l'installazione:

```
additionalValues:
    keycloak-operator:
      livenessProbe:
        initialDelaySeconds: 180
      readinessProbe:
        initialDelaySeconds: 180
```

crds

Le selezioni effettuate in questa sezione determinano il modo in cui Astra Control Center deve gestire i CRD.

Impostazione	Guida	Tipo	Esempio
`crds.externalCertManager`	Se si utilizza un gestore esterno dei certificati, cambiare `externalCertManager` a. `true`. L'impostazione predefinita `false` Fa in modo che Astra Control Center installi i propri CRD di gestione dei certificati durante l'installazione. I CRDS sono oggetti a livello di cluster e l'installazione potrebbe avere un impatto su altre parti del cluster. È possibile utilizzare questo indicatore per segnalare ad Astra Control Center che questi CRD verranno installati e gestiti dall'amministratore del cluster al di fuori di Astra Control Center.	Booleano	`False` (valore predefinito)
`crds.externalTraefik`	Per impostazione predefinita, Astra Control Center installerà i CRD Traefik richiesti. I CRDS sono oggetti a livello di cluster e l'installazione potrebbe avere un impatto su altre parti del cluster. È possibile utilizzare questo indicatore per segnalare ad Astra Control Center che questi CRD verranno installati e gestiti dall'amministratore del cluster al di fuori di Astra Control Center.	Booleano	`False` (valore predefinito)

Assicurarsi di aver selezionato la classe di storage e il tipo di ingresso corretti per la configurazione prima di completare l'installazione.

esempio astra_control_center.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: "[cr.astra.netapp.io or your_registry_path]"
    secret: "astra-registry-cred"
  storageClass: "ontap-gold"
  volumeReclaimPolicy: "Retain"
  ingressType: "Generic"
  scaleSize: "Medium"
  astraResourcesScaler: "Default"
  additionalValues:
    keycloak-operator:
      livenessProbe:
        initialDelaySeconds: 180
      readinessProbe:
        initialDelaySeconds: 180
  crds:
    externalTraefik: false
    externalCertManager: false
```

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]
   ```

2. Se si utilizza una mesh di servizio con Astra Control Center, aggiungere la seguente etichetta al `netapp-acc` o namespace personalizzato:

Il tipo di ingresso (`ingressType`) deve essere impostato su `Generic` In Astra Control Center CR prima di procedere con questo comando.

```
kubectl label ns [netapp-acc or custom namespace] istio-
injection:enabled
```

3. (Consigliato) "Attivare Strict MTLS" Per la mesh di servizio Istio:

```
kubectl apply -n istio-system -f - <<EOF
apiVersion: security.istio.io/v1beta1
kind: PeerAuthentication
metadata:
  name: default
spec:
  mtls:
    mode: STRICT
EOF
```

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

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

L'operatore di Astra Control Center esegue un controllo automatico dei requisiti ambientali. Mancante "requisiti" Può causare problemi di installazione o il funzionamento non corretto di Astra Control Center. Vedere sezione successiva per verificare la presenza di messaggi di avvertenza relativi al controllo automatico del sistema.

Verificare lo stato del sistema

È possibile verificare lo stato del sistema utilizzando i comandi kubectl. Se preferisci utilizzare OpenShift, puoi utilizzare comandi oc paragonabili per le fasi di verifica.

Fasi

1. Verificare che il processo di installazione non abbia prodotto messaggi di avviso relativi ai controlli di convalida:

```
kubectl get acc [astra or custom Astra Control Center CR name] -n
[netapp-acc or custom namespace] -o yaml
```

Ulteriori messaggi di avviso sono riportati anche nei registri dell'operatore di Astra Control Center.

2. Correggere eventuali problemi dell'ambiente segnalati dai controlli automatici dei requisiti.

> È possibile correggere i problemi assicurandosi che l'ambiente soddisfi i requisiti "requisiti" Per Astra Control Center.

3. 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.

Espandere per la risposta del campione

```
acc-helm-repo-5bd77c9ddd-8wxm2             1/1     Running     0
1h
activity-5bb474dc67-8l9ss                  1/1     Running     0
1h
activity-5bb474dc67-qbrtq                  1/1     Running     0
1h
api-token-authentication-6wbj2             1/1     Running     0
1h
api-token-authentication-9pgw6             1/1     Running     0
1h
api-token-authentication-tqf6d             1/1     Running     0
1h
asup-5495f44dbd-z4kft                      1/1     Running     0
1h
authentication-6fdd899858-5x45s            1/1     Running     0
1h
bucketservice-84d47487d-n9xgp              1/1     Running     0
1h
bucketservice-84d47487d-t5jhm              1/1     Running     0
1h
cert-manager-5dcb7648c4-hbldc              1/1     Running     0
1h
cert-manager-5dcb7648c4-nr9qf              1/1     Running     0
1h
cert-manager-cainjector-59b666fb75-bk2tf   1/1     Running     0
1h
cert-manager-cainjector-59b666fb75-pfnck   1/1     Running     0
1h
cert-manager-webhook-c6f9b6796-ngz2x       1/1     Running     0
1h
cert-manager-webhook-c6f9b6796-rwtbn       1/1     Running     0
1h
certificates-5f5b7b4dd-52tnj               1/1     Running     0
1h
certificates-5f5b7b4dd-gtjbx               1/1     Running     0
1h
certificates-expiry-check-28477260-dz5vw   0/1     Completed   0
1h
cloud-extension-6f58cc579c-lzfmv           1/1     Running     0
1h
cloud-extension-6f58cc579c-zw2km           1/1     Running     0
1h
cluster-orchestrator-79dd5c8d95-qjg92      1/1     Running     0
1h
```

```
composite-compute-85dc84579c-nz82f         1/1     Running     0
1h
composite-compute-85dc84579c-wx2z2         1/1     Running     0
1h
composite-volume-bff6f4f76-789nj           1/1     Running     0
1h
composite-volume-bff6f4f76-kwnd4           1/1     Running     0
1h
credentials-79fd64f788-m7m8f               1/1     Running     0
1h
credentials-79fd64f788-qnc6c               1/1     Running     0
1h
entitlement-f69cdbd77-4p2kn                1/1     Running     0
1h
entitlement-f69cdbd77-hswm6                1/1     Running     0
1h
features-7b9585444c-7xd7m                  1/1     Running     0
1h
features-7b9585444c-dcqwc                  1/1     Running     0
1h
fluent-bit-ds-crq8m                        1/1     Running     0
1h
fluent-bit-ds-gmgq8                        1/1     Running     0
1h
fluent-bit-ds-gzr4f                        1/1     Running     0
1h
fluent-bit-ds-j6sf6                        1/1     Running     0
1h
fluent-bit-ds-v4t9f                        1/1     Running     0
1h
fluent-bit-ds-x7j59                        1/1     Running     0
1h
graphql-server-6cc684fb46-2x8lr            1/1     Running     0
1h
graphql-server-6cc684fb46-bshbd            1/1     Running     0
1h
hybridauth-84599f79fd-fjc7k                1/1     Running     0
1h
hybridauth-84599f79fd-s9pmn                1/1     Running     0
1h
identity-95df98cb5-dvlmz                   1/1     Running     0
1h
identity-95df98cb5-krf59                   1/1     Running     0
1h
influxdb2-0                                1/1     Running     0
1h
```

```
keycloak-operator-6d4d688697-cfq8b         1/1     Running     0
1h
krakend-5d5c8f4668-7bq8g                   1/1     Running     0
1h
krakend-5d5c8f4668-t8hbn                   1/1     Running     0
1h
license-689cdd4595-2gsc8                   1/1     Running     0
1h
license-689cdd4595-g6vwk                   1/1     Running     0
1h
login-ui-57bb599956-4fwgz                  1/1     Running     0
1h
login-ui-57bb599956-rhztb                  1/1     Running     0
1h
loki-0                                     1/1     Running     0
1h
metrics-facade-846999bdd4-f7jdm            1/1     Running     0
1h
metrics-facade-846999bdd4-lnsxl            1/1     Running     0
1h
monitoring-operator-6c9d6c4b8c-ggkrl       2/2     Running     0
1h
nats-0                                     1/1     Running     0
1h
nats-1                                     1/1     Running     0
1h
nats-2                                     1/1     Running     0
1h
natssync-server-6df7d6cc68-9v2gd           1/1     Running     0
1h
nautilus-64b7fbdd98-bsgwb                  1/1     Running     0
1h
nautilus-64b7fbdd98-djlhw                  1/1     Running     0
1h
openapi-864584bccc-75nlv                   1/1     Running     0
1h
openapi-864584bccc-zh6bx                   1/1     Running     0
1h
polaris-consul-consul-server-0             1/1     Running     0
1h
polaris-consul-consul-server-1             1/1     Running     0
1h
polaris-consul-consul-server-2             1/1     Running     0
1h
polaris-keycloak-0                         1/1     Running     2 (1h
ago)      1h
```

```
polaris-keycloak-1                          1/1     Running     0
1h
polaris-keycloak-db-0                       1/1     Running     0
1h
polaris-keycloak-db-1                       1/1     Running     0
1h
polaris-keycloak-db-2                       1/1     Running     0
1h
polaris-mongodb-0                           1/1     Running     0
1h
polaris-mongodb-1                           1/1     Running     0
1h
polaris-mongodb-2                           1/1     Running     0
1h
polaris-ui-66476dcf87-f6s8j                 1/1     Running     0
1h
polaris-ui-66476dcf87-ztjk7                 1/1     Running     0
1h
polaris-vault-0                             1/1     Running     0
1h
polaris-vault-1                             1/1     Running     0
1h
polaris-vault-2                             1/1     Running     0
1h
public-metrics-bfc4fc964-x4m79              1/1     Running     0
1h
storage-backend-metrics-7dbb88d4bc-g78cj    1/1     Running     0
1h
storage-provider-5969b5df5-hjvcm            1/1     Running     0
1h
storage-provider-5969b5df5-r79ld            1/1     Running     0
1h
task-service-5fc9dc8d99-4q4f4               1/1     Running     0
1h
task-service-5fc9dc8d99-8l5zl               1/1     Running     0
1h
task-service-task-purge-28485735-fdzkd      1/1     Running     0
12m
telegraf-ds-2rgm4                           1/1     Running     0
1h
telegraf-ds-4qp6r                           1/1     Running     0
1h
telegraf-ds-77frs                           1/1     Running     0
1h
telegraf-ds-bc725                           1/1     Running     0
1h
```

```
telegraf-ds-cvmxf                          1/1     Running     0
1h
telegraf-ds-tqzgj                          1/1     Running     0
1h
telegraf-rs-5wtd8                          1/1     Running     0
1h
telemetry-service-6747866474-5djnc         1/1     Running     0
1h
telemetry-service-6747866474-thb7r         1/1     Running     1 (1h
ago)      1h
tenancy-5669854fb6-gzdzf                   1/1     Running     0
1h
tenancy-5669854fb6-xvsm2                   1/1     Running     0
1h
traefik-8f55f7d5d-4lgfw                    1/1     Running     0
1h
traefik-8f55f7d5d-j4wt6                    1/1     Running     0
1h
traefik-8f55f7d5d-p6gcq                    1/1     Running     0
1h
trident-svc-7cb5bb4685-54cnq               1/1     Running     0
1h
trident-svc-7cb5bb4685-b28xh               1/1     Running     0
1h
vault-controller-777b9bbf88-b5bqt          1/1     Running     0
1h
vault-controller-777b9bbf88-fdfd8          1/1     Running     0
1h
```

4. (Facoltativo) guardare `acc-operator` registri per monitorare l'avanzamento:

```
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 tramite "Aggiungere il flusso di lavoro del cluster nell'interfaccia utente" O API.

5. Una volta eseguiti tutti i pod, verificare che l'installazione sia stata eseguita correttamente (`READY` è `True`) E ottieni la password di configurazione iniziale che userai quando accedi ad Astra Control Center:

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

Risposta:

```
NAME    UUID                                   VERSION     ADDRESS
READY
astra   9aa5fdae-4214-4cb7-9976-5d8b4c0ce27f  24.02.0-69
10.111.111.111  True
```

> Copiare il valore UUID. La password è `ACC-` Seguito dal valore UUID (`ACC-[UUID]` oppure, in questo esempio, `ACC-9aa5fdae-4214-4cb7-9976-5d8b4c0ce27f`).

Impostare l'ingresso per il bilanciamento del carico

È possibile configurare un controller di ingresso Kubernetes che gestisce l'accesso esterno ai servizi. Queste procedure forniscono esempi di configurazione per un controller di ingresso se si utilizza il valore predefinito di `ingressType: "Generic"` Nella risorsa personalizzata di Astra Control Center (`astra_control_center.yaml`). Non è necessario utilizzare questa procedura, se specificato `ingressType: "AccTraefik"` Nella risorsa personalizzata di Astra Control Center (`astra_control_center.yaml`).

Dopo l'implementazione di Astra Control Center, è necessario configurare il controller di ingresso per esporre Astra Control Center con un URL.

Le fasi di installazione variano a seconda del tipo di controller di ingresso utilizzato. Astra Control Center supporta molti tipi di controller di ingresso. Queste procedure di configurazione forniscono alcuni esempi di passaggi per alcuni tipi di controller di ingresso comuni.

Prima di iniziare

- Il necessario "controller di ingresso" dovrebbe essere già implementato.
- Il "classe di ingresso" corrispondente al controller di ingresso dovrebbe già essere creato.

Passaggi per l'ingresso di Istio

1. Configurare l'ingresso Istio.

   > Questa procedura presuppone che Istio venga distribuito utilizzando il profilo di configurazione "predefinito".

2. Raccogliere o creare il certificato e il file della chiave privata desiderati per Ingress Gateway.

   È possibile utilizzare un certificato CA o autofirmato. Il nome comune deve essere l'indirizzo Astra (FQDN).

   Esempio di comando:

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

3. Crea un segreto `tls secret name` di tipo `kubernetes.io/tls` Per una chiave privata TLS e un

certificato in `istio-system namespace` Come descritto in TLS secrets (segreti TLS).

Esempio di comando:

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

> Il nome del segreto deve corrispondere a. `spec.tls.secretName` fornito in `istio-ingress.yaml` file.

4. Implementare una risorsa di ingresso in `netapp-acc` namespace (o personalizzato) che utilizza il tipo di risorsa v1 per uno schema (`istio-Ingress.yaml` in questo esempio):

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

5. Applicare le modifiche:

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

6. Controllare lo stato dell'ingresso:

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

Risposta:

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

7. Completare l'installazione di Astra Control Center.

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` namespace (o personalizzato) che utilizza il tipo di risorsa v1 per uno schema (`nginx-Ingress.yaml` in questo esempio):

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

3. Applicare le modifiche:

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

> NetApp consiglia di installare il controller nginx come implementazione piuttosto che come `daemonSet`.

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, modificherai la password per l'amministratore predefinito ed effettuerai l'accesso alla dashboard dell'interfaccia utente di Astra Control Center.

Fasi

1. In un browser, immettere l'FQDN (compreso il `https://` prefisso) utilizzato in `astraAddress` in `astra_control_center.yaml` CR quando Astra Control Center è stato installato.
2. Accettare i certificati autofirmati, se 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.yaml` CR quando Astra Control Center è stato installato, seguito dalla password di configurazione iniziale (`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 stati ancora creati altri account utente amministrativi, contattare "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.

Opzioni

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

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

- Per controllare l'output di Astra Control Center CR:

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

Procedure di installazione alternative

- **Installa con Red Hat OpenShift OperatorHub**: USA questo "procedura alternativa" Per installare Astra Control Center su OpenShift utilizzando OperatorHub.
- **Installare nel cloud pubblico con backend Cloud Volumes ONTAP**: Utilizzare "queste procedure" Per installare Astra Control Center in Amazon Web Services (AWS), Google Cloud Platform (GCP) o Microsoft Azure con un backend di storage Cloud Volumes ONTAP.

Cosa succederà

- (Opzionale) a seconda dell'ambiente, completare la post-installazione "fasi di configurazione".
- "Dopo aver installato Astra Control Center, effettuato l'accesso all'interfaccia utente e modificato la password, è necessario impostare una licenza, aggiungere cluster, abilitare l'autenticazione, gestire lo storage e aggiungere bucket".

Configurare un gestore esterno dei certificati

Se nel cluster Kubernetes esiste già un cert manager, è necessario eseguire alcuni passaggi preliminari in modo che Astra Control Center non installi il proprio cert manager.

Fasi

1. Verificare che sia installato un gestore dei certificati:

```
kubectl get pods -A | grep 'cert-manager'
```

Esempio di risposta:

```
cert-manager   essential-cert-manager-84446f49d5-sf2zd              1/1
Running    0      6d5h
cert-manager   essential-cert-manager-cainjector-66dc99cc56-9ldmt   1/1
Running    0      6d5h
cert-manager   essential-cert-manager-webhook-56b76db9cc-fjqrq      1/1
Running    0      6d5h
```

2. Creare una coppia certificato/chiave per `astraAddress` FQDN:

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

Esempio di risposta:

```
Generating a 2048 bit RSA private key
..................+++
.........................................+++
writing new private key to 'tls.key'
```

3. Creare un segreto con i file generati in precedenza:

```
kubectl create secret tls selfsigned-tls --key tls.key --cert tls.crt -n
<cert-manager-namespace>
```

Esempio di risposta:

```
secret/selfsigned-tls created
```

4. Creare un `ClusterIssuer` file che è **esattamente** il seguente, ma include la posizione dello spazio dei nomi in cui si trova il `cert-manager` i pod sono installati:

```
apiVersion: cert-manager.io/v1
kind: ClusterIssuer
metadata:
  name: astra-ca-clusterissuer
  namespace: <cert-manager-namespace>
spec:
  ca:
    secretName: selfsigned-tls
```

```
kubectl apply -f ClusterIssuer.yaml
```

Esempio di risposta:

```
clusterissuer.cert-manager.io/astra-ca-clusterissuer created
```

5. Verificare che il `ClusterIssuer` è venuto in su correttamente. `Ready` deve essere `True` prima di procedere:

```
kubectl get ClusterIssuer
```

Esempio di risposta:

```
NAME                      READY   AGE
astra-ca-clusterissuer    True    9s
```

6. Completare il "Processo di installazione di Astra Control Center". Esiste un "Fase di configurazione richiesta per il cluster Astra Control Center YAML" In cui si modifica il valore CRD per indicare che il gestore dei certificati è installato esternamente. È necessario completare questa fase durante l'installazione in modo che Astra Control Center riconosca il cert manager esterno.

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.

Prima di iniziare

- **Soddisfare i requisiti ambientali**: "Prima di iniziare l'installazione, preparare l'ambiente per l'implementazione di Astra Control Center".

> Implementare Astra Control Center in un terzo dominio di errore o in un sito secondario. Questa opzione è consigliata per la replica delle applicazioni e il disaster recovery perfetto.

- **Assicurare operatori di cluster e servizi API sani**:
  - Dal cluster OpenShift, assicurati che tutti gli operatori del cluster siano in buono stato:

```
oc get clusteroperators
```

- Dal cluster OpenShift, assicurati che tutti i servizi API siano in buono stato:

```
oc get apiservices
```

- **Assicurarsi che un FQDN instradabile**: Il FQDN Astra che si intende utilizzare può essere instradato al cluster. Ciò significa che si dispone di una voce DNS nel server DNS interno o si sta utilizzando un percorso URL principale già registrato.
- **Ottieni autorizzazioni OpenShift**: Avrai bisogno di tutte le autorizzazioni necessarie e dell'accesso a Red Hat OpenShift Container Platform per eseguire i passaggi di installazione descritti.
- **Configura un cert manager**: Se nel cluster esiste già un cert manager, è necessario eseguirne alcuni "fasi preliminari" In modo che Astra Control Center non installi il proprio cert manager. Per impostazione predefinita, Astra Control Center installa il proprio cert manager durante l'installazione.
- **Configura controller ingresso Kubernetes**: Se si dispone di un controller ingresso Kubernetes che gestisce l'accesso esterno a servizi, come il bilanciamento del carico in un cluster, è necessario configurarlo per l'utilizzo con Astra Control Center:
  a. Creare lo spazio dei nomi dell'operatore:

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

  b. "Completare la configurazione" per il proprio tipo di controller di ingresso.
- * (Solo driver SAN ONTAP) Abilita multipath*: Se stai utilizzando un driver SAN ONTAP, assicurati che multipath sia abilitato su tutti i tuoi cluster Kubernetes.

È inoltre necessario considerare quanto segue:

- **Ottenere l'accesso al Registro di sistema dell'immagine di controllo Astra di NetApp**:

  È possibile ottenere le immagini di installazione e i miglioramenti delle funzionalità per Astra Control, come Astra Control provisioner, dal registro delle immagini di NetApp.

  a. Registrare l'ID dell'account Astra Control necessario per accedere al Registro di sistema.

     Puoi visualizzare l'ID dell'account nell'interfaccia utente Web di Astra Control Service. Selezionare l'icona a forma di figura in alto a destra nella pagina, selezionare **accesso API** e annotare l'ID account.

  b. Nella stessa pagina, selezionare **generate API token**, copiare la stringa del token API negli Appunti e salvarla nell'editor.
  c. Accedere al registro Astra Control:

```
docker login cr.astra.netapp.io -u <account-id> -p <api-token>
```

- **Installare una mesh di servizio per comunicazioni sicure**: Si consiglia vivamente di proteggere i canali di comunicazione del cluster host Astra Control utilizzando un "mesh di servizio supportata".

> L'integrazione di Astra Control Center con una mesh di servizio può essere eseguita solo durante Astra Control Center "installazione" e non indipendente da questo processo. Il passaggio da un ambiente con mesh a un ambiente senza mesh non è supportato.

Per l'uso della mesh del servizio Istio, è necessario effettuare le seguenti operazioni:

- Aggiungere un `istio-injection:enabled` Etichetta nel namespace Astra prima di implementare Astra Control Center.
- Utilizzare `Generic` impostazione ingresso e fornire un ingresso alternativo per "bilanciamento del carico esterno".
- Per i cluster Red Hat OpenShift, è necessario definire `NetworkAttachmentDefinition` Su tutti i namespace Astra Control Center associati (`netapp-acc-operator`, `netapp-acc`, `netapp-monitoring` per i cluster di applicazioni o qualsiasi namespace personalizzato che sia stato sostituito).

```
cat <<EOF | oc -n netapp-acc-operator create -f -
apiVersion: "k8s.cni.cncf.io/v1"
kind: NetworkAttachmentDefinition
metadata:
  name: istio-cni
EOF

cat <<EOF | oc -n netapp-acc create -f -
apiVersion: "k8s.cni.cncf.io/v1"
kind: NetworkAttachmentDefinition
metadata:
  name: istio-cni
EOF

cat <<EOF | oc -n netapp-monitoring create -f -
apiVersion: "k8s.cni.cncf.io/v1"
kind: NetworkAttachmentDefinition
metadata:
  name: istio-cni
EOF
```

Fasi

- Scarica ed estrai Astra Control Center
- Completare ulteriori passaggi se si utilizza un registro locale
- Individuare la pagina di installazione dell'operatore
- Installare l'operatore
- Installare Astra Control Center

> Non eliminare l'operatore di Astra Control Center (ad esempio, `kubectl delete -f astra_control_center_operator_deploy.yaml`) In qualsiasi momento durante l'installazione o il funzionamento di Astra Control Center per evitare di eliminare i pod.

Scarica ed estrai Astra Control Center

Scarica le immagini di Astra Control Center da una delle seguenti posizioni:

- **Registro di sistema dell'immagine del servizio di controllo Astra**: Utilizzare questa opzione se non si utilizza un registro locale con le immagini del centro di controllo Astra o se si preferisce questo metodo per il download del pacchetto dal sito di supporto NetApp.
- **Sito di supporto NetApp**: Utilizzare questa opzione se si utilizza un registro locale con le immagini del Centro di controllo Astra.

Registro delle immagini di Astra Control

1. Effettua l'accesso ad Astra Control Service.
2. Nella Dashboard, selezionare **distribuire un'istanza autogestita di Astra Control**.
3. Seguire le istruzioni per accedere al registro delle immagini di Astra Control, estrarre l'immagine di installazione di Astra Control Center ed estrarre l'immagine.

Sito di supporto NetApp

1. Scarica il bundle contenente Astra Control Center (`astra-control-center-[version].tar.gz`) da "Pagina di download di Astra Control Center".
2. (Consigliato ma opzionale) Scarica il bundle di certificati e firme per Astra Control Center (`astra-control-center-certs-[version].tar.gz`) per verificare la firma del bundle.

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

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

   Viene visualizzato l'output `Verified OK` una volta completata la verifica.

3. Estrarre le immagini dal bundle Astra Control Center:

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

Completare ulteriori passaggi se si utilizza un registro locale

Se si intende inviare il pacchetto Astra Control Center al registro locale, è necessario utilizzare il plugin della riga di comando di NetApp Astra kubectl.

Installare il plug-in NetApp Astra kubectl

Completare questi passaggi per installare il più recente plugin della riga di comando di NetApp Astra kubectl.

Prima di iniziare

NetApp fornisce binari per 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.

Se il plug-in è già stato installato da un'installazione precedente, "assicurarsi di disporre della versione più recente" prima di completare questa procedura.

Fasi

1. Elencare i binari del plugin NetApp Astra kubectl disponibili e annotare il nome del file necessario per il sistema operativo e l'architettura della CPU:

   > La libreria di plugin kubectl fa parte del bundle tar e viene estratta nella cartella `kubectl-astra`.

   ```
   ls kubectl-astra/
   ```

2. Spostare il binario corretto nel percorso corrente e rinominarlo `kubectl-astra`:

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

Aggiungere le immagini al registro

1. Se si prevede di inviare il pacchetto Astra Control Center al registro locale, completare la sequenza di passaggi appropriata per il motore del contenitore:

Docker

a. Passare alla directory root del tarball. Viene visualizzata la `acc.manifest.bundle.yaml` file e queste directory:

```
acc/
kubectl-astra/
acc.manifest.bundle.yaml
```

b. Trasferire le immagini del pacchetto nella directory delle immagini di Astra Control Center nel registro locale. Eseguire le seguenti sostituzioni prima di eseguire `push-images` comando:

- Sostituire <BUNDLE_FILE> con il nome del file bundle di controllo Astra (`acc.manifest.bundle.yaml`).
- Sostituire <MY_FULL_REGISTRY_PATH> con l’URL del repository Docker; ad esempio, "<a href="https://<docker-registry>"" class="bare">https://<docker-registry>".
- Sostituire <MY_REGISTRY_USER> con il nome utente.
- Sostituire <MY_REGISTRY_TOKEN> con un token autorizzato per il registro.

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

Podman

a. Passare alla directory root del tarball. Vengono visualizzati il file e la directory seguenti:

```
acc/
kubectl-astra/
acc.manifest.bundle.yaml
```

b. Accedere al Registro di sistema:

```
podman login <YOUR_REGISTRY>
```

c. Preparare ed eseguire uno dei seguenti script personalizzato per la versione di Podman utilizzata. Sostituire <MY_FULL_REGISTRY_PATH> con l'URL del repository che include le sottodirectory.

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

```
export REGISTRY=<MY_FULL_REGISTRY_PATH>
export PACKAGENAME=acc
export PACKAGEVERSION=24.02.0-69
export DIRECTORYNAME=acc
for astraImageFile in $(ls ${DIRECTORYNAME}/images/*.tar) ; do
astraImage=$(podman load --input ${astraImageFile} | sed
's/Loaded image: //')
astraImageNoPath=$(echo ${astraImage} | sed 's:.*/::')
podman tag ${astraImageNoPath} ${REGISTRY}/netapp/astra/
${PACKAGENAME}/${PACKAGEVERSION}/${astraImageNoPath}
podman push ${REGISTRY}/netapp/astra/${PACKAGENAME}/
${PACKAGEVERSION}/${astraImageNoPath}
done
```

```
<strong>Podman 3</strong>
```

```
export REGISTRY=<MY_FULL_REGISTRY_PATH>
export PACKAGENAME=acc
export PACKAGEVERSION=24.02.0-69
export DIRECTORYNAME=acc
for astraImageFile in $(ls ${DIRECTORYNAME}/images/*.tar) ; do
astraImage=$(podman load --input ${astraImageFile} | sed
's/Loaded image: //')
astraImageNoPath=$(echo ${astraImage} | sed 's:.*/::')
podman tag ${astraImageNoPath} ${REGISTRY}/netapp/astra/
${PACKAGENAME}/${PACKAGEVERSION}/${astraImageNoPath}
podman push ${REGISTRY}/netapp/astra/${PACKAGENAME}/
${PACKAGEVERSION}/${astraImageNoPath}
done
```

Il percorso dell'immagine creato dallo script deve essere simile al seguente, a seconda della configurazione del Registro di sistema:

```
https://downloads.example.io/docker-astra-control-
prod/netapp/astra/acc/24.02.0-69/image:version
```

2. Modificare la directory:

```
cd manifests
```

Individuare la pagina di installazione dell’operatore

1. Completare una delle seguenti procedure per accedere alla pagina di installazione dell’operatore:

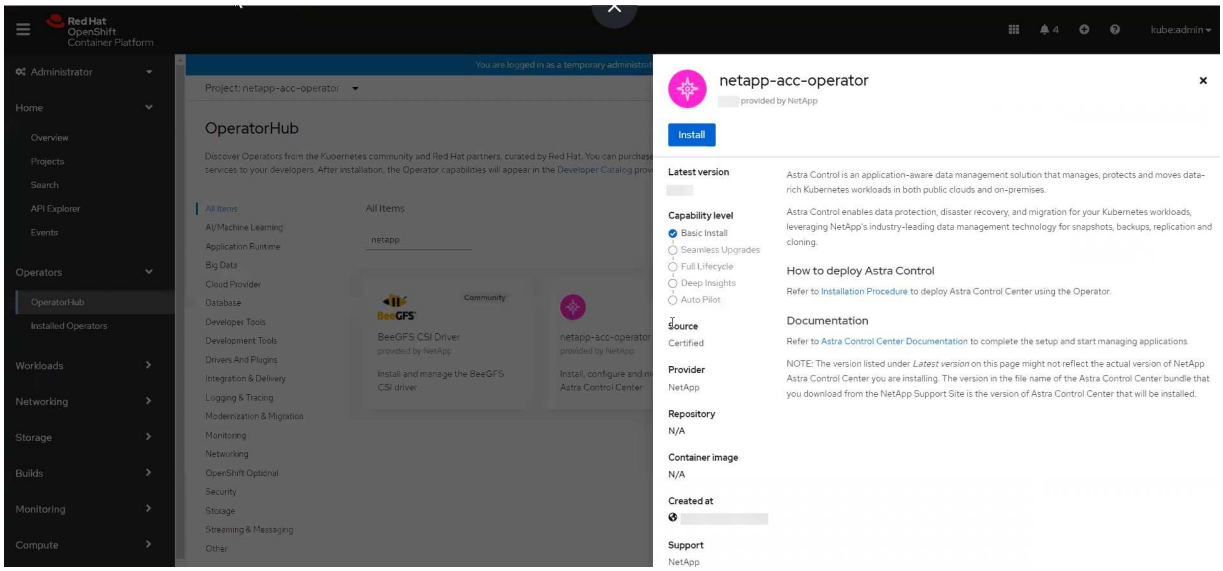
Console Web Red Hat OpenShift

a. Accedere all'interfaccia utente di OpenShift Container Platform.

b. Dal menu laterale, selezionare **Operator (operatori) > OperatorHub**.

> Con questo operatore è possibile eseguire l'aggiornamento solo alla versione corrente di Astra Control Center.

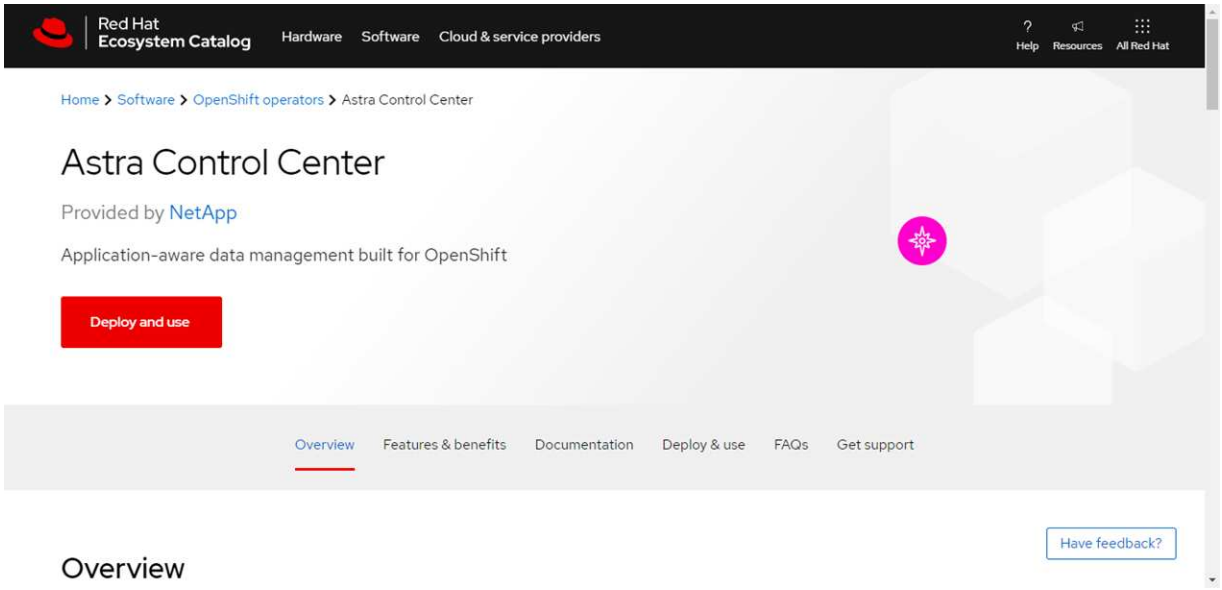
c. Cercare `netapp-acc` E selezionare l'operatore NetApp Astra Control Center.



Catalogo Red Hat Ecosystem

a. Selezionare NetApp Astra Control Center "operatore".

b. Selezionare **Deploy and Use** (distribuzione 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**.

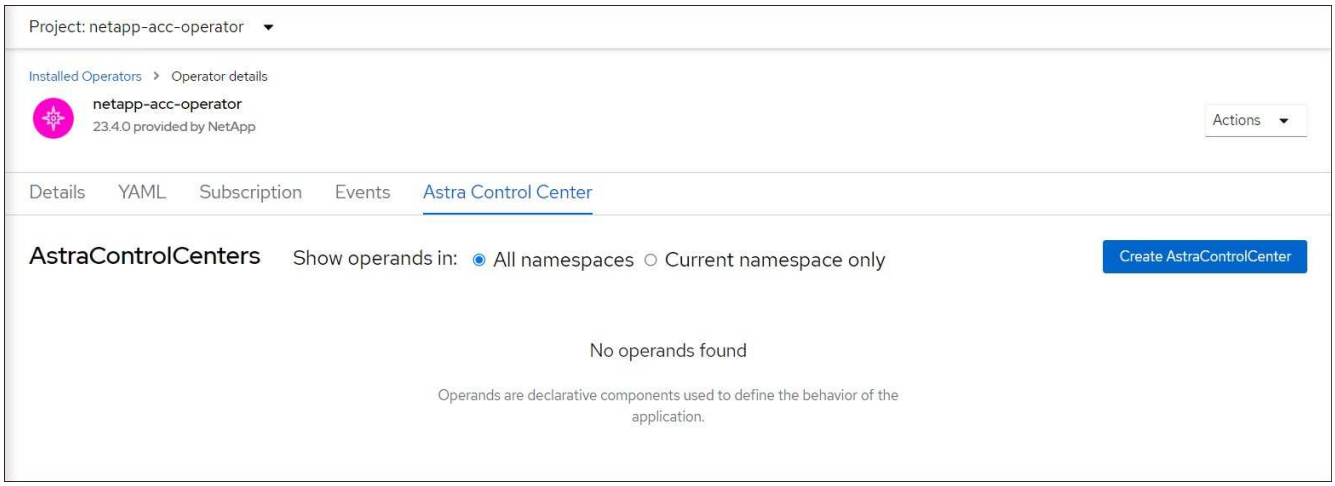
      Se è stata selezionata una strategia di approvazione manuale, verrà richiesto di approvare il piano di installazione manuale per questo operatore.

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 all'interno della scheda **Astra Control Center** dell'operatore Astra Control Center, selezionare **Create AstraControlCenter**

   .



2. Completare il `Create AstraControlCenter` campo del modulo:

   a. Mantenere o regolare il nome di Astra Control Center.

   b. Aggiungere etichette per Astra Control Center.

   c. Attiva o disattiva il supporto automatico. Si consiglia di mantenere la funzionalità di supporto automatico.

   d. Inserire il nome FQDN o l'indirizzo IP di Astra Control Center. Non entrare `http://` oppure `https://` nel campo dell'indirizzo.

   e. Immettere la versione di Astra Control Center, ad esempio 24.02.0-69.

f. Immettere un nome account, un indirizzo e-mail e un cognome amministratore.

g. Scegliere una policy di recupero dei volumi di `Retain`, `Recycle`, o. `Delete`. Il valore predefinito è `Retain`.

h. Selezionare la dimensione della scala dell'installazione.

> Per impostazione predefinita, Astra utilizza High Availability (ha) `scaleSize` di `Medium`, Che implementa la maggior parte dei servizi in ha e implementa più repliche per la ridondanza. Con `scaleSize` come `Small`, Astra ridurrà il numero di repliche per tutti i servizi ad eccezione dei servizi essenziali per ridurre il consumo.

i. selezionare il tipo di ingresso:

- **Generico** (`ingressType: "Generic"`) (Impostazione predefinita)

  Utilizzare questa opzione quando si utilizza un altro controller di ingresso o si preferisce utilizzare un controller di ingresso personalizzato. Dopo aver implementato Astra Control Center, è necessario configurare "controller di ingresso" Per esporre Astra Control Center con un URL.

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

  Utilizzare questa opzione quando si preferisce non configurare un controller di ingresso. 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, fare riferimento a. "Requisiti".

a. In **Registro immagini**, utilizzare il valore predefinito a meno che non sia stato configurato un registro locale. Per un registro locale, sostituire questo valore con il percorso del Registro di sistema dell'immagine locale in cui sono state inserite le immagini in un passaggio precedente. Non entrare `http://` oppure `https://` nel campo dell'indirizzo.

b. Se si utilizza un registro di immagini che richiede l'autenticazione, inserire il segreto dell'immagine.

> Se si utilizza un registro che richiede l'autenticazione, creare un segreto sul cluster.

c. Inserire il nome admin.

d. Configurare la scalabilità delle risorse.

e. Fornire la classe di storage predefinita.

> Se è configurata una classe di storage predefinita, assicurarsi che sia l'unica classe di storage con l'annotazione predefinita.

f. Definire le preferenze di gestione CRD.

3. Selezionare la vista YAML per rivedere le impostazioni selezionate.
4. Selezionare `Create`.

Creare un segreto di registro

Se si utilizza un registro che richiede l'autenticazione, creare un segreto nel cluster OpenShift e immettere il nome segreto nel `Create AstraControlCenter` campo del modulo.

1. Creare uno spazio dei nomi per l'operatore Astra Control Center:

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

2. Creare un segreto in questo namespace:

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

> Astra Control supporta solo i segreti del Registro di sistema di Docker.

3. Completare i campi rimanenti in Il campo Create AstraControlCenter Form (Crea modulo AstraControlCenter).

Cosa succederà

Completare il "fasi rimanenti" Per verificare che Astra Control Center sia stato installato correttamente, configurare un controller di ingresso (opzionale) e accedere all'interfaccia utente. Inoltre, sarà necessario eseguire "attività di installazione" al termine dell'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), Google Cloud Platform (GCP) 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 nella piattaforma Google Cloud
- Implementare Astra Control Center in Microsoft Azure

Puoi gestire le tue applicazioni nelle distribuzioni con cluster Kubernetes autogestiti, come OpenShift Container Platform (OCP). Solo i cluster OCP autogestiti sono validati per l'implementazione di Astra Control Center.

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).

Ciò di cui hai bisogno per AWS

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

- Licenza Astra Control Center. Fare riferimento a. "Requisiti di licenza di Astra Control Center".
- "Soddisfare i requisiti di Astra Control Center".
- Account NetApp Cloud Central
- Se si utilizza OCP, autorizzazioni Red Hat OpenShift Container Platform (OCP) (a livello di spazio dei nomi per creare i 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 richiesta la zona ospitata di AWS e la voce Amazon Route 53

Requisiti dell'ambiente operativo per AWS

Astra Control Center richiede il seguente ambiente operativo per AWS:

- Red Hat OpenShift Container Platform dalla versione 4.11 alla 4.13

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

Astra Control Center richiede risorse specifiche oltre ai requisiti delle risorse dell'ambiente. Fare riferimento a. "Requisiti dell'ambiente operativo di Astra Control Center".

> Il token di registro AWS scade tra 12 ore, dopodiché sarà necessario rinnovare la password di registro dell'immagine di 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 BlueXP per AWS.
5. Installare Astra Control Center per AWS.

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 BlueXP (in precedenza 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 e creare un bucket AWS S3. Se non si riesce ad accedere al registro delle immagini del Centro di controllo Astra di NetApp, è necessario anche creare un ECR (Elastic Container Registry) per ospitare le immagini del Centro di controllo Astra e inviare le immagini al Registro di sistema.

Seguire la documentazione di AWS per completare i seguenti passaggi. Vedere "Documentazione di installazione di AWS".

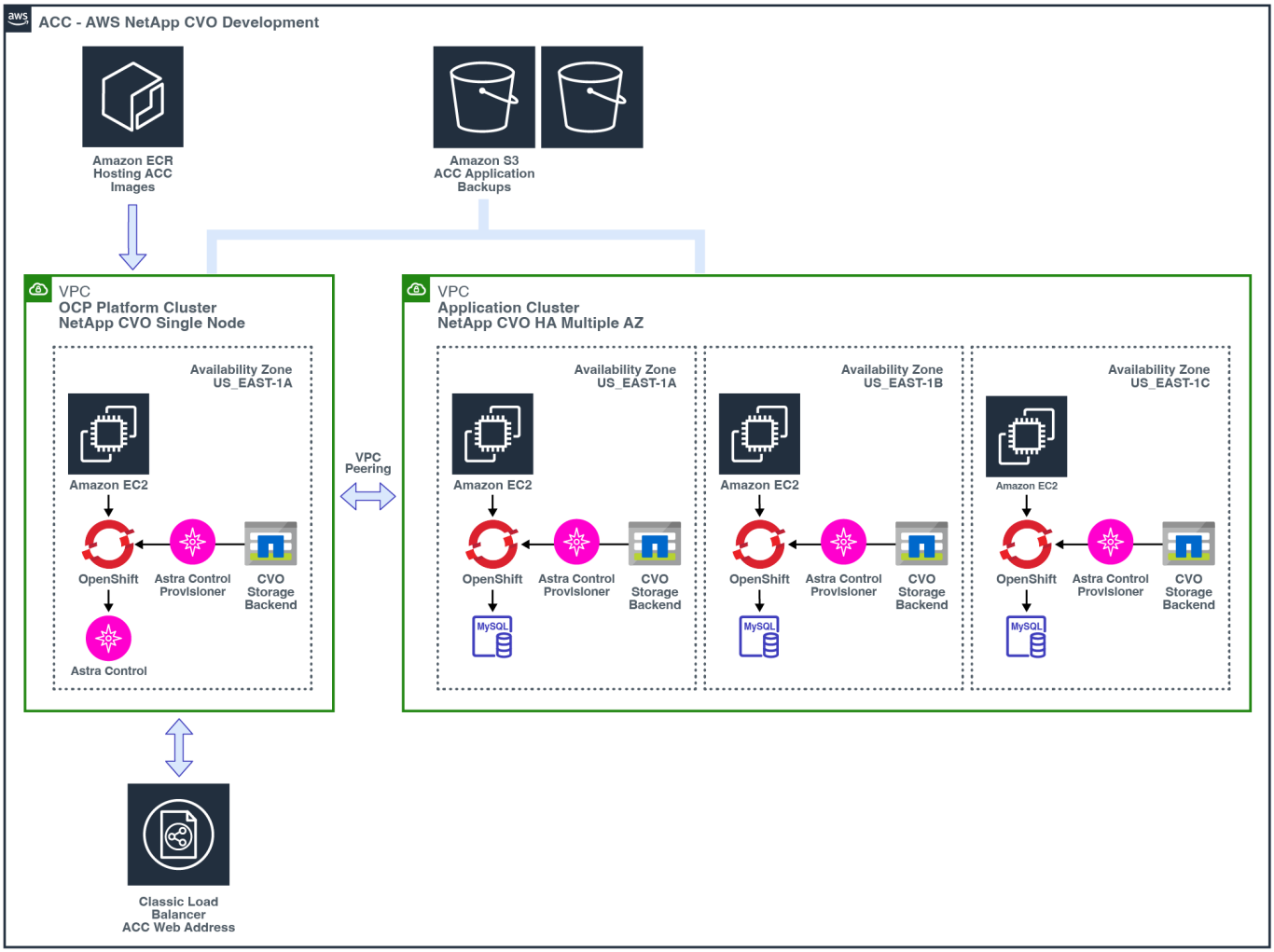
1. Creare una rete virtuale AWS.
2. Esaminare le istanze di calcolo EC2. Può trattarsi di un server bare metal o di macchine virtuali in AWS.
3. Se il tipo di istanza non corrisponde già ai requisiti minimi di risorsa Astra per i nodi master e worker, modificare il tipo di istanza in AWS per soddisfare i requisiti Astra. Fare riferimento a. "Requisiti di Astra Control Center".
4. Creare almeno un bucket AWS S3 per memorizzare i backup.
5. (Facoltativo) se non è possibile accedere al registro delle immagini di NetApp, procedere come segue:
   a. Creare un AWS Elastic Container Registry (ECR) per ospitare tutte le immagini di Astra Control Center.

      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.

   b. Trasferire le immagini di Astra Control Center 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 BlueXP per AWS

Utilizzando NetApp BlueXP (in precedenza Cloud Manager), creare uno spazio di lavoro, aggiungere un connettore ad AWS, creare un ambiente di lavoro e importare il cluster.

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

- "Introduzione a Cloud Volumes ONTAP in AWS".
- "Creare un connettore in AWS utilizzando BlueXP"

Fasi

1. Aggiungi le tue credenziali a BlueXP.
2. Creare un'area di lavoro.
3. Aggiungere un connettore per AWS. Scegliere AWS come provider.
4. Crea 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 connetterà 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, osserva la versione di Astra Control Provisioner.
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.
   Astra Control provisioner viene installato automaticamente nell'ambito 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 per AWS

Seguire lo standard "Istruzioni di installazione di Astra Control Center".

> AWS utilizza il tipo di bucket S3 generico.

Implementare Astra Control Center nella piattaforma Google Cloud

Puoi implementare Astra Control Center su un cluster Kubernetes autogestito ospitato su un cloud pubblico Google Cloud Platform (GCP).

Cosa ti serve per GCP

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

- Licenza Astra Control Center. Fare riferimento a. "Requisiti di licenza di Astra Control Center".
- "Soddisfare i requisiti di Astra Control Center".
- Account NetApp Cloud Central
- Se si utilizza OCP, Red Hat OpenShift Container Platform (OCP) da 4.11 a 4.13
- Se si utilizza OCP, autorizzazioni Red Hat OpenShift Container Platform (OCP) (a livello di spazio dei nomi per creare i pod)
- GCP Service account con autorizzazioni che consentono di creare bucket e connettori

Requisiti dell'ambiente operativo per GCP

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

Astra Control Center richiede risorse specifiche oltre ai requisiti delle risorse dell'ambiente. Fare riferimento a. "Requisiti dell'ambiente operativo di Astra Control Center".

Panoramica dell'implementazione per GCP

Di seguito viene fornita una panoramica del processo di installazione di Astra Control Center su un cluster OCP autogestiti in GCP con Cloud Volumes ONTAP come backend di storage.

Ciascuna di queste fasi viene illustrata più dettagliatamente di seguito.

1. Installare un cluster RedHat OpenShift su GCP.
2. Crea un progetto GCP e un cloud privato virtuale.
3. Assicurarsi di disporre di autorizzazioni IAM sufficienti.
4. Configurare GCP.
5. Configurare NetApp BlueXP per GCP.
6. Installare Astra Control Center per GCP.

Installare un cluster RedHat OpenShift su GCP

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

Per istruzioni sull'installazione, consultare quanto segue:

- "Installazione di un cluster OpenShift in GCP"
- "Creazione di un account di servizio GCP"

Crea un progetto GCP e un cloud privato virtuale

Creare almeno un progetto GCP e Virtual Private Cloud (VPC).

> OpenShift potrebbe creare i propri gruppi di risorse. Inoltre, è necessario definire un VPC GCP. 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 BlueXP (in precedenza Cloud Manager).

Vedere "Credenziali e permessi GCP iniziali".

Configurare GCP

Quindi, configurare GCP per creare un VPC, configurare istanze di calcolo e creare un Google Cloud Object Storage. Se non è possibile accedere al registro delle immagini di NetApp Astra Control Center, è necessario creare un registro dei contenitori di Google per ospitare le immagini di Astra Control Center e inviare le immagini a questo registro.

Seguire la documentazione GCP per completare i seguenti passaggi. Vedere Installazione del cluster OpenShift in GCP.

1. Creare un progetto GCP e un VPC nel GCP che si intende utilizzare per il cluster OCP con backend CVO.
2. Esaminare le istanze di calcolo. Questo può essere un server bare metal o VM in GCP.
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 GCP per soddisfare i requisiti Astra. Fare riferimento a. "Requisiti di Astra Control Center".

4. Crea almeno un bucket di storage cloud GCP per memorizzare i tuoi backup.
5. Creare un segreto, necessario per l'accesso al bucket.
6. (Facoltativo) se non è possibile accedere al registro delle immagini di NetApp, procedere come segue:

   a. Creare un Google Container Registry per ospitare le immagini di Astra Control Center.
   b. Impostare l'accesso al Google Container Registry per il push/pull di Docker per tutte le immagini di Astra Control Center.

      Esempio: Le immagini di Astra Control Center possono essere inviate a questo registro inserendo il seguente script:

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

      Questo script richiede un file manifesto di Astra Control Center e la posizione del Google Image Registry. Esempio:

```
manifestfile=acc.manifest.bundle.yaml
GCP_CR_REGISTRY=<target GCP image registry>
ASTRA_REGISTRY=<source Astra Control Center image registry>

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

7. Impostare le zone DNS.

Configurare NetApp BlueXP per GCP

Utilizzando NetApp BlueXP (in precedenza Cloud Manager), creare uno spazio di lavoro, aggiungere un connettore a GCP, creare un ambiente di lavoro e importare il cluster.

Seguire la documentazione di BlueXP per completare i seguenti passaggi. Vedere "Introduzione a Cloud Volumes ONTAP in GCP".

Prima di iniziare

- Accesso all'account di servizio GCP con i ruoli e le autorizzazioni IAM richiesti

Fasi

1. Aggiungi le tue credenziali a BlueXP. Vedere "Aggiunta di account GCP".
2. Aggiungere un connettore per GCP.

a. Scegliere "GCP" come provider.
b. Immettere le credenziali GCP. Vedere "Creazione di un connettore in GCP da BlueXP".
c. Assicurarsi che il connettore sia in funzione e passare a tale connettore.

3. Crea un ambiente di lavoro per il tuo ambiente cloud.
   a. Location: Italy
   b. Tipo: "Cloud Volumes ONTAP ha"
4. Importare il cluster OpenShift. Il cluster si connetterà 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, osserva la versione di Astra Control Provisioner.
   c. Si noti che le classi di storage del 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.
      Astra Control provisioner viene installato automaticamente nell'ambito del processo di importazione e rilevamento.
5. Tenere presenti tutti i volumi e i volumi persistenti in questa implementazione di Cloud Volumes ONTAP.

> Cloud Volumes ONTAP può operare come un singolo nodo o in alta disponibilità (ha). Se ha è attivato, annotare lo stato ha e lo stato di implementazione del nodo in esecuzione in GCP.

Installare Astra Control Center per GCP

Seguire lo standard "Istruzioni di installazione di Astra Control Center".

> GCP utilizza il tipo di bucket S3 generico.

1. Generare il Docker Secret per estrarre le immagini per l'installazione di Astra Control Center:

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

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. Fare riferimento a. "Requisiti di licenza di Astra Control Center".
- "Soddisfare i requisiti di Astra Control Center".
- Account NetApp Cloud Central

- Se si utilizza OCP, Red Hat OpenShift Container Platform (OCP) da 4.11 a 4.13
- Se si utilizza OCP, autorizzazioni Red Hat OpenShift Container Platform (OCP) (a livello di spazio dei nomi per creare i 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 delle risorse di base descritti nella documentazione ufficiale dell'ambiente.

Astra Control Center richiede risorse specifiche oltre ai requisiti delle risorse dell'ambiente. Fare riferimento a. "Requisiti dell'ambiente operativo di Astra Control Center".

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 BlueXP (in precedenza Cloud Manager) per Azure.
6. Installare e configurare Astra Control Center per Azure.

Installare un cluster RedHat OpenShift su Azure

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

Per istruzioni sull'installazione, consultare quanto segue:

- "Installazione del cluster OpenShift su Azure".
- "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 l'installazione di un cluster RedHat OpenShift e di un connettore NetApp BlueXP.

Vedere "Credenziali e permessi di Azure".

Configurare Azure

Quindi, configurare Azure per creare una rete virtuale, configurare istanze di calcolo e creare un container Azure Blob. Se non è possibile accedere al registro delle immagini del Centro di controllo Astra di NetApp, è necessario creare anche un ACR (Azure Container Registry) per ospitare le immagini del Centro di controllo Astra e inviare le immagini al Registro di sistema.

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. Fare riferimento a. "Requisiti di Astra Control Center".
4. Creare almeno un container Azure Blob per memorizzare i backup.
5. Creare un account storage. Ti servirà un account di storage per creare un container da utilizzare come bucket in Astra Control Center.
6. Creare un segreto, necessario per l'accesso al bucket.
7. (Facoltativo) se non è possibile accedere al registro delle immagini di NetApp, procedere come segue:

   a. Creare un Azure Container Registry (ACR) per ospitare le immagini di Astra Control Center.
   b. Impostare l'accesso ACR per la funzione push/pull di Docker per tutte le immagini di Astra Control Center.
   c. Inviare le immagini di Astra Control Center a questo registro utilizzando il seguente script:

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

   Esempio:

```
manifestfile=acc.manifest.bundle.yaml
AZ_ACR_REGISTRY=<target Azure ACR image registry>
ASTRA_REGISTRY=<source Astra Control Center image registry>

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 < acc.manifest.bundle.yaml
```

8. Impostare le zone DNS.

Configurare NetApp BlueXP (in precedenza Cloud Manager) per Azure

Utilizzando BlueXP (in precedenza 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 BlueXP per completare i seguenti passaggi. Vedere "Introduzione a BlueXP in Azure".

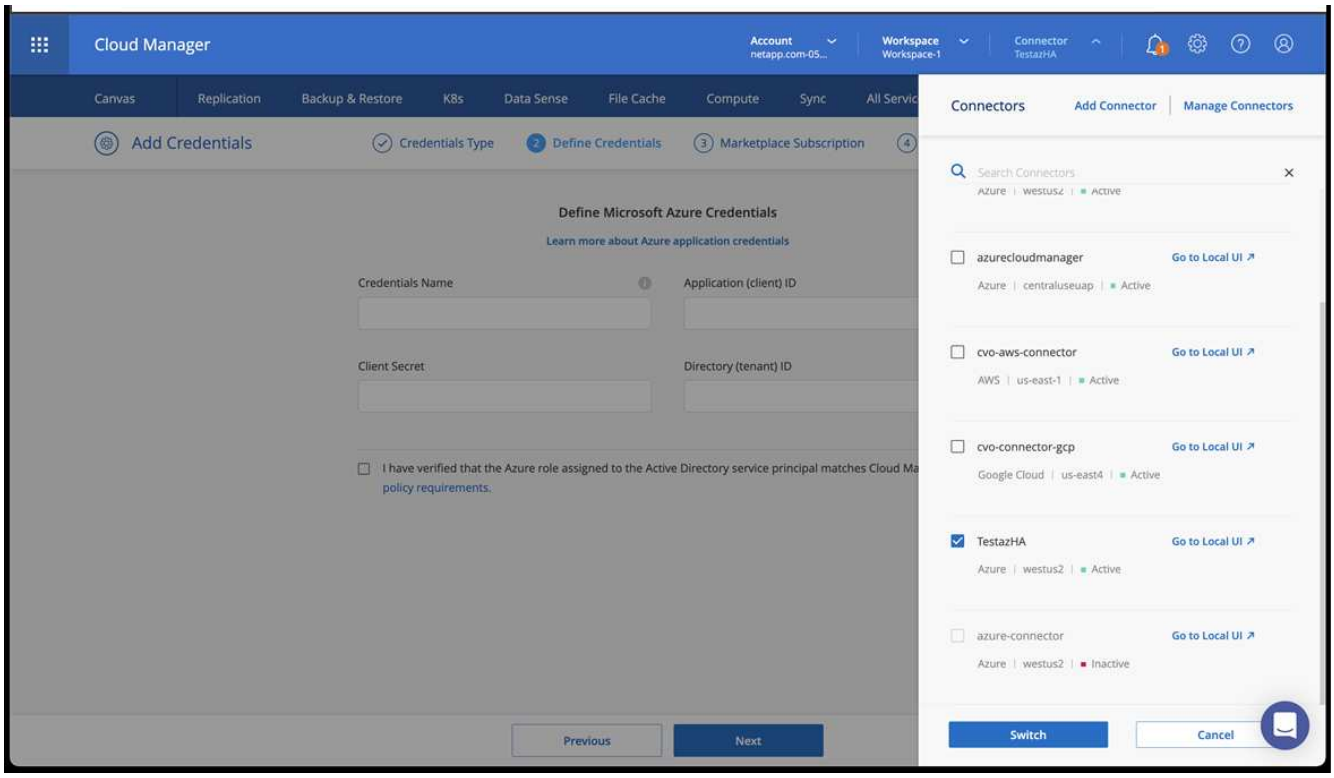
Prima di iniziare

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

Fasi

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

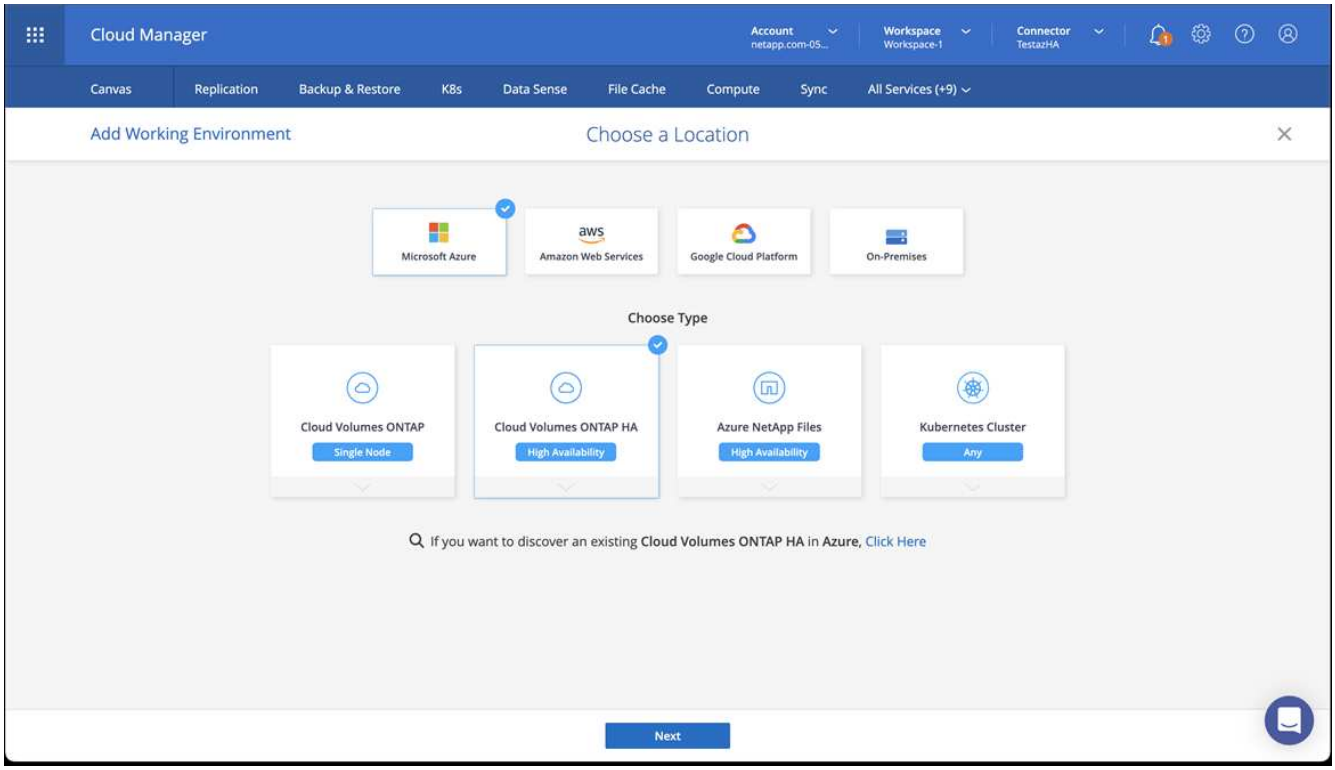
      Vedere "Creazione di un connettore in Azure da BlueXPr".
3. Assicurarsi che il connettore sia in funzione e passare a tale connettore.



4. Crea un ambiente di lavoro per il tuo ambiente cloud.

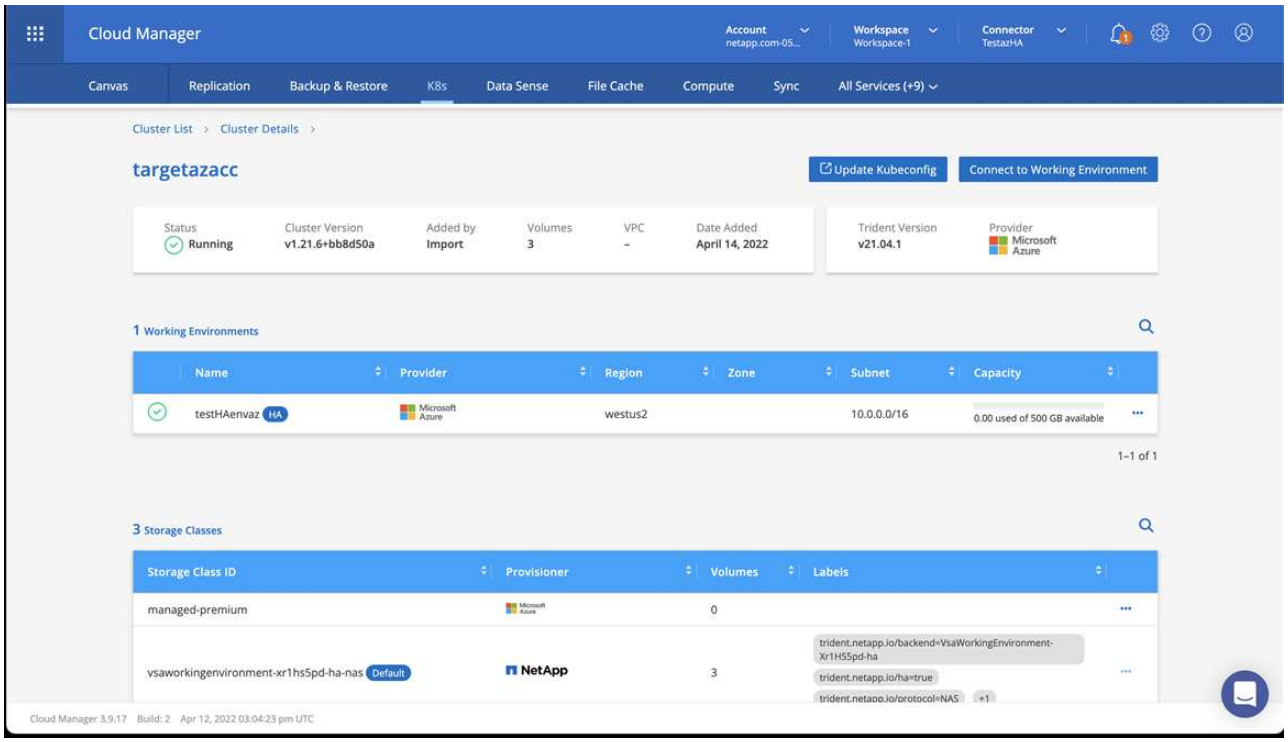
   a. Percorso: "Microsoft Azure".

   b. Tipo: "Cloud Volumes ONTAP ha".



5. Importare il cluster OpenShift. Il cluster si connetterà 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, osserva la versione di Astra Control Provisioner.

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.
Astra Control provisioner viene installato automaticamente nell'ambito 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 per Azure

Installare Astra Control Center con lo standard "istruzioni per l'installazione".

Utilizzando Astra Control Center, aggiungere un bucket Azure. Fare riferimento a. "Configurare Astra Control Center e aggiungere i bucket".

Configurare Astra Control Center dopo l'installazione

A seconda dell'ambiente in uso, potrebbe essere necessaria una configurazione aggiuntiva dopo l'installazione di Astra Control Center.

Rimuovere le limitazioni delle risorse

Alcuni ambienti utilizzano gli oggetti ResourceQuotas e LimitRanges per impedire alle risorse di uno spazio dei nomi di consumare tutta la CPU e la memoria disponibili nel cluster. Astra Control Center non imposta limiti massimi, pertanto non sarà conforme a tali risorse. Se l'ambiente è configurato in questo modo, è necessario rimuovere tali risorse dagli spazi dei nomi in cui si intende installare Astra Control Center.

Per recuperare e rimuovere le quote e i limiti, procedere come segue. In questi esempi, l'output del comando viene visualizzato immediatamente dopo il comando.

Fasi

1. Ottenere le quote delle risorse in `netapp-acc` namespace (o personalizzato):

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

Risposta:

```
NAME          AGE   REQUEST                                          LIMIT
pods-high     16s   requests.cpu: 0/20, requests.memory: 0/100Gi
limits.cpu: 0/200, limits.memory: 0/1000Gi
pods-low      15s   requests.cpu: 0/1, requests.memory: 0/1Gi
limits.cpu: 0/2, limits.memory: 0/2Gi
pods-medium   16s   requests.cpu: 0/10, requests.memory: 0/20Gi
limits.cpu: 0/20, limits.memory: 0/200Gi
```

2. Eliminare tutte le quote delle risorse in base al nome:

```
kubectl delete resourcequota pods-high -n [netapp-acc or custom
namespace]
```

```
kubectl delete resourcequota pods-low -n [netapp-acc or custom
namespace]
```

```
kubectl delete resourcequota pods-medium -n [netapp-acc or custom
namespace]
```

3. Ottenere gli intervalli di limite in `netapp-acc` namespace (o personalizzato):

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

Risposta:

```
NAME              CREATED AT
cpu-limit-range   2022-06-27T19:01:23Z
```

4. Eliminare gli intervalli di limiti in base al nome:

```
kubectl delete limitrange cpu-limit-range -n [netapp-acc or custom
namespace]
```

Aggiungere un certificato TLS personalizzato

Astra Control Center utilizza per impostazione predefinita un certificato TLS autofirmato per il traffico dei controller di ingresso (solo in alcune configurazioni) e l'autenticazione dell'interfaccia utente Web con i browser Web. Per l'utilizzo in produzione, è necessario rimuovere il certificato TLS autofirmato esistente e sostituirlo con un certificato TLS firmato da un'autorità di certificazione (CA).

> Il certificato autofirmato predefinito viene utilizzato per due tipi di connessione:
>
> - Connessioni HTTPS all'interfaccia utente Web di Astra Control Center
> - Traffico del controller di ingresso (solo se `ingressType: "AccTraefik"` la proprietà è stata impostata in `astra_control_center.yaml` Durante l'installazione di Astra Control Center)
>
> La sostituzione del certificato TLS predefinito sostituisce il certificato utilizzato per l'autenticazione di queste connessioni.

Prima di iniziare

- Kubernetes cluster con Astra Control Center installato
- Accesso amministrativo a una shell dei comandi sul cluster da eseguire `kubectl` comandi
- Chiave privata e file di certificato dalla CA

Rimuovere il certificato autofirmato

Rimuovere il certificato TLS autofirmato esistente.

1. Utilizzando SSH, accedere al cluster Kubernetes che ospita Astra Control Center come utente amministrativo.
2. Individuare il segreto TLS associato al certificato corrente utilizzando il seguente comando, sostituendo `<ACC-deployment-namespace>` Con lo spazio dei nomi di implementazione di Astra Control Center:

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

3. Eliminare il certificato e il segreto attualmente installati utilizzando i seguenti comandi:

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

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

Aggiungere un nuovo certificato utilizzando la riga di comando

Aggiungere un nuovo certificato TLS firmato da una CA.

1. Utilizzare il seguente comando per creare il nuovo segreto TLS con la chiave privata e i file di certificato della CA, sostituendo gli argomenti tra parentesi <> con le informazioni appropriate:

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

2. Utilizzare il seguente comando e l'esempio per modificare il file CRD (Custom Resource Definition) del cluster e modificare `spec.selfSigned` valore a. `spec.ca.secretName` Per fare riferimento al segreto TLS creato in precedenza:

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

CRD:

```
#spec:
#  selfSigned: {}

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

3. Utilizzare il seguente comando e l'output di esempio per confermare che le modifiche sono corrette e che il cluster è pronto per validare i certificati, sostituendo `<ACC-deployment-namespace>` Con lo spazio dei nomi di implementazione di Astra Control Center:

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

Risposta:

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

4. Creare il `certificate.yaml` file utilizzando il seguente esempio, sostituendo i valori segnaposto tra parentesi <> con le informazioni appropriate:

> In questo esempio viene utilizzato il `dnsNames` Per specificare l'indirizzo DNS di Astra Control Center. Astra Control Center non supporta l'utilizzo della proprietà Common Name (CN) per specificare l'indirizzo DNS.

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

5. Creare il certificato utilizzando il seguente comando:

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

6. Utilizzando il seguente comando e l'output di esempio, verificare che il certificato sia stato creato correttamente e con gli argomenti specificati durante la creazione (ad esempio nome, durata, scadenza di rinnovo e nomi DNS).

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

Risposta:

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

7. Modificare il CRD degli archivi TLS in modo che punti al nuovo nome segreto del certificato utilizzando il seguente comando ed esempio, sostituendo i valori segnaposto tra parentesi <> con le informazioni appropriate

```
kubectl edit tlsstores.traefik.io -n <ACC-deployment-namespace>
```

CRD:

```
...
spec:
  defaultCertificate:
    secretName: <certificate-secret-name>
```

8. Modificare l'opzione TLS CRD di ingresso per indicare il nuovo segreto del certificato utilizzando il seguente comando ed esempio, sostituendo i valori segnaposto tra parentesi <> con le informazioni appropriate:

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

CRD:

```
...
 tls:
    secretName: <certificate-secret-name>
```

9. Utilizzando un browser Web, accedere all'indirizzo IP di implementazione di Astra Control Center.
10. Verificare che i dettagli del certificato corrispondano ai dettagli del certificato installato.
11. Esportare il certificato e importare il risultato nel gestore dei certificati nel browser Web.