Inizia subito

Astra Control Center

NetApp
November 21, 2023

Sommario

Inizia subito

Requisiti di Astra Control Center

Inizia verificando la preparazione del tuo ambiente operativo, dei cluster di applicazioni, delle applicazioni, delle licenze e del browser Web.

Requisiti dell'ambiente operativo

Astra Control Center richiede uno dei seguenti tipi di ambienti operativi:

- Red Hat OpenShift Container Platform 4.6.8, 4.7 o 4.8
- Rancher 2.5
- Kubernetes da 1.19 a 1.21 (incluso 1.21.x)

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 le seguenti risorse oltre ai requisiti delle risorse dell'ambiente:

Componente	Requisito
Capacità di storage ONTAP back-end	Almeno 300 GB disponibili
Nodi di lavoro	Almeno 3 nodi di lavoro in totale, con 4 core CPU e 12 GB di RAM ciascuno
Bilanciamento del carico	Tipo di servizio "LoadBalancer" disponibile per il traffico in ingresso da inviare ai servizi nel cluster dell'ambiente operativo
Risoluzione FQDN	Metodo per indirizzare l'FQDN di Astra Control Center all'indirizzo IP con bilanciamento del carico
Astra Trident	• Astra Trident 21.04 o versione successiva installata e configurata se NetApp ONTAP versione 9.5 o successiva verrà utilizzato come backend di storage • Astra Trident 21.10.1 o versione successiva installata e configurata se l'anteprima di Astra Data Store verrà utilizzata come backend di storage

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

- **Registro immagini**: È necessario disporre di un registro di immagini Docker privato in cui è possibile trasferire le immagini di build di Astra Control Center. È necessario fornire l'URL del registro delle immagini in cui verranno caricate le immagini.
- **Astra Trident / ONTAP Configuration**: Astra Control Center richiede la creazione e l'impostazione di una classe di storage come classe di storage predefinita. Centro di controllo Astra supporta i seguenti driver

ONTAP forniti da Astra Trident:

- ontap-nas
- ontap-san
- ontap-san-economy

> Durante la clonazione delle applicazioni in ambienti OpenShift, Astra Control Center deve consentire a OpenShift di montare volumi e modificare la proprietà dei file. Per questo motivo, è necessario configurare un criterio di esportazione dei volumi ONTAP per consentire queste operazioni. Puoi farlo con i seguenti comandi:
>
> 1. `export-policy rule modify -vserver <storage virtual machine name> -policyname <policy name> -ruleindex 1 -superuser sys`
> 2. `export-policy rule modify -vserver <storage virtual machine name> -policyname <policy name> -ruleindex 1 -anon 65534`

> Se si prevede di aggiungere un secondo ambiente operativo OpenShift come risorsa di calcolo gestita, è necessario assicurarsi che la funzione Astra Trident Volume Snapshot sia attivata. Per abilitare e testare le snapshot dei volumi con Astra Trident, "Consulta le istruzioni ufficiali di Astra Trident".

Requisiti del cluster di applicazioni

Astra Control Center ha i seguenti requisiti per i cluster che si intende gestire da Astra Control Center. Questi requisiti si applicano anche se il cluster che si intende gestire è il cluster dell'ambiente operativo che ospita Astra Control Center.

- La versione più recente di Kubernetes "componente snapshot-controller" è installato
- Un tridente Astra "oggetto volumesnapshotclass" è stato definito da un amministratore
- Nel cluster esiste una classe di storage Kubernetes predefinita
- Almeno una classe di storage è configurata per utilizzare Astra Trident

> Il cluster di applicazioni deve disporre di un `kubeconfig.yaml` file che definisce un solo elemento *context*. Visitare la documentazione Kubernetes per "informazioni sulla creazione di file kubeconfig".

> Quando si gestiscono i cluster di applicazioni in un ambiente Rancher, modificare il contesto predefinito del cluster di applicazioni in `kubeconfig` File fornito da Rancher per utilizzare un contesto del piano di controllo invece del contesto del server API Rancher. In questo modo si riduce il carico sul server API Rancher e si migliorano le performance.

Requisiti di gestione delle applicazioni

Astra Control ha i seguenti requisiti di gestione delle applicazioni:

- **Licensing**: Per gestire le applicazioni utilizzando Astra Control Center, è necessaria una licenza Astra Control Center.
- **Namespaces**: Astra Control richiede che un'applicazione non si estende più di un singolo namespace, ma uno spazio dei nomi può contenere più di un'applicazione.

- **StorageClass**: Se si installa un'applicazione con un StorageClass esplicitamente impostato ed è necessario clonare l'applicazione, il cluster di destinazione per l'operazione di clone deve avere la StorageClass specificata in origine. Il cloning di un'applicazione con un StorageClass esplicitamente impostato su un cluster che non ha lo stesso StorageClass avrà esito negativo.
- **Kubernetes resources**: Le applicazioni che utilizzano risorse Kubernetes non raccolte da Astra Control potrebbero non disporre di funzionalità complete di gestione dei dati delle applicazioni. Astra Control raccoglie le seguenti risorse Kubernetes:
  - ClusterRole
  - ClusterRoleBinding
  - ConfigMap
  - CustomResourceDefinition
  - CustomResource
  - DemonSet
  - Implementazione
  - DeploymentConfig
  - Ingresso
  - MutatingWebhook
  - PersistentVolumeClaim
  - Pod
  - ReplicaSet
  - RoleBinding
  - Ruolo
  - Percorso
  - Segreto
  - Servizio
  - ServiceAccount
  - StatefulSet
  - ValidatingWebhook

Metodi di installazione delle applicazioni supportati

Astra Control supporta i seguenti metodi di installazione dell'applicazione:

- **Manifest file**: Astra Control supporta le applicazioni installate da un file manifest utilizzando kubectl. Ad esempio:

  ```
  kubectl apply -f myapp.yaml
  ```

- **Helm 3**: Se utilizzi Helm per installare le app, Astra Control richiede Helm versione 3. La gestione e la clonazione delle applicazioni installate con Helm 3 (o aggiornate da Helm 2 a Helm 3) sono completamente supportate. La gestione delle applicazioni installate con Helm 2 non è supportata.
- **Applicazioni distribuite dall'operatore**: Astra Control supporta le applicazioni installate con operatori con

ambito namespace. Di seguito sono riportate alcune applicazioni che sono state validate per questo modello di installazione:

- "Apache K8ssandra"
- "Ci Jenkins"
- "Cluster XtraDB Percona"

> Un operatore e l'applicazione che installa devono utilizzare lo stesso namespace; potrebbe essere necessario modificare il file .yaml di implementazione per l'operatore per assicurarsi che questo sia il caso.

Accesso a Internet

È necessario determinare se si dispone di un accesso esterno a Internet. In caso contrario, alcune funzionalità potrebbero essere limitate, ad esempio la ricezione di dati di monitoraggio e metriche da NetApp Cloud Insights o l'invio di pacchetti di supporto a "Sito di supporto NetApp".

Licenza

Astra Control Center richiede una licenza Astra Control Center per una funzionalità completa. Ottenere una licenza di valutazione o una licenza completa da NetApp. Senza una licenza, non sarà possibile:

- Definire applicazioni personalizzate
- Creare snapshot o cloni di applicazioni esistenti
- Configurare le policy di protezione dei dati

Se si desidera provare Astra Control Center, è possibile "utilizzare una licenza di valutazione di 90 giorni".

Tipo di servizio "LoadBalancer" per cluster Kubernetes on-premise

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 i bilanciatori di carico sono consentiti nel tuo ambiente e non ne hai già configurati uno, puoi utilizzare "MetalLB" Per assegnare automaticamente 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.

Requisiti di rete

L'ambiente operativo che ospita Astra Control Center comunica utilizzando le seguenti porte TCP. Assicurarsi che queste porte siano consentite attraverso qualsiasi firewall e configurare i firewall in modo da consentire qualsiasi traffico HTTPS in uscita dalla rete Astra. Alcune porte richiedono la connettività tra l'ambiente che ospita Astra Control Center e ciascun cluster gestito (annotato dove applicabile).

Origine	Destinazione	Porta	Protocollo	Scopo
PC client	Centro di controllo Astra	443	HTTPS	Accesso UI/API - assicurarsi che questa porta sia aperta in entrambi i modi tra il cluster che ospita Astra Control Center e ciascun cluster gestito
Metriche consumer	Nodo di lavoro Astra Control Center	9090	HTTPS	Comunicazione dei dati delle metriche - garantire che ciascun cluster gestito possa accedere a questa porta sul cluster che ospita Astra Control Center (è richiesta una comunicazione bidirezionale)
Centro di controllo Astra	Servizio Hosted Cloud Insights (https://cloudinsights.netapp.com)	443	HTTPS	Comunicazione Cloud Insights
Centro di controllo Astra	Provider di bucket di storage Amazon S3 (https://my-bucket.s3.us-west-2.amazonaws.com/)	443	HTTPS	Comunicazione con lo storage Amazon S3
Centro di controllo Astra	NetApp ActiveIQ (https://activeiq.solidfire.com)	443	HTTPS	Comunicazione NetApp ActiveIQ

Browser Web supportati

Astra Control Center supporta versioni recenti di Firefox, Safari e Chrome con una risoluzione minima di 1280 x 720.

Cosa succederà

Visualizzare il "avvio rapido" panoramica.

Avvio rapido per Astra Control Center

Questa pagina fornisce una panoramica generale dei passaggi necessari per iniziare a utilizzare Astra Control Center. I collegamenti all'interno di ogni passaggio consentono di accedere a una pagina che fornisce ulteriori dettagli.

Provalo! Se si desidera provare Astra Control Center, è possibile utilizzare una licenza di valutazione di 90

giorni. Vedere "informazioni sulle licenze" per ulteriori informazioni.

1 Esaminare i requisiti del cluster Kubernetes

- Astra funziona con i cluster Kubernetes con un backend di storage ONTAP configurato con Trident o un backend di storage di anteprima Astra Data Store.
- I cluster devono essere in esecuzione in condizioni di salute, con almeno tre nodi di lavoro online.
- Il cluster deve eseguire Kubernetes.

"Scopri di più sui requisiti di Astra Control Center".

2 Scaricare e installare Astra Control Center

- Scaricare Astra Control Center dal "Sito di supporto NetApp pagina Download di Astra Control Center".
- Installare Astra Control Center nell'ambiente locale.

  Se lo si desidera, installare Astra Control Center utilizzando Red Hat OperatorHub.

- Scopri la tua configurazione Trident supportata dal backend dello storage ONTAP. Oppure, scopri il tuo "Anteprima di Astra Data Store" cluster come back-end dello storage.

  Le immagini vengono installate in un registro OpenShift o utilizzate il registro locale.

"Scopri di più sull'installazione di Astra Control Center".

3 Completare alcune attività di configurazione iniziali

- Aggiungere una licenza.
- Aggiungi un cluster Kubernetes e Astra Control Center scopre i dettagli.
- Aggiungi un backend di storage per l'anteprima del data store ONTAP o Astra.
- Facoltativamente, Aggiungi un bucket di store di oggetti che memorizzerà i backup delle app.

"Scopri di più sul processo di configurazione iniziale".

4 Utilizzare Astra Control Center

Dopo aver completato la configurazione di Astra Control Center, ecco cosa fare:

- Gestire un'applicazione. "Scopri di più su come gestire le app".
- Se lo si desidera, connettersi a NetApp Cloud Insights per visualizzare le metriche sullo stato di salute del sistema, sulla capacità e sul throughput all'interno dell'interfaccia utente di Astra Control Center. "Scopri di più sulla connessione a Cloud Insights".

5 Continuare da questa guida di avvio rapido

"Installare Astra Control Center".

Trova ulteriori informazioni

- "Utilizzare l'API di controllo Astra"

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 Astra Control Center utilizzando il processo standard

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

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

Di cosa hai bisogno

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

  Esempio di OpenShift:

  ```
  oc get clusteroperators
  ```

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

  Esempio di OpenShift:

  ```
  oc get apiservices
  ```

- Hai creato un indirizzo FQDN per Astra Control Center nel tuo data center.

A proposito di questa attività

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

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

- Installa l'interfaccia utente Astra.

> I comandi Podman possono essere utilizzati al posto dei comandi Docker se si utilizza il Podman di Red Hat invece di Docker Engine.

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

Fasi

Per installare Astra Control Center, procedere come segue:

- Scarica il bundle Astra Control Center
- Disimballare il bundle e modificare la directory
- Aggiungere le immagini al registro locale
- Impostare namespace e secret per i registri con requisiti di autenticazione
- Installare l'operatore del centro di controllo Astra
- Configurare Astra Control Center
- Completare l'installazione dell'Astra Control Center e dell'operatore
- Verificare lo stato del sistema
- Accedere all'interfaccia utente di Astra Control Center

Completare l'implementazione eseguendo "attività di installazione".

Scarica il bundle Astra Control Center

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

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

Disimballare il bundle e modificare la directory

1. Estrarre le immagini:

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

2. Passare alla directory Astra.

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

Aggiungere le immagini al registro locale

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

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

   a. Accedere al Registro di sistema:

      Docker:

      ```
      docker login [your_registry_path]
      ```

      Podman:

      ```
      podman login [your_registry_path]
      ```

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

      Docker:

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

      Podman:

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

Impostare namespace e secret per i registri con requisiti di autenticazione

1. Se si utilizza un registro che richiede l'autenticazione, è necessario effettuare le seguenti operazioni:

   a. Creare il `netapp-acc-operator` spazio dei nomi:

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

   Risposta:

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

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

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

   Esempio di risposta:

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

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

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

   Esempio di risposta:

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

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

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

Risposta

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

Installare l'operatore del centro di controllo Astra

1. Modificare l'YAML di implementazione dell'operatore di Astra Control Center (`astra_control_center_operator_deploy.yaml`) per fare riferimento al registro locale e al segreto.

```
vim astra_control_center_operator_deploy.yaml
```

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

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

b. Cambiare `[your_registry_path]` per `kube-rbac-proxy` al percorso del registro in cui sono state inviate le immagini in a. passaggio precedente.

c. Cambiare `[your_registry_path]` per `acc-operator-controller-manager` al percorso del registro in cui sono state inviate le immagini in a. passaggio precedente.

d. (Per le installazioni che utilizzano l'anteprima di Astra Data Store) vedere questo problema noto relativo a. "Provisioning delle classi di storage e modifiche aggiuntive da apportare al programma YAML".

```
apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    control-plane: controller-manager
  name: acc-operator-controller-manager
  namespace: netapp-acc-operator
spec:
  replicas: 1
  selector:
    matchLabels:
      control-plane: controller-manager
  template:
    metadata:
      labels:
        control-plane: controller-manager
    spec:
      containers:
      - args:
        - --secure-listen-address=0.0.0.0:8443
        - --upstream=http://127.0.0.1:8080/
        - --logtostderr=true
        - --v=10
        image: [your_registry_path]/kube-rbac-proxy:v4.8.0
        name: kube-rbac-proxy
        ports:
        - containerPort: 8443
          name: https
      - args:
        - --health-probe-bind-address=:8081
        - --metrics-bind-address=127.0.0.1:8080
        - --leader-elect
        command:
        - /manager
        env:
        - name: ACCOP_LOG_LEVEL
          value: "2"
        image: [your_registry_path]/acc-operator:[version x.y.z]
        imagePullPolicy: IfNotPresent
      imagePullSecrets: []
```

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

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

Esempio di risposta:

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

Configurare Astra Control Center

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

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

```
vim astra_control_center_min.yaml
```

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

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

a. Cambiare `[your_registry_path]` al percorso del registro di sistema in cui sono state inviate le immagini nel passaggio precedente.

b. Modificare il `accountName` stringa al nome che si desidera associare all'account.

c. Modificare il `astraAddress` Stringa all'FQDN che si desidera utilizzare nel browser per accedere ad Astra. Non utilizzare `http://` oppure `https://` nell'indirizzo. Copiare questo FQDN per utilizzarlo in

un passo successivo.

d. Modificare il `email` stringa all'indirizzo iniziale predefinito dell'amministratore. Copiare questo indirizzo e-mail per utilizzarlo in passo successivo.

e. Cambiare `enrolled` Per AutoSupport a. `false` per i siti senza connettività internet o senza retain `true` per i siti connessi.

f. (Facoltativo) aggiungere un nome `firstName` e cognome `lastName` dell'utente associato all'account. È possibile eseguire questo passaggio ora o in un secondo momento all'interno dell'interfaccia utente.

g. (Facoltativo) modificare `storageClass` Valore per un'altra risorsa Astra Trident StorageClass, se richiesto dall'installazione.

h. (Per le installazioni che utilizzano l'anteprima di Astra Data Store) vedere questo problema noto per "ulteriori modifiche richieste" Al programma YAML.

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

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

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

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

Esempio di risposta:

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

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

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

Esempio di risposta:

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

Verificare lo stato del sistema

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

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

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

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

Esempio di risposta:

```
NAME                                        READY   STATUS    RESTARTS
AGE
acc-helm-repo-5f75c5f564-bzqmt              1/1     Running   0
11m
activity-6b8f7cccb9-mlrn4                   1/1     Running   0
9m2s
api-token-authentication-6hznt              1/1     Running   0
8m50s
api-token-authentication-qpfgb              1/1     Running   0
8m50s
api-token-authentication-sqnb7              1/1     Running   0
8m50s
asup-5578bbdd57-dxkbp                       1/1     Running   0
9m3s
authentication-56bff4f95d-mspmq             1/1     Running   0
7m31s
bucketservice-6f7968b95d-9rrrl              1/1     Running   0
8m36s
cert-manager-5f6cf4bc4b-82khn               1/1     Running   0
6m19s
cert-manager-cainjector-76cf976458-sdrbc    1/1     Running   0
6m19s
cert-manager-webhook-5b7896bfd8-2n45j       1/1     Running   0
```

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

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

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

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

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

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

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

4. Controllare `status.deploymentState` nella risposta per `Deployed` valore. Se l'implementazione non ha avuto esito positivo, viene visualizzato un messaggio di errore.

> Verrà utilizzato il `uuid` nella fase successiva.

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

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

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

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

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

5. Per ottenere la password monouso da utilizzare quando si accede ad Astra Control Center, copiare il `status.uuid` valore della risposta nella fase precedente. La password è `ACC-` Seguito dal valore UUID (`ACC-[UUID]` oppure, in questo esempio, `ACC-c49008a5-4ef1-4c5d-a53e-830daf994116`).

Accedere all'interfaccia utente di Astra Control Center

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

Fasi

1. In un browser, immettere l'FQDN utilizzato in `astraAddress` in `astra_control_center_min.yaml` CR quando Astra Control Center è stato installato.

2. Accettare i certificati autofirmati quando richiesto.

> È possibile creare un certificato personalizzato dopo l'accesso.

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

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

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

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

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

Risolvere i problemi di installazione

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

Fasi

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

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

Cosa succederà

Completare l'implementazione eseguendo "attività di installazione".

Installare Astra Control Center utilizzando OpenShift OperatorHub

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

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

Di cosa hai bisogno

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

`true`):

```
oc get clusteroperators
```

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

```
oc get apiservices
```

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

Fasi

- Scarica il bundle Astra Control Center
- Disimballare il bundle e modificare la directory
- Aggiungere le immagini al registro locale
- Individuare la pagina di installazione dell'operatore
- Installare l'operatore
- Installare Astra Control Center

Scarica il bundle Astra Control Center

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

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

Disimballare il bundle e modificare la directory

1. Estrarre le immagini:

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

2. Passare alla directory Astra.

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

Aggiungere le immagini al registro locale

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

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

   a. Accedere al Registro di sistema:

      Docker:

      ```
      docker login [your_registry_path]
      ```

      Podman:

      ```
      podman login [your_registry_path]
      ```

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

      Docker:

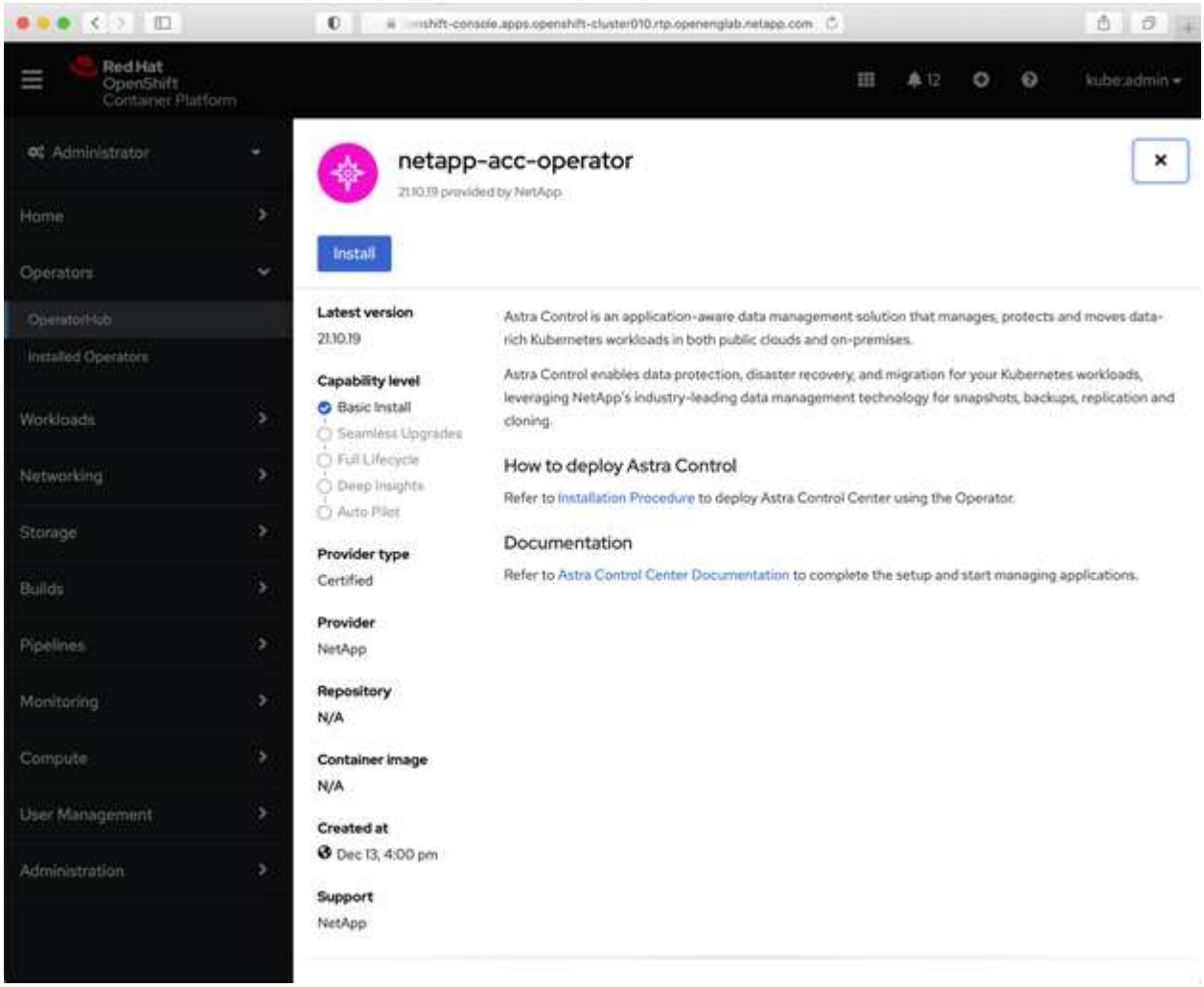
      ```
      export REGISTRY=[Docker_registry_path]
      for astraImageFile in $(ls images/*.tar) ; do
        # Load to local cache. And store the name of the loaded image
      trimming the 'Loaded images: '
        astraImage=$(docker load --input ${astraImageFile} | sed 's/Loaded
      image: //')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

      Podman:

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

Individuare la pagina di installazione dell'operatore

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



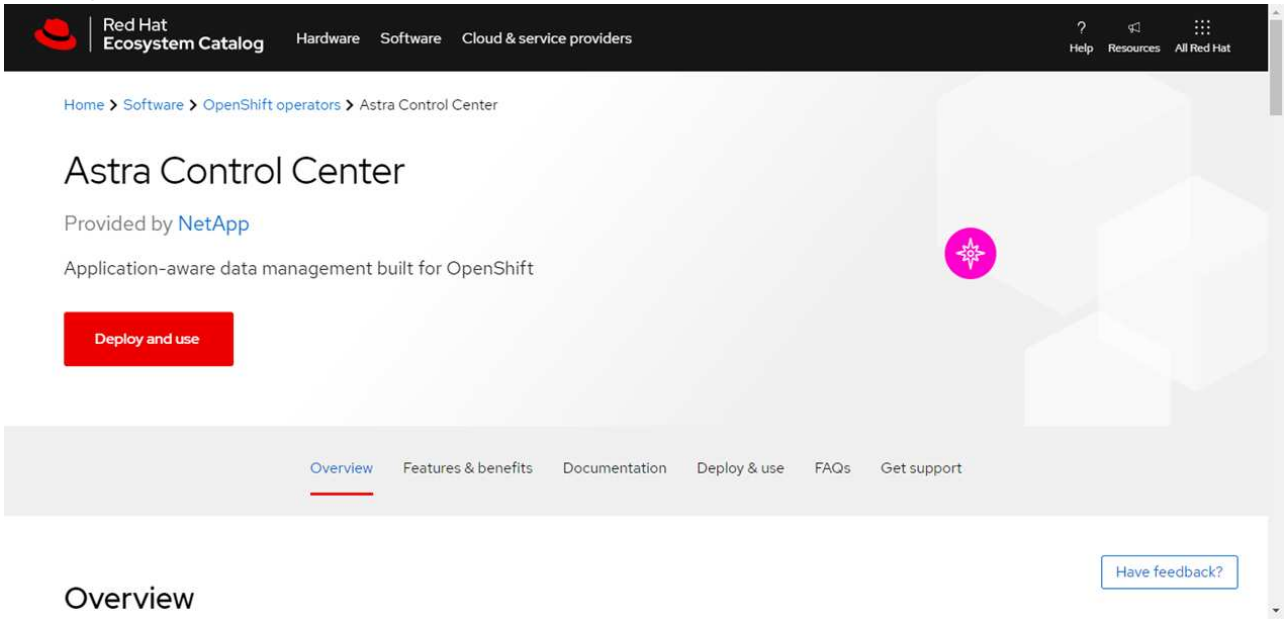
     i. Accedere all'interfaccia utente di OpenShift Container Platform.

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

iii. Selezionare l'operatore di NetApp Astra Control Center.

iv. Selezionare **Installa**.

- Dal Red Hat Ecosystem Catalog:



i. Selezionare NetApp Astra Control Center "operatore".

ii. Selezionare **Deploy and Use** (implementazione e utilizzo).

Installare l'operatore

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

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

a. Selezionare lo spazio dei nomi dell'operatore o. `netapp-acc-operator` lo spazio dei nomi verrà creato automaticamente come parte dell'installazione dell'operatore.

b. Selezionare una strategia di approvazione manuale o automatica.

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

c. Selezionare **Installa**.

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

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

Installare Astra Control Center

1. Dalla console nella vista dettagli dell'operatore Astra Control Center, selezionare `Create instance`

Nella sezione API fornite.

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

Cosa succederà

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

Configurare Astra Control Center

Il centro di controllo Astra supporta e monitora l'archivio dati ONTAP e Astra come back-end dello storage. Dopo aver installato Astra Control Center, aver effettuato l'accesso all'interfaccia utente e aver modificato la password, sarà necessario impostare una licenza, aggiungere cluster, gestire lo storage e aggiungere bucket.

Attività

- Aggiungere una licenza per Astra Control Center
- Aggiungere il cluster
- Aggiungere un backend di storage
- Aggiungi un bucket

Aggiungere una licenza per Astra Control Center

È possibile aggiungere una nuova licenza utilizzando l'interfaccia utente o. "API" Per ottenere la funzionalità completa di Astra Control Center. Senza una licenza, l'utilizzo di Astra Control Center è limitato alla gestione degli utenti e all'aggiunta di nuovi cluster.

Di cosa hai bisogno

Quando si scarica Astra Control Center da "Sito di supporto NetApp", Inoltre, è stato scaricato il file di licenza NetApp (NLF). Assicurarsi di avere accesso a questo file di licenza.

Per aggiornare una licenza di valutazione o una licenza completa, vedere "Aggiornare una licenza esistente".

Aggiungere una licenza completa o di valutazione

Le licenze di Astra Control Center misurano le risorse della CPU utilizzando le unità CPU di Kubernetes. La licenza deve tenere conto delle risorse CPU assegnate ai nodi di lavoro di tutti i cluster Kubernetes gestiti. Prima di aggiungere una licenza, è necessario ottenere il file di licenza (NLF) da "Sito di supporto NetApp".

Puoi anche provare Astra Control Center con una licenza di valutazione, che ti consente di utilizzare Astra Control Center per 90 giorni dalla data di download della licenza. Puoi iscriverti per una prova gratuita registrandoti "qui".

Se l'installazione supera il numero concesso in licenza di unità CPU, Astra Control Center impedisce la gestione di nuove applicazioni. Quando viene superata la capacità, viene visualizzato un avviso.

Fasi

1. Accedere all'interfaccia utente di Astra Control Center.
2. Selezionare **account** > **licenza**.
3. Selezionare **Aggiungi licenza**.
4. Individuare il file di licenza (NLF) scaricato.
5. Selezionare **Aggiungi licenza**.

La pagina **account** > **licenza** visualizza le informazioni sulla licenza, la data di scadenza, il numero di serie della licenza, l'ID account e le unità CPU utilizzate.

Se si dispone di una licenza di valutazione, assicurarsi di memorizzare l'ID account per evitare la perdita di dati in caso di guasto di Astra Control Center se non si inviano ASUP.

Aggiungere il cluster

Per iniziare a gestire le tue applicazioni, Aggiungi un cluster Kubernetes e gestilo come risorsa di calcolo. Devi aggiungere un cluster per Astra Control Center per scoprire le tue applicazioni Kubernetes. Per l'anteprima di Astra Data Store, aggiungere il cluster di applicazioni Kubernetes che contiene applicazioni che utilizzano volumi forniti da Astra Data Store Preview.

Si consiglia ad Astra Control Center di gestire il cluster su cui viene implementato prima di aggiungere altri cluster ad Astra Control Center da gestire. La gestione del cluster iniziale è necessaria per inviare i dati Kublemetrics e i dati associati al cluster per metriche e troubleshooting. È possibile utilizzare la funzione **Add Cluster** per gestire un cluster con Astra Control Center.

Quando Astra Control gestisce un cluster, tiene traccia della StorageClass predefinita del cluster. Se si modifica StorageClass utilizzando `kubectl` Comandi, Astra Control ripristina la modifica. Per modificare la classe di storage predefinita in un cluster gestito da Astra Control, utilizzare uno dei seguenti metodi:

- Utilizzare l'API di controllo Astra `PUT /managedClusters` E assegnare un diverso StorageClass predefinito con `DefaultStorageClass` parametro
- Rimuovere il cluster dalla gestione di Astra Control e aggiungerlo nuovamente con un diverso StorageClass predefinito selezionato

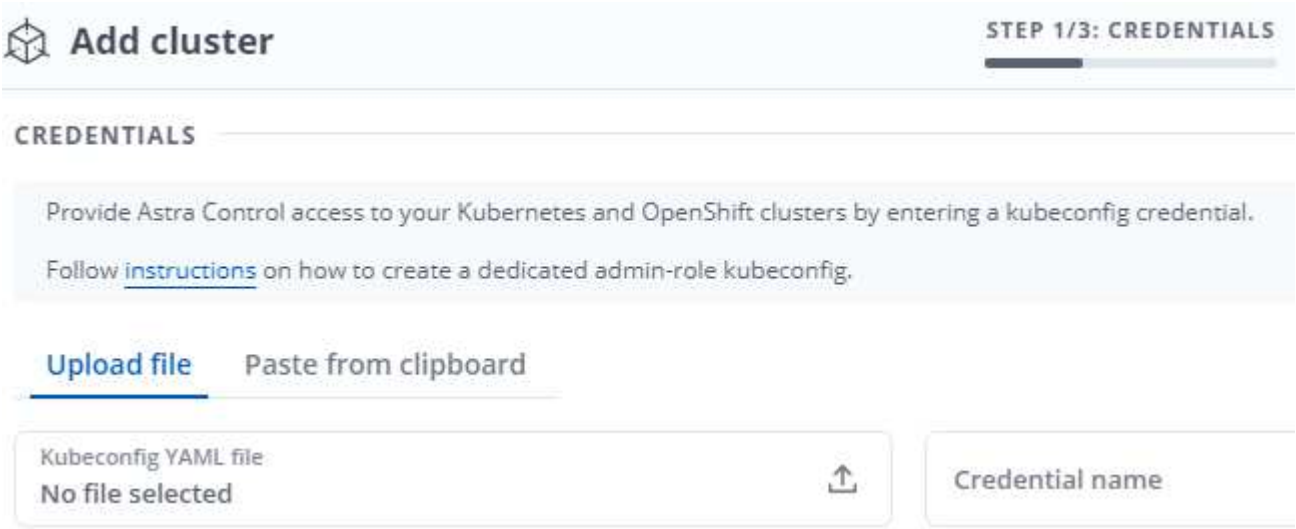
Cosa ti serve? 8217

Prima di aggiungere un cluster, esaminare ed eseguire le operazioni necessarie "attività prerequisite".

Fasi

1. Dal pannello **Dashboard** dell'interfaccia utente di Astra Control Center, selezionare **Add** (Aggiungi) nella sezione Clusters (Clusters).
2. Nella finestra **Add Cluster** che si apre, caricare un `kubeconfig.yaml` archiviare o incollare il contenuto di a. `kubeconfig.yaml` file.

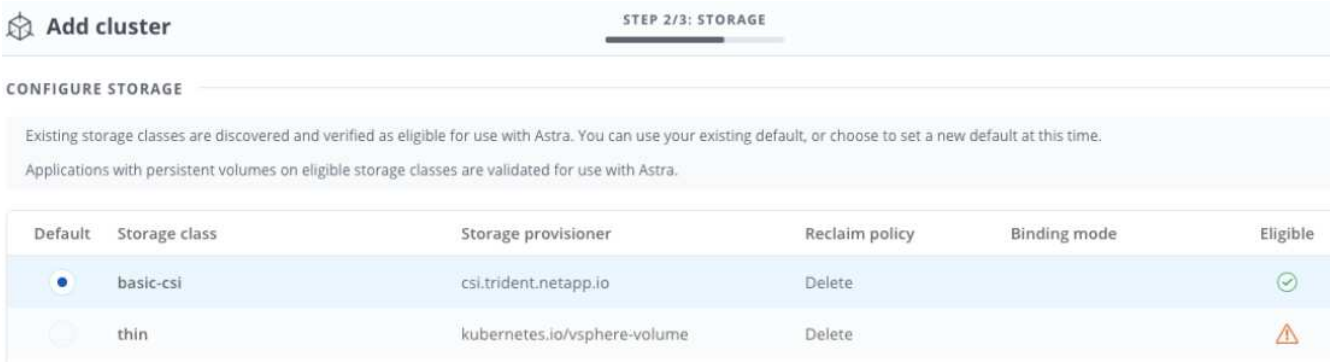
   Il `kubeconfig.yaml` il file deve includere **solo le credenziali del cluster per un cluster**.



   Se crei il tuo `kubeconfig` file, è necessario definire solo **un** elemento di contesto al suo interno. Vedere "Documentazione Kubernetes" per informazioni sulla creazione `kubeconfig` file.

3. Fornire un nome di credenziale. Per impostazione predefinita, il nome della credenziale viene compilato automaticamente come nome del cluster.
4. Selezionare **Configura storage**.
5. Selezionare la classe di storage da utilizzare per questo cluster Kubernetes e selezionare **Review**.

È necessario selezionare una classe di storage Trident supportata dallo storage ONTAP o dall'archivio dati Astra.



6. Esaminare le informazioni e, se l'aspetto è soddisfacente, selezionare **Aggiungi cluster**.

Risultato

Il cluster passa allo stato **rilevamento**, quindi passa a **in esecuzione**. Hai aggiunto un cluster Kubernetes e lo stai gestendo in Astra Control Center.

Dopo aver aggiunto un cluster da gestire in Astra Control Center, l'implementazione dell'operatore di monitoraggio potrebbe richiedere alcuni minuti. Fino a quel momento, l'icona di notifica diventa rossa e registra un evento **Monitoring Agent Status Check Failed** (controllo stato agente non riuscito). È possibile ignorarlo, perché il problema si risolve quando Astra Control Center ottiene lo stato corretto. Se il problema non si risolve in pochi minuti, accedere al cluster ed eseguire `oc get pods -n netapp-monitoring` come punto di partenza. Per eseguire il debug del problema, consultare i log dell'operatore di monitoraggio.

Aggiungere un backend di storage

È possibile aggiungere un backend di storage in modo che Astra Control possa gestire le proprie risorse. La gestione dei cluster di storage in Astra Control come back-end dello storage consente di ottenere collegamenti tra volumi persistenti (PVS) e il back-end dello storage, oltre a metriche di storage aggiuntive.

È possibile aggiungere un backend di storage rilevato navigando tra le richieste dal Dashboard o dal menu Backend.

Di cosa hai bisogno

- Lo hai fatto "aggiunto un cluster" Ed è gestito da Astra Control.

  Al cluster gestito è associato un backend supportato che può essere rilevato da Astra Control.

- Per le installazioni di anteprima di Astra Data Store: Hai aggiunto il cluster dell'app Kubernetes.

> Dopo aver aggiunto il cluster di applicazioni Kubernetes per Astra Data Store, il cluster viene visualizzato come `unmanaged` nell'elenco dei backend rilevati. È quindi necessario aggiungere il cluster di calcolo che contiene Astra Data Store e che si trova sotto il cluster di applicazioni Kubernetes. È possibile eseguire questa operazione da **Backend** nell'interfaccia utente. Selezionare il menu Actions (azioni) per il cluster, quindi scegliere `Manage`, e. "aggiungere il cluster". Dopo lo stato del cluster di `unmanaged` Modifiche al nome del cluster Kubernetes, è possibile procedere con l'aggiunta di un backend.

Fasi

1. Effettuare una delle seguenti operazioni:
   - Da **Dashboard**:
     i. Dalla sezione backend Dashboard Storage, selezionare **Manage** (Gestisci).
     ii. Dalla sezione Dashboard Resource Summary > Storage Backend, selezionare **Add** (Aggiungi).
   - Da **backend**:
     i. Nell'area di navigazione a sinistra, selezionare **Backend**.
     ii. Selezionare **Gestisci**.
2. Eseguire una delle seguenti operazioni in base al tipo di backend:
   - **Archivio dati Astra**:
     i. Selezionare la scheda **Astra Data Store**.
     ii. Selezionare il cluster di calcolo gestito e selezionare **Avanti**.
     iii. Confermare i dettagli del back-end e selezionare **Manage storage backend**.
   - **ONTAP**:
     i. Immettere le credenziali di amministratore di ONTAP e selezionare **Rivedi**.
     ii. Confermare i dettagli del back-end e selezionare **Manage** (Gestisci).

   Il backend viene visualizzato in `available` indicare nell'elenco le informazioni di riepilogo.

> Potrebbe essere necessario aggiornare la pagina per visualizzare il backend.

Aggiungi un bucket

L'aggiunta di provider di bucket di archivi di oggetti è essenziale se si desidera eseguire il backup delle applicazioni e dello storage persistente o se si desidera clonare le applicazioni tra cluster. Astra Control memorizza i backup o i cloni nei bucket dell'archivio di oggetti definiti dall'utente.

Quando si aggiunge un bucket, Astra Control contrassegna un bucket come indicatore di bucket predefinito. Il primo bucket creato diventa quello predefinito.

Non è necessario un bucket se si clonano la configurazione dell'applicazione e lo storage persistente sullo stesso cluster.

Utilizzare uno dei seguenti tipi di bucket:

- NetApp ONTAP S3
- NetApp StorageGRID S3

- Generico S3

> Sebbene Astra Control Center supporti Amazon S3 come provider di bucket S3 generico, Astra Control Center potrebbe non supportare tutti i vendor di archivi di oggetti che sostengono il supporto S3 di Amazon.

Per istruzioni su come aggiungere bucket utilizzando l'API Astra Control, vedere "Astra Automation e informazioni API".

Fasi

1. Nell'area di navigazione a sinistra, selezionare **Bucket**.
   a. Selezionare **Aggiungi**.
   b. Selezionare il tipo di bucket.

   > Quando si aggiunge un bucket, selezionare il bucket provider corretto e fornire le credenziali corrette per tale provider. Ad esempio, l'interfaccia utente accetta come tipo NetApp ONTAP S3 e accetta le credenziali StorageGRID; tuttavia, questo causerà l'errore di tutti i backup e ripristini futuri dell'applicazione che utilizzano questo bucket.

   c. Creare un nuovo nome di bucket o inserire un nome di bucket esistente e una descrizione opzionale.

   > Il nome e la descrizione del bucket vengono visualizzati come percorso di backup che è possibile scegliere in seguito quando si crea un backup. Il nome viene visualizzato anche durante la configurazione del criterio di protezione.

   d. Inserire il nome o l'indirizzo IP dell'endpoint S3.
   e. Se si desidera che questo bucket sia il bucket predefinito per tutti i backup, selezionare `Make this bucket the default bucket for this private cloud` opzione.

   > Questa opzione non viene visualizzata per il primo bucket creato.

   f. Continuare aggiungendo informazioni sulle credenziali.

Aggiungere le credenziali di accesso S3

Aggiungi credenziali di accesso S3 in qualsiasi momento.

Fasi

1. Dalla finestra di dialogo bucket, selezionare la scheda **Add** (Aggiungi) o **Use existing** (Usa esistente).
   a. Immettere un nome per la credenziale che la distingue dalle altre credenziali in Astra Control.
   b. Inserire l'ID di accesso e la chiave segreta incollando il contenuto dagli Appunti.

Quali sono le prossime novità?

Ora che hai effettuato l'accesso e aggiunto i cluster ad Astra Control Center, sei pronto per iniziare a utilizzare le funzionalità di gestione dei dati delle applicazioni di Astra Control Center.

- "Gestire gli utenti"
- "Inizia a gestire le app"

- "Proteggi le app"
- "Clonare le applicazioni"
- "Gestire le notifiche"
- "Connettersi a Cloud Insights"
- "Aggiungere un certificato TLS personalizzato"

Trova ulteriori informazioni

- "Utilizzare l'API di controllo Astra"
- "Problemi noti"

Prerequisiti per l'aggiunta di un cluster

Prima di aggiungere un cluster, assicurarsi che le condizioni preliminari siano soddisfatte. È inoltre necessario eseguire i controlli di idoneità per assicurarsi che il cluster sia pronto per essere aggiunto ad Astra Control Center.

Cosa serve prima di aggiungere un cluster

- Uno dei seguenti tipi di cluster:
  - Cluster che eseguono OpenShift 4.6, 4.7 o 4.8, con Astra Trident StorageClasses supportato da Astra Data Store o ONTAP 9.5 o versione successiva
  - Cluster che eseguono Rancher 2.5
  - Cluster che eseguono Kubernetes da 1.19 a 1.21 (inclusa la versione 1.21.x)

    Assicurarsi che i cluster dispongano di uno o più nodi di lavoro con almeno 1 GB di RAM disponibile per l'esecuzione dei servizi di telemetria.

    > Se si intende aggiungere un secondo cluster OpenShift 4.6, 4.7 o 4.8 come risorsa di calcolo gestita, assicurarsi che la funzione Astra Trident Volume Snapshot sia attivata. Vedi l'Astra Trident ufficiale "istruzioni" Per attivare e testare le istantanee dei volumi con Astra Trident.

- Il superuser e l'ID utente impostati sul sistema ONTAP di backup per eseguire il backup e il ripristino delle applicazioni con Centro di controllo Astra. Eseguire il seguente comando nella riga di comando di ONTAP: `export-policy rule modify -vserver <storage virtual machine name> -policyname <policy name> -ruleindex 1 -superuser sysm --anon 65534`
- Un tridente Astra `volumesnapshotclass` oggetto definito da un amministratore. Vedi Astra Trident "istruzioni" Per attivare e testare le istantanee dei volumi con Astra Trident.
- Assicurarsi di avere definito solo una singola classe di storage predefinita per il cluster Kubernetes.

Eseguire i controlli di idoneità

Eseguire i seguenti controlli di idoneità per assicurarsi che il cluster sia pronto per essere aggiunto ad Astra Control Center.

Fasi

1. Controllare la versione di Trident.

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

Se Trident esiste, l'output è simile a quanto segue:

```
NAME      VERSION
trident   21.04.0
```

Se Trident non esiste, viene visualizzato un output simile a quanto segue:

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

> Se Trident non è installato o se la versione installata non è la più recente, è necessario installare la versione più recente di Trident prima di procedere. Vedere "Documentazione di Trident" per istruzioni.

2. Controllare se le classi di storage utilizzano i driver Trident supportati. Il nome del provider deve essere `csi.trident.netapp.io`. Vedere il seguente esempio:

```
kubectl get sc
NAME                    PROVISIONER                    RECLAIMPOLICY
VOLUMEBINDINGMODE   ALLOWVOLUMEEXPANSION   AGE
ontap-gold (default)    csi.trident.netapp.io          Delete
Immediate           true                   5d23h
thin                    kubernetes.io/vsphere-volume   Delete
Immediate           false                  6d
```

Creare un kubeconfig con ruolo di amministratore

Prima di eseguire la procedura, assicurarsi di disporre dei seguenti elementi sul computer:

- `kubectl` v1.19 o versione successiva installata
- Un kubeconfig attivo con diritti di amministratore del cluster per il contesto attivo

Fasi

1. Creare un account di servizio come segue:

   a. Creare un file di account del servizio denominato `astracontrol-service-account.yaml`.

      Regolare il nome e lo spazio dei nomi in base alle esigenze. Se le modifiche vengono apportate qui, è necessario applicare le stesse modifiche nei passaggi seguenti.

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

+

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

a. Applicare l'account del servizio:

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

2. Concedere le autorizzazioni di amministratore del cluster come segue:

a. Creare un `ClusterRoleBinding` file chiamato `astracontrol-clusterrolebinding.yaml`.

Modificare i nomi e gli spazi dei nomi modificati quando si crea l'account del servizio, in base alle necessità.

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

+

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

a. Applicare l'associazione del ruolo del cluster:

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

3. Elencare i segreti dell'account di servizio, sostituendo `<context>` con il contesto corretto per l'installazione:

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

La fine dell'output dovrebbe essere simile a quanto segue:

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

Gli indici di ciascun elemento in `secrets` l'array inizia con 0. Nell'esempio precedente, l'indice per `astracontrol-service-account-dockercfg-vhz87` sarebbe 0 e l'indice per `astracontrol-service-account-token-r59kr` sarebbe 1. Nell'output, annotare l'indice del nome dell'account del servizio che contiene la parola "token".

4. Generare il kubeconfig come segue:
   a. Creare un `create-kubeconfig.sh` file. Sostituire `TOKEN_INDEX` all'inizio del seguente script con il valore corretto.

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

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

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

CONTEXT=$(kubectl config current-context)

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

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

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

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

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

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

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

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

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

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

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

b. Eseguire la sorgente dei comandi per applicarli al cluster Kubernetes.

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

5. (**opzionale**) rinominare il kubeconfig con un nome significativo per il cluster. Proteggi la tua credenziale del

cluster.

```
chmod 700 create-kubeconfig.sh
mv kubeconfig-sa.txt YOUR_CLUSTER_NAME_kubeconfig
```

Quali sono le prossime novità?

Ora che hai verificato che i prerequisiti sono stati soddisfatti, sei pronto "aggiungere un cluster".

Trova ulteriori informazioni

- "Documentazione di Trident"
- "Utilizzare l'API di controllo Astra"

Aggiungere un certificato TLS personalizzato

È possibile rimuovere il certificato TLS autofirmato esistente e sostituirlo con un certificato TLS firmato da un'autorità di certificazione (CA).

Di cosa hai bisogno

- 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

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>
....

#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>
....

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:

```
apiVersion: cert-manager.io/v1
kind: Certificate
metadata:
  name: <certificate-name>
  namespace: <ACC-deployment-namespace>
spec:
  secretName: <certificate-secret-name>
  duration: 2160h # 90d
  renewBefore: 360h # 15d
  dnsNames:
  - <astra.dnsname.example.com> #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>
....

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 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.containo.us -n <ACC-deployment-
namespace>
....

# tls:
#    options:
#      name: default
#    secretName: secure-testing-cert
#    store:
#      name: default

 tls:
    options:
      name: default
    secretName: <certificate-secret-name>
    store:
      name: default
```

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

Domande frequenti per Astra Control Center

Queste FAQ possono essere utili se stai cercando una risposta rapida a una domanda.

Panoramica

Le sezioni seguenti forniscono risposte ad alcune domande aggiuntive che potrebbero essere presentate durante l'utilizzo di Astra Control Center. Per ulteriori chiarimenti, contatta il sito astra.feedback@netapp.com

Accesso al centro di controllo Astra

Cos'è l'URL di Astra Control?

Astra Control Center utilizza l'autenticazione locale e un URL specifico per ciascun ambiente.

Per l'URL, in un browser, immettere il nome di dominio completo (FQDN) impostato nel campo spec.astraAddress nel file Astra_Control_Center_min.yaml custom resource Definition (CRD) al momento dell'installazione di Astra Control Center. Il messaggio di posta elettronica è il valore impostato nel campo spec.email nel CRD Astra_Control_Center_min.yaml.

Utilizzo la licenza di valutazione. Come si passa alla licenza completa?

È possibile passare facilmente a una licenza completa ottenendo il file di licenza NetApp (NLF).

Fasi

- Dalla barra di navigazione a sinistra, selezionare **account** > **licenza**.
- Selezionare **Aggiungi licenza**.
- Individuare il file di licenza scaricato e selezionare **Aggiungi**.

Utilizzo la licenza di valutazione. Posso comunque gestire le applicazioni?

Sì, puoi testare la funzionalità di gestione delle app con la licenza Evaluation.

Registrazione dei cluster Kubernetes

Devo aggiungere nodi di lavoro al cluster Kubernetes dopo l'aggiunta ad Astra Control. Cosa devo fare?

È possibile aggiungere nuovi nodi di lavoro ai pool esistenti. Questi verranno rilevati automaticamente da Astra Control. Se i nuovi nodi non sono visibili in Astra Control, controllare se i nuovi nodi di lavoro eseguono il tipo di immagine supportato. È inoltre possibile verificare lo stato dei nuovi nodi di lavoro utilizzando `kubectl get nodes` comando.

Come si annulla la gestione corretta di un cluster?

1. "Annulla la gestione delle applicazioni da Astra Control".
2. "Annullare la gestione del cluster da Astra Control".

Cosa succede alle mie applicazioni e ai miei dati dopo aver rimosso il cluster Kubernetes da Astra Control?

La rimozione di un cluster da Astra Control non apporta alcuna modifica alla configurazione del cluster (applicazioni e storage persistente). Eventuali snapshot di Astra Control o backup delle applicazioni su quel cluster non saranno disponibili per il ripristino. I backup persistenti dello storage creati da Astra Control rimangono all'interno di Astra Control, ma non sono disponibili per il ripristino.

> Rimuovere sempre un cluster da Astra Control prima di eliminarlo con altri metodi. L'eliminazione di un cluster utilizzando un altro tool mentre viene ancora gestito da Astra Control può causare problemi all'account Astra Control.

NetApp Trident verrà disinstallato quando rimuoverò un cluster Kubernetes da Astra Control?

Trident non verrà disinstallato da un cluster quando viene rimosso da Astra Control.

Gestione delle applicazioni

Astra Control può implementare un'applicazione?

Astra Control non implementa le applicazioni. Le applicazioni devono essere implementate all'esterno di Astra Control.

Cosa succede alle applicazioni dopo che non li gestisco da Astra Control?

Eventuali backup o snapshot esistenti verranno eliminati. Le applicazioni e i dati rimangono disponibili. Le operazioni di gestione dei dati non saranno disponibili per le applicazioni non gestite o per eventuali backup o snapshot ad esse appartenenti.

- Astra Control può gestire un’applicazione su storage non NetApp?*

No Mentre Astra Control è in grado di rilevare applicazioni che utilizzano storage non NetApp, non può gestire un’applicazione che utilizza storage non NetApp.

Dovrei gestire Astra Control da solo? No, non dovresti gestire Astra Control perché è un’applicazione di sistema.

Operazioni di gestione dei dati

Nel mio account sono presenti snapshot che non ho creato. Da dove sono venuti?

In alcune situazioni, Astra Control crea automaticamente uno snapshot come parte di un processo di backup, clonazione o ripristino.

La mia applicazione utilizza diversi PVS. Astra Control eseguirà snapshot e backup di tutti questi PVC?

Sì. Un’operazione snapshot su un’applicazione di Astra Control include l’istantanea di tutti i PVS associati ai PVC dell’applicazione.

È possibile gestire le snapshot acquisite da Astra Control direttamente attraverso un’interfaccia o un’archiviazione a oggetti diversa?

No Le snapshot e i backup eseguiti da Astra Control possono essere gestiti solo con Astra Control.