Configurare Astra Control Center

Astra Control Center

NetApp
November 21, 2023

Sommario

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.

Per ulteriori informazioni sul calcolo delle licenze, vedere "Licensing".

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

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.

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.

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 Astra Data Store, si desidera aggiungere il cluster di applicazioni Kubernetes che contiene applicazioni che utilizzano volumi forniti da Astra Data Store.

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 classe di storage predefinita del cluster. Se si modifica la classe di storage 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 una classe di storage predefinita diversa con `DefaultStorageClass` parametro.
- Utilizzare l'interfaccia utente Web di Astra Control per assegnare una classe di storage predefinita diversa. Vedere Modificare la classe di storage predefinita.

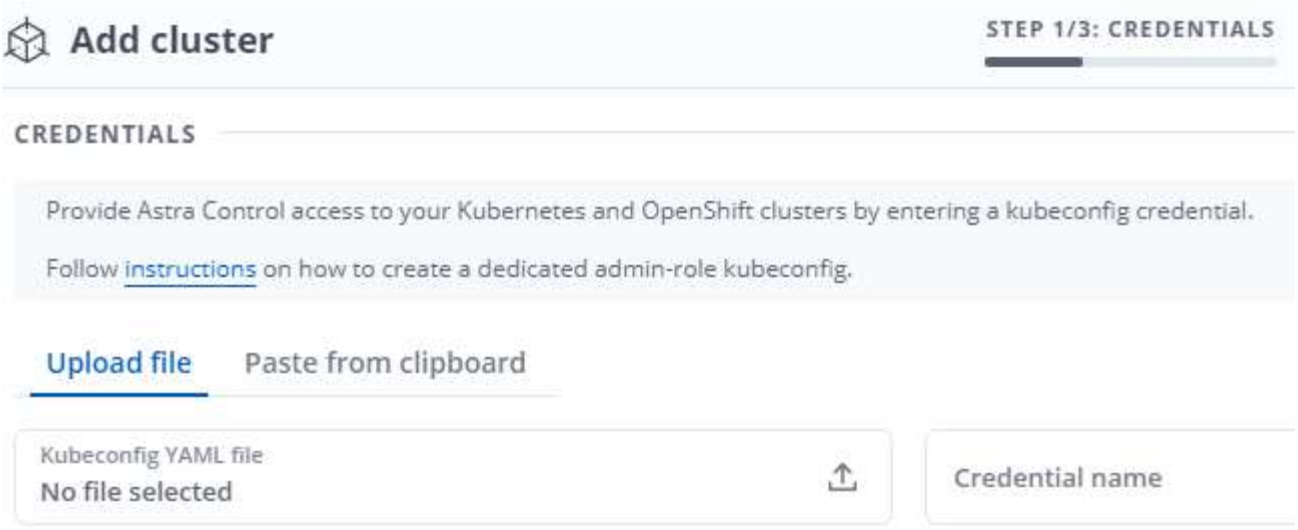
Di cosa hai bisogno

- 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.

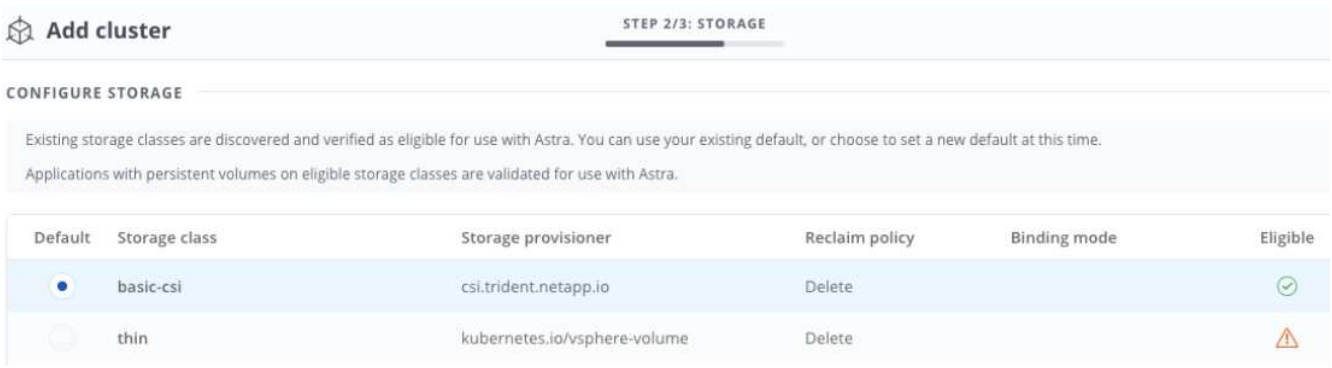
   Il `kubeconfig.yaml` il file deve includere **solo le credenziali del cluster per un cluster**.



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. È possibile implementare un backend di storage su un cluster gestito o utilizzare un backend di storage esistente.

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.

Ciò di cui hai bisogno per le implementazioni di Astra Data Store esistenti

- Hai aggiunto il cluster di applicazioni Kubernetes e il cluster di calcolo sottostante.

Dopo aver aggiunto il cluster di applicazioni Kubernetes per Astra Data Store ed essere gestito da Astra Control, 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.

Ciò di cui hai bisogno per le nuove implementazioni di Astra Data Store

- Lo hai fatto "ha caricato la versione del bundle di installazione che si intende implementare" In una posizione accessibile da Astra Control.
- È stato aggiunto il cluster Kubernetes che si intende utilizzare per la distribuzione.
- Hai caricato Licenza Astra Data Store Per l'implementazione in una posizione accessibile ad Astra Control.

Opzioni

- Implementare le risorse di storage
- Utilizzare un backend di storage esistente

Implementare le risorse di storage

È possibile implementare un nuovo archivio dati Astra e gestire il backend dello storage associato.

Fasi

1. Spostarsi dal menu Dashboard o Backend:
    - Da **Dashboard**: Dal riepilogo delle risorse, selezionare un collegamento dal riquadro Storage Backends e selezionare **Add** dalla sezione Backend.
    - Da **backend**:

i. Nell'area di navigazione a sinistra, selezionare **Backend**.

ii. Selezionare **Aggiungi**.

2. Selezionare l'opzione di implementazione **Astra Data Store** nella scheda **Deploy**.
3. Selezionare il pacchetto Astra Data Store da implementare:
   a. Immettere un nome per l'applicazione Astra Data Store.
   b. Scegli la versione di Astra Data Store che desideri implementare.

> Se non è stata ancora caricata la versione che si intende distribuire, è possibile utilizzare l'opzione **Add package** (Aggiungi pacchetto) o uscire dalla procedura guidata e utilizzarla "gestione dei pacchetti" per caricare il bundle di installazione.

4. Selezionare una licenza Astra Data Store precedentemente caricata oppure utilizzare l'opzione **Add License** (Aggiungi licenza) per caricare una licenza da utilizzare con l'applicazione.

> Le licenze di Astra Data Store con autorizzazioni complete sono associate al cluster Kubernetes e i cluster associati dovrebbero essere visualizzati automaticamente. Se non è presente alcun cluster gestito, è possibile selezionare l'opzione **Aggiungi un cluster** per aggiungerne uno alla gestione di Astra Control. Per le licenze Astra Data Store, se non è stata effettuata alcuna associazione tra la licenza e il cluster, è possibile definire questa associazione nella pagina successiva della procedura guidata.

5. Se non hai aggiunto un cluster Kubernetes alla gestione di Astra Control, devi farlo dalla pagina **Kubernetes cluster**. Selezionare un cluster esistente dall'elenco o selezionare **add the underlying cluster** (Aggiungi cluster sottostante) per aggiungere un cluster alla gestione di Astra Control.
6. Selezionare la dimensione del modello di implementazione per il cluster Kubernetes che fornirà le risorse per Astra Data Store.

> Quando si sceglie un modello, selezionare nodi più grandi con più memoria e core per carichi di lavoro più grandi o un numero maggiore di nodi per carichi di lavoro più piccoli. Selezionare un modello in base a quanto consentito dalla licenza. Ogni opzione di modello suggerisce il numero di nodi idonei che soddisfano lo schema di modello per memoria, core e capacità per ciascun nodo.

7. Configurare i nodi:
   a. Aggiungere un'etichetta di nodo per identificare il pool di nodi di lavoro che supporta questo cluster Astra Data Store.

> L'etichetta deve essere aggiunta a ogni singolo nodo del cluster che verrà utilizzato per l'implementazione di Astra Data Store prima dell'inizio dell'implementazione, altrimenti l'implementazione non avrà esito positivo.

   b. Configurare manualmente la capacità (GiB) per nodo o selezionare la capacità massima consentita per nodo.
   c. Configurare un numero massimo di nodi consentiti nel cluster o consentire il numero massimo di nodi nel cluster.
8. (Solo per le licenze complete di Astra Data Store) inserire la chiave dell'etichetta che si desidera utilizzare per i domini di protezione.

Creare almeno tre etichette univoche per la chiave per ciascun nodo. Ad esempio, se la chiave è `astra.datastore.protection.domain`, è possibile creare le seguenti etichette: `astra.datastore.protection.domain=domain1` ,`astra.datastore.protection.domain=domain2`, e. `astra.datastore.protection.domain=domain3`.

9. Configurare la rete di gestione:
   a. Inserire un indirizzo IP di gestione per la gestione interna di Astra Data Store che si trova sulla stessa sottorete degli indirizzi IP del nodo di lavoro.
   b. Scegliere di utilizzare la stessa scheda NIC per reti di gestione e dati o configurarle separatamente.
   c. Inserire il pool di indirizzi IP della rete dati, la subnet mask e il gateway per l'accesso allo storage.
10. Esaminare la configurazione e selezionare **Deploy** per iniziare l'installazione.

Risultato

Una volta completata l'installazione, il backend viene visualizzato in `available` indicare nell'elenco backend insieme alle informazioni sulle performance attive.

Potrebbe essere necessario aggiornare la pagina per visualizzare il backend.

Utilizzare un backend di storage esistente

Puoi portare un backend di storage ONTAP o Astra Data Store scoperto nella gestione del centro di controllo Astra.

Fasi

1. Spostarsi dal menu Dashboard o Backend:
   - Da **Dashboard**: Dal riepilogo delle risorse, selezionare un collegamento dal riquadro Storage Backends e selezionare **Add** dalla sezione Backend.
   - Da **backend**:
     i. Nell'area di navigazione a sinistra, selezionare **Backend**.
     ii. Selezionare **Gestisci** su un backend rilevato dal cluster gestito oppure selezionare **Aggiungi** per gestire un backend esistente aggiuntivo.
2. Selezionare la scheda **Usa esistente**.
3. Eseguire una delle seguenti operazioni in base al tipo di backend:
   - **Archivio dati Astra**:
     i. Selezionare **Astra Data Store**.
     ii. Selezionare il cluster di calcolo gestito e selezionare **Avanti**.
     iii. Confermare i dettagli del back-end e selezionare **Add storage backend**.
   - **ONTAP**:
     i. Selezionare **ONTAP**.
     ii. Immettere le credenziali di amministratore di ONTAP e selezionare **Rivedi**.
     iii. Confermare i dettagli del back-end e selezionare **Add storage backend**.

Risultato

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.

Modificare la classe di storage predefinita

È possibile modificare la classe di storage predefinita per un cluster.

Fasi

1. Nell'interfaccia utente Web di Astra Control Center, selezionare **Clusters**.
2. Nella pagina **Clusters**, selezionare il cluster che si desidera modificare.
3. Selezionare la scheda **Storage**.
4. Selezionare la categoria **classi di storage**.
5. Selezionare il menu **azioni** per la classe di storage che si desidera impostare come predefinita.
6. Selezionare **Imposta come predefinito**.

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 4.6.8, 4.7, 4.8 o 4.9
  - Cluster che eseguono Rancher 2.5.8, 2.5.9 o 2.6 con RKE1
  - Cluster che eseguono Kubernetes da 1.20 a 1.23
  - Cluster che eseguono VMware Tanzu Kubernetes Grid 1.4
  - Cluster che eseguono VMware Tanzu Kubernetes Grid Integrated Edition 1.12.2

    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.

- Astra Trident StorageClasses configurato con un "back-end di storage supportato" (richiesto per qualsiasi tipo di cluster)
- 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. (Facoltativo) se il cluster utilizza una policy di sicurezza del pod restrittiva che non consente la creazione di pod privilegiati o consente l'esecuzione di processi all'interno dei container del pod come utente root, creare una policy di sicurezza del pod personalizzata per il cluster che consenta ad Astra Control di creare e gestire i pod. Per istruzioni, vedere "Creare una policy di sicurezza pod personalizzata".
3. 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
```

4. 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".

5. 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
```

6. (**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.

Creare una policy di sicurezza pod personalizzata

Astra Control deve creare e gestire i pod Kubernetes sui cluster gestiti. Se il cluster utilizza una policy di sicurezza del pod restrittiva che non consente la creazione di pod con privilegi o l'esecuzione di processi all'interno dei container del pod come utente root, è necessario creare una policy di sicurezza del pod meno restrittiva per consentire ad Astra Control di creare e gestire questi pod.

Fasi

1. Creare un criterio di protezione pod per il cluster meno restrittivo di quello predefinito e salvarlo in un file. Ad esempio:

```
apiVersion: policy/v1beta1
kind: PodSecurityPolicy
metadata:
  name: astracontrol
  annotations:
    seccomp.security.alpha.kubernetes.io/allowedProfileNames: '*'
spec:
  privileged: true
  allowPrivilegeEscalation: true
  allowedCapabilities:
  - '*'
  volumes:
  - '*'
  hostNetwork: true
  hostPorts:
  - min: 0
    max: 65535
  hostIPC: true
  hostPID: true
  runAsUser:
    rule: 'RunAsAny'
  seLinux:
    rule: 'RunAsAny'
  supplementalGroups:
    rule: 'RunAsAny'
  fsGroup:
    rule: 'RunAsAny'
```

2. Creare un nuovo ruolo per la policy di sicurezza del pod.

```
kubectl-admin create role psp:astracontrol \
    --verb=use \
    --resource=podsecuritypolicy \
    --resource-name=astracontrol
```

3. Associare il nuovo ruolo all'account del servizio.

```
kubectl-admin create rolebinding default:psp:astracontrol \
    --role=psp:astracontrol \
    --serviceaccount=astracontrol-service-account:default
```