Aggiungere un cluster

Astra Control Service

NetApp
October 21, 2024

Sommario

Aggiungere un cluster a Astra Control Service

Dopo aver configurato l'ambiente, è possibile creare un cluster Kubernetes e aggiungerlo ad Astra Control Service. Questo consente di utilizzare Astra Control Service per proteggere le applicazioni sul cluster.

A seconda del tipo di cluster da aggiungere ad Astra Control Service, è necessario utilizzare diversi passaggi per aggiungere il cluster.

- "Aggiungere un cluster gestito da un provider pubblico ad Astra Control Service": Utilizzare questa procedura per aggiungere un cluster con un indirizzo IP pubblico e gestito da un provider cloud. È necessario disporre dell'account Service Principal, dell'account del servizio o dell'account utente per il provider cloud.
- "Aggiungere un cluster gestito da provider privato ad Astra Control Service": Utilizzare questa procedura per aggiungere un cluster che dispone di un indirizzo IP privato ed è gestito da un provider cloud. È necessario disporre dell'account Service Principal, dell'account del servizio o dell'account utente per il provider cloud.
- "Aggiungere un cluster pubblico autogestato ad Astra Control Service": Utilizzare questa procedura per aggiungere un cluster con un indirizzo IP pubblico e gestito dall'organizzazione. Sarà necessario creare un file kubeconfig per il cluster che si desidera aggiungere.
- "Aggiungere un cluster privato autogestato ad Astra Control Service": Utilizzare questa procedura per aggiungere un cluster con un indirizzo IP privato e gestito dall'organizzazione. Sarà necessario creare un file kubeconfig per il cluster che si desidera aggiungere.

Installa Astra Connector per gestire i cluster

Astra Connector è un software che risiede nei cluster gestiti e facilita la comunicazione tra il cluster gestito e Astra Control. Per i cluster gestiti mediante Astra Control Service, sono disponibili due versioni di Astra Connector:

- **Versione precedente del connettore Astra**: "Installare la versione precedente del connettore Astra" Sul tuo cluster, se intendi gestire il cluster con flussi di lavoro non nativi di Kubernetes.
- [Tech preview] * connettore dichiarativo di Kubernetes Astra*: "Installa Astra Connector per i cluster gestiti con flussi di lavoro Kubernetes dichiarativi" Sul cluster, se si intende gestire il cluster utilizzando flussi di lavoro Kubernetes dichiarativi. Dopo aver installato Astra Connector sul cluster, il cluster viene aggiunto automaticamente ad Astra Control.

> Il connettore dichiarativo Kubernetes Astra è disponibile solo come parte del programma Astra Control Early Adopter Program (EAP). Per informazioni sulla partecipazione al programma EAP, contattare il rappresentante commerciale NetApp di zona.

Installare la versione precedente del connettore Astra

Astra Control Service utilizza la versione precedente di Astra Connector per consentire la comunicazione tra Astra Control Service e i cluster privati gestiti con flussi di lavoro non nativi per Kubernetes. Devi installare Astra Connector su cluster privati che vuoi gestire con flussi di lavoro non nativi di Kubernetes.

La versione precedente di Astra Connector supporta i seguenti tipi di cluster privati gestiti con flussi di lavoro non nativi di Kubernetes:

- Amazon Elastic Kubernetes Service (EKS)
- Servizio Azure Kubernetes (AKS)
- Google Kubernetes Engine (GKE)
- Red Hat OpenShift Service su AWS (ROSA)
- ROSA con AWS PrivateLink
- Piattaforma Red Hat OpenShift Container all'interno dell'hotel

A proposito di questa attività

- Per eseguire questi passaggi, esegui questi comandi sul cluster privato che desideri gestire con Astra Control Service.
- Se si utilizza un host Bastion, eseguire questi comandi dalla riga di comando dell'host Bastion.

Prima di iniziare

- Devi accedere al cluster privato da gestire con Astra Control Service.
- Sono necessarie autorizzazioni di amministratore di Kubernetes per installare l'operatore Astra Connector sul cluster.

Fasi

1. Installa l'operatore Astra Connector precedente sul cluster privato che desideri gestire con flussi di lavoro non nativi di Kubernetes. Quando si esegue questo comando, lo spazio dei nomi `astra-connector-operator` viene creato e la configurazione viene applicata allo spazio dei nomi:

   ```
   kubectl apply -f https://github.com/NetApp/astra-connector-operator/releases/download/23.07.0-202310251519/astraconnector_operator.yaml
   ```

2. Verificare che l'operatore sia installato e pronto:

   ```
   kubectl get all -n astra-connector-operator
   ```

3. Ottieni un token API da Astra Control. Fare riferimento a. "Documentazione di Astra Automation" per istruzioni.
4. Creare lo spazio dei nomi astra-Connector:

   ```
   kubectl create ns astra-connector
   ```

5. Creare il file Astra Connector CR e assegnargli un nome `astra-connector-cr.yaml`. Aggiorna i valori tra parentesi <> per farli corrispondere all'ambiente Astra Control e alla configurazione del cluster:
   - **<ASTRA_CONTROL_SERVICE_URL>**: L'URL dell'interfaccia utente web del servizio di controllo Astra. Ad esempio:

```
https://astra.netapp.io
```

- **<ASTRA_CONTROL_SERVICE_API_TOKEN>**: Il token dell'API di controllo Astra ottenuto nel passaggio precedente.
- **<PRIVATE_AKS_CLUSTER_NAME>**: (Solo cluster AKS) - il nome del cluster del cluster privato Azure Kubernetes Service. Annullare il commento e popolare questa riga solo se si aggiunge un cluster AKS privato.
- **<ASTRA_CONTROL_ACCOUNT_ID>**: Ottenuto dall'interfaccia utente web Astra Control. Selezionare l'icona a forma di figura in alto a destra nella pagina e selezionare **accesso API**.

```yaml
apiVersion: netapp.astraconnector.com/v1
kind: AstraConnector
metadata:
  name: astra-connector
  namespace: astra-connector
spec:
  natssync-client:
    cloud-bridge-url: <ASTRA_CONTROL_SERVICE_URL>
  imageRegistry:
    name: theotw
    secret: ""
  astra:
    token: <ASTRA_CONTROL_SERVICE_API_TOKEN>
    #clusterName: <PRIVATE_AKS_CLUSTER_NAME>
    accountId: <ASTRA_CONTROL_ACCOUNT_ID>
    acceptEULA: yes
```

6. Dopo aver popolato il `astra-connector-cr.yaml` File con i valori corretti, applicare il CR:

```
kubectl apply -f astra-connector-cr.yaml
```

7. Verificare che il connettore Astra sia completamente distribuito:

```
kubectl get all -n astra-connector
```

8. Verifica che il cluster sia registrato con Astra Control:

```
kubectl get astraconnector -n astra-connector
```

L'output dovrebbe essere simile a quanto segue:

```
NAME                REGISTERED   ASTRACONNECTORID
STATUS
astra-connector     true         be475ae5-1511-4eaa-9b9e-712f09b0d065
Registered with Astra
```

Prendere nota di ASTRACONNECTORID; sarà necessario quando si aggiunge il cluster ad Astra Control.

Quali sono le prossime novità?

Una volta installato Astra Connector, puoi aggiungere il cluster privato ad Astra Control Service.

- "Aggiungere un cluster gestito da provider privato ad Astra Control Service": Utilizzare questa procedura per aggiungere un cluster che dispone di un indirizzo IP privato ed è gestito da un provider cloud. È necessario disporre dell'account Service Principal, dell'account del servizio o dell'account utente per il provider cloud.
- "Aggiungere un cluster privato autogestato ad Astra Control Service": Utilizzare questa procedura per aggiungere un cluster con un indirizzo IP privato e gestito dall'organizzazione. Sarà necessario creare un file kubeconfig per il cluster che si desidera aggiungere.

Per ulteriori informazioni

- "Aggiungere un cluster"

(Anteprima tecnica) Installa il connettore dichiarativo di Kubernetes Astra

I cluster gestiti utilizzando flussi di lavoro Kubernetes dichiarativi utilizzano Astra Connector per consentire la comunicazione tra il cluster gestito e Astra Control. Devi installare Astra Connector su tutti i cluster che verranno gestiti con flussi di lavoro Kubernetes dichiarativi.

Viene installato il connettore Astra dichiarativo di Kubernetes utilizzando i comandi di Kubernetes e i file Custom Resource (CR).

A proposito di questa attività

- Quando esegui questi passaggi, esegui questi comandi sul cluster che desideri gestire con Astra Control.
- Se si utilizza un host Bastion, eseguire questi comandi dalla riga di comando dell'host Bastion.

Prima di iniziare

- Devi accedere al cluster da gestire con Astra Control.
- Sono necessarie autorizzazioni di amministratore di Kubernetes per installare l'operatore Astra Connector sul cluster.

Se il cluster è configurato con l'imposizione dell'ammissione di sicurezza pod, che è l'impostazione predefinita per i cluster Kubernetes 1,25 e versioni successive, è necessario abilitare le restrizioni PSA sugli spazi dei nomi appropriati. Fare riferimento a. "Prepara il tuo ambiente per la gestione dei cluster utilizzando Astra Control" per istruzioni.

Fasi

1. Installare l'operatore Astra Connector sul cluster che si desidera gestire con flussi di lavoro Kubernetes dichiarativi. Quando si esegue questo comando, lo spazio dei nomi `astra-connector-operator` viene creato e la configurazione viene applicata allo spazio dei nomi:

```
kubectl apply -f https://github.com/NetApp/astra-connector-
operator/releases/download/24.02.0-
202403151353/astraconnector_operator.yaml
```

2. Verificare che l'operatore sia installato e pronto:

```
kubectl get all -n astra-connector-operator
```

3. Ottieni un token API da Astra Control. Fare riferimento a. "Documentazione di Astra Automation" per istruzioni.
4. Creare un segreto utilizzando il token. Sostituisci <API_TOKEN> con il token ricevuto da Astra Control:

```
kubectl create secret generic astra-token \
--from-literal=apiToken=<API_TOKEN> \
-n astra-connector
```

5. Crea un Docker Secret da usare per estrarre l'immagine di Astra Connector. Sostituire i valori tra parentesi <> con le informazioni dell'ambiente:

> Puoi trovare il <ASTRA_CONTROL_ACCOUNT_ID> nell'interfaccia utente web di Astra Control. Nell'interfaccia utente Web, selezionare l'icona della figura in alto a destra nella pagina e selezionare **accesso API**.

```
kubectl create secret docker-registry regcred \
--docker-username=<ASTRA_CONTROL_ACCOUNT_ID> \
--docker-password=<API_TOKEN> \
-n astra-connector \
--docker-server=cr.astra.netapp.io
```

6. Creare il file Astra Connector CR e assegnargli un nome `astra-connector-cr.yaml`. Aggiorna i valori tra parentesi <> per farli corrispondere all'ambiente Astra Control e alla configurazione del cluster:
   - <ASTRA_CONTROL_ACCOUNT_ID>: Ottenuto dall'interfaccia utente web Astra Control durante la fase precedente.
   - <CLUSTER_NAME>: Il nome che il cluster deve essere assegnato in Astra Control.
   - <ASTRA_CONTROL_URL>: L'URL dell'interfaccia utente web di Astra Control. Ad esempio:

```
https://astra.control.url
```

```yaml
apiVersion: astra.netapp.io/v1
kind: AstraConnector
metadata:
  name: astra-connector
  namespace: astra-connector
spec:
  astra:
    accountId: <ASTRA_CONTROL_ACCOUNT_ID>
    clusterName: <CLUSTER_NAME>
    #Only set `skipTLSValidation` to `true` when using the default
self-signed
    #certificate in a proof-of-concept environment.
    skipTLSValidation: false #Should be set to false in production
environments
    tokenRef: astra-token
  natsSyncClient:
    cloudBridgeURL: <ASTRA_CONTROL_HOST_URL>
  imageRegistry:
    name: cr.astra.netapp.io
    secret: regcred
```

7. Dopo aver popolato il `astra-connector-cr.yaml` File con i valori corretti, applicare il CR:

```
kubectl apply -n astra-connector -f astra-connector-cr.yaml
```

8. Verificare che il connettore Astra sia completamente distribuito:

```
kubectl get all -n astra-connector
```

9. Verifica che il cluster sia registrato con Astra Control:

```
kubectl get astraconnectors.astra.netapp.io -A
```

L'output dovrebbe essere simile a quanto segue:

```
NAMESPACE          NAME               REGISTERED   ASTRACONNECTORID
STATUS
astra-connector    astra-connector    true         00ac8-2cef-41ac-8777-
ed0583e   Registered with Astra
```

10. Verificare che il cluster compaia nell'elenco dei cluster gestiti nella pagina **cluster** dell'interfaccia utente Web Astra Control.

Aggiungere un cluster gestito dal provider

Aggiungere un cluster gestito da un provider pubblico ad Astra Control Service

Dopo aver configurato l'ambiente cloud, sei pronto per creare un cluster Kubernetes e aggiungerlo ad Astra Control Service.

- Creare un cluster Kubernetes
- Aggiungere il cluster ad Astra Control Service
- Modificare la classe di storage predefinita

Creare un cluster Kubernetes

Se non si dispone ancora di un cluster, è possibile crearne uno che soddisfi le esigenze "Requisiti del servizio Astra Control per Amazon Elastic Kubernetes Service (EKS)". Se non si dispone ancora di un cluster, è possibile crearne uno che soddisfi le esigenze "Requisiti del servizio Astra Control per Google Kubernetes Engine (GKE)". Se non si dispone ancora di un cluster, è possibile crearne uno che soddisfi le esigenze "Requisiti del servizio di controllo Astra per il servizio Azure Kubernetes (AKS) con Azure NetApp Files" oppure "Requisiti del servizio di controllo Astra per Azure Kubernetes Service (AKS) con dischi gestiti Azure".

> Astra Control Service supporta i cluster AKS che utilizzano Azure Active Directory (Azure ad) per l'autenticazione e la gestione delle identità. Quando si crea il cluster, seguire le istruzioni in "documentazione ufficiale" Per configurare il cluster per l'utilizzo di Azure ad. È necessario assicurarsi che i cluster soddisfino i requisiti per l'integrazione di Azure ad gestita da AKS.

Aggiungere il cluster ad Astra Control Service

Dopo aver effettuato l'accesso ad Astra Control Service, il primo passo è iniziare a gestire i cluster. Prima di aggiungere un cluster ad Astra Control Service, è necessario eseguire attività specifiche e assicurarsi che il cluster soddisfi determinati requisiti.

Quando gestisci i cluster Azure Kubernetes Service e Google Kubernetes Engine, tieni presente che hai due opzioni per l'installazione di Astra Control Provivisioner e la gestione del ciclo di vita:

- Puoi utilizzare Astra Control Service per gestire automaticamente il ciclo di vita di Astra Control Provisioner. Per fare questo, assicurati che Astra Trident non sia installato e Astra Control provisioner non sia abilitato nel cluster che vuoi gestire con Astra Control Service. In questo caso, Astra Control Service abilita automaticamente Astra Control Provisioner quando inizi a gestire il cluster e gli aggiornamenti di Astra Control Provisioner vengono gestiti automaticamente.
- Puoi gestire tu stesso il ciclo di vita di Astra Control Provisioner. A tale scopo, abilita Astra Control Provisioner sul cluster prima di gestire il cluster con Astra Control Service. In questo caso, Astra Control

Service rileva che Astra Control Provisioner è già abilitato e non lo reinstalla né gestisce gli aggiornamenti di Astra Control Provisioner. Fare riferimento a. "Abilita Astra Control Provisioner" Per i passaggi, abilita Astra Control provisioner.

Se gestisci i cluster di Amazon Web Services con Astra Control Service, se hai bisogno di backend di storage che possono essere utilizzati solo con Astra Control Provivisioner, devi abilitare Astra Control Provivisioner manualmente sul cluster prima di gestirlo con Astra Control Service. Fare riferimento a. "Abilita Astra Control Provisioner" Per informazioni su come attivare Astra Control Provivioner.

Prima di iniziare

Amazon Web Services

- Il file JSON contiene le credenziali dell'utente IAM che ha creato il cluster. "Scopri come creare un utente IAM".
- Per Amazon FSX per NetApp ONTAP è necessario Astra Control Provisioner. Se intendi usare Amazon FSX per NetApp ONTAP come back-end dello storage per il tuo cluster EKS, fai riferimento alle informazioni Astra Control Provivisioner nel "Requisiti del cluster EKS".
- (Facoltativo) se è necessario fornire `kubectl` Accesso ai comandi per un cluster ad altri utenti IAM che non sono i creatori del cluster, fare riferimento alle istruzioni in "Come posso fornire l'accesso ad altri utenti e ruoli IAM dopo la creazione del cluster in Amazon EKS?".
- Se intendi utilizzare NetApp Cloud Volumes ONTAP come backend di storage, devi configurare Cloud Volumes ONTAP per l'utilizzo con Amazon Web Services. Fare riferimento alla Cloud Volumes ONTAP "documentazione di installazione".

Microsoft Azure

- Il file JSON che contiene l'output della CLI di Azure deve essere presente al momento della creazione dell'entità del servizio. "Scopri come configurare un service principal".

  Avrai inoltre bisogno del tuo ID di abbonamento Azure, se non lo hai aggiunto al file JSON.

- Se si intende utilizzare NetApp Cloud Volumes ONTAP come back-end per lo storage, è necessario configurare Cloud Volumes ONTAP per l'utilizzo con Microsoft Azure. Fare riferimento alla Cloud Volumes ONTAP "documentazione di installazione".

Google Cloud

- È necessario disporre del file della chiave dell'account di servizio per un account di servizio che dispone delle autorizzazioni necessarie. "Scopri come configurare un account di servizio".
- Se si intende utilizzare NetApp Cloud Volumes ONTAP come back-end per lo storage, è necessario configurare Cloud Volumes ONTAP per l'utilizzo con Google Cloud. Fare riferimento alla Cloud Volumes ONTAP "documentazione di installazione".

Fasi

1. (Facoltativo) se stai aggiungendo un cluster Amazon EKS o vuoi gestire da solo l'installazione e gli aggiornamenti di Astra Control Provivisioner, abilita Astra Control Provivisioner sul cluster. Fare riferimento a. "Abilita Astra Control Provisioner" per i passaggi di abilitazione.
2. Aprire l'interfaccia utente Web di Astra Control Service in un browser.
3. Nella dashboard, selezionare **Manage Kubernetes cluster** (Gestisci cluster Kubernetes).

   Seguire le istruzioni per aggiungere il cluster.

4. **Provider**: Seleziona il tuo cloud provider e fornisci le credenziali necessarie per creare una nuova istanza di cloud oppure seleziona un'istanza di cloud esistente da utilizzare.
5. **Amazon Web Services**: Fornisci i dettagli del tuo account utente IAM Amazon Web Services caricando un file JSON o incollando il contenuto del file JSON dagli Appunti.

   Il file JSON deve contenere le credenziali dell'utente IAM che ha creato il cluster.

6. **Microsoft Azure**: Fornisci dettagli sull'entità del servizio Azure caricando un file JSON o incollando il contenuto di tale file JSON dagli Appunti.

   Il file JSON deve contenere l'output dell'interfaccia CLI di Azure al momento della creazione dell'entità del servizio. Può anche includere il tuo ID di abbonamento per aggiungerlo automaticamente ad Astra. In caso contrario, è necessario inserire manualmente l'ID dopo aver fornito il codice JSON.

7. **Google Cloud Platform**: Fornire il file delle chiavi dell'account di servizio caricando il file o incollando il contenuto dagli Appunti.

   Astra Control Service utilizza l'account del servizio per rilevare i cluster in esecuzione in Google Kubernetes Engine.

8. **Altro**: Questa scheda è destinata solo ai cluster a gestione automatica.
   a. **Nome istanza cloud**: Fornire un nome per la nuova istanza cloud che verrà creata quando si aggiunge questo cluster. Scopri di più "istanze cloud".
   b. Selezionare **Avanti**.

      Astra Control Service visualizza un elenco di cluster tra i quali è possibile scegliere.

   c. **Cluster**: Selezionare un cluster dall'elenco da aggiungere ad Astra Control Service.

      > Durante la selezione dall'elenco dei cluster, prestare attenzione alla colonna **Eligiblity**. Se un cluster è "non idoneo" o "parzialmente idoneo", passare il mouse sullo stato per determinare se si è verificato un problema con il cluster. Ad esempio, potrebbe identificare che il cluster non dispone di un nodo di lavoro.

   d. Selezionare **Avanti**.
   e. (Facoltativo) **Storage**: Facoltativamente, selezionare la classe di storage che si desidera utilizzare per impostazione predefinita per le applicazioni Kubernetes distribuite in questo cluster.
9. Per selezionare una nuova classe di storage predefinita per il cluster, attivare la casella di controllo **Assegna una nuova classe di storage predefinita**.
10. Selezionare una nuova classe di storage predefinita dall'elenco.

Ogni servizio di storage del cloud provider visualizza le seguenti informazioni su prezzo, performance e resilienza:

- Cloud Volumes Service per Google Cloud: Informazioni su prezzi, performance e resilienza
- Google Persistent Disk: Non sono disponibili informazioni su prezzi, performance o resilienza
- Azure NetApp Files: Informazioni su performance e resilienza
- Dischi gestiti Azure: Non sono disponibili informazioni su prezzi, performance o resilienza
- Amazon Elastic Block Store: Nessuna informazione su prezzi, performance o resilienza disponibile
- Amazon FSX per NetApp ONTAP: Nessuna informazione su prezzi, performance o resilienza disponibile
- NetApp Cloud Volumes ONTAP: Non sono disponibili informazioni su prezzi, performance o resilienza

Ogni classe di storage può utilizzare uno dei seguenti servizi:

- "Cloud Volumes Service per Google Cloud"
- "Disco persistente di Google"
  - "Azure NetApp Files"
  - "Dischi gestiti da Azure"
  - "Amazon Elastic Block Store"
  - "Amazon FSX per NetApp ONTAP"
  - "NetApp Cloud Volumes ONTAP"

  Scopri di più "Classi di storage per cluster Amazon Web Services". Scopri di più "Classi di storage per cluster AKS". Scopri di più "Classi di storage per cluster GKE".

  a. Selezionare **Avanti**.
  b. **Review & Approve** (Rivedi e approva): Verifica dei dettagli della configurazione.
  c. Selezionare **Add** per aggiungere il cluster ad Astra Control Service.

Risultato

Se si tratta del primo cluster aggiunto per questo provider cloud, Astra Control Service crea un archivio di oggetti per il provider cloud per i backup delle applicazioni in esecuzione sui cluster idonei. (Quando si aggiungono cluster successivi per questo provider cloud, non vengono creati ulteriori archivi di oggetti). Se è stata specificata una classe di storage predefinita, Astra Control Service imposta la classe di storage predefinita specificata. Per i cluster gestiti in Amazon Web Services o Google Cloud Platform, Astra Control Service crea anche un account admin sul cluster. Queste operazioni possono richiedere alcuni minuti.

Modificare la classe di storage predefinita

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

Modificare la classe di storage predefinita utilizzando Astra Control

È possibile modificare la classe di storage predefinita per un cluster da Astra Control. Se il cluster utilizza un servizio backend di storage precedentemente installato, potrebbe non essere possibile utilizzare questo metodo per modificare la classe di storage predefinita (l'azione **Set as default** non è selezionabile). In questo caso, è possibile Modificare la classe di storage predefinita utilizzando la riga di comando.

Fasi

1. Nell'interfaccia utente di Astra Control Service, 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**.

Modificare la classe di storage predefinita utilizzando la riga di comando

È possibile modificare la classe di storage predefinita per un cluster utilizzando i comandi Kubernetes. Questo metodo funziona indipendentemente dalla configurazione del cluster.

Fasi

1. Accedere al cluster Kubernetes.
2. Elencare le classi di storage nel cluster:

```
kubectl get storageclass
```

3. Rimuovere la designazione predefinita dalla classe di storage predefinita. Sostituire <SC_NAME> con il nome della classe di storage:

```
kubectl patch storageclass <SC_NAME> -p '{"metadata":
{"annotations":{"storageclass.kubernetes.io/is-default-
class":"false"}}}'
```

4. Contrassegna una classe di storage diversa come predefinita. Sostituire <SC_NAME> con il nome della classe di storage:

```
kubectl patch storageclass <SC_NAME> -p '{"metadata":
{"annotations":{"storageclass.kubernetes.io/is-default-class":"true"}}}'
```

5. Confermare la nuova classe di storage predefinita:

```
kubectl get storageclass
```

Aggiungere un cluster gestito da provider privato ad Astra Control Service

Puoi utilizzare Astra Control Service per gestire cluster privati di Google Kubernetes Engine (GKE). Queste istruzioni presuppongono che sia già stato creato un cluster AKS o OpenShift privato e che sia stato preparato un metodo sicuro per accedervi in remoto; per ulteriori informazioni sulla creazione e l'accesso a cluster AKS o OpenShift privati, fare riferimento alla seguente documentazione:

- "Documentazione di Azure per cluster AKS privati"
- "Documentazione di Azure per cluster OpenShift privati"

Puoi utilizzare Astra Control Service per gestire cluster privati Azure Kubernetes Service (AKS) e cluster privati Red Hat OpenShift in AKS. Queste istruzioni presuppongono che sia già stato creato un cluster AKS o OpenShift privato e che sia stato preparato un metodo sicuro per accedervi in remoto; per ulteriori informazioni sulla creazione e l'accesso a cluster AKS o OpenShift privati, fare riferimento alla seguente documentazione:

- "Documentazione di Azure per cluster AKS privati"
- "Documentazione di Azure per cluster OpenShift privati"

Puoi utilizzare Astra Control Service per gestire cluster privati Amazon Elastic Kubernetes Service (EKS). Queste istruzioni presuppongono che sia già stato creato un cluster EKS privato e che sia stato preparato un metodo sicuro per accedervi in remoto; per ulteriori informazioni sulla creazione e l'accesso a cluster EKS privati, fare riferimento a. "Documentazione Amazon EKS".

Per aggiungere il cluster privato ad Astra Control Service, è necessario eseguire le seguenti operazioni:

1. Installare il connettore Astra
2. Configurare lo storage persistente
3. Aggiungere il cluster gestito dal provider privato ad Astra Control Service

Installare il connettore Astra

Prima di aggiungere un cluster privato, devi installare Astra Connector sul cluster in modo che Astra Control possa comunicare con esso. Fare riferimento a. "Installa la versione precedente di Astra Connector per cluster privati gestiti con flussi di lavoro non nativi di Kubernetes" per istruzioni.

Configurare lo storage persistente

Configurare lo storage persistente per il cluster. Fare riferimento alla documentazione introduttiva per ulteriori informazioni sulla configurazione dello storage persistente:

- "Configurare Microsoft Azure con Azure NetApp Files"
- "Configurare Microsoft Azure con dischi gestiti Azure"
- "Configurare Amazon Web Services"
- "Configurare Google Cloud"

Aggiungere il cluster gestito dal provider privato ad Astra Control Service

È ora possibile aggiungere il cluster privato ad Astra Control Service.

Quando gestisci i cluster Azure Kubernetes Service e Google Kubernetes Engine, tieni presente che hai due opzioni per l'installazione di Astra Control Provivisioner e la gestione del ciclo di vita:

- Puoi utilizzare Astra Control Service per gestire automaticamente il ciclo di vita di Astra Control Provisioner. Per fare questo, assicurati che Astra Trident non sia installato e Astra Control provisioner non sia abilitato nel cluster che vuoi gestire con Astra Control Service. In questo caso, Astra Control Service abilita automaticamente Astra Control Provisioner quando inizi a gestire il cluster e gli aggiornamenti di Astra Control Provisioner vengono gestiti automaticamente.
- Puoi gestire tu stesso il ciclo di vita di Astra Control Provisioner. A tale scopo, abilita Astra Control Provisioner sul cluster prima di gestire il cluster con Astra Control Service. In questo caso, Astra Control Service rileva che Astra Control Provisioner è già abilitato e non lo reinstalla né gestisce gli aggiornamenti di Astra Control Provisioner. Fare riferimento a. "Abilita Astra Control Provisioner" Per i passaggi, abilita Astra Control provisioner.

Se gestisci i cluster di Amazon Web Services con Astra Control Service, se hai bisogno di backend di storage che possono essere utilizzati solo con Astra Control Provivisioner, devi abilitare Astra Control Provivisioner manualmente sul cluster prima di gestirlo con Astra Control Service. Fare riferimento a. "Abilita Astra Control Provisioner" Per informazioni su come attivare Astra Control Provivioner.

Prima di iniziare

Amazon Web Services

- Il file JSON contiene le credenziali dell'utente IAM che ha creato il cluster. "Scopri come creare un utente IAM".
- Per Amazon FSX per NetApp ONTAP è necessario Astra Control Provisioner. Se intendi usare Amazon FSX per NetApp ONTAP come back-end dello storage per il tuo cluster EKS, fai riferimento alle informazioni Astra Control Provivisioner nel "Requisiti del cluster EKS".
- (Facoltativo) se è necessario fornire `kubectl` Accesso ai comandi per un cluster ad altri utenti IAM che non sono i creatori del cluster, fare riferimento alle istruzioni in "Come posso fornire l'accesso ad altri utenti e ruoli IAM dopo la creazione del cluster in Amazon EKS?".
- Se intendi utilizzare NetApp Cloud Volumes ONTAP come backend di storage, devi configurare Cloud Volumes ONTAP per l'utilizzo con Amazon Web Services. Fare riferimento alla Cloud Volumes ONTAP "documentazione di installazione".

Microsoft Azure

- Il file JSON che contiene l'output della CLI di Azure deve essere presente al momento della creazione dell'entità del servizio. "Scopri come configurare un service principal".

  Avrai inoltre bisogno del tuo ID di abbonamento Azure, se non lo hai aggiunto al file JSON.

- Se si intende utilizzare NetApp Cloud Volumes ONTAP come back-end per lo storage, è necessario configurare Cloud Volumes ONTAP per l'utilizzo con Microsoft Azure. Fare riferimento alla Cloud Volumes ONTAP "documentazione di installazione".

Google Cloud

- È necessario disporre del file della chiave dell'account di servizio per un account di servizio che dispone delle autorizzazioni necessarie. "Scopri come configurare un account di servizio".
- Se il cluster è privato, il "reti autorizzate" Deve consentire l'indirizzo IP di Astra Control Service:

  52.188.218.166/32

- Se si intende utilizzare NetApp Cloud Volumes ONTAP come back-end per lo storage, è necessario configurare Cloud Volumes ONTAP per l'utilizzo con Google Cloud. Fare riferimento alla Cloud Volumes ONTAP "documentazione di installazione".

Fasi

1. (Facoltativo) se stai aggiungendo un cluster Amazon EKS o vuoi gestire da solo l'installazione e gli aggiornamenti di Astra Control Provivisioner, abilita Astra Control Provivisioner sul cluster. Fare riferimento a. "Abilita Astra Control Provisioner" per i passaggi di abilitazione.
2. Aprire l'interfaccia utente Web di Astra Control Service in un browser.
3. Nella dashboard, selezionare **Manage Kubernetes cluster** (Gestisci cluster Kubernetes).

   Seguire le istruzioni per aggiungere il cluster.

4. **Provider**: Seleziona il tuo cloud provider e fornisci le credenziali necessarie per creare una nuova istanza di cloud oppure seleziona un'istanza di cloud esistente da utilizzare.
5. **Amazon Web Services**: Fornisci i dettagli del tuo account utente IAM Amazon Web Services caricando un file JSON o incollando il contenuto del file JSON dagli Appunti.

Il file JSON deve contenere le credenziali dell'utente IAM che ha creato il cluster.

6. **Microsoft Azure**: Fornisci dettagli sull'entità del servizio Azure caricando un file JSON o incollando il contenuto di tale file JSON dagli Appunti.

   Il file JSON deve contenere l'output dell'interfaccia CLI di Azure al momento della creazione dell'entità del servizio. Può anche includere il tuo ID di abbonamento per aggiungerlo automaticamente ad Astra. In caso contrario, è necessario inserire manualmente l'ID dopo aver fornito il codice JSON.

7. **Google Cloud Platform**: Fornire il file delle chiavi dell'account di servizio caricando il file o incollando il contenuto dagli Appunti.

   Astra Control Service utilizza l'account del servizio per rilevare i cluster in esecuzione in Google Kubernetes Engine.

8. **Altro**: Questa scheda è destinata solo ai cluster a gestione automatica.
   a. **Nome istanza cloud**: Fornire un nome per la nuova istanza cloud che verrà creata quando si aggiunge questo cluster. Scopri di più "istanze cloud".
   b. Selezionare **Avanti**.

      Astra Control Service visualizza un elenco di cluster tra i quali è possibile scegliere.

   c. **Cluster**: Selezionare un cluster dall'elenco da aggiungere ad Astra Control Service.

      > Durante la selezione dall'elenco dei cluster, prestare attenzione alla colonna **Eligiblity**. Se un cluster è "non idoneo" o "parzialmente idoneo", passare il mouse sullo stato per determinare se si è verificato un problema con il cluster. Ad esempio, potrebbe identificare che il cluster non dispone di un nodo di lavoro.

9. Selezionare **Avanti**.
10. (Facoltativo) **Storage**: Facoltativamente, selezionare la classe di storage che si desidera utilizzare per impostazione predefinita per le applicazioni Kubernetes distribuite in questo cluster.
    a. Per selezionare una nuova classe di storage predefinita per il cluster, attivare la casella di controllo **Assegna una nuova classe di storage predefinita**.
    b. Selezionare una nuova classe di storage predefinita dall'elenco.

Ogni servizio di storage del cloud provider visualizza le seguenti informazioni su prezzo, performance e resilienza:

- Cloud Volumes Service per Google Cloud: Informazioni su prezzi, performance e resilienza
- Google Persistent Disk: Non sono disponibili informazioni su prezzi, performance o resilienza
- Azure NetApp Files: Informazioni su performance e resilienza
- Dischi gestiti Azure: Non sono disponibili informazioni su prezzi, performance o resilienza
- Amazon Elastic Block Store: Nessuna informazione su prezzi, performance o resilienza disponibile
- Amazon FSX per NetApp ONTAP: Nessuna informazione su prezzi, performance o resilienza disponibile
- NetApp Cloud Volumes ONTAP: Non sono disponibili informazioni su prezzi, performance o resilienza

Ogni classe di storage può utilizzare uno dei seguenti servizi:

- "Cloud Volumes Service per Google Cloud"
- "Disco persistente di Google"
- "Azure NetApp Files"
- "Dischi gestiti da Azure"
- "Amazon Elastic Block Store"
- "Amazon FSX per NetApp ONTAP"
- "NetApp Cloud Volumes ONTAP"

Scopri di più "Classi di storage per cluster Amazon Web Services". Scopri di più "Classi di storage per cluster AKS". Scopri di più "Classi di storage per cluster GKE".

c. Selezionare **Avanti**.

d. **Review & Approve** (Rivedi e approva): Verifica dei dettagli della configurazione.

e. Selezionare **Add** per aggiungere il cluster ad Astra Control Service.

Risultato

Se si tratta del primo cluster aggiunto per questo provider cloud, Astra Control Service crea un archivio di oggetti per il provider cloud per i backup delle applicazioni in esecuzione sui cluster idonei. (Quando si aggiungono cluster successivi per questo provider cloud, non vengono creati ulteriori archivi di oggetti). Se è stata specificata una classe di storage predefinita, Astra Control Service imposta la classe di storage predefinita specificata. Per i cluster gestiti in Amazon Web Services o Google Cloud Platform, Astra Control Service crea anche un account admin sul cluster. Queste operazioni possono richiedere alcuni minuti.

Modificare la classe di storage predefinita

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

Modificare la classe di storage predefinita utilizzando Astra Control

È possibile modificare la classe di storage predefinita per un cluster da Astra Control. Se il cluster utilizza un servizio backend di storage precedentemente installato, potrebbe non essere possibile utilizzare questo metodo per modificare la classe di storage predefinita (l'azione **Set as default** non è selezionabile). In questo caso, è possibile Modificare la classe di storage predefinita utilizzando la riga di comando.

Fasi

1. Nell'interfaccia utente di Astra Control Service, 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**.

Modificare la classe di storage predefinita utilizzando la riga di comando

È possibile modificare la classe di storage predefinita per un cluster utilizzando i comandi Kubernetes. Questo metodo funziona indipendentemente dalla configurazione del cluster.

Fasi

1. Accedere al cluster Kubernetes.
2. Elencare le classi di storage nel cluster:

```
kubectl get storageclass
```

3. Rimuovere la designazione predefinita dalla classe di storage predefinita. Sostituire <SC_NAME> con il nome della classe di storage:

```
kubectl patch storageclass <SC_NAME> -p '{"metadata":
{"annotations":{"storageclass.kubernetes.io/is-default-
class":"false"}}}'
```

4. Contrassegna una classe di storage diversa come predefinita. Sostituire <SC_NAME> con il nome della classe di storage:

```
kubectl patch storageclass <SC_NAME> -p '{"metadata":
{"annotations":{"storageclass.kubernetes.io/is-default-class":"true"}}}'
```

5. Confermare la nuova classe di storage predefinita:

```
kubectl get storageclass
```

Aggiungere un cluster a gestione automatica

Aggiungere un cluster pubblico autogestato ad Astra Control Service

Dopo aver configurato l'ambiente, è possibile creare un cluster Kubernetes e aggiungerlo ad Astra Control Service.

Un cluster a gestione automatica è un cluster che viene direttamente provisioning e gestione. Astra Control Service supporta cluster autogestiti eseguiti in un ambiente di cloud pubblico. È possibile aggiungere un cluster a gestione automatica ad Astra Control Service caricando un `kubeconfig.yaml` file. È necessario assicurarsi che il cluster soddisfi i requisiti descritti di seguito.

Distribuzioni Kubernetes supportate

È possibile utilizzare Astra Control Service per gestire i seguenti tipi di cluster pubblici e autogestati:

Distribuzione Kubernetes	Versioni supportate
Kubernetes (upstream)	da 1,27 a 1,29
Rancher Kubernetes Engine (RKE)	RKE 1: Versioni 1.24.17, 1.25.13, 1.26.8 con Rancher Manager 2.7.9 RKE 2: Versioni 1.23.16 e 1.24.13 con Rancher Manager 2.6.13 RKE 2: Versioni 1.24.17, 1.25.14, 1.26.9 con Rancher Manager 2.7.9
Red Hat OpenShift Container Platform	da 4,12 a 4,14

Queste istruzioni presuppongono che sia già stato creato un cluster a gestione automatica.

- Aggiungere il cluster ad Astra Control Service
- Modificare la classe di storage predefinita

Aggiungere il cluster ad Astra Control Service

Dopo aver effettuato l'accesso ad Astra Control Service, il primo passo è iniziare a gestire i cluster. Prima di aggiungere un cluster ad Astra Control Service, è necessario eseguire attività specifiche e assicurarsi che il cluster soddisfi determinati requisiti.

Prima di iniziare

Un cluster a gestione automatica è un cluster che viene direttamente provisioning e gestione. Astra Control Service supporta cluster autogestiti eseguiti in un ambiente di cloud pubblico. I tuoi cluster a gestione autonoma possono utilizzare Astra Control Provivisioner per interfacciarsi con i servizi storage NetApp, oppure possono utilizzare driver Container Storage Interface (CSI) per interfacciarsi con Amazon Elastic Block Store (EBS), Azure Managed Disks e Google Persistent Disk.

Astra Control Service supporta cluster autogestiti che utilizzano le seguenti distribuzioni Kubernetes:

- Red Hat OpenShift Container Platform
- Motore di rancher Kubernetes
- Kubernetes upstream

Il cluster a gestione automatica deve soddisfare i seguenti requisiti:

- Il cluster deve essere accessibile via Internet.
- Se si utilizza o si prevede di utilizzare lo storage abilitato con i driver CSI, i driver CSI appropriati devono essere installati sul cluster. Per ulteriori informazioni sull'utilizzo dei driver CSI per integrare lo storage, consultare la documentazione del servizio di storage.
- Si ha accesso al file cluster kubeconfig che include un solo elemento di contesto. Segui "queste istruzioni" per generare un file kubeconfig.
- Se si aggiunge il cluster utilizzando un file kubeconfig che fa riferimento a un'autorità di certificazione privata, aggiungere la riga seguente al `cluster` sezione del file kubeconfig. In questo modo si permette ad Astra Control di aggiungere il cluster:

  ```
  insecure-skip-tls-verify: true
  ```

- **Solo Rancher**: Quando si gestiscono i cluster di applicazioni in un ambiente Rancher, modificare il contesto predefinito del cluster di applicazioni nel file kubeconfig 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 Astra Control Provisioner**: Dovresti avere un Astra Control Provisioner configurato correttamente, inclusi i suoi componenti Astra Trident, per gestire i cluster.
  - **Rivedi i requisiti dell'ambiente Astra Trident**: Prima di installare o aggiornare Astra Control provisioner, consulta "frontend, backend e configurazioni host supportati".
  - **Abilitare la funzionalità Astra Control Provisioner**: Si consiglia vivamente di installare Astra Trident 23,10 o versione successiva e di abilitare "Astra Control Provivisioner funzionalità di storage avanzate". Nelle prossime release, Astra Control non supporterà Astra Trident se anche Astra Control Provisioner non è abilitato.
  - **Configurare un backend di archiviazione**: Deve essere presente almeno un backend di archiviazione "Configurato in Astra Trident" sul cluster.
  - **Configurare una classe di archiviazione**: Deve essere presente almeno una classe di archiviazione "Configurato in Astra Trident" sul cluster. Se è configurata una classe di archiviazione predefinita, assicurarsi che sia la classe di archiviazione **only** con l'annotazione predefinita.
  - **Configurare un controller snapshot volume e installare una classe snapshot volume**: "Installare un controller per lo snapshot del volume" In modo che le snapshot possano essere

create in Astra Control. "Creare" almeno uno `VolumeSnapshotClass` Utilizzando Astra Trident.

Fasi

1. Nella dashboard, selezionare **Manage Kubernetes cluster** (Gestisci cluster Kubernetes).

   Seguire le istruzioni per aggiungere il cluster.

2. **Provider**: Selezionare la scheda **Other** per aggiungere dettagli sul cluster a gestione automatica.
   a. **Altro**: Fornisci dettagli sul tuo cluster autogestito caricando un `kubeconfig.yaml` o incollando il contenuto di `kubeconfig.yaml` file dagli appunti.

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

3. **Nome credenziale**: Fornire un nome per la credenziale del cluster a gestione automatica che si sta caricando su Astra Control. Per impostazione predefinita, il nome della credenziale viene compilato automaticamente come nome del cluster.
4. **Private route identifier**: Questo campo può essere utilizzato solo con cluster privati.
5. Selezionare **Avanti**.
6. (Facoltativo) **Storage**: Facoltativamente, selezionare la classe di storage che si desidera utilizzare per impostazione predefinita per le applicazioni Kubernetes distribuite in questo cluster.
   a. Per selezionare una nuova classe di storage predefinita per il cluster, attivare la casella di controllo **Assegna una nuova classe di storage predefinita**.
   b. Selezionare una nuova classe di storage predefinita dall'elenco.

   > Ogni servizio di storage del cloud provider visualizza le seguenti informazioni su prezzo, performance e resilienza:
   >
   > - Cloud Volumes Service per Google Cloud: Informazioni su prezzi, performance e resilienza
   > - Google Persistent Disk: Non sono disponibili informazioni su prezzi, performance o resilienza
   > - Azure NetApp Files: Informazioni su performance e resilienza
   > - Dischi gestiti Azure: Non sono disponibili informazioni su prezzi, performance o resilienza
   > - Amazon Elastic Block Store: Nessuna informazione su prezzi, performance o resilienza disponibile
   > - Amazon FSX per NetApp ONTAP: Nessuna informazione su prezzi, performance o resilienza disponibile
   > - NetApp Cloud Volumes ONTAP: Non sono disponibili informazioni su prezzi, performance o resilienza

   Ogni classe di storage può utilizzare uno dei seguenti servizi:

   - "Cloud Volumes Service per Google Cloud"

- "Disco persistente di Google"
  - "Azure NetApp Files"
  - "Dischi gestiti da Azure"
  - "Amazon Elastic Block Store"
  - "Amazon FSX per NetApp ONTAP"
  - "NetApp Cloud Volumes ONTAP"

    Scopri di più "Classi di storage per cluster Amazon Web Services". Scopri di più "Classi di storage per cluster AKS". Scopri di più "Classi di storage per cluster GKE".

c. Selezionare **Avanti**.

d. **Review & Approve** (Rivedi e approva): Verifica dei dettagli della configurazione.

e. Selezionare **Add** per aggiungere il cluster ad Astra Control Service.

Modificare la classe di storage predefinita

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

Modificare la classe di storage predefinita utilizzando Astra Control

È possibile modificare la classe di storage predefinita per un cluster da Astra Control. Se il cluster utilizza un servizio backend di storage precedentemente installato, potrebbe non essere possibile utilizzare questo metodo per modificare la classe di storage predefinita (l'azione **Set as default** non è selezionabile). In questo caso, è possibile Modificare la classe di storage predefinita utilizzando la riga di comando.

Fasi

1. Nell'interfaccia utente di Astra Control Service, 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**.

Modificare la classe di storage predefinita utilizzando la riga di comando

È possibile modificare la classe di storage predefinita per un cluster utilizzando i comandi Kubernetes. Questo metodo funziona indipendentemente dalla configurazione del cluster.

Fasi

1. Accedere al cluster Kubernetes.
2. Elencare le classi di storage nel cluster:

```
kubectl get storageclass
```

3. Rimuovere la designazione predefinita dalla classe di storage predefinita. Sostituire <SC_NAME> con il nome della classe di storage:

```
kubectl patch storageclass <SC_NAME> -p '{"metadata":
{"annotations":{"storageclass.kubernetes.io/is-default-
class":"false"}}}'
```

4. Contrassegna una classe di storage diversa come predefinita. Sostituire <SC_NAME> con il nome della classe di storage:

```
kubectl patch storageclass <SC_NAME> -p '{"metadata":
{"annotations":{"storageclass.kubernetes.io/is-default-class":"true"}}}'
```

5. Confermare la nuova classe di storage predefinita:

```
kubectl get storageclass
```

Aggiungere un cluster privato autogestato ad Astra Control Service

Dopo aver configurato l'ambiente, è possibile creare un cluster Kubernetes e aggiungerlo ad Astra Control Service.

Un cluster a gestione automatica è un cluster che viene direttamente provisioning e gestione. Astra Control Service supporta cluster autogestiti eseguiti in un ambiente di cloud pubblico. È possibile aggiungere un cluster a gestione automatica ad Astra Control Service caricando un `kubeconfig.yaml` file. È necessario assicurarsi che il cluster soddisfi i requisiti descritti di seguito.

Distribuzioni Kubernetes supportate

È possibile utilizzare Astra Control Service per gestire i seguenti tipi di cluster privati a gestione automatica:

Distribuzione Kubernetes	Versioni supportate
Kubernetes (upstream)	da 1,27 a 1,29
Rancher Kubernetes Engine (RKE)	RKE 1: Versioni 1.24.17, 1.25.13, 1.26.8 con Rancher Manager 2.7.9 RKE 2: Versioni 1.23.16 e 1.24.13 con Rancher Manager 2.6.13 RKE 2: Versioni 1.24.17, 1.25.14, 1.26.9 con Rancher Manager 2.7.9
Red Hat OpenShift Container Platform	da 4,12 a 4,14

Queste istruzioni presuppongono che sia già stato creato un cluster privato e che sia stato preparato un metodo sicuro per accedervi in remoto.

Per aggiungere il cluster privato ad Astra Control Service, è necessario eseguire le seguenti operazioni:

1. Installare il connettore Astra
2. Configurare lo storage persistente

3. Aggiungere il cluster privato autogestato ad Astra Control Service

Installare il connettore Astra

Prima di aggiungere un cluster privato, devi installare Astra Connector sul cluster in modo che Astra Control possa comunicare con esso. Fare riferimento a. "Installa la versione precedente di Astra Connector per cluster privati gestiti con flussi di lavoro non nativi di Kubernetes" per istruzioni.

Configurare lo storage persistente

Configurare lo storage persistente per il cluster. Fare riferimento alla documentazione introduttiva per ulteriori informazioni sulla configurazione dello storage persistente:

- "Configurare Microsoft Azure con Azure NetApp Files"
- "Configurare Microsoft Azure con dischi gestiti Azure"
- "Configurare Amazon Web Services"
- "Configurare Google Cloud"

Aggiungere il cluster privato autogestato ad Astra Control Service

È ora possibile aggiungere il cluster privato ad Astra Control Service.

Prima di iniziare

Un cluster a gestione automatica è un cluster che viene direttamente provisioning e gestione. Astra Control Service supporta cluster autogestiti eseguiti in un ambiente di cloud pubblico. I tuoi cluster a gestione autonoma possono utilizzare Astra Control Provivisioner per interfacciarsi con i servizi storage NetApp, oppure possono utilizzare driver Container Storage Interface (CSI) per interfacciarsi con Amazon Elastic Block Store (EBS), Azure Managed Disks e Google Persistent Disk.

Astra Control Service supporta cluster autogestiti che utilizzano le seguenti distribuzioni Kubernetes:

- Red Hat OpenShift Container Platform
- Motore di rancher Kubernetes
- Kubernetes upstream

Il cluster a gestione automatica deve soddisfare i seguenti requisiti:

- Il cluster deve essere accessibile via Internet.
- Se si utilizza o si prevede di utilizzare lo storage abilitato con i driver CSI, i driver CSI appropriati devono essere installati sul cluster. Per ulteriori informazioni sull'utilizzo dei driver CSI per integrare lo storage, consultare la documentazione del servizio di storage.
- Si ha accesso al file cluster kubeconfig che include un solo elemento di contesto. Segui "queste istruzioni" per generare un file kubeconfig.
- Se si aggiunge il cluster utilizzando un file kubeconfig che fa riferimento a un'autorità di certificazione privata, aggiungere la riga seguente al `cluster` sezione del file kubeconfig. In questo modo si permette ad Astra Control di aggiungere il cluster:

  ```
  insecure-skip-tls-verify: true
  ```

- **Solo Rancher**: Quando si gestiscono i cluster di applicazioni in un ambiente Rancher, modificare il contesto predefinito del cluster di applicazioni nel file kubeconfig 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 Astra Control Provisioner**: Dovresti avere un Astra Control Provisioner configurato correttamente, inclusi i suoi componenti Astra Trident, per gestire i cluster.
  - **Rivedi i requisiti dell'ambiente Astra Trident**: Prima di installare o aggiornare Astra Control provisioner, consulta "frontend, backend e configurazioni host supportati".
  - **Abilitare la funzionalità Astra Control Provisioner**: Si consiglia vivamente di installare Astra Trident 23,10 o versione successiva e di abilitare "Astra Control Provivisioner funzionalità di storage avanzate". Nelle prossime release, Astra Control non supporterà Astra Trident se anche Astra Control Provisioner non è abilitato.
  - **Configurare un backend di archiviazione**: Deve essere presente almeno un backend di archiviazione "Configurato in Astra Trident" sul cluster.
  - **Configurare una classe di archiviazione**: Deve essere presente almeno una classe di archiviazione "Configurato in Astra Trident" sul cluster. Se è configurata una classe di archiviazione predefinita, assicurarsi che sia la classe di archiviazione **only** con l'annotazione predefinita.
  - **Configurare un controller snapshot volume e installare una classe snapshot volume**: "Installare un controller per lo snapshot del volume" In modo che le snapshot possano essere

create in Astra Control. "Creare" almeno uno `VolumeSnapshotClass` Utilizzando Astra Trident.

Fasi

1. Nella dashboard, selezionare **Manage Kubernetes cluster** (Gestisci cluster Kubernetes).

   Seguire le istruzioni per aggiungere il cluster.

2. **Provider**: Selezionare la scheda **Other** per aggiungere dettagli sul cluster a gestione automatica.
3. **Altro**: Fornisci dettagli sul tuo cluster autogestito caricando un `kubeconfig.yaml` o incollando il contenuto di `kubeconfig.yaml` file dagli appunti.

   > Se crei il tuo `kubeconfig` file, è necessario definire solo **un** elemento di contesto al suo interno. Fare riferimento a. "queste istruzioni" per informazioni sulla creazione `kubeconfig` file.

4. **Nome credenziale**: Fornire un nome per la credenziale del cluster a gestione automatica che si sta caricando su Astra Control. Per impostazione predefinita, il nome della credenziale viene compilato automaticamente come nome del cluster.
5. **Private route identifier**: Immettere l'identificativo di percorso privato, che è possibile ottenere da Astra Connector. Se si esegue una query su Astra Connector tramite `kubectl get astraconnector -n astra-connector` l'identificatore di route privato viene definito `ASTRACONNECTORID`.

   > L'identificatore di route privato è il nome associato al connettore Astra che consente la gestione di un cluster Kubernetes privato da parte di Astra. In questo contesto, un cluster privato è un cluster Kubernetes che non espone il proprio server API a Internet.

6. Selezionare **Avanti**.
7. (Facoltativo) **Storage**: Facoltativamente, selezionare la classe di storage che si desidera utilizzare per impostazione predefinita per le applicazioni Kubernetes distribuite in questo cluster.
   a. Per selezionare una nuova classe di storage predefinita per il cluster, attivare la casella di controllo **Assegna una nuova classe di storage predefinita**.
   b. Selezionare una nuova classe di storage predefinita dall'elenco.

Ogni servizio di storage del cloud provider visualizza le seguenti informazioni su prezzo, performance e resilienza:

- Cloud Volumes Service per Google Cloud: Informazioni su prezzi, performance e resilienza
- Google Persistent Disk: Non sono disponibili informazioni su prezzi, performance o resilienza
- Azure NetApp Files: Informazioni su performance e resilienza
- Dischi gestiti Azure: Non sono disponibili informazioni su prezzi, performance o resilienza
- Amazon Elastic Block Store: Nessuna informazione su prezzi, performance o resilienza disponibile
- Amazon FSX per NetApp ONTAP: Nessuna informazione su prezzi, performance o resilienza disponibile
- NetApp Cloud Volumes ONTAP: Non sono disponibili informazioni su prezzi, performance o resilienza

Ogni classe di storage può utilizzare uno dei seguenti servizi:

- "Cloud Volumes Service per Google Cloud"
- "Disco persistente di Google"
- "Azure NetApp Files"
- "Dischi gestiti da Azure"
- "Amazon Elastic Block Store"
- "Amazon FSX per NetApp ONTAP"
- "NetApp Cloud Volumes ONTAP"

Scopri di più "Classi di storage per cluster Amazon Web Services". Scopri di più "Classi di storage per cluster AKS". Scopri di più "Classi di storage per cluster GKE".

c. Selezionare **Avanti**.

d. **Review & Approve** (Rivedi e approva): Verifica dei dettagli della configurazione.

e. Selezionare **Add** per aggiungere il cluster ad Astra Control Service.

Modificare la classe di storage predefinita

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

Modificare la classe di storage predefinita utilizzando Astra Control

È possibile modificare la classe di storage predefinita per un cluster da Astra Control. Se il cluster utilizza un servizio backend di storage precedentemente installato, potrebbe non essere possibile utilizzare questo metodo per modificare la classe di storage predefinita (l'azione **Set as default** non è selezionabile). In questo caso, è possibile Modificare la classe di storage predefinita utilizzando la riga di comando.

Fasi

1. Nell'interfaccia utente di Astra Control Service, 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**.

Modificare la classe di storage predefinita utilizzando la riga di comando

È possibile modificare la classe di storage predefinita per un cluster utilizzando i comandi Kubernetes. Questo metodo funziona indipendentemente dalla configurazione del cluster.

Fasi

1. Accedere al cluster Kubernetes.
2. Elencare le classi di storage nel cluster:

```
kubectl get storageclass
```

3. Rimuovere la designazione predefinita dalla classe di storage predefinita. Sostituire <SC_NAME> con il nome della classe di storage:

```
kubectl patch storageclass <SC_NAME> -p '{"metadata":
{"annotations":{"storageclass.kubernetes.io/is-default-
class":"false"}}}'
```

4. Contrassegna una classe di storage diversa come predefinita. Sostituire <SC_NAME> con il nome della classe di storage:

```
kubectl patch storageclass <SC_NAME> -p '{"metadata":
{"annotations":{"storageclass.kubernetes.io/is-default-class":"true"}}}'
```

5. Confermare la nuova classe di storage predefinita:

```
kubectl get storageclass
```

Controllare la versione di Astra Trident

Per aggiungere un cluster a gestione autonoma che utilizzi Astra Control Provivisioner o Astra Trident per i servizi di storage, assicurati che la versione installata di Astra Trident sia la 23,10 o più recente.

Fasi

1. Determina la versione di Astra Trident che stai utilizzando:

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

Se Astra Trident è installato, viene visualizzato un output simile a quanto segue:

```
NAME      VERSION
trident   24.02.0
```

Se Astra Trident non è installato, viene visualizzato un output simile a quanto segue:

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

2. Effettuare una delle seguenti operazioni:
   - Se utilizzi Astra Trident 23,01 o versione precedente, utilizza questi elementi "istruzioni" Per effettuare l'aggiornamento a una versione più recente di Astra Trident prima di effettuare l'aggiornamento a Astra Control Provivisioner. È possibile "eseguire un aggiornamento diretto" A Astra Control Provivisioner 24,02 se il tuo Astra Trident si trova all'interno di una finestra a quattro release della versione 24,02. Ad esempio, puoi eseguire l'upgrade direttamente da Astra Trident 23,04 a Astra Control Provisioner 24,02.
   - Se stai eseguendo Astra Trident 23,10 o versione successiva, verifica che Astra Control provisioner sia stato "attivato". Astra Control Provisioner non funzionerà con le versioni di Astra Control Center precedenti alla 23,10. "Aggiorna Astra Control provisioner" In modo che abbia la stessa versione di Astra Control Center che stai effettuando l'aggiornamento per accedere alle funzionalità più recenti.
3. Assicurarsi che i pod siano in funzione:

```
kubectl get pods -n trident
```

4. Controllare se le classi di storage utilizzano i driver Astra Trident supportati. Il nome del provider deve essere `csi.trident.netapp.io`. Fare riferimento al seguente esempio:

```
kubectl get sc
```

Esempio di risposta:

```
NAME                    PROVISIONER                    RECLAIMPOLICY
VOLUMEBINDINGMODE   ALLOWVOLUMEEXPANSION   AGE
ontap-gold (default)    csi.trident.netapp.io          Delete
Immediate           true                   5d23h
```

Creare un file kubeconfig

È possibile aggiungere un cluster ad Astra Control Service utilizzando un file kubeconfig. A seconda del tipo di cluster che si desidera aggiungere, potrebbe essere necessario creare manualmente un file kubeconfig per il cluster utilizzando passaggi specifici.

- Creare un file kubeconfig per i cluster Amazon EKS
- Creare un file kubeconfig per Red Hat OpenShift Service su cluster AWS (ROSA)
- Creare un file kubeconfig per altri tipi di cluster

Creare un file kubeconfig per i cluster Amazon EKS

Segui queste istruzioni per creare un file kubeconfig e un token secret permanente per i cluster Amazon EKS. Per i cluster ospitati in EKS è necessario un token secret permanente.

Fasi

1. Seguire le istruzioni nella documentazione di Amazon per generare un file kubeconfig:

   "Creazione o aggiornamento di un file kubeconfig per un cluster Amazon EKS"

2. Creare un account di servizio come segue:

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

      Modificare il nome dell'account di servizio in base alle necessità. Lo spazio dei nomi `kube-system` è necessario per questi passaggi. Se si modifica il nome dell'account di servizio, è necessario apportare le stesse modifiche nei seguenti passaggi.

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

+

```yaml
apiVersion: v1
kind: ServiceAccount
metadata:
  name: astra-admin-account
  namespace: kube-system
```

3. Applicare l'account del servizio:

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

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

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

```yaml
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: astra-admin-binding
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: cluster-admin
subjects:
- kind: ServiceAccount
  name: astra-admin-account
  namespace: kube-system
```

5. Applicare l'associazione del ruolo del cluster:

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

6. Creare un file token secret dell'account di servizio chiamato `astracontrol-secret.yaml`.

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

```yaml
apiVersion: v1
kind: Secret
metadata:
  annotations:
    kubernetes.io/service-account.name: astra-admin-account
  name: astra-admin-account
  namespace: kube-system
type: kubernetes.io/service-account-token
```

7. Applicare il token secret:

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

8. Recuperare il token secret:

```
kubectl get secret astra-admin-account -n kube-system -o
jsonpath='{.data.token}' | base64 -d
```

9. Sostituire `user` Sezione del file kubeconfig AWS EKS con il token, come mostrato nell'esempio seguente:

```
user:
    token: k8s-aws-
v1.aHR0cHM6Ly9zdHMudXMtd2VzdC0yLmFtYXpvbmF3cy5jb20vP0FjdGlvbj1HZXRDYWxsZ
XJJZGVudGl0eSZWZXJzaW9uPTIwMTEtMDYtMTUmWC1BbXotQWxnb3JpdGhtPUFXUzQtSE1BQ
y1TSEEyNTYmWC1BbXotQ3JlZGVudGlhbD1BS0lBM1JEWDdKU0haWU9LSEQ2SyUyRjIwMjMwN
DAzJTJGdXMtd2VzdC0yJTJGc3RzJTJGYXdzNF9yZXF1ZXN0JlgtQW16LURhdGU9MjAyMzA0M
DNUMjA0MzQwWiZYLUFtei1FeHBpcmVzPTYwJlgtQW16LVNpZ25lZEhlYWRlcnM9aG9zdCUzQ
ngtazhzLWF3cy1pZCZYLUFtei1TaWduYXR1cmU9YjU4ZWM0NzdiM2NkZGYxNGRhNzU4MGI2Z
WQ2zY2NzI2YWIwM2UyNThjMjRhNTJjNmVhNjc4MTRlNjJkOTg2Mg
```

Creare un file kubeconfig per Red Hat OpenShift Service su cluster AWS (ROSA)

Segui queste istruzioni per creare un file kubeconfig per Red Hat OpenShift Service su cluster AWS (ROSA).

Fasi

1. Accedere al cluster ROSA.
2. Creare un account di servizio:

```
oc create sa astracontrol-service-account
```

3. Aggiungere un ruolo cluster:

```
oc adm policy add-cluster-role-to-user cluster-admin -z astracontrol-
service-account
```

4. Utilizzando l'esempio seguente, creare un file di configurazione segreto dell'account di servizio:

```
<strong>secret-astra-sa.yaml</strong>
```

```
apiVersion: v1
kind: Secret
metadata:
  name: secret-astracontrol-service-account
  annotations:
    kubernetes.io/service-account.name: "astracontrol-service-account"
type: kubernetes.io/service-account-token
```

5. Creare il segreto:

```
oc create -f secret-astra-sa.yaml
```

6. Modificare l'account di servizio creato e aggiungere il nome segreto dell'account del servizio Astra Control
   a. `secrets` sezione:

```
oc edit sa astracontrol-service-account
```

```
apiVersion: v1
imagePullSecrets:
- name: astracontrol-service-account-dockercfg-dvfcd
kind: ServiceAccount
metadata:
  creationTimestamp: "2023-08-04T04:18:30Z"
  name: astracontrol-service-account
  namespace: default
  resourceVersion: "169770"
  uid: 965fa151-923f-4fbd-9289-30cad15998ac
secrets:
- name: astracontrol-service-account-dockercfg-dvfcd
- name: secret-astracontrol-service-account ####ADD THIS ONLY####
```

7. 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-dvfcd"},
{ "name": "secret-astracontrol-service-account"}
]
```

Gli indici di ciascun elemento in `secrets` l'array inizia con 0. Nell'esempio precedente, l'indice per `astracontrol-service-account-dockercfg-dvfcd` sarebbe 0 e l'indice per `secret-astracontrol-service-account` sarebbe 1. Nell'output, annotare il numero dell'indice per il segreto dell'account del servizio. Questo numero di indice sarà necessario nella fase successiva.

8. 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
```

9. (Facoltativo) rinominare il kubeconfig con un nome significativo per il cluster.

```
mv kubeconfig-sa YOUR_CLUSTER_NAME_kubeconfig
```

Creare un file kubeconfig per altri tipi di cluster

Segui queste istruzioni per creare un file kubeconfig con ruolo limitato o esteso per i cluster Rancher, Upstream Kubernetes e Red Hat OpenShift.

Per i cluster gestiti utilizzando kubeconfig, è possibile creare un'autorizzazione limitata o un ruolo di amministratore di autorizzazioni esteso per Astra Control Service.

Questa procedura consente di creare una configurazione separata se uno dei seguenti scenari si applica al proprio ambiente:

- Si desidera limitare le autorizzazioni di Astra Control sui cluster gestiti
- Si utilizzano più contesti e non è possibile utilizzare il kubeconfig di Astra Control predefinito configurato durante l'installazione oppure un ruolo limitato con un singolo contesto non funziona nell'ambiente

Prima di iniziare

Prima di completare la procedura, assicurarsi di disporre dei seguenti elementi per il cluster che si desidera gestire:

- R "versione supportata" di kubectl è installato.
- Kubectl accesso al cluster che si intende aggiungere e gestire con Astra Control Service

> Per questa procedura, non è necessario l'accesso kubectl al cluster che esegue Astra Control Service.

- Un kubeconfig attivo per il cluster che si intende gestire con i diritti di amministratore del cluster per il contesto attivo

Fasi

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

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

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

   b. Applicare l'account del servizio:

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

2. Creare uno dei seguenti ruoli del cluster con autorizzazioni sufficienti per la gestione di un cluster da parte di Astra Control:

Ruolo cluster limitato

Questo ruolo contiene le autorizzazioni minime necessarie per gestire un cluster da Astra Control:

a. Creare un `ClusterRole` file chiamato, ad esempio, `astra-admin-account.yaml`.

```
<strong>astra-admin-account.yaml</strong>
```

```yaml
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: astra-admin-account
rules:

# Get, List, Create, and Update all resources
# Necessary to backup and restore all resources in an app
- apiGroups:
  - '*'
  resources:
  - '*'
  verbs:
  - get
  - list
  - create
  - patch

# Delete Resources
# Necessary for in-place restore and AppMirror failover
- apiGroups:
  - ""
  - apps
  - autoscaling
  - batch
  - crd.projectcalico.org
  - extensions
  - networking.k8s.io
  - policy
  - rbac.authorization.k8s.io
  - snapshot.storage.k8s.io
  - trident.netapp.io
  resources:
  - configmaps
  - cronjobs
  - daemonsets
  - deployments
```

```
    - horizontalpodautoscalers
    - ingresses
    - jobs
    - namespaces
    - networkpolicies
    - persistentvolumeclaims
    - poddisruptionbudgets
    - pods
    - podtemplates
    - replicasets
    - replicationcontrollers
    - replicationcontrollers/scale
    - rolebindings
    - roles
    - secrets
    - serviceaccounts
    - services
    - statefulsets
    - tridentmirrorrelationships
    - tridentsnapshotinfos
    - volumesnapshots
    - volumesnapshotcontents
    verbs:
    - delete

  # Watch resources
  # Necessary to monitor progress
  - apiGroups:
    - ""
    resources:
    - pods
    - replicationcontrollers
    - replicationcontrollers/scale
    verbs:
    - watch

  # Update resources
  - apiGroups:
    - ""
    - build.openshift.io
    - image.openshift.io
    resources:
    - builds/details
    - replicationcontrollers
    - replicationcontrollers/scale
    - imagestreams/layers
```

```
  - imagestreamtags
  - imagetags
  verbs:
  - update
```

b. (Solo per i cluster OpenShift) aggiungere quanto segue alla fine di `astra-admin-account.yaml` file:

```
# OpenShift security
- apiGroups:
  - security.openshift.io
  resources:
  - securitycontextconstraints
  verbs:
  - use
  - update
```

c. Applicare il ruolo del cluster:

```
kubectl apply -f astra-admin-account.yaml
```

Ruolo cluster esteso

Questo ruolo contiene autorizzazioni estese per un cluster da gestire con Astra Control. È possibile utilizzare questo ruolo se si utilizzano più contesti e non è possibile utilizzare il kubeconfig di Astra Control predefinito configurato durante l'installazione oppure se un ruolo limitato con un singolo contesto non funziona nell'ambiente:

> Quanto segue `ClusterRole` I passaggi sono un esempio generale di Kubernetes. Consultare la documentazione della distribuzione Kubernetes per istruzioni specifiche sull'ambiente in uso.

a. Creare un `ClusterRole` file chiamato, ad esempio, `astra-admin-account.yaml`.

```
<strong>astra-admin-account.yaml</strong>
```

```yaml
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: astra-admin-account
rules:
- apiGroups:
  - '*'
  resources:
  - '*'
  verbs:
  - '*'
- nonResourceURLs:
  - '*'
  verbs:
  - '*'
```

b. Applicare il ruolo del cluster:

```
kubectl apply -f astra-admin-account.yaml
```

3. Creare l'associazione del ruolo del cluster all'account del servizio per il ruolo del cluster:

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

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

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

b. Applicare l'associazione del ruolo del cluster:

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

4. Creare e applicare il token secret:
   a. Creare un file token secret chiamato `secret-astracontrol-service-account.yaml`.

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

```yaml
apiVersion: v1
kind: Secret
metadata:
  name: secret-astracontrol-service-account
  namespace: default
  annotations:
    kubernetes.io/service-account.name: "astracontrol-service-
account"
type: kubernetes.io/service-account-token
```

   b. Applicare il token secret:

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

5. Aggiungere il token secret all'account del servizio aggiungendo il nome a `secrets` array (l'ultima riga dell'esempio seguente):

```
kubectl edit sa astracontrol-service-account
```

```
apiVersion: v1
imagePullSecrets:
- name: astracontrol-service-account-dockercfg-48xhx
kind: ServiceAccount
metadata:
  annotations:
    kubectl.kubernetes.io/last-applied-configuration: |

{"apiVersion":"v1","kind":"ServiceAccount","metadata":{"annotations":{},
"name":"astracontrol-service-account","namespace":"default"}}
  creationTimestamp: "2023-06-14T15:25:45Z"
  name: astracontrol-service-account
  namespace: default
  resourceVersion: "2767069"
  uid: 2ce068c4-810e-4a96-ada3-49cbf9ec3f89
secrets:
- name: astracontrol-service-account-dockercfg-48xhx
<strong>- name: secret-astracontrol-service-account</strong>
```

6. 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-48xhx"},
{ "name": "secret-astracontrol-service-account"}
]
```

Gli indici di ciascun elemento in `secrets` l'array inizia con 0. Nell'esempio precedente, l'indice per `astracontrol-service-account-dockercfg-48xhx` sarebbe 0 e l'indice per `secret-astracontrol-service-account` sarebbe 1. Nell'output, annotare il numero dell'indice per il segreto dell'account del servizio. Questo numero di indice è necessario nel passaggio successivo.

7. Generare il kubeconfig come segue:
   a. Creare un `create-kubeconfig.sh` file.
   b. 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
```

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

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

8. (Facoltativo) rinominare il kubeconfig con un nome significativo per il cluster.

```
mv kubeconfig-sa YOUR_CLUSTER_NAME_kubeconfig
```