Configurare Astra Control Center

Astra Control Center

NetApp
April 25, 2024

Sommario

Configurare Astra Control Center

Aggiungere una licenza per Astra Control Center

Quando si installa Astra Control Center, è già installata una licenza di valutazione integrata. Se stai valutando Astra Control Center, puoi saltare questo passaggio.

È possibile aggiungere una nuova licenza utilizzando l'interfaccia utente di Astra Control o. "API di controllo Astra".

Le licenze di Astra Control Center misurano le risorse CPU utilizzando le unità CPU di Kubernetes e tengono conto delle risorse CPU assegnate ai nodi di lavoro di tutti i cluster Kubernetes gestiti. Le licenze si basano sull'utilizzo di vCPU. Per ulteriori informazioni sul calcolo delle licenze, fare riferimento a. "Licensing".

> 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.

> Per aggiornare una licenza di valutazione o una licenza completa, fare riferimento a. "Aggiornare una licenza esistente".

Prima di iniziare

- Accesso a un'istanza di Astra Control Center appena installata.
- Autorizzazioni per il ruolo di amministratore.
- R "File di licenza NetApp" (NLF).

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 e non si inviano dati a AutoSupport, assicurarsi di memorizzare l'ID account per evitare la perdita di dati in caso di guasto del centro di controllo Astra.

Abilita Astra Control Provisioner

Astra Trident le versioni 23,10 e successive includono la possibilità di utilizzare Astra Control Provivisioner, che consente agli utenti dotati di licenza Astra Control di accedere a funzionalità avanzate di provisioning dello storage. Astra Control Provisioner fornisce questa funzionalità estesa oltre alle funzionalità standard basate su CSI Astra Trident.

In arrivo gli update di Astra Control, Astra Control Provivisioner sostituirà Astra Trident come provisioner di storage e orchestrator e sarà obbligatorio per l'utilizzo di Astra Control. Per questo motivo, si consiglia vivamente agli utenti di Astra Control di attivare Astra Control Provisioner. Astra Trident continuerà a rimanere open source e ad essere rilasciato, mantenuto, supportato e aggiornato con le nuove CSI e altre funzionalità di NetApp.

A proposito di questa attività

È necessario seguire questa procedura se si è un utente di Astra Control Center con licenza e si sta cercando di utilizzare la funzionalità di Astra Control Provisioner. Devi seguire questa procedura anche se sei un utente di Astra Trident e desideri utilizzare le funzionalità aggiuntive fornite da Astra Control Provisioner senza utilizzare Astra Control.

Per ogni caso, la funzionalità di provisioning non è abilitata per impostazione predefinita in Astra Trident 24,02 e deve essere abilitata.

Prima di iniziare

Se stai abilitando Astra Control provisioner, esegui prima quanto segue:

Utenti di Astra Control Provisioners con Astra Control Center

- **Ottenere una licenza Astra Control Center**: È necessario un "Licenza Astra Control Center" Per abilitare Astra Control Provisioner e accedere alle funzionalità fornite.
- **Installa o esegui l'aggiornamento ad Astra Control Center 23,10 o versione successiva**: Se intendi utilizzare la funzionalità più recente di Astra Control Center (24,02) 24,02 con Astra Control.
- **Confirmi che il tuo cluster ha un'architettura di sistema AMD64**: L'immagine Astra Control Provivisioner è fornita in entrambe le architetture CPU AMD64 e ARM64, ma solo AMD64 è supportato da Astra Control Center.
- **Ottenere un account del Servizio di controllo Astra per l'accesso al Registro di sistema**: Se si intende utilizzare il Registro di sistema di controllo Astra piuttosto che il Sito di supporto NetApp per scaricare l'immagine del revisioner di controllo Astra, completare la registrazione per un "Account Astra Control Service". Dopo aver completato e inviato il modulo e creato un account BlueXP, riceverai un'email di benvenuto con Astra Control Service.
- **Se Astra Trident è installato, conferma che la sua versione si trovi all'interno di una finestra a quattro release**: Puoi eseguire un aggiornamento diretto a Astra Trident 24,02 con Astra Control Provisioner 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 24,02.

Solo utenti di Astra Control provisioner

- **Ottenere una licenza Astra Control Center**: È necessario un "Licenza Astra Control Center" Per abilitare Astra Control Provisioner e accedere alle funzionalità fornite.
- **Se Astra Trident è installato, conferma che la sua versione si trovi all'interno di una finestra a quattro release**: Puoi eseguire un aggiornamento diretto a Astra Trident 24,02 con Astra Control Provisioner 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 24,02.
- **Prendi un account Astra Control Service per l'accesso al Registro di sistema**: Per scaricare le immagini di Astra Control provisioner, è necessario accedere al Registro di sistema. Per iniziare, completa la registrazione per un "Account Astra Control Service". Dopo aver completato e inviato il modulo e creato un account BlueXP, riceverai un'email di benvenuto con Astra Control Service.

(Fase 1) ottenere l’immagine di Astra Control provisioner

Gli utenti di Astra Control Center possono ottenere l’immagine di Astra Control Provisioner utilizzando il Registro di sistema di Astra Control o il metodo del sito di supporto di NetApp. Gli utenti di Astra Trident che desiderano utilizzare Astra Control Protivisioner senza Astra Control devono utilizzare il metodo del Registro di sistema.

Registro delle immagini di Astra Control

> È possibile utilizzare Podman invece di Docker per i comandi di questa procedura. Se si utilizza un ambiente Windows, si consiglia di utilizzare PowerShell.

1. Accedere al Registro di sistema dell'immagine di controllo Astra di NetApp:
   a. Accedere all'interfaccia utente Web di Astra Control Service e selezionare l'icona raffigurata in alto a destra nella pagina.
   b. Selezionare **API access**.
   c. Annotare l'ID account.
   d. Nella stessa pagina, selezionare **generate API token**, copiare la stringa del token API negli Appunti e salvarla nell'editor.
   e. Accedere al registro Astra Control utilizzando il metodo preferito:

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

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

2. (Solo registri personalizzati) attenersi alla seguente procedura per spostare l'immagine nel registro personalizzato. Se non si utilizza un registro, seguire i passaggi dell'operatore Trident nel "sezione successiva".
   a. Estrarre l'immagine di Astra Control provisioner dal Registro di sistema:

> L'immagine estratta non supporta più piattaforme e supporta solo la stessa piattaforma dell'host che ha estratto l'immagine, ad esempio Linux AMD64.

```
docker pull cr.astra.netapp.io/astra/trident-acp:24.02.0
--platform <cluster platform>
```

Esempio:

```
docker pull cr.astra.netapp.io/astra/trident-acp:24.02.0 --platform
linux/amd64
```

   a. Contrassegnare l'immagine:

```
docker tag cr.astra.netapp.io/astra/trident-acp:24.02.0
<my_custom_registry>/trident-acp:24.02.0
```

b. Inviare l'immagine al registro personalizzato:

```
docker push <my_custom_registry>/trident-acp:24.02.0
```

> È possibile utilizzare la copia di Crane come alternativa all'esecuzione dei seguenti comandi di Docker:
> `crane copy cr.astra.netapp.io/astra/trident-acp:24.02.0 <my_custom_registry>/trident-acp:24.02.0`

Sito di supporto NetApp

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

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

```
openssl dgst -sha256 -verify certs/AstraControlCenterDockerImages-public.pub -signature certs/trident-acp-[version].tar.sig trident-acp-[version].tar
```

3. Caricare l'immagine di Astra Control provisioner:

```
docker load < trident-acp-24.02.0.tar
```

Risposta:

```
Loaded image: trident-acp:24.02.0-linux-amd64
```

4. Contrassegnare l'immagine:

```
docker tag trident-acp:24.02.0-linux-amd64 <my_custom_registry>/trident-acp:24.02.0
```

5. Inviare l'immagine al registro personalizzato:

```
docker push <my_custom_registry>/trident-acp:24.02.0
```

(Fase 2) attiva Astra Control Provisioner in Astra Trident

Determinare se il metodo di installazione originale ha utilizzato un "Operatore (manualmente o con Helm) o tridentctl" e completare i passaggi appropriati in base al metodo originale.

Operatore Astra Trident

1. "Scaricare il programma di installazione di Astra Trident ed estrarlo".
2. Completa questi passaggi se non hai ancora installato Astra Trident o se hai rimosso l'operatore dall'implementazione originale di Astra Trident:
   a. Creare il CRD:

   ```
   kubectl create -f
   deploy/crds/trident.netapp.io_tridentorchestrators_crd_post1.16.y
   aml
   ```

   b. Creare lo spazio dei nomi tridente (`kubectl create namespace trident`) o confermare che lo spazio dei nomi tridente esiste ancora (`kubectl get all -n trident`). Se lo spazio dei nomi è stato rimosso, crearlo di nuovo.
3. Aggiorna Astra Trident alla versione 24.02.0:

   > Per i cluster che eseguono Kubernetes 1.24 o versioni precedenti, utilizzare `bundle_pre_1_25.yaml`. Per i cluster che eseguono Kubernetes 1.25 o versioni successive, utilizzare `bundle_post_1_25.yaml`.

   ```
   kubectl -n trident apply -f trident-installer/deploy/<bundle-
   name.yaml>
   ```

4. Verificare che Astra Trident sia in esecuzione:

   ```
   kubectl get torc -n trident
   ```

   Risposta:

   ```
   NAME      AGE
   trident   21m
   ```

5. se si dispone di un registro che utilizza segreti, creare un segreto da utilizzare per estrarre l'immagine di Astra Control Provisioner:

   ```
   kubectl create secret docker-registry <secret_name> -n trident
   --docker-server=<my_custom_registry> --docker-username=<username>
   --docker-password=<token>
   ```

6. Modificare il TridentOrchestrator CR e apportare le seguenti modifiche:

```
kubectl edit torc trident -n trident
```

a. Impostare una posizione del Registro di sistema personalizzata per l'immagine Astra Trident o estrarla dal Registro di sistema Astra Control (`tridentImage: <my_custom_registry>/trident:24.02.0` oppure `tridentImage: netapp/trident:24.02.0`).

b. Abilita Astra Control Provisioner (`enableACP: true`).

c. Impostare la posizione del Registro di sistema personalizzata per l'immagine Astra Control Provivioner o estrarla dal Registro di sistema Astra Control (`acpImage: <my_custom_registry>/trident-acp:24.02.0` oppure `acpImage: cr.astra.netapp.io/astra/trident-acp:24.02.0`).

d. Se stabilito segreti di estrazione delle immagini in precedenza, è possibile impostarle qui (`imagePullSecrets: - <secret_name>`). Usare lo stesso nome segreto che hai stabilito nei passaggi precedenti.

```
apiVersion: trident.netapp.io/v1
kind: TridentOrchestrator
metadata:
  name: trident
spec:
  debug: true
  namespace: trident
  tridentImage: <registry>/trident:24.02.0
  enableACP: true
  acpImage: <registry>/trident-acp:24.02.0
  imagePullSecrets:
  - <secret_name>
```

7. Salvare e uscire dal file. Il processo di distribuzione si avvia automaticamente.
8. Verificare che l'operatore, la distribuzione e i replicaset siano stati creati.

```
kubectl get all -n trident
```

> In un cluster Kubernetes dovrebbe esserci solo **un'istanza** dell'operatore. Non creare implementazioni multiple dell'operatore Astra Trident.

9. Verificare `trident-acp` il container è in esecuzione e così `acpVersion` è `24.02.0` con stato di `Installed`:

```
kubectl get torc -o yaml
```

Risposta:

```
status:
  acpVersion: 24.02.0
  currentInstallationParams:
    ...
    acpImage: <registry>/trident-acp:24.02.0
    enableACP: "true"
    ...
  ...
  status: Installed
```

tridentctl

1. "Scaricare il programma di installazione di Astra Trident ed estrarlo".
2. "Se disponi già di un Astra Trident, disinstallarlo dal cluster che lo ospita".
3. Installa Astra Trident con Astra Control Provisioner abilitato (`--enable-acp=true`):

```
./tridentctl -n trident install --enable-acp=true --acp
-image=mycustomregistry/trident-acp:24.02
```

4. Confermare che Astra Control Provisioner è stato abilitato:

```
./tridentctl -n trident version
```

Risposta:

```
+----------------+----------------+-------------+ | SERVER VERSION |
CLIENT VERSION | ACP VERSION | +----------------+----------------
+-------------+ | 24.02.0 | 24.02.0 | 24.02.0. | +----------------
+----------------+-------------+
```

Timone

1. Se hai installato Astra Trident 23.07.1 o una versione precedente, "disinstallazione" l'operatore e gli altri componenti.
2. Se il cluster Kubernetes esegue la versione 1,24 o precedente, elimina psp:

```
kubectl delete psp tridentoperatorpod
```

3. Aggiungere il repository Astra Trident Helm:

```
helm repo add netapp-trident https://netapp.github.io/trident-helm-
chart
```

4. Aggiornare il grafico Helm:

```
helm repo update netapp-trident
```

Risposta:

```
Hang tight while we grab the latest from your chart repositories...
...Successfully got an update from the "netapp-trident" chart
repository
Update Complete. □Happy Helming!□
```

5. Elencare le immagini:

```
./tridentctl images -n trident
```

Risposta:

```
| v1.28.0            | netapp/trident:24.02.0|
|                    | docker.io/netapp/trident-autosupport:24.02|
|                    | registry.k8s.io/sig-storage/csi-
provisioner:v4.0.0|
|                    | registry.k8s.io/sig-storage/csi-
attacher:v4.5.0|
|                    | registry.k8s.io/sig-storage/csi-
resizer:v1.9.3|
|                    | registry.k8s.io/sig-storage/csi-
snapshotter:v6.3.3|
|                    | registry.k8s.io/sig-storage/csi-node-driver-
registrar:v2.10.0 |
|                    | netapp/trident-operator:24.02.0 (optional)
```

6. Assicurarsi che l'operatore di tridente 24.02.0 sia disponibile:

```
helm search repo netapp-trident/trident-operator --versions
```

Risposta:

```
NAME                            CHART VERSION   APP VERSION
DESCRIPTION
netapp-trident/trident-operator 100.2402.0      24.02.0         A
```

7. Utilizzare `helm install` ed eseguire una delle seguenti opzioni che includono queste impostazioni:
   - Un nome per la posizione di distribuzione
   - La versione di Astra Trident
   - Il nome dell'immagine di Astra Control provisioner
   - Il flag per abilitare il provisioner
   - (Facoltativo) percorso del Registro di sistema locale. Se si utilizza un registro locale, il "Immagini Trident" Può trovarsi in un registro o in registri diversi, ma tutte le immagini CSI devono trovarsi nello stesso registro.
   - Il namespace Trident

Opzioni

- Immagini senza registro

```
helm install trident netapp-trident/trident-operator --version
100.2402.0 --set acpImage=cr.astra.netapp.io/astra/trident-acp:24.02.0
--set enableACP=true --set operatorImage=netapp/trident-
operator:24.02.0 --set
tridentAutosupportImage=docker.io/netapp/trident-autosupport:24.02
--set tridentImage=netapp/trident:24.02.0 --namespace trident
```

- Immagini in uno o più registri

```
helm install trident netapp-trident/trident-operator --version
100.2402.0 --set acpImage=<your-registry>:<acp image> --set
enableACP=true --set imageRegistry=<your-registry>/sig-storage --set
operatorImage=netapp/trident-operator:24.02.0 --set
tridentAutosupportImage=docker.io/netapp/trident-autosupport:24.02
--set tridentImage=netapp/trident:24.02.0 --namespace trident
```

È possibile utilizzare `helm list` per rivedere i dettagli dell'installazione, ad esempio nome, spazio dei nomi, grafico, stato, versione dell'applicazione, e numero di revisione.

Se hai problemi nell'implementazione di Trident utilizzando Helm, esegui questo comando per disinstallare completamente Astra Trident:

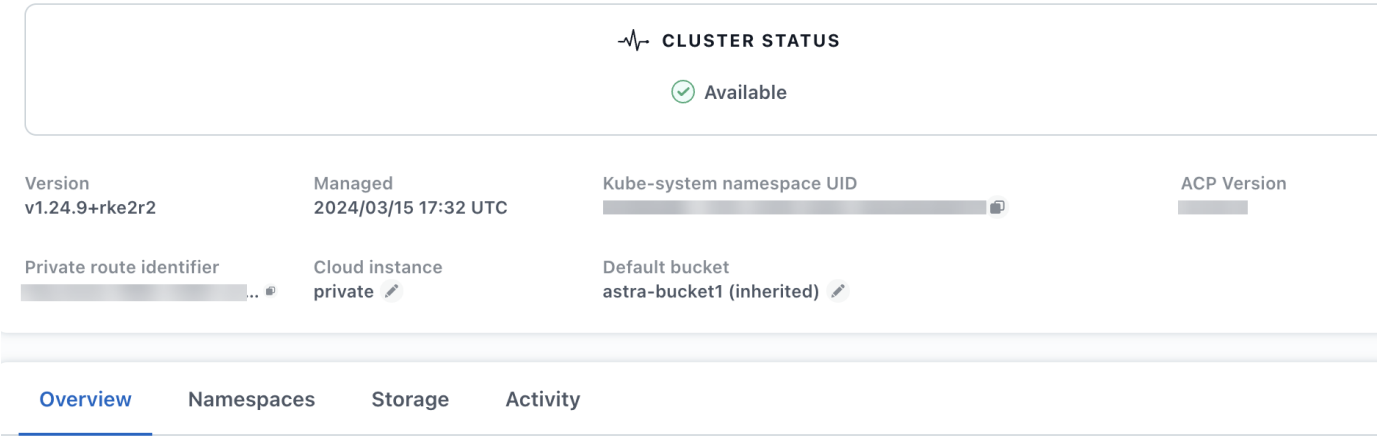
```
./tridentctl uninstall -n trident
```

Non fare "Rimuovere completamente i CRD Astra Trident" Come parte della disinstallazione prima di tentare di attivare nuovamente Astra Control Provivisioner.

Risultato

La funzionalità Astra Control Provisioner è abilitata ed è possibile utilizzare qualsiasi funzionalità disponibile per la versione in esecuzione.

(Solo per gli utenti di Astra Control Center) dopo l'installazione di Astra Control provisioner, il cluster che ospita il provisioner nell'interfaccia utente di Astra Control Center mostrerà un `ACP version` piuttosto che `Trident version` campo e numero della versione installata corrente.



Per ulteriori informazioni

- "Documentazione sugli aggiornamenti di Astra Trident"

Prepara il tuo ambiente per la gestione dei cluster utilizzando Astra Control

Prima di aggiungere un cluster, assicurarsi che siano soddisfatte le seguenti condizioni preliminari. È inoltre necessario eseguire controlli di idoneità per assicurarsi che il cluster sia pronto per essere aggiunto ad Astra Control Center e creare ruoli cluster kubeconfig secondo necessità.

Astra Control consente di aggiungere cluster gestiti da risorse personalizzate (CR) o kubeconfig, a seconda dell'ambiente e delle preferenze.

Prima di iniziare

- **Soddisfare i requisiti ambientali**: Il vostro ambiente soddisfa "requisiti dell'ambiente operativo" Per Astra Control Center.
- **Configura nodi di lavoro**: Assicurarsi che "configurare i nodi di lavoro" nel cluster con i driver di storage appropriati, in modo che i pod possano interagire con lo storage backend.
- **Abilita restrizioni PSA**: Se il cluster ha abilitato l'applicazione di accesso di sicurezza pod, che è standard per i cluster Kubernetes 1,25 e versioni successive, è necessario abilitare le restrizioni PSA nei seguenti spazi dei nomi:

- `netapp-acc-operator` spazio dei nomi:

```
kubectl label --overwrite ns netapp-acc-operator pod-
security.kubernetes.io/enforce=privileged
```

- `netapp monitoring` spazio dei nomi:

```
kubectl label --overwrite ns netapp-monitoring pod-
security.kubernetes.io/enforce=privileged
```

- **Credenziali ONTAP**: Per eseguire il backup e il ripristino delle applicazioni con il centro di controllo Astra sono necessarie le credenziali ONTAP e un ID utente e un superutente impostati sul sistema ONTAP di backup.

  Eseguire i seguenti comandi nella riga di comando di ONTAP:

```
export-policy rule modify -vserver <storage virtual machine name>
-policyname <policy name> -ruleindex 1 -superuser sys
export-policy rule modify -vserver <storage virtual machine name>
-policyname <policy name> -ruleindex 1 -anon 65534
```

- **Requisiti dei cluster gestiti da kubeconfig**: Questi requisiti sono specifici per i cluster di app gestiti da kubeconfig.
  - **Rendere accessibile kubeconfig**: Si ha accesso al "default cluster kubeconfig" quello "la configurazione è stata eseguita durante l'installazione".
  - **Considerazioni sull'autorità di certificazione**: Se si aggiunge il cluster utilizzando un file kubeconfig che fa riferimento a un'autorità di certificazione (CA) privata, aggiungere la seguente riga 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.

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. Determina la versione di Astra Trident che stai utilizzando:

```
kubectl get tridentversion -n trident
```

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

```
NAME      VERSION
trident   24.02.0
```

Se Astra Trident non esiste, viene visualizzato un output simile al seguente:

```
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 tutti i pod (inclusi `trident-acp`) in esecuzione:

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

4. Determinare se le classi di storage utilizzano i driver Astra Trident supportati. Il nome del provider deve

essere `csi.trident.netapp.io`. Vedere il 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 ruolo cluster kubeconfig

Per i cluster gestiti utilizzando kubeconfig, è possibile creare un'autorizzazione limitata o un ruolo di amministratore di autorizzazioni esteso per Astra Control Center. Questa procedura non è necessaria per la configurazione di Astra Control Center, in quanto è già stata configurata una configurazione come parte di "processo di installazione".

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:

- kubectl v1.23 o versione successiva installata
- Accesso kubectl al cluster che si intende aggiungere e gestire con Astra Control Center

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

- 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
```

(Anteprima tecnica) Installa Astra Connector per cluster gestiti

I cluster gestiti da Astra Control Center utilizzano Astra Connector per consentire la comunicazione tra il cluster gestito e Astra Control Center. Devi installare Astra Connector su tutti i cluster che desideri gestire.

Installare il connettore Astra

Installi Astra Connector 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. Installa l'operatore Astra Connector sul cluster che desideri gestire con Astra Control. 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

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.

> 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.

Prima di iniziare

- Prima di aggiungere un cluster, esaminare ed eseguire le operazioni necessarie "attività prerequisite".
- Se stai utilizzando un driver SAN ONTAP, assicurati che multipath sia abilitato su tutti i tuoi cluster Kubernetes.

Fasi

1. Spostarsi dal menu Dashboard o Clusters:
   - Da **Dashboard** in Resource Summary (Riepilogo risorse), selezionare **Add** (Aggiungi) dal pannello Clusters (Clusters).
   - Nell'area di navigazione a sinistra, selezionare **Clusters**, quindi selezionare **Add Cluster** (Aggiungi cluster) dalla pagina Clusters (Cluster).
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. Fare riferimento a. "Documentazione Kubernetes" per informazioni sulla creazione `kubeconfig` file. Se hai creato un kubeconfig per un ruolo cluster limitato utilizzando "questo processo", assicurarsi di caricare o incollare il kubeconfig in questa fase.

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

> Scegli una classe di storage configurata in Astra Control Provivisioner e supportata dallo storage ONTAP.

6. Esaminare le informazioni e, se tutto sembra buono, selezionare **Aggiungi**.

Risultato

Il cluster passa allo stato **Discovering** e quindi passa a **Healthy**. Ora stai gestendo il cluster con 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, è necessario esaminare i registri dell'operatore di monitoraggio.

Abilitare l'autenticazione su un backend di storage ONTAP

Il centro di controllo Astra offre due modalità di autenticazione di un backend ONTAP:

- **Autenticazione basata su credenziali**: Nome utente e password di un utente ONTAP con le autorizzazioni richieste. Per garantire la massima compatibilità con le versioni di ONTAP, è necessario utilizzare un ruolo di accesso di sicurezza predefinito, ad esempio admin o vsadmin.
- **Autenticazione basata su certificato**: Il centro di controllo Astra può anche comunicare con un cluster ONTAP utilizzando un certificato installato sul back-end. Utilizzare il certificato client, la chiave e il certificato CA attendibile, se utilizzato (consigliato).

È possibile aggiornare in seguito i back-end esistenti per passare da un tipo di autenticazione a un altro metodo. È supportato un solo metodo di autenticazione alla volta.

Abilitare l'autenticazione basata su credenziali

Astra Control Center richiede le credenziali per un cluster con ambito `admin` Per comunicare con il backend ONTAP. È necessario utilizzare ruoli standard predefiniti, ad esempio `admin`. Ciò garantisce la compatibilità con le future release di ONTAP che potrebbero esporre le API delle funzionalità da utilizzare nelle future release di Astra Control Center.

> Un ruolo di accesso di sicurezza personalizzato può essere creato e utilizzato con Astra Control Center, ma non è consigliato.

Un esempio di definizione di backend è simile al seguente:

```
{
  "version": 1,
  "backendName": "ExampleBackend",
  "storageDriverName": "ontap-nas",
  "managementLIF": "10.0.0.1",
  "dataLIF": "10.0.0.2",
  "svm": "svm_nfs",
  "username": "admin",
  "password": "secret"
}
```

La definizione di backend è l'unica posizione in cui le credenziali vengono memorizzate in testo normale. La creazione o l'aggiornamento di un backend è l'unico passaggio che richiede la conoscenza delle credenziali. Pertanto, si tratta di un'operazione di sola amministrazione, che deve essere eseguita da Kubernetes o dall'amministratore dello storage.

Abilitare l'autenticazione basata su certificato

Il centro di controllo Astra può utilizzare i certificati per comunicare con i backend ONTAP nuovi ed esistenti. Inserire le seguenti informazioni nella definizione di backend.

- `clientCertificate`: Certificato del client.
- `clientPrivateKey`: Chiave privata associata.
- `trustedCACertificate`: Certificato CA attendibile. Se si utilizza una CA attendibile, è necessario fornire questo parametro. Questa operazione può essere ignorata se non viene utilizzata alcuna CA attendibile.

È possibile utilizzare uno dei seguenti tipi di certificati:

- Certificato autofirmato
- Certificato di terze parti

Abilitare l'autenticazione con un certificato autofirmato

Un workflow tipico prevede i seguenti passaggi.

Fasi

1. Generare un certificato e una chiave del client. Durante la generazione, impostare il nome comune (CN) sull'utente ONTAP per l'autenticazione come.

   ```
   openssl req -x509 -nodes -days 1095 -newkey rsa:2048 -keyout k8senv.key
   -out k8senv.pem -subj "/C=US/ST=NC/L=RTP/O=NetApp/CN=<common-name>"
   ```

2. Installare il certificato client di tipo `client-ca` E sul cluster ONTAP.

```
security certificate install -type client-ca -cert-name <certificate-
name> -vserver <vserver-name>
security ssl modify -vserver <vserver-name> -client-enabled true
```

3. Verificare che il ruolo di accesso di sicurezza di ONTAP supporti il metodo di autenticazione del certificato.

```
security login create -user-or-group-name vsadmin -application ontapi
-authentication-method cert -vserver <vserver-name>
security login create -user-or-group-name vsadmin -application http
-authentication-method cert -vserver <vserver-name>
```

4. Verificare l'autenticazione utilizzando il certificato generato. Sostituire <LIF di gestione ONTAP> e <vserver name> con l'IP LIF di gestione e il nome SVM. Assicurarsi che la politica di servizio di LIF sia impostata su `default-data-management`.

```
curl -X POST -Lk https://<ONTAP-Management-
LIF>/servlets/netapp.servlets.admin.XMLrequest_filer --key k8senv.key
--cert ~/k8senv.pem -d '<?xml version="1.0" encoding="UTF-8"?><netapp
xmlns=http://www.netapp.com/filer/admin version="1.21" vfiler="<vserver-
name>"><vserver-get></vserver-get></netapp>
```

5. Utilizzando i valori ottenuti dal passaggio precedente, aggiungere il backend di storage nell'interfaccia utente di Astra Control Center.

Abilitare l'autenticazione con un certificato di terze parti

Se si dispone di un certificato di terze parti, è possibile configurare l'autenticazione basata su certificato con questa procedura.

Fasi

1. Generare la chiave privata e la CSR:

```
openssl req -new -newkey rsa:4096 -nodes -sha256 -subj "/" -outform pem
-out ontap_cert_request.csr -keyout ontap_cert_request.key -addext
"subjectAltName = DNS:<ONTAP_CLUSTER_FQDN_NAME>,IP:<ONTAP_MGMT_IP>"
```

2. Passare la CSR alla CA di Windows (CA di terze parti) e rilasciare il certificato firmato.
3. Scarica il certificato firmato e chiamalo `ontap_signed_cert.crt'
4. Esportare il certificato root dalla CA di Windows (CA di terze parti).
5. Assegnare un nome al file `ca_root.crt`

   A questo punto, sono disponibili i seguenti tre file:

   - **Chiave privata**: `ontap_signed_request.key` (Chiave corrispondente al certificato del server in

ONTAP). È necessario durante l'installazione del certificato del server).

- **Certificato firmato**: `ontap_signed_cert.crt` (Questo è anche chiamato *certificato del server* in ONTAP).
- **Certificato CA root**: `ca_root.crt` (Questo è anche chiamato *certificato server-ca* in ONTAP).

6. Installare questi certificati in ONTAP. Generare e installare `server` e. `server-ca` Certificati su ONTAP.

Espandere per sample.yaml

```
# Copy the contents of ca_root.crt and use it here.

security certificate install -type server-ca

Please enter Certificate: Press <Enter> when done

-----BEGIN CERTIFICATE-----
<certificate details>
-----END CERTIFICATE-----


You should keep a copy of the CA-signed digital certificate for
future reference.

The installed certificate's CA and serial number for reference:

CA:
serial:

The certificate's generated name for reference:


===

# Copy the contents of ontap_signed_cert.crt and use it here. For
key, use the contents of ontap_cert_request.key file.
security certificate install -type server
Please enter Certificate: Press <Enter> when done

-----BEGIN CERTIFICATE-----
<certificate details>
-----END CERTIFICATE-----

Please enter Private Key: Press <Enter> when done

-----BEGIN PRIVATE KEY-----
<private key details>
-----END PRIVATE KEY-----

Enter certificates of certification authorities (CA) which form the
certificate chain of the server certificate. This starts with the
issuing CA certificate of the server certificate and can range up to
the root CA certificate.
Do you want to continue entering root and/or intermediate
```

```
certificates {y|n}: n

The provided certificate does not have a common name in the subject
field.
Enter a valid common name to continue installation of the
certificate: <ONTAP_CLUSTER_FQDN_NAME>

You should keep a copy of the private key and the CA-signed digital
certificate for future reference.
The installed certificate's CA and serial number for reference:
CA:
serial:
The certificate's generated name for reference:


==
# Modify the vserver settings to enable SSL for the installed
certificate

ssl modify -vserver <vserver_name> -ca <CA>  -server-enabled true
-serial <serial number>       (security ssl modify)

==
# Verify if the certificate works fine:

openssl s_client -CAfile ca_root.crt -showcerts -servername server
-connect <ONTAP_CLUSTER_FQDN_NAME>:443
CONNECTED(00000005)
depth=1 DC = local, DC = umca, CN = <CA>
verify return:1
depth=0
verify return:1
write W BLOCK
---
Certificate chain
0 s:
   i:/DC=local/DC=umca/<CA>

-----BEGIN CERTIFICATE-----
<Certificate details>
```

7. Creare il certificato client per lo stesso host per le comunicazioni senza password. Il centro di controllo Astra utilizza questo processo per comunicare con ONTAP.
8. Generare e installare i certificati client su ONTAP:

Espandere per sample.yaml

```
# Use /CN=admin or use some other account which has privileges.
openssl req -x509 -nodes -days 1095 -newkey rsa:2048 -keyout
ontap_test_client.key -out ontap_test_client.pem -subj "/CN=admin"

Copy the content of ontap_test_client.pem file and use it in the
below command:
security certificate install -type client-ca -vserver <vserver_name>

Please enter Certificate: Press <Enter> when done

-----BEGIN CERTIFICATE-----
<Certificate details>
-----END CERTIFICATE-----

You should keep a copy of the CA-signed digital certificate for
future reference.
The installed certificate's CA and serial number for reference:

CA:
serial:
The certificate's generated name for reference:


==

ssl modify -vserver <vserver_name> -client-enabled true
(security ssl modify)

# Setting permissions for certificates
security login create -user-or-group-name admin -application ontapi
-authentication-method cert -role admin -vserver <vserver_name>

security login create -user-or-group-name admin -application http
-authentication-method cert -role admin -vserver <vserver_name>

==

#Verify passwordless communication works fine with the use of only
certificates:

curl --cacert ontap_signed_cert.crt  --key ontap_test_client.key
--cert ontap_test_client.pem
https://<ONTAP_CLUSTER_FQDN_NAME>/api/storage/aggregates
{
```

```
"records": [
{
"uuid": "f84e0a9b-e72f-4431-88c4-4bf5378b41bd",
"name": "<aggr_name>",
"node": {
"uuid": "7835876c-3484-11ed-97bb-d039ea50375c",
"name": "<node_name>",
"_links": {
"self": {
"href": "/api/cluster/nodes/7835876c-3484-11ed-97bb-d039ea50375c"
}
}
},
"_links": {
"self": {
"href": "/api/storage/aggregates/f84e0a9b-e72f-4431-88c4-
4bf5378b41bd"
}
}
}
],
"num_records": 1,
"_links": {
"self": {
"href": "/api/storage/aggregates"
}
}
}%
```


9. Aggiungere il backend dello storage nell'interfaccia utente di Astra Control Center e fornire i seguenti valori:
   - **Certificato client**: ontap_test_client.pem
   - **Chiave privata**: ontap_test_client.key
   - **Certificato CA attendibile**: ontap_signed_cert.crt

Aggiungere un backend di storage

Dopo aver impostato le credenziali o le informazioni di autenticazione del certificato, è possibile aggiungere un backend di storage ONTAP esistente a Astra Control Center per gestire le risorse.

La gestione dei cluster di storage in Astra Control come back-end dello storage consente di ottenere collegamenti tra volumi persistenti (PVS) e il back-end dello storage, oltre a metriche di storage aggiuntive.

L'aggiunta e la gestione dei backend di storage ONTAP in Astra Control Center sono opzionali quando si

utilizza la tecnologia NetApp SnapMirror, se hai abilitato Astra Control Provivisioner.

Fasi

1. Dal pannello di controllo nell'area di navigazione a sinistra, selezionare **Backend**.
2. Selezionare **Aggiungi**.
3. Nella sezione Use existing della pagina Add storage backend, selezionare **ONTAP**.
4. Selezionare una delle seguenti opzioni:
   - **Usa credenziali amministratore**: Inserire l'indirizzo IP di gestione del cluster ONTAP e le credenziali di amministratore. Le credenziali devono essere credenziali a livello di cluster.

     > L'utente di cui si inseriscono le credenziali deve disporre di `ontapi` Metodo di accesso all'accesso dell'utente abilitato in Gestione di sistema di ONTAP sul cluster ONTAP. Se si intende utilizzare la replica SnapMirror, applicare le credenziali utente con il ruolo "admin", che dispone dei metodi di accesso `ontapi` e. `http`, Sui cluster ONTAP di origine e di destinazione. Fare riferimento a. "Gestire gli account utente nella documentazione di ONTAP" per ulteriori informazioni.

   - **Usa un certificato**: Carica il certificato `.pem` file, la chiave del certificato `.key` e, facoltativamente, il file dell'autorità di certificazione.
5. Selezionare **Avanti**.
6. Confermare i dettagli del back-end e selezionare **Manage** (Gestisci).

Risultato

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

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

Aggiungi un bucket

È possibile aggiungere un bucket utilizzando l'interfaccia utente di Astra Control o. "API di controllo Astra". 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.

Non è necessario un bucket in Astra Control se si esegue il cloning della configurazione dell'applicazione e dello storage persistente sullo stesso cluster. La funzionalità di snapshot delle applicazioni non richiede un bucket.

Prima di iniziare

- Assicurati di avere un bucket raggiungibile dai cluster gestiti da Astra Control Center.
- Assicurarsi di disporre delle credenziali per il bucket.
- Assicurarsi che la benna sia di uno dei seguenti tipi:
  - NetApp ONTAP S3
  - NetApp StorageGRID S3

- Microsoft Azure
- Generico S3

> Amazon Web Services (AWS) e Google Cloud Platform (GCP) utilizzano il tipo di bucket S3 generico.

> 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 rivendicano il supporto S3 di Amazon.

Fasi

1. Nell'area di navigazione a sinistra, selezionare **Bucket**.
2. Selezionare **Aggiungi**.
3. 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.

4. Inserire un nome bucket esistente e una descrizione opzionale.

> Il nome e la descrizione del bucket vengono visualizzati come una posizione 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.

5. Inserire il nome o l'indirizzo IP dell'endpoint S3.
6. In **Seleziona credenziali**, selezionare la scheda **Aggiungi** o **Usa esistente**.
   - Se si sceglie **Aggiungi**:
     i. Immettere un nome per la credenziale che la distingue dalle altre credenziali in Astra Control.
     ii. Inserire l'ID di accesso e la chiave segreta incollando il contenuto dagli Appunti.
   - Se si sceglie **Usa esistente**:
     i. Selezionare le credenziali esistenti che si desidera utilizzare con il bucket.
7. Selezionare `Add`.

> Quando si aggiunge un bucket, Astra Control contrassegna un bucket con l'indicatore bucket predefinito. Il primo bucket creato diventa quello predefinito. Con l'aggiunta di bucket, è possibile decidere in un secondo momento "impostare un altro bucket predefinito".