Inizia subito

Astra Control Center

NetApp
November 20, 2023

Sommario

Inizia subito

Requisiti di Astra Control Center

Inizia verificando il supporto per cluster, applicazioni, licenze e browser web Kubernetes.

Requisiti generali del cluster Kubernetes

Un cluster Kubernetes deve soddisfare i seguenti requisiti generali in modo da poterlo individuare e gestire da Astra Control Center.

- **Registro immagini**: È necessario disporre di un registro di immagini Docker privato in cui è possibile trasferire le immagini di build di Astra Control Center. È necessario disporre dell'URL del registro delle immagini in cui caricare le immagini e contrassegnare le immagini per il registro dei container privati.
- **Configurazione dello storage Trident / ONTAP**: Il centro di controllo Astra richiede che Trident versione 21.01 o 21.04 sia già installato e configurato per funzionare con NetApp ONTAP versione 9.5 o successiva come backend dello storage. Astra Control Center richiede la creazione e l'impostazione di una classe di storage come classe di storage predefinita. Centro di controllo Astra supporta i seguenti driver ONTAP forniti da Trident:
  - ontap-nas
  - ontap-nas-flexgroup
  - ontap-san
  - ontap-san-economy

Se si intende gestire il cluster Kubernetes da Astra Control Center e utilizzare il cluster per ospitare l'installazione di Astra Control Center, il cluster presenta i seguenti requisiti aggiuntivi:

- La versione più recente di Kubernetes "componente snapshot-controller" è installato
- Un Trident "oggetto volumesnapshotclass" è stato definito da un amministratore
- Nel cluster esiste una classe di storage Kubernetes predefinita
- Almeno una classe di storage è configurata per utilizzare Trident
- Metodo per indirizzare l'FQDN di Astra Control Center all'indirizzo IP esterno del servizio Astra Control Center

Cluster OpenShift

Centro di controllo Astra richiede un cluster Red Hat OpenShift Container Platform 4.6.8 o 4.7 con classi di storage Trident supportate da ONTAP 9.5 o versione successiva, con i seguenti attributi:

- Almeno 300 GB di capacità di storage ONTAP disponibile
- 3 nodi controller con 4 core CPU, 16 GB di RAM e 120 GB di storage disponibili ciascuno
- 3 nodi di lavoro con almeno 12 core CPU, 32 GB di RAM e 50 GB di storage disponibili ciascuno
- Kubernetes versione 1.19 o 1.20
- Tipo di servizio "LoadBalancer" disponibile per il traffico in ingresso da inviare ai servizi nel cluster OpenShift
- Metodo per indirizzare l'FQDN di Astra Control Center all'indirizzo IP con bilanciamento del carico

Questi requisiti minimi presuppongono che Astra Control Center sia l'unica applicazione in esecuzione sul cluster OpenShift. Se il cluster esegue applicazioni aggiuntive, è necessario modificare di conseguenza questi requisiti minimi.

Assicurarsi che il cluster soddisfi i requisiti minimi e seguire le Best practice di Kubernetes in modo che Astra Control Center sia altamente disponibile nel cluster Kubernetes.

OpenShift 4.8 non è supportato.

Durante la clonazione dell'applicazione, Astra Control Center deve consentire a OpenShift di montare volumi e modificare la proprietà dei file. Per questo motivo, ONTAP deve essere configurato in modo da consentire il completamento corretto delle operazioni del volume utilizzando i seguenti comandi:

1. `export-policy rule modify -vserver svm0 -policyname default -ruleindex 1 -superuser sys`
2. `export-policy rule modify -policyname default -ruleindex 1 -anon 65534`

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

Requisiti di gestione delle applicazioni

Astra Control Center ha i seguenti requisiti di gestione delle applicazioni:

- **Licensing**: È necessaria una licenza Astra Control Center per gestire le applicazioni utilizzando Astra Control Center.
- **Helm 3**: Se utilizzi Helm per implementare le app, Astra Control Center richiede Helm versione 3. La gestione e la clonazione delle applicazioni implementate con Helm 3 (o aggiornate da Helm 2 a Helm 3) sono completamente supportate. Le app implementate con Helm 2 non sono supportate.
- **Gestione degli operatori**: Astra Control Center non supporta le applicazioni implementate con operatori abilitati per Operator Lifecycle Manager (OLM) o con gli operatori con ambito cluster.

Accesso a Internet

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

Licenza

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

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

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

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

Astra Control Center utilizza un servizio del tipo "LoadBalancer" (svc/traefik nello spazio dei nomi di Astra Control Center) e richiede l'assegnazione di un indirizzo IP esterno accessibile. Per i cluster OpenShift on-premise, è possibile utilizzare "MetalLB" Per assegnare automaticamente un indirizzo IP esterno al servizio. Nella configurazione del server DNS interno, puntare il nome DNS scelto per Astra Control Center sull'indirizzo IP con bilanciamento del carico.

Requisiti di rete

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

Prodotto	Porta	Protocollo	Direzione	Scopo
Centro di controllo Astra	443	HTTPS	Ingresso	Accesso UI/API - assicurarsi che questa porta sia aperta in entrambi i modi tra il cluster che ospita Astra Control Center e ciascun cluster gestito
Centro di controllo Astra	9090	HTTPS	• Ingresso (al cluster che ospita Astra Control Center) • Egress (porta casuale dall'indirizzo IP del nodo di ciascun nodo di lavoro di ciascun cluster gestito)	Dati delle metriche per il cliente: Assicurarsi che ogni cluster gestito possa accedere a questa porta sul cluster che ospita Astra Control Center
Trident	34571	HTTPS	Ingresso	Comunicazione del nodo pod
Trident	9220	HTTP	Ingresso	Endpoint delle metriche

Browser Web supportati

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

Cosa succederà

Visualizzare il "avvio rapido" panoramica.

Avvio rapido per Astra Control Center

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

Provalo! Se si desidera provare Astra Control Center, è possibile utilizzare una licenza di valutazione di 90 giorni. Vedere "informazioni sulle licenze" per ulteriori informazioni.

1 Esaminare i requisiti del cluster Kubernetes

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

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

2 Scaricare e installare Astra Control Center

- Scarica Astra Control Center dal NetApp Support Site.
- Installare Astra Control Center nell'ambiente locale.
- Scopri la tua configurazione Trident supportata dal backend dello storage ONTAP.

  Per la prima release, installerai le immagini su un registro OpenShift o utilizzerai il registro locale.

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

3 Completare alcune attività di configurazione iniziali

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

"Scopri di più sul processo di configurazione iniziale".

4 Utilizzare Astra Control Center

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

- Gestire un'applicazione. "Scopri di più su come gestire le app".
- Se lo si desidera, connettersi a NetApp Cloud Insights per visualizzare le metriche sullo stato di salute del

sistema, sulla capacità e sul throughput all'interno dell'interfaccia utente di Astra Control Center. "Scopri di più sulla connessione a Cloud Insights".

5 **Continuare da questa guida di avvio rapido**

"Installare Astra Control Center".

Trova ulteriori informazioni

- "Utilizzare l'API Astra"

Installare Astra Control Center

Per installare Astra Control Center, procedere come segue:

- Installare Astra Control Center
- Accedere all'interfaccia utente di Astra Control Center

Installare Astra Control Center

Per installare Astra Control Center, scarica il pacchetto di installazione dal NetApp Support Site ed esegui una serie di comandi per installare Astra Control Center Operator e Astra Control Center nel tuo ambiente. È possibile utilizzare questa procedura per installare Astra Control Center in ambienti connessi a Internet o con connessione ad aria.

Di cosa hai bisogno

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

```
oc get clusteroperators
```

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

```
oc get apiservices
```

A proposito di questa attività

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

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

- Consente di determinare se tutti i pod Astra Control Center sono in esecuzione.
- Installa l'interfaccia utente Astra.

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

Fasi

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

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

4. Estrarre le immagini:

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

5. Passare alla directory Astra.

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

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

> Di seguito viene riportato uno script di esempio per il caricamento automatico delle immagini.

   a. Accedere al registro di sistema di Docker:

```
docker login [Docker_registry_path]
```

   b. Caricare le immagini in Docker.
   c. Contrassegnare le immagini.
   d. Trasferire le immagini nel registro locale.

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

7. (Solo per i registri con requisiti di autenticazione) se si utilizza un registro che richiede l'autenticazione, è necessario effettuare le seguenti operazioni:

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

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

   Risposta:

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

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

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

   Esempio di risposta:

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

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

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

   Esempio di risposta:

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

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

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

Risposta

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

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

```
vim astra_control_center_operator_deploy.yaml
```

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

```
imagePullSecrets:
- name: astra-registry-cred
```

b. Cambiare `[Docker_registry_path]` per `kube-rbac-prox` immagine al percorso del registro in cui sono state inviate le immagini in un passaggio precedente.

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

```
apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    control-plane: controller-manager
  name: acc-operator-controller-manager
  namespace: netapp-acc-operator
spec:
  replicas: 1
  selector:
    matchLabels:
      control-plane: controller-manager
  template:
    metadata:
      labels:
        control-plane: controller-manager
    spec:
      containers:
      - args:
        - --secure-listen-address=0.0.0.0:8443
        - --upstream=http://127.0.0.1:8080/
        - --logtostderr=true
        - --v=10
        image: [Docker_registry_path]/kube-rbac-proxy:v0.5.0
        name: kube-rbac-proxy
        ports:
        - containerPort: 8443
          name: https
      - args:
        - --health-probe-bind-address=:8081
        - --metrics-bind-address=127.0.0.1:8080
        - --leader-elect
        command:
        - /manager
        env:
        - name: ACCOP_LOG_LEVEL
          value: "2"
        image: [Docker_registry_path]/acc-operator:[version x.y.z]
        imagePullPolicy: IfNotPresent
      imagePullSecrets: []
```

9. Modificare il file delle risorse personalizzate (CR) di Astra Control Center (`astra_control_center_min.yaml`):

```
vim astra_control_center_min.yaml
```

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

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

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

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

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

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

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

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

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

h. Se non si utilizza un registro che richiede l'autorizzazione, eliminare `secret` linea.

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

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

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

Esempio di risposta:

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

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

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

Esempio di risposta:

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

12. Eseguire la seguente patch per correggere "associazione dei ruoli del cluster".
13. Installare Astra Control Center in `netapp-acc` spazio dei nomi (o personalizzato):

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

Esempio di risposta:

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

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

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

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

Esempio di risposta:

```
NAME                                        READY   STATUS    RESTARTS
AGE
acc-helm-repo-5fdfff786f-gkv6z              1/1     Running   0
4m58s
activity-649f869bf7-jn5gs                   1/1     Running   0
3m14s
asup-79846b5fdc-s9s97                       1/1     Running   0
3m10s
authentication-84c78f5cf4-qhx9t             1/1     Running   0
118s
billing-9b8496787-v8rzv                     1/1     Running   0
2m54s
bucketservice-5fb876d9d5-wkfvz              1/1     Running   0
3m26s
cloud-extension-f9f4f59c6-dz6s6             1/1     Running   0
3m
cloud-insights-service-5676b8c6d4-6q7lv     1/1     Running   0
2m52s
composite-compute-7dcc9c6d6c-lxdr6          1/1     Running   0
2m50s
composite-volume-74dbfd7577-cd42b           1/1     Running   0
3m2s
credentials-75dbf46f9d-5qm2b                1/1     Running   0
3m32s
entitlement-6cf875cb48-gkvhp                1/1     Running   0
3m12s
features-74fd97bb46-vss2n                   1/1     Running   0
3m6s
fluent-bit-ds-2g9jb                         1/1     Running   0
113s
fluent-bit-ds-5tg5h                         1/1     Running   0
113s
fluent-bit-ds-qfxb8                         1/1     Running   0
113s
graphql-server-7769f98b86-p4qrv             1/1     Running   0
90s
```

```
identity-566c566cd5-ntfj6                    1/1     Running   0
3m16s
influxdb2-0                                  1/1     Running   0
4m43s
krakend-5cb8d56978-44q66                     1/1     Running   0
93s
license-66cbbc6f48-27kgf                     1/1     Running   0
3m4s
login-ui-584f7fd84b-dmdrp                    1/1     Running   0
87s
loki-0                                       1/1     Running   0
4m44s
metrics-ingestion-service-6dcfddf45f-mhnvh   1/1     Running   0
3m8s
monitoring-operator-78d67b4d4-nxs6v          2/2     Running   0
116s
nats-0                                       1/1     Running   0
4m40s
nats-1                                       1/1     Running   0
4m26s
nats-2                                       1/1     Running   0
4m15s
nautilus-9b664bc55-rn9t8                     1/1     Running   0
2m56s
openapi-dc5ddfb7d-6q8vh                      1/1     Running   0
3m20s
polaris-consul-consul-5tjs7                  1/1     Running   0
4m43s
polaris-consul-consul-5wbnx                  1/1     Running   0
4m43s
polaris-consul-consul-bfvl7                  1/1     Running   0
4m43s
polaris-consul-consul-server-0               1/1     Running   0
4m43s
polaris-consul-consul-server-1               1/1     Running   0
4m43s
polaris-consul-consul-server-2               1/1     Running   0
4m43s
polaris-mongodb-0                            2/2     Running   0
4m49s
polaris-mongodb-1                            2/2     Running   0
4m22s
polaris-mongodb-arbiter-0                    1/1     Running   0
4m49s
polaris-ui-6648875998-75d98                  1/1     Running   0
92s
```

```
polaris-vault-0                              1/1     Running   0
4m41s
polaris-vault-1                              1/1     Running   0
4m41s
polaris-vault-2                              1/1     Running   0
4m41s
storage-backend-metrics-69546f4fc8-m7lfj     1/1     Running   0
3m22s
storage-provider-5d46f755b-qfv89             1/1     Running   0
3m30s
support-5dc579865c-z4pwq                     1/1     Running   0
3m18s
telegraf-ds-4452f                            1/1     Running   0
113s
telegraf-ds-gnqxl                            1/1     Running   0
113s
telegraf-ds-jhw74                            1/1     Running   0
113s
telegraf-rs-gg6m4                            1/1     Running   0
113s
telemetry-service-6dcc875f98-zft26           1/1     Running   0
3m24s
tenancy-7f7f77f699-q7l6w                     1/1     Running   0
3m28s
traefik-769d846f9b-c9crt                     1/1     Running   0
83s
traefik-769d846f9b-l9n4k                     1/1     Running   0
67s
trident-svc-8649c8bfc5-pdj79                 1/1     Running   0
2m57s
vault-controller-745879f98b-49c5v            1/1     Running   0
4m51s
```

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

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

16. Una volta eseguiti tutti i pod, verificare che l'installazione sia riuscita recuperando l'istanza di AstraControlCenter installata dall'operatore ACC.

```
kubectl get acc -o yaml -n netapp-acc
```

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

> Verrà utilizzato il `uuid` nella fase successiva.

```
apiVersion: v1
items:
- apiVersion: astra.netapp.io/v1
  kind: AstraControlCenter
  metadata:
    creationTimestamp: "2021-07-28T21:36:49Z"
    finalizers:
    - astracontrolcenter.netapp.io/finalizer
   generation: 1
    name: astra
    namespace: netapp-acc
    resourceVersion: "27797604"
    selfLink: /apis/astra.netapp.io/v1/namespaces/netapp-
acc/astracontrolcenters/astra
    uid: 61cd8b65-047b-431a-ba35-510afcb845f1
  spec:
    accountName: Example
    astraAddress: astra.example.com
    astraResourcesScaler: "Off"
    astraVersion: 21.08.52
    autoSupport:
      enrolled: false
    email: admin@example.com
    firstName: SRE
    lastName: Admin
    imageRegistry:
      name: registry_name/astra
  status:
    certManager: deploy
    deploymentState: Deployed
    observedGeneration: 1
    observedVersion: 21.08.52
    postInstall: Complete
    uuid: c49008a5-4ef1-4c5d-a53e-830daf994116
kind: List
metadata:
  resourceVersion: ""
  selfLink: ""
```

18. Per ottenere la password monouso da utilizzare quando si accede ad Astra Control Center, copiare il

`status.uuid` valore della risposta nella fase precedente. La password è `ACC-` Seguito dal valore UUID (`ACC-[UUID]` oppure, in questo esempio, `ACC-c49008a5-4ef1-4c5d-a53e-830daf994116`).

Accedere all'interfaccia utente di Astra Control Center

Dopo aver installato ACC, si modifica la password dell'amministratore predefinito e si accede alla dashboard dell'interfaccia utente ACC.

Fasi

1. In un browser, immettere l'FQDN utilizzato in `astraAddress` in `astra_control_center_min.yaml` CR quando ACC è stato installato.
2. Accettare i certificati autofirmati quando richiesto.

   È possibile creare un certificato personalizzato dopo l'accesso.

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

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

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

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

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

Risolvere i problemi di installazione

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

Fasi

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

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

Cosa succederà

Completare l'implementazione eseguendo "attività di installazione".

Configurare Astra Control Center

Dopo aver installato Astra Control Center, aver effettuato l'accesso all'interfaccia utente e aver modificato la password, sarà necessario impostare una licenza, aggiungere cluster, gestire lo storage e aggiungere bucket.

Attività

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

Aggiungere una licenza per Astra Control Center

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

Di cosa hai bisogno

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

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

Aggiungere una licenza completa o di valutazione

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

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

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

Fasi

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

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

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

Aggiungere il cluster

Per iniziare a gestire le tue applicazioni, Aggiungi un cluster Kubernetes e gestilo come risorsa di calcolo. Devi aggiungere un cluster per Astra Control Center per scoprire le tue applicazioni Kubernetes.

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

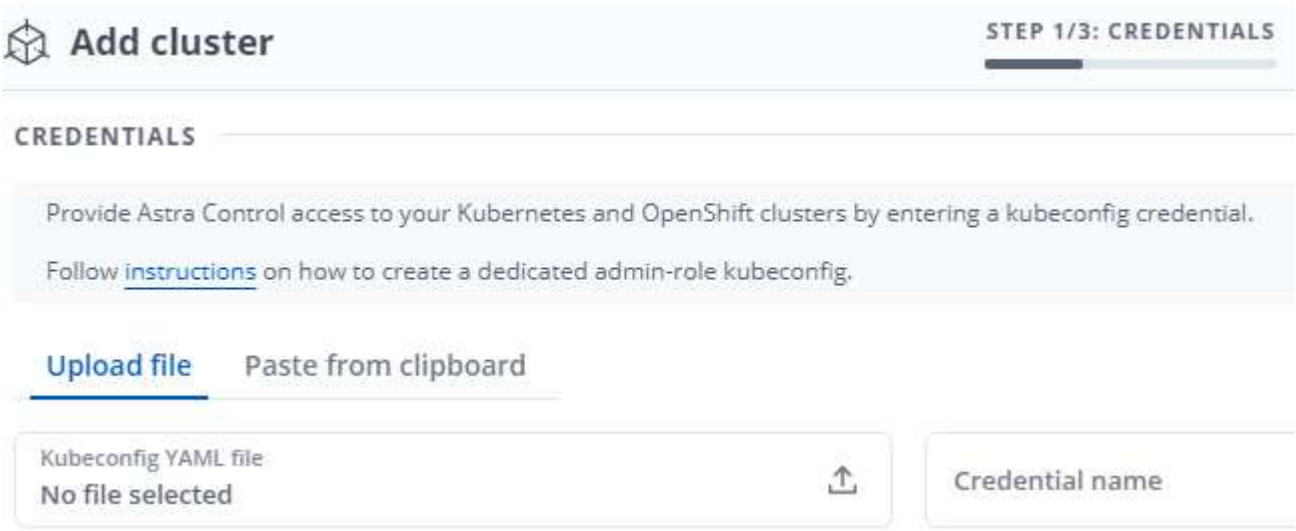
Cosa ti serve? 8217

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

Fasi

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

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

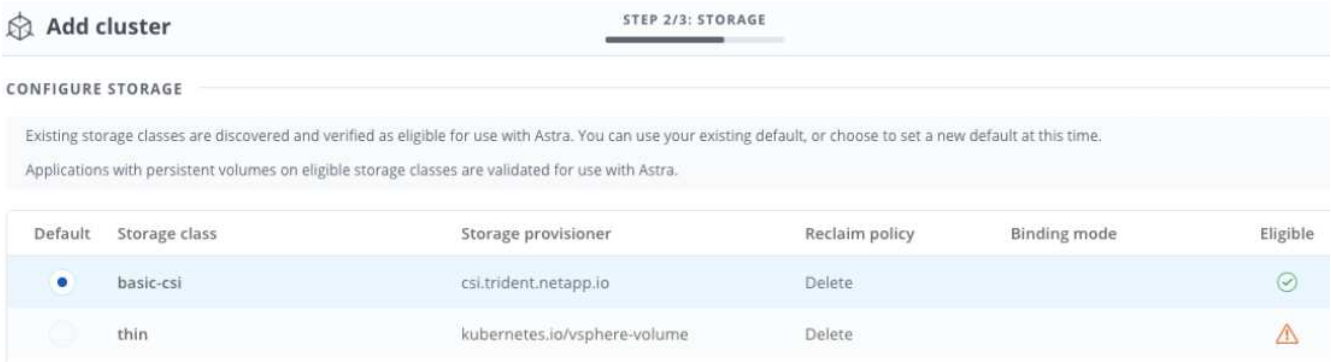


   Se crei il tuo `kubeconfig` file, è necessario definire solo **un** elemento di contesto al suo interno. Vedere "Documentazione Kubernetes" per informazioni sulla creazione `kubeconfig` file.

3. Fornire un nome di credenziale. Per impostazione predefinita, il nome della credenziale viene compilato automaticamente come nome del cluster.
4. Selezionare **Configura storage**.

5. Selezionare la classe di storage da utilizzare per questo cluster Kubernetes e selezionare **Review**.

> Selezionare una classe di storage Trident supportata dallo storage ONTAP.



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

Risultato

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

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

Aggiungere un backend di storage

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

È possibile aggiungere un backend di storage nei seguenti modi:

- Configurare lo storage quando si aggiunge un cluster. Vedere "Aggiungere il cluster".
- Aggiungere un backend di storage rilevato utilizzando la dashboard o l'opzione Backend.

È possibile aggiungere un backend di storage già rilevato utilizzando le seguenti opzioni:

- Aggiungere il back-end di storage utilizzando Dashboard
- Aggiungere il backend di storage utilizzando l'opzione Backend

Aggiungere il back-end di storage utilizzando Dashboard

1. Dalla dashboard eseguire una delle seguenti operazioni:
   a. Dalla sezione backend Dashboard Storage, selezionare **Manage** (Gestisci).
   b. Dalla sezione Dashboard Resource Summary > Storage Backend, selezionare **Add** (Aggiungi).

2. Immettere le credenziali di amministratore di ONTAP e selezionare **Rivedi**.
3. Confermare i dettagli del back-end e selezionare **Manage** (Gestisci).

   Il backend viene visualizzato nell'elenco con le informazioni di riepilogo.

Aggiungere il backend di storage utilizzando l'opzione Backend

1. Nell'area di navigazione a sinistra, selezionare **Backend**.
2. Selezionare **Gestisci**.
3. Immettere le credenziali di amministratore di ONTAP e selezionare **Rivedi**.
4. Confermare i dettagli del back-end e selezionare **Manage** (Gestisci).

   Il backend viene visualizzato nell'elenco con le informazioni di riepilogo.

5. Per visualizzare i dettagli dello storage back-end, selezionarlo.

   Vengono visualizzati anche i volumi persistenti utilizzati dalle applicazioni nel cluster di calcolo gestito.

Aggiungi un bucket

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

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

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

Utilizzare uno dei seguenti tipi di bucket:

- NetApp ONTAP S3
- NetApp StorageGRID S3
- Generico S3

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

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

Fasi

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

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

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

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

d. Immettere il nome o l'indirizzo IP del server S3.

e. Se si desidera che questo bucket sia il bucket predefinito per tutti i backup, selezionare `Make this bucket the default bucket for this private cloud` opzione.

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

f. Continuare aggiungendo informazioni sulle credenziali.

Aggiungere le credenziali di accesso S3

Aggiungi credenziali di accesso S3 in qualsiasi momento.

Fasi

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

Quali sono le prossime novità?

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

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

Trova ulteriori informazioni

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

Prerequisiti per l'aggiunta di un cluster

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

Cosa serve prima di aggiungere un cluster

- Un cluster che esegue OpenShift 4.6 o 4.7, con Trident StorageClasses supportato da ONTAP 9.5 o versione successiva.
  - Uno o più nodi di lavoro con almeno 1 GB di RAM disponibile per l'esecuzione dei servizi di telemetria.

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

- Il superuser e l'ID utente impostati sul sistema ONTAP di backup per eseguire il backup e il ripristino delle applicazioni con il centro di controllo Astra (ACC). Eseguire i seguenti comandi nella riga di comando di ONTAP:
  `export policy rule modify -vserver svm0 -policyname default -ruleindex 1 -superuser sys`
  `export-policy rule modify -policyname default -ruleindex 1 -anon 65534` (valore predefinito)

Eseguire i controlli di idoneità

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

Fasi

1. Controllare la versione di Trident.

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

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

```
NAME       VERSION
trident    21.04.0
```

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

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

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

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

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

Creare un kubeconfig con ruolo di amministratore

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

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

Fasi

1. Creare un account di servizio come segue:

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

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

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

+

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

   a. Applicare l'account del servizio:

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

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

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

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

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

+

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

a. Applicare l'associazione del ruolo del cluster:

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

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

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

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

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

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

4. Generare il kubeconfig come segue:

a. Creare un `create-kubeconfig.sh` file. Se l'indice del token annotato nel passaggio precedente non era 0, sostituire il valore per `TOKEN_INDEX` all'inizio del seguente script con il valore corretto.

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

```
# Update these to match your environment. Replace the value for
TOKEN_INDEX from
# the output in the previous step if it was not 0. 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'
TOKEN_INDEX=0

CONTEXT=$(kubectl config current-context)

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Quali sono le prossime novità?

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

Trova ulteriori informazioni

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

Aggiungere un certificato TLS personalizzato

È possibile rimuovere il certificato TLS autofirmato esistente e sostituirlo con un certificato TLS firmato da

un'autorità di certificazione (CA).

Di cosa hai bisogno

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

Rimuovere il certificato autofirmato

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

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

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

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

Aggiungere un nuovo certificato

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

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

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

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

#spec:
#  selfSigned: {}

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

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

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

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

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

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

5. Creare il certificato utilizzando il seguente comando:

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

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

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

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

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

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

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

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

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

Domande frequenti per Astra Control Center

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

Panoramica

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

Accesso al centro di controllo Astra

Cos'è l'URL di Astra Control?

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

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

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

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

Fasi

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

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

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

Registrazione dei cluster Kubernetes

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

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

Come si annulla la gestione corretta di un cluster?

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

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

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

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

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

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

Gestione delle applicazioni

Astra Control può implementare un'applicazione?

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

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

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

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

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

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

Operazioni di gestione dei dati

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

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

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

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

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

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