



Installare Astra Control Center

Astra Control Center

NetApp

November 21, 2023

Sommario

Installare Astra Control Center utilizzando il processo standard	1
Scarica ed estrai Astra Control Center	2
Installare il plug-in NetApp Astra kubectl	3
Aggiungere le immagini al registro locale	3
Impostare namespace e secret per i registri con requisiti di autenticazione	6
Installare l'operatore del centro di controllo Astra	7
Configurare Astra Control Center	10
Completare l'installazione dell'Astra Control Center e dell'operatore	25
Verificare lo stato del sistema	26
Impostare l'ingresso per il bilanciamento del carico	31
Accedere all'interfaccia utente di Astra Control Center	35
Risolvere i problemi di installazione	35
Cosa succederà	36
Configurare un gestore esterno dei certificati	36

Installare Astra Control Center utilizzando il processo standard

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

Altre procedure di installazione

- **Installa con RedHat OpenShift OperatorHub:** Utilizza questo ["procedura alternativa"](#) Per installare Astra Control Center su OpenShift utilizzando OperatorHub.
- **Installare nel cloud pubblico con backend Cloud Volumes ONTAP:** Utilizzare ["queste procedure"](#) Per installare Astra Control Center in Amazon Web Services (AWS), Google Cloud Platform (GCP) o Microsoft Azure con un backend di storage Cloud Volumes ONTAP.

Per una dimostrazione del processo di installazione di Astra Control Center, vedere ["questo video"](#).

Prima di iniziare

- ["Prima di iniziare l'installazione, preparare l'ambiente per l'implementazione di Astra Control Center"](#).
- Se hai configurato o vuoi configurare le policy di sicurezza dei pod nel tuo ambiente, familiarizza con le policy di sicurezza dei pod e con il modo in cui influiscono sull'installazione di Astra Control Center. Fare riferimento a ["Comprendere le restrizioni delle policy di sicurezza del pod"](#).
- Assicurarsi che tutti i servizi API siano in buono stato e disponibili:

```
kubectl get apiservices
```

- Assicurarsi che l'FQDN Astra che si intende utilizzare sia instradabile a questo cluster. Ciò significa che si dispone di una voce DNS nel server DNS interno o si sta utilizzando un percorso URL principale già registrato.
- Se nel cluster esiste già un gestore dei certificati, è necessario eseguirne alcuni ["fasi preliminari"](#) In modo che Astra Control Center non tenti di installare il proprio cert manager. Per impostazione predefinita, Astra Control Center installa il proprio cert manager durante l'installazione.



Implementare Astra Control Center in un terzo dominio di errore o in un sito secondario. Questa opzione è consigliata per la replica delle applicazioni e il disaster recovery perfetto.

A proposito di questa attività

Il processo di installazione di Astra Control Center consente di effettuare le seguenti operazioni:

- Installare i componenti Astra in `netapp-acc` namespace (o personalizzato).
- Creare un account di amministrazione proprietario di Astra Control predefinito.
- Stabilire un indirizzo e-mail dell'utente amministrativo e una password di configurazione iniziale predefinita. A questo utente viene assegnato il ruolo Owner (Proprietario) necessario per il primo accesso all'interfaccia utente.

- Verificare che tutti i pod Astra Control Center siano in esecuzione.
- Installare l'interfaccia utente di Astra Control Center.



Non eliminare l'operatore di Astra Control Center (ad esempio, `kubectl delete -f astra_control_center_operator_deploy.yaml`) In qualsiasi momento durante l'installazione o il funzionamento di Astra Control Center per evitare di eliminare i pod.

Fasi

Per installare Astra Control Center, procedere come segue:

- [Scarica ed estrai Astra Control Center](#)
- [Installare il plug-in NetApp Astra kubectl](#)
- [Aggiungere le immagini al registro locale](#)
- [Impostare namespace e secret per i registri con requisiti di autenticazione](#)
- [Installare l'operatore del centro di controllo Astra](#)
- [Configurare Astra Control Center](#)
- [Completare l'installazione dell'Astra Control Center e dell'operatore](#)
- [Verificare lo stato del sistema](#)
- [Impostare l'ingresso per il bilanciamento del carico](#)
- [Accedere all'interfaccia utente di Astra Control Center](#)

Scarica ed estrai Astra Control Center

1. Accedere alla "[Pagina di download di Astra Control Center](#)" Sul sito di supporto NetApp.
2. Scarica il bundle contenente Astra Control Center (`astra-control-center-[version].tar.gz`).
3. (Consigliato ma opzionale) Scarica il bundle di certificati e firme per Astra Control Center (`astra-control-center-certs-[version].tar.gz`) per verificare la firma del bundle:

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

```
openssl dgst -sha256 -verify certs/AstraControlCenter-public.pub  
-signature certs/astra-control-center-[version].tar.gz.sig astra-  
control-center-[version].tar.gz
```

Viene visualizzato l'output `Verified OK` una volta completata la verifica.

4. Estrarre le immagini dal bundle Astra Control Center:

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

Installare il plug-in NetApp Astra kubectl

È possibile utilizzare il plug-in della riga di comando di NetApp Astra kubectl per inviare immagini a un repository Docker locale.

Prima di iniziare

NetApp fornisce binari per plug-in per diverse architetture CPU e sistemi operativi. Prima di eseguire questa attività, è necessario conoscere la CPU e il sistema operativo in uso.

Se il plug-in è già stato installato da un'installazione precedente, "[assicurarsi di disporre della versione più recente](#)" prima di completare questa procedura.

Fasi

1. Elencare i binari del plugin NetApp Astra kubectl disponibili e annotare il nome del file necessario per il sistema operativo e l'architettura della CPU:



La libreria di plugin kubectl fa parte del bundle tar e viene estratta nella cartella `kubectl-astra`.

```
ls kubectl-astra/
```

2. Spostare il binario corretto nel percorso corrente e rinominarlo `kubectl-astra`:

```
cp kubectl-astra/<binary-name> /usr/local/bin/kubectl-astra
```

Aggiungere le immagini al registro locale

1. Completare la sequenza di passaggi appropriata per il motore dei container:

Docker

1. Passare alla directory root del tarball. Vengono visualizzati il file e la directory seguenti:

```
acc.manifest.bundle.yaml
acc/
```

2. Trasferire le immagini del pacchetto nella directory delle immagini di Astra Control Center nel registro locale. Eseguire le seguenti sostituzioni prima di eseguire `push-images` comando:
 - Sostituire `<BUNDLE_FILE>` con il nome del file bundle di controllo Astra (`acc.manifest.bundle.yaml`).
 - Sostituire `<MY_FULL_REGISTRY_PATH>` con l'URL del repository Docker; ad esempio, "`<a href='\"https://<docker-registry>\"\" class='\"bare\"\">https://<docker-registry>\"`".
 - Sostituire `<MY_REGISTRY_USER>` con il nome utente.
 - Sostituire `<MY_REGISTRY_TOKEN>` con un token autorizzato per il registro.

```
kubectl astra packages push-images -m <BUNDLE_FILE> -r
<MY_FULL_REGISTRY_PATH> -u <MY_REGISTRY_USER> -p
<MY_REGISTRY_TOKEN>
```

Podman

1. Passare alla directory root del tarball. Vengono visualizzati il file e la directory seguenti:

```
acc.manifest.bundle.yaml
acc/
```

2. Accedere al Registro di sistema:

```
podman login <YOUR_REGISTRY>
```

3. Preparare ed eseguire uno dei seguenti script personalizzato per la versione di Podman utilizzata. Sostituire `<MY_FULL_REGISTRY_PATH>` con l'URL del repository che include le sottodirectory.

```
<strong>Podman 4</strong>
```

```

export REGISTRY=<MY_FULL_REGISTRY_PATH>
export PACKAGENAME=acc
export PACKAGEVERSION=23.04.2-7
export DIRECTORYNAME=acc
for astraImageFile in $(ls ${DIRECTORYNAME}/images/*.tar) ; do
astraImage=$(podman load --input ${astraImageFile} | sed 's/Loaded
image: //'')
astraImageNoPath=$(echo ${astraImage} | sed 's:.*://:')
podman tag ${astraImageNoPath} ${REGISTRY}/netapp/astra/
${PACKAGENAME}/${PACKAGEVERSION}/${astraImageNoPath}
podman push ${REGISTRY}/netapp/astra/${PACKAGENAME}/${
PACKAGEVERSION}/${astraImageNoPath}
done

```

Podman 3

```

export REGISTRY=<MY_FULL_REGISTRY_PATH>
export PACKAGENAME=acc
export PACKAGEVERSION=23.04.2-7
export DIRECTORYNAME=acc
for astraImageFile in $(ls ${DIRECTORYNAME}/images/*.tar) ; do
astraImage=$(podman load --input ${astraImageFile} | sed 's/Loaded
image: //'')
astraImageNoPath=$(echo ${astraImage} | sed 's:.*://:')
podman tag ${astraImageNoPath} ${REGISTRY}/netapp/astra/
${PACKAGENAME}/${PACKAGEVERSION}/${astraImageNoPath}
podman push ${REGISTRY}/netapp/astra/${PACKAGENAME}/${
PACKAGEVERSION}/${astraImageNoPath}
done

```



Il percorso dell'immagine creato dallo script deve essere simile al seguente, a seconda della configurazione del Registro di sistema:

<https://netappdownloads.jfrog.io/docker-astra-control-prod/netapp/astra/acc/23.04.2-7/image:version>

Impostare namespace e secret per i registri con requisiti di autenticazione

1. Esportare il KUBECONFIG per il cluster host Astra Control Center:

```
export KUBECONFIG=[file_path]
```



Prima di completare l'installazione, assicurarsi che KUBECONFIG punti al cluster in cui si desidera installare Astra Control Center. KUBECONFIG può contenere un solo contesto.

2. 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:



Il segnaposto `your_registry_path` deve corrispondere alla posizione delle immagini caricate in precedenza (ad esempio, `[Registry_URL]/netapp/astra/astracc/23.04.2-7`).

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

Esempio di risposta:

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



Se si elimina lo spazio dei nomi dopo la generazione del segreto, ricreare lo spazio dei nomi e rigenerare il segreto per lo spazio dei nomi.

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


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

Esempio di risposta:

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

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

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

Risposta

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

Installare l'operatore del centro di controllo Astra

1. Modificare la directory:

```
cd manifests
```

2. 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
```



Un YAML di esempio annotato segue questi passaggi.

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

```
<strong>astra_control_center_operator_deploy.yaml</strong>
```

```
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
  strategy:
    type: Recreate
  template:
    metadata:
      labels:
        control-plane: controller-manager
    spec:
      containers:
      - args:
        - --secure-listen-address=0.0.0.0:8443
        - --upstream=http://127.0.0.1:8080/
        - --logtostderr=true
        - --v=10
        image: [your_registry_path]/kube-rbac-proxy:v4.8.0
        name: kube-rbac-proxy
        ports:
        - containerPort: 8443
          name: https
      - args:
        - --health-probe-bind-address=:8081
        - --metrics-bind-address=127.0.0.1:8080
        - --leader-elect
        env:
        - name: ACCOP_LOG_LEVEL
          value: "2"
        - name: ACCOP_HELM_INSTALLTIMEOUT
          value: 5m
        image: [your_registry_path]/acc-operator:23.04.36
        imagePullPolicy: IfNotPresent
        livenessProbe:
```

```
    httpGet:
      path: /healthz
      port: 8081
      initialDelaySeconds: 15
      periodSeconds: 20
  name: manager
  readinessProbe:
    httpGet:
      path: /readyz
      port: 8081
      initialDelaySeconds: 5
      periodSeconds: 10
  resources:
    limits:
      cpu: 300m
      memory: 750Mi
    requests:
      cpu: 100m
      memory: 75Mi
  securityContext:
    allowPrivilegeEscalation: false
imagePullSecrets: []
  securityContext:
    runAsUser: 65532
  terminationGracePeriodSeconds: 10
```

3. 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
```

4. Verificare che i pod siano in esecuzione:

```
kubectl get pods -n netapp-acc-operator
```

Configurare Astra Control Center

1. Modificare il file delle risorse personalizzate (CR) di Astra Control Center (`astra_control_center.yaml`) per creare account, supporto, registro e altre configurazioni necessarie:

```
vim astra_control_center.yaml
```



Un YAML di esempio annotato segue questi passaggi.

2. Modificare o confermare le seguenti impostazioni:

`<code>accountName</code>`

Impostazione	Guida	Tipo	Esempio
accountName	Modificare il <code>accountName</code> Stringa al nome che si desidera associare all'account Astra Control Center. Può essere presente un solo nome account.	stringa	Example

`<code>astraVersion</code>`

Impostazione	Guida	Tipo	Esempio
astraVersion	La versione di Astra Control Center da implementare. Non è necessaria alcuna azione per questa impostazione, in quanto il valore verrà pre-compilato.	stringa	23.04.2-7

`<code>astraAddress</code>`

Impostazione	Guida	Tipo	Esempio
<code>astraAddress</code>	<p>Modificare il <code>astraAddress</code></p> <p>Inserire l'FQDN (consigliato) o l'indirizzo IP che si desidera utilizzare nel browser per accedere ad Astra Control Center. Questo indirizzo definisce il modo in cui Astra Control Center verrà trovato nel data center e corrisponde allo stesso FQDN o indirizzo IP fornito dal bilanciamento del carico al termine dell'operazione "Requisiti di Astra Control Center".</p> <p>NOTA: Non utilizzare <code>http://</code> oppure <code>https://</code> nell'indirizzo. Copiare questo FQDN per utilizzarlo in un passo successivo.</p>	stringa	<code>astra.example.com</code>

<code>autoSupport</code>

Le selezioni effettuate in questa sezione determinano se parteciperai all'applicazione di supporto proattivo di NetApp, NetApp Active IQ, e dove verranno inviati i dati. È necessaria una connessione a Internet (porta 442) e tutti i dati di supporto sono resi anonimi.

Impostazione	Utilizzare	Guida	Tipo	Esempio
<code>autoSupport.enrolled</code>	Entrambi <code>enrolled</code> oppure <code>url</code> i campi devono essere selezionati	Cambiare <code>enrolled</code> Per <code>AutoSupport a.false</code> per i siti senza connettività internet o senza <code>retain true</code> per i siti connessi. Un'impostazione di <code>true</code> Consente l'invio di dati anonimi a NetApp a scopo di supporto. L'elezione predefinita è <code>false</code> E indica che non verranno inviati dati di supporto a NetApp.	Booleano	<code>false</code> (valore predefinito)
<code>autoSupport.url</code>	Entrambi <code>enrolled</code> oppure <code>url</code> i campi devono essere selezionati	Questo URL determina dove verranno inviati i dati anonimi.	stringa	https://support.netapp.com/asupprod/post/1.0/postAsup

`<code>email</code>`

Impostazione	Guida	Tipo	Esempio
email	Modificare il email stringa all'indirizzo iniziale predefinito dell'amministratore. Copiare questo indirizzo e-mail per utilizzarlo in passo successivo . Questo indirizzo e-mail verrà utilizzato come nome utente per l'account iniziale per accedere all'interfaccia utente e verrà notificato degli eventi in Astra Control.	stringa	admin@example.com

`<code>firstName</code>`

Impostazione	Guida	Tipo	Esempio
firstName	Il nome dell'amministratore iniziale predefinito associato all'account Astra. Il nome utilizzato qui sarà visibile in un'intestazione dell'interfaccia utente dopo il primo accesso.	stringa	SRE

`<code>LastName</code>`

Impostazione	Guida	Tipo	Esempio
lastName	Il cognome dell'amministratore iniziale predefinito associato all'account Astra. Il nome utilizzato qui sarà visibile in un'intestazione dell'interfaccia utente dopo il primo accesso.	stringa	Admin

<code>imageRegistry</code>

Le selezioni effettuate in questa sezione definiscono il registro delle immagini container che ospita le immagini dell'applicazione Astra, Astra Control Center Operator e il repository Astra Control Center Helm.

Impostazione	Utilizzare	Guida	Tipo	Esempio
<code>imageRegistry.name</code>	Obbligatorio	Il nome del registro delle immagini in cui sono state inviate le immagini in passaggio precedente . Non utilizzare <code>http://</code> oppure <code>https://</code> nel nome del registro di sistema.	stringa	<code>example.registry.com/astra</code>
<code>imageRegistry.secret</code>	Obbligatorio se la stringa immessa per <code>imageRegistry.name</code> requires a secret. IMPORTANT: If you are using a registry that does not require authorization, you must delete this <code>secret</code> linea entro <code>imageRegistry</code> in caso negativo, l'installazione non riesce.	Il nome del segreto Kubernetes utilizzato per l'autenticazione con il registro delle immagini.	stringa	<code>astra-registry-cred</code>

`<code>storageClass</code>`

Impostazione	Guida	Tipo	Esempio
<code>storageClass</code>	<p>Modificare il <code>storageClass</code> valore da <code>ontap-gold</code> A un'altra risorsa Astra Trident <code>storageClass</code> come richiesto dall'installazione. Eseguire il comando <code>kubectl get sc</code> per determinare le classi di storage configurate esistenti. Una delle classi di storage basate su Astra Trident deve essere inserita nel file manifest (<code>astra-control-center-<version>.manifest</code>) E verranno utilizzati per Astra PVS. Se non è impostata, viene utilizzata la classe di storage predefinita.</p> <p>NOTA: Se è configurata una classe di storage predefinita, assicurarsi che sia l'unica classe di storage con l'annotazione predefinita.</p>	stringa	<code>ontap-gold</code>

`<code>volumeReclaimPolicy</code>`

Impostazione	Guida	Tipo	Opzioni
<code>volumeReclaimPolicy</code>	In questo modo viene impostata la policy di recupero per il PVS di Astra. Impostare questo criterio su <code>Retain</code> Conserva i volumi persistenti dopo l'eliminazione di Astra. Impostare questo criterio su <code>Delete</code> elimina i volumi persistenti dopo l'eliminazione di astra. Se questo valore non viene impostato, il PVS viene mantenuto.	stringa	<ul style="list-style-type: none">• <code>Retain</code> (Valore predefinito)• <code>Delete</code>

`<code>ingressType</code>`





Impostazione	Guida	Tipo	Opzioni
ingressType	<p>Utilizzare uno dei seguenti tipi di ingresso:</p> <p>Generic (ingressType: "Generic") (Impostazione predefinita) Utilizzare questa opzione quando si utilizza un altro controller di ingresso o si preferisce utilizzare un controller di ingresso personalizzato. Una volta implementato Astra Control Center, è necessario configurare "controller di ingresso" Per esporre Astra Control Center con un URL.</p> <p>AccTraefik (ingressType: "AccTraefik") Utilizzare questa opzione quando si preferisce non configurare un controller di ingresso. In questo modo viene implementato l'Astra Control Center traefik Gateway come servizio di tipo Kubernetes LoadBalancer.</p> <p>Astra Control Center utilizza un servizio del tipo "LoadBalancer" (svc/traefik Nello spazio dei nomi di Astra Control Center) e richiede l'assegnazione di un indirizzo IP esterno accessibile. Se nel proprio ambiente sono consentiti i bilanciatori di carico e non ne è già configurato uno, è possibile utilizzare MetalLB o un</p>	stringa	<ul style="list-style-type: none"> • Generic (valore predefinito) • AccTraefik

`scaleSize`

Impostazione	Guida	Tipo	Opzioni
<code>scaleSize</code>	<p>Per impostazione predefinita, Astra utilizza High Availability (ha) <code>scaleSize</code> di Medium, Che implementa la maggior parte dei servizi in ha e implementa più repliche per la ridondanza. Con <code>scaleSize</code> come Small, Astra ridurrà il numero di repliche per tutti i servizi ad eccezione dei servizi essenziali per ridurre il consumo.</p> <p>SUGGERIMENTO: Medium le implementazioni sono costituite da circa 100 pod (non inclusi i carichi di lavoro transitori. 100 pod si basa su una configurazione a tre nodi master e tre nodi worker). Tenere a conoscenza dei limiti di rete per pod che potrebbero rappresentare un problema nell'ambiente, in particolare quando si prendono in considerazione scenari di disaster recovery.</p>	stringa	<ul style="list-style-type: none">• Small• Medium (Valore predefinito)

`<code>astraResourcesScaler</code>`

Impostazione	Guida	Tipo	Opzioni
<code>astraResourcesScaler</code>	<p>Opzioni di scalabilità per i limiti delle risorse di AstraControlCenter. Per impostazione predefinita, Astra Control Center implementa le richieste di risorse impostate per la maggior parte dei componenti all'interno di Astra. Questa configurazione consente allo stack software Astra Control Center di migliorare le prestazioni in ambienti con maggiore carico e scalabilità delle applicazioni.</p> <p>Tuttavia, negli scenari che utilizzano cluster di sviluppo o test più piccoli, il campo <code>CR astraResourcesScaler</code> può essere impostato su <code>Off</code>. In questo modo vengono disattivate le richieste di risorse e viene eseguita l'implementazione su cluster più piccoli.</p>	stringa	<ul style="list-style-type: none">• <code>Default</code> (Valore predefinito)• <code>Off</code>

<code>additionalValues</code>

- Per le comunicazioni Cloud Insights e Centro di controllo Astral, la verifica del certificato TLS è disattivata per impostazione predefinita. È possibile attivare la verifica della certificazione TLS per la comunicazione tra Cloud Insights e il cluster host e il cluster gestito di Astra Control Center aggiungendo la seguente sezione in `additionalValues`.

```
additionalValues:  
  netapp-monitoring-operator:  
    config:  
      ciSkipTlsVerify: false  
  cloud-insights-service:  
    config:  
      ciSkipTlsVerify: false  
  telemetry-service:  
    config:  
      ciSkipTlsVerify: false
```

`<code>crds</code>`

Le selezioni effettuate in questa sezione determinano il modo in cui Astra Control Center deve gestire i CRD.

Impostazione	Guida	Tipo	Esempio
<code>crds.externalCertManager</code>	<p>Se si utilizza un gestore esterno dei certificati, cambiare <code>externalCertManager</code> a <code>true</code>. L'impostazione predefinita <code>false</code> Fa in modo che Astra Control Center installi i propri CRD di gestione dei certificati durante l'installazione.</p> <p>I CRDS sono oggetti a livello di cluster e l'installazione potrebbe avere un impatto su altre parti del cluster. È possibile utilizzare questo indicatore per segnalare ad Astra Control Center che questi CRD verranno installati e gestiti dall'amministratore del cluster al di fuori di Astra Control Center.</p>	Booleano	<code>False</code> (valore predefinito)
<code>crds.externalTraefik</code>	<p>Per impostazione predefinita, Astra Control Center installerà i CRD Traefik richiesti. I CRDS sono oggetti a livello di cluster e l'installazione potrebbe avere un impatto su altre parti del cluster. È possibile utilizzare questo indicatore per segnalare ad Astra Control Center che questi CRD verranno installati e gestiti dall'amministratore del cluster al di fuori di Astra Control Center.</p>	Booleano	<code>False</code> (valore predefinito)



Assicurarsi di aver selezionato la classe di storage e il tipo di ingresso corretti per la configurazione prima di completare l'installazione.

```
<strong>astra_control_center.yaml</strong>
```

```
apiVersion: astra.netapp.io/v1
kind: AstraControlCenter
metadata:
  name: astra
spec:
  accountName: "Example"
  astraVersion: "ASTRA_VERSION"
  astraAddress: "astra.example.com"
  autoSupport:
    enrolled: true
  email: "[admin@example.com]"
  firstName: "SRE"
  lastName: "Admin"
  imageRegistry:
    name: "[your_registry_path]"
    secret: "astra-registry-cred"
  storageClass: "ontap-gold"
  volumeReclaimPolicy: "Retain"
  ingressType: "Generic"
  scaleSize: "Medium"
  astraResourcesScaler: "Default"
  additionalValues: {}
  crds:
    externalTraefik: false
    externalCertManager: false
```

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

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

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

Esempio di risposta:

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

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

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

Esempio di risposta:

```
astracenter.astra.netapp.io/astra created
```



L'operatore di Astra Control Center esegue un controllo automatico dei requisiti ambientali. Mancante "requisiti" Può causare problemi di installazione o il funzionamento non corretto di Astra Control Center. Vedere [sezione successiva](#) per verificare la presenza di messaggi di avvertenza relativi al controllo automatico del sistema.

Verificare lo stato del sistema

È possibile verificare lo stato del sistema utilizzando i comandi `kubectl`. Se preferisci utilizzare OpenShift, puoi utilizzare comandi `oc` paragonabili per le fasi di verifica.

Fasi

1. Verificare che il processo di installazione non abbia prodotto messaggi di avviso relativi ai controlli di convalida:

```
kubectl get acc [astra or custom Astra Control Center CR name] -n [netapp-acc or custom namespace] -o yaml
```



Ulteriori messaggi di avviso sono riportati anche nei registri dell'operatore di Astra Control Center.

2. Correggere eventuali problemi dell'ambiente segnalati dai controlli automatici dei requisiti.



È possibile correggere i problemi assicurandosi che l'ambiente soddisfi i requisiti "requisiti" Per Astra Control Center.

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

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

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

Esempio di risposta

NAME	READY	STATUS	
RESTARTS	AGE		
acc-helm-repo-6cc7696d8f-pmhm8	1/1	Running	0
9h			
activity-597fb656dc-5rd4l	1/1	Running	0
9h			
activity-597fb656dc-mqmcw	1/1	Running	0
9h			
api-token-authentication-62f84	1/1	Running	0
9h			
api-token-authentication-68nlf	1/1	Running	0
9h			
api-token-authentication-ztgrm	1/1	Running	0
9h			
asup-669d4ddbc4-fnmwp	1/1	Running	1
(9h ago) 9h			
authentication-78789d7549-lk686	1/1	Running	0
9h			
bucket-service-65c7d95496-24x7l	1/1	Running	3
(9h ago) 9h			
cert-manager-c9f9fbf9f-k8zq2	1/1	Running	0
9h			
cert-manager-c9f9fbf9f-qj1zm	1/1	Running	0
9h			
cert-manager-cainjector-dbbbd8447-b5q1l	1/1	Running	0
9h			
cert-manager-cainjector-dbbbd8447-p5whs	1/1	Running	0
9h			
cert-manager-webhook-6f97bb7d84-4722b	1/1	Running	0
9h			
cert-manager-webhook-6f97bb7d84-86kv5	1/1	Running	0
9h			
certificates-59d9f6f4bd-2j899	1/1	Running	0
9h			
certificates-59d9f6f4bd-9d9k6	1/1	Running	0
9h			
certificates-expiry-check-28011180--1-81kxz	0/1	Completed	0
9h			
cloud-extension-5c9c9958f8-jdhrp	1/1	Running	0
9h			
cloud-insights-service-5cdd5f7f-pp8r5	1/1	Running	0
9h			
composite-compute-66585789f4-hxn5w	1/1	Running	0
9h			

composite-volume-68649f68fd-tb7p4 9h	1/1	Running	0
credentials-dfc844c57-jsx92 9h	1/1	Running	0
credentials-dfc844c57-xw26s 9h	1/1	Running	0
entitlement-7b47769b87-4jb6c 9h	1/1	Running	0
features-854d8444cc-c24b7 9h	1/1	Running	0
features-854d8444cc-dv6sm 9h	1/1	Running	0
fluent-bit-ds-9tlv4 9h	1/1	Running	0
fluent-bit-ds-bpkcb 9h	1/1	Running	0
fluent-bit-ds-cxmxw 9h	1/1	Running	0
fluent-bit-ds-jgnhc 9h	1/1	Running	0
fluent-bit-ds-vtr6k 9h	1/1	Running	0
fluent-bit-ds-vxqd5 9h	1/1	Running	0
graphql-server-7d4b9d44d5-zdbf5 9h	1/1	Running	0
identity-6655c48769-4pwk8 9h	1/1	Running	0
influxdb2-0 9h	1/1	Running	0
keycloak-operator-55479d6fc6-slvmt 9h	1/1	Running	0
krakend-f487cb465-78679 9h	1/1	Running	0
krakend-f487cb465-rjsxx 9h	1/1	Running	0
license-64cbc7cd9c-qxsr8 9h	1/1	Running	0
login-ui-5db89b5589-ndb96 9h	1/1	Running	0
loki-0 9h	1/1	Running	0
metrics-facade-8446f64c94-x8h7b 9h	1/1	Running	0
monitoring-operator-6b44586965-pvcl4 9h	2/2	Running	0

nats-0	1/1	Running	0
9h			
nats-1	1/1	Running	0
9h			
nats-2	1/1	Running	0
9h			
nautilus-85754d87d7-756qb	1/1	Running	0
9h			
nautilus-85754d87d7-q8j7d	1/1	Running	0
9h			
openapi-5f9cc76544-7fnjm	1/1	Running	0
9h			
openapi-5f9cc76544-vzr7b	1/1	Running	0
9h			
packages-5db49f8b5-lrzhd	1/1	Running	0
9h			
polaris-consul-consul-server-0	1/1	Running	0
9h			
polaris-consul-consul-server-1	1/1	Running	0
9h			
polaris-consul-consul-server-2	1/1	Running	0
9h			
polaris-keycloak-0	1/1	Running	2
(9h ago) 9h			
polaris-keycloak-1	1/1	Running	0
9h			
polaris-keycloak-2	1/1	Running	0
9h			
polaris-keycloak-db-0	1/1	Running	0
9h			
polaris-keycloak-db-1	1/1	Running	0
9h			
polaris-keycloak-db-2	1/1	Running	0
9h			
polaris-mongodb-0	1/1	Running	0
9h			
polaris-mongodb-1	1/1	Running	0
9h			
polaris-mongodb-2	1/1	Running	0
9h			
polaris-ui-66fb99479-qp9gq	1/1	Running	0
9h			
polaris-vault-0	1/1	Running	0
9h			
polaris-vault-1	1/1	Running	0
9h			

polaris-vault-2 9h	1/1	Running	0
public-metrics-76fbf9594d-zmxzw 9h	1/1	Running	0
storage-backend-metrics-7d7fbc9cb9-lmd25 9h	1/1	Running	0
storage-provider-5bdd456c4b-2fftc 9h	1/1	Running	0
task-service-87575df85-dnn2q (9h ago) 9h	1/1	Running	3
task-service-task-purge-28011720--1-q6w4r 28m	0/1	Completed	0
task-service-task-purge-28011735--1-vk6pd 13m	1/1	Running	0
telegraf-ds-2r2kw 9h	1/1	Running	0
telegraf-ds-6s9d5 9h	1/1	Running	0
telegraf-ds-96jl7 9h	1/1	Running	0
telegraf-ds-hbp84 9h	1/1	Running	0
telegraf-ds-plwzv 9h	1/1	Running	0
telegraf-ds-sr22c 9h	1/1	Running	0
telegraf-rs-4sbg8 9h	1/1	Running	0
telemetry-service-fb9559f7b-mk917 (9h ago) 9h	1/1	Running	3
tenancy-559bbc6b48-5msgg 9h	1/1	Running	0
traefik-d997b8877-7xpf4 9h	1/1	Running	0
traefik-d997b8877-9xv96 9h	1/1	Running	0
trident-svc-585c97548c-d25z5 9h	1/1	Running	0
vault-controller-88484b454-2d6sr 9h	1/1	Running	0
vault-controller-88484b454-fc5cz 9h	1/1	Running	0
vault-controller-88484b454-jktld 9h	1/1	Running	0

4. (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
```



`accHost` la registrazione del cluster è una delle ultime operazioni e, in caso di errore, la distribuzione non avrà esito negativo. In caso di errore di registrazione del cluster indicato nei registri, è possibile tentare di nuovo la registrazione tramite ["Aggiungere il flusso di lavoro del cluster nell'interfaccia utente"](#) O API.

5. Una volta eseguiti tutti i pod, verificare che l'installazione sia stata eseguita correttamente (`READY` è `True`) E ottenere la password di configurazione iniziale da utilizzare quando si accede ad Astra Control Center:

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

Risposta:

NAME	UUID	VERSION	ADDRESS
READY			
astra	9aa5fdae-4214-4cb7-9976-5d8b4c0ce27f	23.04.2-7	10.111.111.111
True			



Copiare il valore UUID. La password è `ACC-` Seguito dal valore UUID (`ACC-[UUID]` oppure, in questo esempio, `ACC-9aa5fdae-4214-4cb7-9976-5d8b4c0ce27f`).

Impostare l'ingresso per il bilanciamento del carico

È possibile configurare un controller di ingresso Kubernetes che gestisce l'accesso esterno ai servizi. Queste procedure forniscono esempi di configurazione per un controller di ingresso se si utilizza il valore predefinito di `ingressType: "Generic"` Nella risorsa personalizzata di Astra Control Center (`astra_control_center.yaml`). Non è necessario utilizzare questa procedura, se specificato `ingressType: "AccTraefik"` Nella risorsa personalizzata di Astra Control Center (`astra_control_center.yaml`).

Dopo l'implementazione di Astra Control Center, è necessario configurare il controller di ingresso per esporre Astra Control Center con un URL.

Le fasi di installazione variano a seconda del tipo di controller di ingresso utilizzato. Astra Control Center supporta molti tipi di controller di ingresso. Queste procedure di configurazione forniscono alcuni esempi di passaggi per i seguenti tipi di controller di ingresso:

- Ingresso Istio
- Controller di ingresso nginx

- Controller di ingresso OpenShift

Prima di iniziare

- Il necessario "controller di ingresso" dovrebbe essere già implementato.
- Il "classe di ingresso" corrispondente al controller di ingresso dovrebbe già essere creato.

Passaggi per l'ingresso di Istio

1. Configurare l'ingresso Istio.



Questa procedura presuppone che Istio venga distribuito utilizzando il profilo di configurazione "predefinito".

2. Raccogliere o creare il certificato e il file della chiave privata desiderati per Ingress Gateway.

È possibile utilizzare un certificato CA o autofirmato. Il nome comune deve essere l'indirizzo Astra (FQDN).

Esempio di comando:

```
openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout tls.key -out  
tls.crt
```

3. Crea un segreto `tls secret name` di tipo `kubernetes.io/tls` Per una chiave privata TLS e un certificato in `istio-system` namespace Come descritto in [TLS secrets \(segreti TLS\)](#).

Esempio di comando:

```
kubectl create secret tls [tls secret name] --key="tls.key"  
--cert="tls.crt" -n istio-system
```



Il nome del segreto deve corrispondere a `spec.tls.secretName` fornito in `istio-ingress.yaml` file.

4. Implementare una risorsa di ingresso in `netapp-acc` namespace (o personalizzato) che utilizza il tipo di risorsa `v1` per uno schema (`istio-Ingress.yaml` in questo esempio):

```

apiVersion: networking.k8s.io/v1
kind: IngressClass
metadata:
  name: istio
spec:
  controller: istio.io/ingress-controller
---
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: ingress
  namespace: [netapp-acc or custom namespace]
spec:
  ingressClassName: istio
  tls:
    - hosts:
      - <ACC address>
      secretName: [tls secret name]
  rules:
    - host: [ACC address]
      http:
        paths:
          - path: /
            pathType: Prefix
            backend:
              service:
                name: traefik
                port:
                  number: 80

```

5. Applicare le modifiche:

```
kubectl apply -f istio-Ingress.yaml
```

6. Controllare lo stato dell'ingresso:

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

Risposta:

NAME	CLASS	HOSTS	ADDRESS	PORTS	AGE
ingress	istio	astra.example.com	172.16.103.248	80, 443	1h

7. Completare l'installazione di Astra Control Center.

Procedura per il controller di ingresso Nginx

1. Creare un segreto di tipo `kubernetes.io/tls` Per una chiave privata TLS e un certificato in `netapp-acc` (o con nome personalizzato) come descritto in "Segreti TLS".
2. Implementare una risorsa `ingress` in `netapp-acc` namespace (o personalizzato) che utilizza il tipo di risorsa `v1` per uno schema (`nginx-Ingress.yaml` in questo esempio):

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: netapp-acc-ingress
  namespace: [netapp-acc or custom namespace]
spec:
  ingressClassName: [class name for nginx controller]
  tls:
  - hosts:
    - <ACC address>
    secretName: [tls secret name]
  rules:
  - host: <ACC address>
    http:
      paths:
      - path:
        backend:
          service:
            name: traefik
            port:
              number: 80
        pathType: ImplementationSpecific
```

3. Applicare le modifiche:

```
kubectl apply -f nginx-Ingress.yaml
```



NetApp consiglia di installare il controller `nginx` come implementazione piuttosto che come `daemonSet`.

Procedura per il controller di ingresso OpenShift

1. Procurarsi il certificato e ottenere la chiave, il certificato e i file CA pronti per l'uso con il percorso `OpenShift`.
2. Creare il percorso `OpenShift`:

```
oc create route edge --service=traefik --port=web -n [netapp-acc or
custom namespace] --insecure-policy=Redirect --hostname=<ACC address>
--cert=cert.pem --key=key.pem
```

Accedere all'interfaccia utente di Astra Control Center

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

Fasi

1. In un browser, immettere l'FQDN (compreso il `https://` prefisso) utilizzato in `astraAddress` in `astra_control_center.yaml` CR quando [Astra Control Center è stato installato](#).
2. Accettare i certificati autofirmati, se 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.yaml` CR quando [Astra Control Center è stato installato](#), seguito dalla password di configurazione iniziale (`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 stati ancora creati altri account utente amministrativi, contattare ["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.

Opzioni

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

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

- Per controllare l'output di Astra Control Center CR:

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

Cosa succederà

- (Opzionale) a seconda dell'ambiente, completare la post-installazione "[fasi di configurazione](#)".
- Completare l'implementazione eseguendo "[attività di installazione](#)".

Configurare un gestore esterno dei certificati

Se nel cluster Kubernetes esiste già un cert manager, è necessario eseguire alcuni passaggi preliminari in modo che Astra Control Center non installi il proprio cert manager.

Fasi

1. Verificare che sia installato un gestore dei certificati:

```
kubectl get pods -A | grep 'cert-manager'
```

Esempio di risposta:

```
cert-manager   essential-cert-manager-84446f49d5-sf2zd   1/1
Running        0     6d5h
cert-manager   essential-cert-manager-cainjector-66dc99cc56-9ldmt   1/1
Running        0     6d5h
cert-manager   essential-cert-manager-webhook-56b76db9cc-fjqrq     1/1
Running        0     6d5h
```

2. Creare una coppia certificato/chiave per astraAddress FQDN:

```
openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout tls.key -out
tls.crt
```

Esempio di risposta:

```
Generating a 2048 bit RSA private key
.....+++
.....+++
writing new private key to 'tls.key'
```

3. Creare un segreto con i file generati in precedenza:

```
kubectl create secret tls selfsigned-tls --key tls.key --cert tls.crt -n <cert-manager-namespace>
```

Esempio di risposta:

```
secret/selfsigned-tls created
```

4. Creare un ClusterIssuer file che è **esattamente** il seguente, ma include la posizione dello spazio dei nomi in cui si trova il cert-manager i pod sono installati:

```
apiVersion: cert-manager.io/v1
kind: ClusterIssuer
metadata:
  name: astra-ca-clusterissuer
  namespace: <cert-manager-namespace>
spec:
  ca:
    secretName: selfsigned-tls
```

```
kubectl apply -f ClusterIssuer.yaml
```

Esempio di risposta:

```
clusterissuer.cert-manager.io/astra-ca-clusterissuer created
```

5. Verificare che il ClusterIssuer è venuto in su correttamente. Ready deve essere True prima di procedere:

```
kubectl get ClusterIssuer
```

Esempio di risposta:

NAME	READY	AGE
astra-ca-clusterissuer	True	9s

6. Completare il "[Processo di installazione di Astra Control Center](#)". Esiste un "[Fase di configurazione richiesta per il cluster Astra Control Center YAML](#)" In cui si modifica il valore CRD per indicare che il gestore dei certificati è installato esternamente. È necessario completare questa fase durante l'installazione in modo che Astra Control Center riconosca il cert manager esterno.

Informazioni sul copyright

Copyright © 2023 NetApp, Inc. Tutti i diritti riservati. Stampato negli Stati Uniti d'America. Nessuna porzione di questo documento soggetta a copyright può essere riprodotta in qualsiasi formato o mezzo (grafico, elettronico o meccanico, inclusi fotocopie, registrazione, nastri o storage in un sistema elettronico) senza previo consenso scritto da parte del detentore del copyright.

Il software derivato dal materiale sottoposto a copyright di NetApp è soggetto alla seguente licenza e dichiarazione di non responsabilità:

IL PRESENTE SOFTWARE VIENE FORNITO DA NETAPP "COSÌ COM'È" E SENZA QUALSIVOGLIA TIPO DI GARANZIA IMPLICITA O ESPRESSA FRA CUI, A TITOLO ESEMPLIFICATIVO E NON ESAUSTIVO, GARANZIE IMPLICITE DI COMMERCIALIZZABILITÀ E IDONEITÀ PER UNO SCOPO SPECIFICO, CHE VENGONO DECLINATE DAL PRESENTE DOCUMENTO. NETAPP NON VERRÀ CONSIDERATA RESPONSABILE IN ALCUN CASO PER QUALSIVOGLIA DANNO DIRETTO, INDIRETTO, ACCIDENTALE, SPECIALE, ESEMPLARE E CONSEGUENZIALE (COMPRESI, A TITOLO ESEMPLIFICATIVO E NON ESAUSTIVO, PROCUREMENT O SOSTITUZIONE DI MERCI O SERVIZI, IMPOSSIBILITÀ DI UTILIZZO O PERDITA DI DATI O PROFITTI OPPURE INTERRUZIONE DELL'ATTIVITÀ AZIENDALE) CAUSATO IN QUALSIVOGLIA MODO O IN RELAZIONE A QUALUNQUE TEORIA DI RESPONSABILITÀ, SIA ESSA CONTRATTUALE, RIGOROSA O DOVUTA A INSOLVENZA (COMPRESA LA NEGLIGENZA O ALTRO) INSORTA IN QUALSIASI MODO ATTRAVERSO L'UTILIZZO DEL PRESENTE SOFTWARE ANCHE IN PRESENZA DI UN PREAVVISO CIRCA L'EVENTUALITÀ DI QUESTO TIPO DI DANNI.

NetApp si riserva il diritto di modificare in qualsiasi momento qualunque prodotto descritto nel presente documento senza fornire alcun preavviso. NetApp non si assume alcuna responsabilità circa l'utilizzo dei prodotti o materiali descritti nel presente documento, con l'eccezione di quanto concordato espressamente e per iscritto da NetApp. L'utilizzo o l'acquisto del presente prodotto non comporta il rilascio di una licenza nell'ambito di un qualche diritto di brevetto, marchio commerciale o altro diritto di proprietà intellettuale di NetApp.

Il prodotto descritto in questa guida può essere protetto da uno o più brevetti degli Stati Uniti, esteri o in attesa di approvazione.

LEGENDA PER I DIRITTI SOTTOPOSTI A LIMITAZIONE: l'utilizzo, la duplicazione o la divulgazione da parte degli enti governativi sono soggetti alle limitazioni indicate nel sottoparagrafo (b)(3) della clausola Rights in Technical Data and Computer Software del DFARS 252.227-7013 (FEB 2014) e FAR 52.227-19 (DIC 2007).

I dati contenuti nel presente documento riguardano un articolo commerciale (secondo la definizione data in FAR 2.101) e sono di proprietà di NetApp, Inc. Tutti i dati tecnici e il software NetApp forniti secondo i termini del presente Contratto sono articoli aventi natura commerciale, sviluppati con finanziamenti esclusivamente privati. Il governo statunitense ha una licenza irrevocabile limitata, non esclusiva, non trasferibile, non cedibile, mondiale, per l'utilizzo dei Dati esclusivamente in connessione con e a supporto di un contratto governativo statunitense in base al quale i Dati sono distribuiti. Con la sola esclusione di quanto indicato nel presente documento, i Dati non possono essere utilizzati, divulgati, riprodotti, modificati, visualizzati o mostrati senza la previa approvazione scritta di NetApp, Inc. I diritti di licenza del governo degli Stati Uniti per il Dipartimento della Difesa sono limitati ai diritti identificati nella clausola DFARS 252.227-7015(b) (FEB 2014).

Informazioni sul marchio commerciale

NETAPP, il logo NETAPP e i marchi elencati alla pagina <http://www.netapp.com/TM> sono marchi di NetApp, Inc. Gli altri nomi di aziende e prodotti potrebbero essere marchi dei rispettivi proprietari.