



Installare utilizzando l'operatore Trident

Astra Trident

NetApp
December 03, 2024

Sommario

- Installare utilizzando l'operatore Trident 1
 - Implementare manualmente l'operatore Trident (modalità standard) 1
 - Implementare manualmente l'operatore Trident (modalità offline) 6
 - Implementare l'operatore Trident utilizzando Helm (modalità standard) 12
 - Implementare l'operatore Trident utilizzando Helm (modalità offline) 16
 - Personalizzare l'installazione dell'operatore Trident 21

Installare utilizzando l'operatore Trident

Implementare manualmente l'operatore Trident (modalità standard)

È possibile implementare manualmente l'operatore Trident per installare Astra Trident. Questo processo si applica alle installazioni in cui le immagini container richieste da Astra Trident non sono memorizzate in un registro privato. Se si dispone di un registro delle immagini privato, utilizzare ["processo per l'implementazione offline"](#).

Informazioni critiche su Astra Trident 24,06

È necessario leggere le seguenti informazioni critiche su Astra Trident.

informazioni su Astra

- Kubernetes 1,31 è ora supportato in Astra Trident. Aggiornare Trident prima di aggiornare Kubernetes.
- Astra Trident applica rigorosamente l'utilizzo della configurazione multipath negli ambienti SAN, con un valore consigliato `find_multipaths: no` nel file `multipath.conf`.

L'utilizzo di una configurazione non multipathing o di `find_multipaths: yes` un valore OR `find_multipaths: smart` nel file `multipath.conf` determinerà errori di montaggio. Trident ha consigliato l'uso di `find_multipaths: no` fin dalla versione 21,07.

Implementare manualmente l'operatore Trident e installare Trident

Verificare ["panoramica dell'installazione"](#) di aver soddisfatto i prerequisiti di installazione e di aver selezionato l'opzione di installazione corretta per l'ambiente in uso.

Prima di iniziare

Prima di iniziare l'installazione, accedere all'host Linux e verificare che sia in corso la gestione di un funzionante ["Cluster Kubernetes supportato"](#) e che si disponga del Privileges necessario.



Con OpenShift, utilizzare `oc` invece di `kubectl` tutti gli esempi che seguono e accedere come **system:admin** eseguendo prima `oc login -u system:admin` o `oc login -u kube-admin`.

1. Verificare la versione di Kubernetes:

```
kubectl version
```

2. Verificare i privilegi di amministratore del cluster:

```
kubectl auth can-i '*' '*' --all-namespaces
```

3. Verificare che sia possibile avviare un pod che utilizza un'immagine da Docker Hub e raggiungere il sistema di storage tramite la rete pod:

```
kubectl run -i --tty ping --image=busybox --restart=Never --rm -- \
ping <management IP>
```

Fase 1: Scaricare il pacchetto di installazione di Trident

Il pacchetto di installazione di Astra Trident contiene tutto il necessario per implementare l'operatore Trident e installare Astra Trident. Scaricare ed estrarre la versione più recente del programma di installazione di Trident da "[La sezione Assets su GitHub](#)".

```
wget https://github.com/NetApp/trident/releases/download/v24.06.0/trident-
installer-24.06.0.tar.gz
tar -xf trident-installer-24.06.0.tar.gz
cd trident-installer
```

Fase 2: Creare il TridentOrchestrator CRD

Creare la TridentOrchestrator definizione di risorsa personalizzata (CRD). In seguito, verrà creata una TridentOrchestrator risorsa personalizzata. Utilizzare la versione CRD YAML appropriata in per creare il TridentOrchestrator CRD `deploy/crds`.

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

Fase 3: Implementare l'operatore Trident

Il programma di installazione di Astra Trident fornisce un file bundle che può essere utilizzato per installare l'operatore e creare oggetti associati. Il file bundle è un modo semplice per implementare l'operatore e installare Astra Trident utilizzando una configurazione predefinita.

- Per i cluster che eseguono Kubernetes 1,24, utilizzare `bundle_pre_1_25.yaml`.

- Per i cluster che eseguono Kubernetes 1,25 o versione successiva, utilizzare `bundle_post_1_25.yaml`.

Prima di iniziare

- Per impostazione predefinita, il programma di installazione di Trident implementa l'operatore nello `trident` spazio dei nomi. Se lo `trident` spazio dei nomi non esiste, crearlo utilizzando:

```
kubectl apply -f deploy/namespace.yaml
```

- Per distribuire l'operatore in uno spazio dei nomi diverso dallo `trident` spazio dei nomi, aggiornare `serviceaccount.yaml` `clusterrolebinding.yaml` e `operator.yaml` generare il file bundle utilizzando `kustomization.yaml`.

- a. Crea il `kustomization.yaml` usando il seguente comando dove `<bundle.yaml>` è `bundle_pre_1_25.yaml` o `bundle_post_1_25.yaml` si basa sulla tua versione di Kubernetes.

```
cp deploy/kustomization_<bundle.yaml> deploy/kustomization.yaml
```

- b. Compila il bundle usando il seguente comando dove `<bundle.yaml>` si trova `bundle_pre_1_25.yaml` o `bundle_post_1_25.yaml` si basa sulla tua versione di Kubernetes.

```
kubectl kustomize deploy/ > deploy/<bundle.yaml>
```

Fasi

1. Creare le risorse e implementare l'operatore:

```
kubectl create -f deploy/<bundle.yaml>
```

2. Verificare che l'operatore, l'implementazione e i replicaset siano stati creati.

```
kubectl get all -n <operator-namespace>
```



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

Fase 4: Creare `TridentOrchestrator` e installare Trident

Ora puoi creare `TridentOrchestrator` e installare Astra Trident. Facoltativamente, è possibile "[Personalizzare l'installazione di Trident](#)" utilizzare gli attributi nella `TridentOrchestrator` specifica.

```

kubect1 create -f deploy/crds/tridentorchestrator_cr.yaml
tridentorchestrator.trident.netapp.io/trident created

kubect1 describe torc trident

Name:          trident
Namespace:
Labels:        <none>
Annotations:   <none>
API Version:   trident.netapp.io/v1
Kind:          TridentOrchestrator
...
Spec:
  Debug:       true
  Namespace:   trident
Status:
  Current Installation Params:
    IPv6:                false
    Autosupport Hostname:
    Autosupport Image:   netapp/trident-autosupport:24.06
    Autosupport Proxy:
    Autosupport Serial Number:
    Debug:               true
    Image Pull Secrets:
    Image Registry:
    k8sTimeout:         30
    Kubelet Dir:        /var/lib/kubelet
    Log Format:          text
    Silence Autosupport: false
    Trident Image:      netapp/trident:24.06.0
  Message:             Trident installed Namespace:
trident
  Status:              Installed
  Version:             v24.06.0
Events:
  Type Reason Age From Message ---- -
-----
Normal
Installing 74s trident-operator.netapp.io Installing Trident Normal
Installed 67s trident-operator.netapp.io Trident installed

```

Verificare l'installazione

Esistono diversi modi per verificare l'installazione.

Uso dello `TridentOrchestrator` stato

Lo stato di `TridentOrchestrator` indica se l'installazione è riuscita e visualizza la versione di Trident installata. Durante l'installazione, lo stato di `TridentOrchestrator` cambia da `Installing` a `Installed`. Se si osserva `Failed` lo stato e l'operatore non è in grado di recuperare da solo, "[controllare i registri](#)".

Stato	Descrizione
Installazione in corso	L'operatore sta installando Astra Trident utilizzando questo <code>TridentOrchestrator</code> CR.
Installato	Astra Trident è stato installato correttamente.
Disinstallazione in corso	L'operatore sta disinstallando Astra Trident, perché <code>spec.uninstall=true</code> .
Disinstallato	Astra Trident disinstallato.
Non riuscito	L'operatore non ha potuto installare, applicare patch, aggiornare o disinstallare Astra Trident; l'operatore tenterà automaticamente di eseguire il ripristino da questo stato. Se lo stato persiste, è necessario eseguire la risoluzione dei problemi.
Aggiornamento in corso	L'operatore sta aggiornando un'installazione esistente.
Errore	<code>TridentOrchestrator</code> Non viene utilizzato. Un'altra esiste già.

Utilizzo dello stato di creazione del pod

È possibile verificare se l'installazione di Astra Trident è stata completata esaminando lo stato dei pod creati:

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

NAME	READY	STATUS	RESTARTS
AGE			
trident-controller-7d466bf5c7-v4cpw 1m	6/6	Running	0
trident-node-linux-mr6zc 1m	2/2	Running	0
trident-node-linux-xrp7w 1m	2/2	Running	0
trident-node-linux-zh2jt 1m	2/2	Running	0
trident-operator-766f7b8658-ldzsv 3m	1/1	Running	0

Uso `tridentctl`

Puoi usare `tridentctl` per controllare la versione di Astra Trident installata.

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

```
+-----+-----+
| SERVER VERSION | CLIENT VERSION |
+-----+-----+
| 24.06.0        | 24.06.0        |
+-----+-----+
```

Implementare manualmente l'operatore Trident (modalità offline)

È possibile implementare manualmente l'operatore Trident per installare Astra Trident. Questo processo si applica alle installazioni in cui le immagini container richieste da Astra Trident sono memorizzate in un registro privato. Se non si dispone di un registro delle immagini private, utilizzare ["processo per l'implementazione standard"](#).

Informazioni critiche su Astra Trident 24,06

È necessario leggere le seguenti informazioni critiche su Astra Trident.

informazioni su Astra

- Kubernetes 1,31 è ora supportato in Astra Trident. Aggiornare Trident prima di aggiornare Kubernetes.
- Astra Trident applica rigorosamente l'utilizzo della configurazione multipath negli ambienti SAN, con un valore consigliato `find_multipaths: no` nel file `multipath.conf`.

L'utilizzo di una configurazione non multipathing o di `find_multipaths: yes` un valore OR `find_multipaths: smart` nel file `multipath.conf` determinerà errori di montaggio. Trident ha consigliato l'uso di `find_multipaths: no` fin dalla versione 21,07.

Implementare manualmente l'operatore Trident e installare Trident

Verificare ["panoramica dell'installazione"](#) di aver soddisfatto i prerequisiti di installazione e di aver selezionato l'opzione di installazione corretta per l'ambiente in uso.

Prima di iniziare

Accedere all'host Linux e verificare che sia in corso la gestione di un funzionante e ["Cluster Kubernetes supportato"](#) che si disponga del Privileges necessario.



Con OpenShift, utilizzare `oc` invece di `kubectl` tutti gli esempi che seguono e accedere come **system:admin** eseguendo prima `oc login -u system:admin` o `oc login -u kube-admin`.

1. Verificare la versione di Kubernetes:

```
kubectl version
```

2. Verificare i privilegi di amministratore del cluster:

```
kubectl auth can-i '*' '*' --all-namespaces
```

3. Verificare che sia possibile avviare un pod che utilizza un'immagine da Docker Hub e raggiungere il sistema di storage tramite la rete pod:

```
kubectl run -i --tty ping --image=busybox --restart=Never --rm -- \
ping <management IP>
```

Fase 1: Scaricare il pacchetto di installazione di Trident

Il pacchetto di installazione di Astra Trident contiene tutto il necessario per implementare l'operatore Trident e installare Astra Trident. Scaricare ed estrarre la versione più recente del programma di installazione di Trident da ["La sezione Assets su GitHub"](#).

```
wget https://github.com/NetApp/trident/releases/download/v24.06.0/trident-
installer-24.06.0.tar.gz
tar -xf trident-installer-24.06.0.tar.gz
cd trident-installer
```

Fase 2: Creare il TridentOrchestrator CRD

Creare la TridentOrchestrator definizione di risorsa personalizzata (CRD). In seguito, verrà creata una TridentOrchestrator risorsa personalizzata. Utilizzare la versione CRD YAML appropriata in `deploy/crds` per creare il TridentOrchestrator CRD:

```
kubectl create -f deploy/crds/<VERSION>.yaml
```

Fase 3: Aggiornare la posizione del registro nell'operatore

In `/deploy/operator.yaml`, eseguire l'aggiornamento `image: docker.io/netapp/trident-operator:24.06.0` per riflettere la posizione del registro delle immagini. Lo ["Immagini Trident e CSI"](#) può trovarsi in un registro o in registri diversi, ma tutte le immagini CSI devono trovarsi nello stesso registro. Ad esempio:

- `image: <your-registry>/trident-operator:24.06.0` se tutte le immagini si trovano in un unico registro.

- `image: <your-registry>/netapp/trident-operator:24.06.0` Se l'immagine Trident si trova in un registro diverso dalle immagini CSI.

Fase 4: Implementare l'operatore Trident

Il programma di installazione di Astra Trident fornisce un file bundle che può essere utilizzato per installare l'operatore e creare oggetti associati. Il file bundle è un modo semplice per implementare l'operatore e installare Astra Trident utilizzando una configurazione predefinita.

- Per i cluster che eseguono Kubernetes 1,24, utilizzare `bundle_pre_1_25.yaml`.
- Per i cluster che eseguono Kubernetes 1,25 o versione successiva, utilizzare `bundle_post_1_25.yaml`.

Prima di iniziare

- Per impostazione predefinita, il programma di installazione di Trident implementa l'operatore nello `trident` spazio dei nomi. Se lo `trident` spazio dei nomi non esiste, crearlo utilizzando:

```
kubectl apply -f deploy/namespace.yaml
```

- Per distribuire l'operatore in uno spazio dei nomi diverso dallo `trident` spazio dei nomi, aggiornare `serviceaccount.yaml` `clusterrolebinding.yaml` e `operator.yaml` generare il file bundle utilizzando `kustomization.yaml`.

- a. Crea il `kustomization.yaml` usando il seguente comando dove `<bundle.yaml>` è `bundle_pre_1_25.yaml` o `bundle_post_1_25.yaml` si basa sulla tua versione di Kubernetes.

```
cp deploy/kustomization_<bundle.yaml> deploy/kustomization.yaml
```

- b. Compila il bundle usando il seguente comando dove `<bundle.yaml>` si trova `bundle_pre_1_25.yaml` o `bundle_post_1_25.yaml` si basa sulla tua versione di Kubernetes.

```
kubectl kustomize deploy/ > deploy/<bundle.yaml>
```

Fasi

1. Creare le risorse e implementare l'operatore:

```
kubectl create -f deploy/<bundle.yaml>
```

2. Verificare che l'operatore, l'implementazione e i replicaset siano stati creati.

```
kubectl get all -n <operator-namespace>
```



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

Passaggio 5: Aggiornare la posizione del Registro di sistema dell'immagine in `TridentOrchestrator`

Lo "Immagini Trident e CSI" può trovarsi in un registro o in registri diversi, ma tutte le immagini CSI devono trovarsi nello stesso registro. Aggiornare `deploy/crds/tridentorchestrator_cr.yaml` per aggiungere le specifiche di posizione aggiuntive in base alla configurazione del Registro di sistema.

Immagini in un registro

```
imageRegistry: "<your-registry>"
autosupportImage: "<your-registry>/trident-autosupport:24.06"
tridentImage: "<your-registry>/trident:24.06.0"
```

Immagini in diversi registri

È necessario aggiungere `sig-storage` al `imageRegistry` per utilizzare posizioni del Registro di sistema diverse.

```
imageRegistry: "<your-registry>/sig-storage"
autosupportImage: "<your-registry>/netapp/trident-autosupport:24.06"
tridentImage: "<your-registry>/netapp/trident:24.06.0"
```

Fase 6: Creare `TridentOrchestrator` e installare Trident

Ora puoi creare `TridentOrchestrator` e installare Astra Trident. Facoltativamente, è possibile utilizzare ulteriormente "[Personalizzare l'installazione di Trident](#)" gli attributi nella `TridentOrchestrator` specifica. L'esempio seguente mostra un'installazione in cui le immagini Trident e CSI si trovano in diversi registri.

```
kubectl create -f deploy/crds/tridentorchestrator_cr.yaml
tridentorchestrator.trident.netapp.io/trident created
```

```
kubectl describe torc trident
```

```
Name:          trident
Namespace:
Labels:        <none>
Annotations:   <none>
API Version:   trident.netapp.io/v1
Kind:          TridentOrchestrator
...
Spec:
  Autosupport Image: <your-registry>/netapp/trident-autosupport:24.06
  Debug:             true
  Image Registry:    <your-registry>/sig-storage
  Namespace:        trident
  Trident Image:     <your-registry>/netapp/trident:24.06.0
```

```
Status:
```

```
  Current Installation Params:
```

```
    IPv6:                false
    Autosupport Hostname:
    Autosupport Image:    <your-registry>/netapp/trident-
autosupport:24.06
    Autosupport Proxy:
    Autosupport Serial Number:
    Debug:                true
    Http Request Timeout: 90s
    Image Pull Secrets:
    Image Registry:       <your-registry>/sig-storage
    k8sTimeout:           30
    Kubelet Dir:          /var/lib/kubelet
    Log Format:            text
    Probe Port:           17546
    Silence Autosupport:  false
    Trident Image:        <your-registry>/netapp/trident:24.06.0
  Message:                Trident installed
  Namespace:              trident
  Status:                 Installed
  Version:                v24.06.0
```

```
Events:
```

```
  Type Reason Age From Message ---- -
-----
  Installing 74s trident-operator.netapp.io Installing Trident Normal
  Installed 67s trident-operator.netapp.io Trident installed
```

Verificare l'installazione

Esistono diversi modi per verificare l'installazione.

Uso dello `TridentOrchestrator` stato

Lo stato di `TridentOrchestrator` indica se l'installazione è riuscita e visualizza la versione di Trident installata. Durante l'installazione, lo stato di `TridentOrchestrator` cambia da `Installing` a `Installed`. Se si osserva `Failed` lo stato e l'operatore non è in grado di recuperare da solo, "[controllare i registri](#)".

Stato	Descrizione
Installazione in corso	L'operatore sta installando Astra Trident utilizzando questo <code>TridentOrchestrator</code> CR.
Installato	Astra Trident è stato installato correttamente.
Disinstallazione in corso	L'operatore sta disinstallando Astra Trident, perché <code>spec.uninstall=true</code> .
Disinstallato	Astra Trident disinstallato.
Non riuscito	L'operatore non ha potuto installare, applicare patch, aggiornare o disinstallare Astra Trident; l'operatore tenterà automaticamente di eseguire il ripristino da questo stato. Se lo stato persiste, è necessario eseguire la risoluzione dei problemi.
Aggiornamento in corso	L'operatore sta aggiornando un'installazione esistente.
Errore	<code>TridentOrchestrator</code> Non viene utilizzato. Un'altra esiste già.

Utilizzo dello stato di creazione del pod

È possibile verificare se l'installazione di Astra Trident è stata completata esaminando lo stato dei pod creati:

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

NAME	READY	STATUS	RESTARTS
AGE			
trident-controller-7d466bf5c7-v4cpw	6/6	Running	0
1m			
trident-node-linux-mr6zc	2/2	Running	0
1m			
trident-node-linux-xrp7w	2/2	Running	0
1m			
trident-node-linux-zh2jt	2/2	Running	0
1m			
trident-operator-766f7b8658-ldzsv	1/1	Running	0
3m			

Uso tridentctl

Puoi usare `tridentctl` per controllare la versione di Astra Trident installata.

```
./tridentctl -n trident version

+-----+-----+
| SERVER VERSION | CLIENT VERSION |
+-----+-----+
| 24.06.0       | 24.06.0       |
+-----+-----+
```

Implementare l'operatore Trident utilizzando Helm (modalità standard)

È possibile implementare l'operatore Trident e installare Astra Trident utilizzando Helm. Questo processo si applica alle installazioni in cui le immagini container richieste da Astra Trident non sono memorizzate in un registro privato. Se si dispone di un registro delle immagini privato, utilizzare ["processo per l'implementazione offline"](#).

Informazioni critiche su Astra Trident 24,06

È necessario leggere le seguenti informazioni critiche su Astra Trident.

** informazioni su Astra **

- Kubernetes 1,31 è ora supportato in Astra Trident. Aggiornare Trident prima di aggiornare Kubernetes.
- Astra Trident applica rigorosamente l'utilizzo della configurazione multipath negli ambienti SAN, con un valore consigliato `find_multipaths: no` nel file `multipath.conf`.

L'utilizzo di una configurazione non multipathing o di `find_multipaths: yes` un valore OR `find_multipaths: smart` nel file `multipath.conf` determinerà errori di montaggio. Trident ha consigliato l'uso di `find_multipaths: no` fin dalla versione 21,07.

Implementare l'operatore Trident e installare Astra Trident utilizzando Helm

Utilizzando Trident ["Grafico di comando"](#) è possibile distribuire l'operatore Trident e installare Trident in un'unica fase.

Verificare ["panoramica dell'installazione"](#) di aver soddisfatto i prerequisiti di installazione e di aver selezionato l'opzione di installazione corretta per l'ambiente in uso.

Prima di iniziare

Oltre al ["prerequisiti per l'implementazione"](#) necessario ["Helm versione 3"](#).

Fasi

1. Aggiungere il repository Astra Trident Helm:

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

2. USA `helm install` e specifica un nome per la tua implementazione come nel seguente esempio dove si trova `100.2404.0` la versione di Astra Trident che stai installando.

```
helm install <name> netapp-trident/trident-operator --version 100.2406.0  
--create-namespace --namespace <trident-namespace>
```



Se è già stato creato uno spazio dei nomi per Trident, il `--create-namespace` parametro non creerà uno spazio dei nomi aggiuntivo.

È possibile utilizzare `helm list` per esaminare i dettagli dell'installazione come nome, spazio dei nomi, grafico, stato, versione dell'app, e numero di revisione.

Passare i dati di configurazione durante l'installazione

Esistono due modi per passare i dati di configurazione durante l'installazione:

Opzione	Descrizione
<code>--values (o -f)</code>	Specificare un file YAML con override. Questo valore può essere specificato più volte e il file più a destra avrà la precedenza.
<code>--set</code>	Specificare le sostituzioni sulla riga di comando.

Ad esempio, per modificare il valore predefinito di `debug`, eseguire il comando seguente `--set` dove si trova `100.2406.0` la versione di Astra Trident che si sta installando:

```
helm install <name> netapp-trident/trident-operator --version 100.2406.0  
--create-namespace --namespace trident --set tridentDebug=true
```

Opzioni di configurazione

Questa tabella e il `values.yaml` file, che fa parte del grafico Helm, forniscono l'elenco delle chiavi e i loro valori predefiniti.

Opzione	Descrizione	Predefinito
<code>nodeSelector</code>	Etichette dei nodi per l'assegnazione dei pod	
<code>podAnnotations</code>	Annotazioni Pod	

Opzione	Descrizione	Predefinito
deploymentAnnotations	Annotazioni di implementazione	
tolerations	Pedaggi per l'assegnazione del pod	
affinity	Affinità per l'assegnazione del pod	
tridentControllerPluginNodeSelector	Selettori di nodi aggiuntivi per i pod. Per ulteriori informazioni, fare riferimento alla Comprensione dei pod controller e dei pod di nodi sezione.	
tridentControllerPluginTolerations	Ignora le tolleranze Kubernetes per i pod. Per ulteriori informazioni, fare riferimento alla Comprensione dei pod controller e dei pod di nodi sezione.	
tridentNodePluginNodeSelector	Selettori di nodi aggiuntivi per i pod. Per ulteriori informazioni, fare riferimento alla Comprensione dei pod controller e dei pod di nodi sezione.	
tridentNodePluginTolerations	Ignora le tolleranze Kubernetes per i pod. Per ulteriori informazioni, fare riferimento alla Comprensione dei pod controller e dei pod di nodi sezione.	
imageRegistry	Identifica il Registro di sistema per <code>trident-operator</code> , <code>trident</code> e altre immagini. Lasciare vuoto per accettare l'impostazione predefinita.	""
imagePullPolicy	Consente di impostare il criterio di pull dell'immagine per <code>trident-operator</code> .	IfNotPresent
imagePullSecrets	Consente di impostare i segreti di estrazione delle immagini per <code>trident-operator</code> , <code>trident</code> e altre immagini.	
kubeletDir	Consente di ignorare la posizione host dello stato interno del kubelet.	"/var/lib/kubelet"
operatorLogLevel	Consente di impostare il livello di registrazione dell'operatore Trident su: <code>trace</code> , <code>debug</code> , <code>info</code> , <code>warn</code> o <code>fatal</code> .	"info"
operatorDebug	Consente di impostare il livello di log dell'operatore Trident su <code>debug</code> .	true
operatorImage	Consente la sostituzione completa dell'immagine per <code>trident-operator</code> .	""
operatorImageTag	Consente di sovrascrivere il tag dell' <code>trident-operator</code> immagine.	""
tridentIPv6	Consente ad Astra Trident di funzionare nei cluster IPv6.	false
tridentK8sTimeout	Esegue l'override del timeout predefinito di 30 secondi per la maggior parte delle operazioni API di Kubernetes (se diverso da zero, in secondi).	0

Opzione	Descrizione	Predefinito
tridentHttpRequestTimeout	Ignora il timeout predefinito di 90 secondi per le richieste HTTP, con 0s una durata infinita per il timeout. Non sono consentiti valori negativi.	"90s"
tridentSilenceAutoSupport	Consente di disattivare il reporting periodico AutoSupport di Astra Trident.	false
tridentAutoSupportImageTag	Consente di ignorare il tag dell'immagine per il contenitore Astra Trident AutoSupport.	<version>
tridentAutoSupportProxy	Consente al container Astra Trident AutoSupport di telefonare a casa tramite un proxy HTTP.	""
tridentLogFormat	Imposta il formato di registrazione Astra Trident (text o json).	"text"
tridentDisableAuditLog	Disattiva l'audit logger Astra Trident.	true
tridentLogLevel	Consente di impostare il livello di registro di Astra Trident su: trace, debug, info, warn error o fatal.	"info"
tridentDebug	Consente di impostare il livello di registro di Astra Trident su debug.	false
tridentLogWorkflows	Consente di attivare specifici flussi di lavoro di Astra Trident per la registrazione delle tracce o la soppressione dei log.	""
tridentLogLayers	Consente di attivare specifici livelli Astra Trident per la registrazione delle tracce o la soppressione dei log.	""
tridentImage	Consente l'override completo dell'immagine per Astra Trident.	""
tridentImageTag	Consente di ignorare il tag dell'immagine per Astra Trident.	""
tridentProbePort	Consente di ignorare la porta predefinita utilizzata per le sonde liveness/readiness Kubernetes.	""
windows	Consente di installare Astra Trident sul nodo di lavoro Windows.	false
enableForceDetach	Consente di attivare la funzione di distacco forzato.	false
excludePodSecurityPolicy	Esclude la creazione della policy di sicurezza del pod operatore.	false
cloudProvider	Impostato su "Azure" quando si utilizzano identità gestite o un'identità cloud su un cluster AKS. Impostato su "AWS" quando si utilizza un'identità cloud su un cluster EKS.	""

Opzione	Descrizione	Predefinito
cloudIdentity	Impostare sull'identità del workload ("Azure.workload.Identity/client-id: Xxxxxxxx-xxxx-xxxx-xxxx-xxxx-xxxx-xxxx-xxxx-xxxx-xxxx-xxxx") quando si utilizza l'identità cloud su un cluster AKS. Impostato sul ruolo AWS IAM ("eks.amazonaws.com/role-arn: arn:AWS:iam::123456:role/astatadent-role") quando si utilizza l'identità cloud su un cluster EKS.	""
iscsiSelfHealingInterval	Intervallo in cui viene richiamata la riparazione automatica iSCSI.	5m0s
iscsiSelfHealingWaitTime	La durata dopo la quale l'autoriparazione iSCSI avvia un tentativo di risolvere una sessione obsoleta eseguendo una disconnessione e un login successivo.	7m0s

Comprensione dei pod controller e dei pod di nodi

Astra Trident viene eseguito come singolo pod controller, più un pod di nodi su ciascun nodo di lavoro nel cluster. Il pod nodo deve essere in esecuzione su qualsiasi host in cui si desidera montare un volume Astra Trident.

Kubernetes ["selettori di nodi"](#) e ["tollerazioni e contaminazioni"](#) sono utilizzati per vincolare un pod all'esecuzione su un nodo specifico o preferito. Utilizzando `ControllerPlugin` e `NodePlugin`, è possibile specificare vincoli e sostituzioni.

- Il plug-in del controller gestisce il provisioning e la gestione dei volumi, ad esempio snapshot e ridimensionamento.
- Il plug-in del nodo gestisce il collegamento dello storage al nodo.

Implementare l'operatore Trident utilizzando Helm (modalità offline)

È possibile implementare l'operatore Trident e installare Astra Trident utilizzando Helm. Questo processo si applica alle installazioni in cui le immagini container richieste da Astra Trident sono memorizzate in un registro privato. Se non si dispone di un registro delle immagini private, utilizzare ["processo per l'implementazione standard"](#).

Informazioni critiche su Astra Trident 24,06

È necessario leggere le seguenti informazioni critiche su Astra Trident.

 informazioni su Astra

- Kubernetes 1,31 è ora supportato in Astra Trident. Aggiornare Trident prima di aggiornare Kubernetes.
- Astra Trident applica rigorosamente l'utilizzo della configurazione multipath negli ambienti SAN, con un valore consigliato `find_multipaths: no` nel file `multipath.conf`.

L'utilizzo di una configurazione non multipathing o di `find_multipaths: yes` un valore OR `find_multipaths: smart` nel file `multipath.conf` determinerà errori di montaggio. Trident ha consigliato l'uso di `find_multipaths: no` fin dalla versione 21,07.

Implementare l'operatore Trident e installare Astra Trident utilizzando Helm

Utilizzando Trident "[Grafico di comando](#)" è possibile distribuire l'operatore Trident e installare Trident in un'unica fase.

Verificare "[panoramica dell'installazione](#)" di aver soddisfatto i prerequisiti di installazione e di aver selezionato l'opzione di installazione corretta per l'ambiente in uso.

Prima di iniziare

Oltre al "[prerequisiti per l'implementazione](#)" necessario "[Helm versione 3](#)".

Fasi

1. Aggiungere il repository Astra Trident Helm:

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

2. Utilizzare `helm install` e specificare un nome per la distribuzione e la posizione del Registro di sistema dell'immagine. Lo "[Immagine Trident e CSI](#)" può trovarsi in un registro o in registri diversi, ma tutte le immagini CSI devono trovarsi nello stesso registro. Negli esempi, `100.2406.0` è la versione di Astra Trident che stai installando.

Immagini in un registro

```
helm install <name> netapp-trident/trident-operator --version
100.2406.0 --set imageRegistry=<your-registry> --create-namespace
--namespace <trident-namespace>
```

Immagini in diversi registri

È necessario aggiungere sig-storage al imageRegistry per utilizzare posizioni del Registro di sistema diverse.

```
helm install <name> netapp-trident/trident-operator --version
100.2406.0 --set imageRegistry=<your-registry>/sig-storage --set
operatorImage=<your-registry>/netapp/trident-operator:24.06.0 --set
tridentAutosupportImage=<your-registry>/netapp/trident-
autosupport:24.06 --set tridentImage=<your-
registry>/netapp/trident:24.06.0 --create-namespace --namespace
<trident-namespace>
```



Se è già stato creato uno spazio dei nomi per Trident, il --create-namespace parametro non creerà uno spazio dei nomi aggiuntivo.

È possibile utilizzare `helm list` per esaminare i dettagli dell'installazione come nome, spazio dei nomi, grafico, stato, versione dell'app, e numero di revisione.

Passare i dati di configurazione durante l'installazione

Esistono due modi per passare i dati di configurazione durante l'installazione:

Opzione	Descrizione
<code>--values (o -f)</code>	Specificare un file YAML con override. Questo valore può essere specificato più volte e il file più a destra avrà la precedenza.
<code>--set</code>	Specificare le sostituzioni sulla riga di comando.

Ad esempio, per modificare il valore predefinito di debug, eseguire il comando seguente `--set` dove si trova 100.2406.0 la versione di Astra Trident che si sta installando:

```
helm install <name> netapp-trident/trident-operator --version 100.2406.0
--create-namespace --namespace trident --set tridentDebug=true
```

Opzioni di configurazione

Questa tabella e il `values.yaml` file, che fa parte del grafico Helm, forniscono l'elenco delle chiavi e i loro valori predefiniti.

Opzione	Descrizione	Predefinito
<code>nodeSelector</code>	Etichette dei nodi per l'assegnazione dei pod	
<code>podAnnotations</code>	Annotazioni Pod	
<code>deploymentAnnotations</code>	Annotazioni di implementazione	
<code>tolerations</code>	Pedaggi per l'assegnazione del pod	
<code>affinity</code>	Affinità per l'assegnazione del pod	
<code>tridentControllerPluginNodeSelector</code>	Selettori di nodi aggiuntivi per i pod. Per ulteriori informazioni, fare riferimento alla " Comprensione dei pod controller e dei pod di nodi " sezione.	
<code>tridentControllerPluginTolerations</code>	Ignora le tolleranze Kubernetes per i pod. Per ulteriori informazioni, fare riferimento alla " Comprensione dei pod controller e dei pod di nodi " sezione.	
<code>tridentNodePluginNodeSelector</code>	Selettori di nodi aggiuntivi per i pod. Per ulteriori informazioni, fare riferimento alla " Comprensione dei pod controller e dei pod di nodi " sezione.	
<code>tridentNodePluginTolerations</code>	Ignora le tolleranze Kubernetes per i pod. Per ulteriori informazioni, fare riferimento alla " Comprensione dei pod controller e dei pod di nodi " sezione.	
<code>imageRegistry</code>	Identifica il Registro di sistema per <code>trident-operator</code> , <code>trident</code> e altre immagini. Lasciare vuoto per accettare l'impostazione predefinita.	""
<code>imagePullPolicy</code>	Consente di impostare il criterio di pull dell'immagine per <code>trident-operator</code> .	IfNotPresent
<code>imagePullSecrets</code>	Consente di impostare i segreti di estrazione delle immagini per <code>trident-operator</code> , <code>trident</code> e altre immagini.	
<code>kubeletDir</code>	Consente di ignorare la posizione host dello stato interno del kubelet.	"/var/lib/kubelet"

Opzione	Descrizione	Predefinito
operatorLogLevel	Consente di impostare il livello di registrazione dell'operatore Trident su: trace, debug, , info, warn error O fatal.	"info"
operatorDebug	Consente di impostare il livello di log dell'operatore Trident su debug.	true
operatorImage	Consente la sostituzione completa dell'immagine per trident-operator.	""
operatorImageTag	Consente di sovrascrivere il tag dell' `trident-operator` immagine.	""
tridentIPv6	Consente ad Astra Trident di funzionare nei cluster IPv6.	false
tridentK8sTimeout	Esegue l'override del timeout predefinito di 30 secondi per la maggior parte delle operazioni API di Kubernetes (se diverso da zero, in secondi).	0
tridentHttpRequestTimeout	Ignora il timeout predefinito di 90 secondi per le richieste HTTP, con 0s una durata infinita per il timeout. Non sono consentiti valori negativi.	"90s"
tridentSilenceAutosupport	Consente di disattivare il reporting periodico AutoSupport di Astra Trident.	false
tridentAutosupportImageTag	Consente di ignorare il tag dell'immagine per il contenitore Astra Trident AutoSupport.	<version>
tridentAutosupportProxy	Consente al container Astra Trident AutoSupport di telefonare a casa tramite un proxy HTTP.	""
tridentLogFormat	Imposta il formato di registrazione Astra Trident (text`o `json).	"text"
tridentDisableAuditLog	Disattiva l'audit logger Astra Trident.	true
tridentLogLevel	Consente di impostare il livello di registro di Astra Trident su: trace, debug, info, , warn error O fatal.	"info"
tridentDebug	Consente di impostare il livello di registro di Astra Trident su debug.	false

Opzione	Descrizione	Predefinito
<code>tridentLogWorkflows</code>	Consente di attivare specifici flussi di lavoro di Astra Trident per la registrazione delle tracce o la soppressione dei log.	""
<code>tridentLogLayers</code>	Consente di attivare specifici livelli Astra Trident per la registrazione delle tracce o la soppressione dei log.	""
<code>tridentImage</code>	Consente l'override completo dell'immagine per Astra Trident.	""
<code>tridentImageTag</code>	Consente di ignorare il tag dell'immagine per Astra Trident.	""
<code>tridentProbePort</code>	Consente di ignorare la porta predefinita utilizzata per le sonde liveness/readiness Kubernetes.	""
<code>windows</code>	Consente di installare Astra Trident sul nodo di lavoro Windows.	false
<code>enableForceDetach</code>	Consente di attivare la funzione di distacco forzato.	false
<code>excludePodSecurityPolicy</code>	Esclude la creazione della policy di sicurezza del pod operatore.	false

Personalizzare l'installazione dell'operatore Trident

L'operatore Trident consente di personalizzare l'installazione di Astra Trident utilizzando gli attributi nelle `TridentOrchestrator` specifiche. Se si desidera personalizzare l'installazione oltre gli `TridentOrchestrator` argomenti consentiti, utilizzare `tridentctl` per generare manifesti YAML personalizzati da modificare in base alle necessità.

Comprensione dei pod controller e dei pod di nodi

Astra Trident viene eseguito come singolo pod controller, più un pod di nodi su ciascun nodo di lavoro nel cluster. Il pod nodo deve essere in esecuzione su qualsiasi host in cui si desidera montare un volume Astra Trident.

Kubernetes "[selettori di nodi](#)" e "[tollerazioni e contaminazioni](#)" sono utilizzati per vincolare un pod all'esecuzione su un nodo specifico o preferito. Utilizzando `ControllerPlugin` e `NodePlugin`, è possibile specificare vincoli e sostituzioni.

- Il plug-in del controller gestisce il provisioning e la gestione dei volumi, ad esempio snapshot e ridimensionamento.
- Il plug-in del nodo gestisce il collegamento dello storage al nodo.

Opzioni di configurazione



`spec.namespace` Viene specificato in `TridentOrchestrator` per indicare lo spazio dei nomi in cui è installato Astra Trident. Questo parametro **non può essere aggiornato dopo l'installazione di Astra Trident**. Se si tenta di eseguire questa operazione, `TridentOrchestrator` lo stato viene modificato in `Failed`. Astra Trident non deve essere migrato tra spazi dei nomi.

Questa tabella fornisce i dettagli degli `TridentOrchestrator` attributi.

Parametro	Descrizione	Predefinito
<code>namespace</code>	Spazio dei nomi in cui installare Astra Trident	"default"
<code>debug</code>	Attiva il debug per Astra Trident	false
<code>enableForceDetach</code>	<code>ontap-san</code> e <code>ontap-san-economy</code> solo. Funziona con Kubernetes non-Graged Node Shutdown (NGNS) per consentire agli amministratori del cluster di migrare in sicurezza i carichi di lavoro con volumi montati su nuovi nodi in caso di problemi di integrità di un nodo.	false
<code>windows</code>	L'impostazione su <code>true</code> attiva l'installazione sui nodi di lavoro Windows.	false
<code>cloudProvider</code>	Impostato su "Azure" quando si utilizzano identità gestite o un'identità cloud su un cluster AKS. Impostato su "AWS" quando si utilizza un'identità cloud su un cluster EKS.	""
<code>cloudIdentity</code>	Impostare sull'identità del workload ("Azure.workload.Identity/client-id: XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX") quando si utilizza l'identità cloud su un cluster AKS. Impostato sul ruolo AWS IAM ("eks.amazonaws.com/role-arn: arn:AWS:iam::123456:role/astrattadent-role") quando si utilizza l'identità cloud su un cluster EKS.	""
<code>IPv6</code>	Installare Astra Trident su IPv6	false
<code>k8sTimeout</code>	Timeout per le operazioni Kubernetes	30sec
<code>silenceAutosupport</code>	Non inviare pacchetti AutoSupport automaticamente a NetApp	false
<code>autosupportImage</code>	L'immagine del contenitore per la telemetria AutoSupport	"netapp/trident-autosupport:24.06"
<code>autosupportProxy</code>	Indirizzo/porta di un proxy per l'invio di telemetria AutoSupport	"http://proxy.example.com:8888"
<code>uninstall</code>	Flag utilizzato per disinstallare Astra Trident	false
<code>logFormat</code>	Formato di registrazione Astra Trident da utilizzare [text,json]	"text"

Parametro	Descrizione	Predefinito
tridentImage	Immagine Astra Trident da installare	"netapp/trident:24.06"
imageRegistry	Percorso al registro interno, del formato <registry FQDN>[:port][/subpath]	"k8s.gcr.io/sig-storage" (Kubernetes 1,19+) o. "quay.io/k8scsi"
kubeletDir	Percorso della directory del kubelet sull'host	"/var/lib/kubelet"
wipeout	Un elenco di risorse da eliminare per eseguire una rimozione completa di Astra Trident	
imagePullSecrets	Secrets (segreti) per estrarre immagini da un registro interno	
imagePullPolicy	Imposta il criterio di pull dell'immagine per l'operatore Trident. I valori validi sono: Always Per tirare sempre l'immagine. IfNotPresent per estrarre l'immagine solo se non esiste già sul nodo. Never per non tirare mai l'immagine.	IfNotPresent
controllerPluginNodeSelector	Selettori di nodi aggiuntivi per i pod. Segue lo stesso formato di pod.spec.nodeSelector.	Nessuna impostazione predefinita; opzionale
controllerPluginTolerations	Ignora le tolleranze Kubernetes per i pod. Segue lo stesso formato di pod.spec.Tolerations.	Nessuna impostazione predefinita; opzionale
nodePluginNodeSelector	Selettori di nodi aggiuntivi per i pod. Segue lo stesso formato di pod.spec.nodeSelector.	Nessuna impostazione predefinita; opzionale
nodePluginTolerations	Ignora le tolleranze Kubernetes per i pod. Segue lo stesso formato di pod.spec.Tolerations.	Nessuna impostazione predefinita; opzionale



Per ulteriori informazioni sulla formattazione dei parametri del pod, fare riferimento alla sezione ["Assegnazione di pod ai nodi"](#).

Dettagli sulla forza di distacco

Lo stacco forzato è disponibile solo per `ontap-san` e `ontap-san-economy`. Prima di attivare la funzione di forze stach, è necessario attivare la funzione NGNS (non-aggraziate node shutdown) sul cluster Kubernetes. Per ulteriori informazioni, fare riferimento a ["Kubernetes: Shutdown del nodo non aggraziato"](#).



Poiché Astra Trident fa affidamento su Kubernetes NGNS, non rimuovere i `out-of-service` tag da un nodo non integro fino a quando tutti i carichi di lavoro non tollerabili non vengono ripianificati. L'applicazione o la rimozione sconsiderata della contaminazione può compromettere la protezione dei dati back-end.

Quando l'amministratore del cluster Kubernetes ha applicato il `node.kubernetes.io/out-of-service=nodeshutdown:NoExecute` tag taint al nodo ed `enableForceDetach` è impostato su `true`, Astra Trident determina lo stato del nodo e:

1. Interrompere l'accesso i/o back-end per i volumi montati su quel nodo.

2. Contrassegnare l'oggetto nodo Astra Trident come `dirty` (non sicuro per le nuove pubblicazioni).



Il controller Trident rifiuterà le nuove richieste di volume di pubblicazione finché il nodo non viene riqualificato (dopo essere stato contrassegnato come `dirty`) dal pod di nodo Trident. Tutti i carichi di lavoro pianificati con un PVC montato (anche dopo che il nodo del cluster è integro e pronto) non saranno accettati fino a quando Astra Trident non sarà in grado di verificare il nodo `clean` (sicuro per le nuove pubblicazioni).

Quando lo stato del nodo viene ripristinato e la contaminazione viene rimossa, Astra Trident:

1. Identificare e pulire i percorsi pubblicati obsoleti sul nodo.
2. Se il nodo è in uno `cleanable` stato (il tag `out-of-service` è stato rimosso e il nodo è in `Ready` stato) e tutti i percorsi obsoleti e pubblicati sono puliti, Astra Trident riammetterà il nodo come `clean` e consentirà ai nuovi volumi pubblicati di accedere al nodo.

Configurazioni di esempio

È possibile utilizzare gli attributi in [Opzioni di configurazione](#) durante la definizione `TridentOrchestrator` per personalizzare l'installazione.

Configurazione personalizzata di base

Questo è un esempio per un'installazione personalizzata di base.

```
cat deploy/crds/tridentorchestrator_cr_imagepullsecrets.yaml
apiVersion: trident.netapp.io/v1
kind: TridentOrchestrator
metadata:
  name: trident
spec:
  debug: true
  namespace: trident
  imagePullSecrets:
  - thisisasecret
```

Selettori di nodo

Questo esempio installa Astra Trident con selettori di nodo.

```
apiVersion: trident.netapp.io/v1
kind: TridentOrchestrator
metadata:
  name: trident
spec:
  debug: true
  namespace: trident
  controllerPluginNodeSelector:
    nodetype: master
  nodePluginNodeSelector:
    storage: netapp
```

Nodi di lavoro Windows

In questo esempio viene installato Astra Trident su un nodo di lavoro Windows.

```
cat deploy/crds/tridentorchestrator_cr.yaml
apiVersion: trident.netapp.io/v1
kind: TridentOrchestrator
metadata:
  name: trident
spec:
  debug: true
  namespace: trident
  windows: true
```

Identità gestite su un cluster AKS

Questo esempio installa Astra Trident per abilitare le identità gestite su un cluster AKS.

```
apiVersion: trident.netapp.io/v1
kind: TridentOrchestrator
metadata:
  name: trident
spec:
  debug: true
  namespace: trident
  cloudProvider: "Azure"
```

Identità cloud su un cluster AKS

Questo esempio installa Astra Trident per l'utilizzo con un'identità cloud su un cluster AKS.

```
apiVersion: trident.netapp.io/v1
kind: TridentOrchestrator
metadata:
  name: trident
spec:
  debug: true
  namespace: trident
  cloudProvider: "Azure"
  cloudIdentity: 'azure.workload.identity/client-id: xxxxxxxx-xxxx-
xxxx-xxxx-xxxxxxxxxxxxx'
```

Identità cloud su un cluster EKS

Questo esempio installa Astra Trident per l'utilizzo con un'identità cloud su un cluster AKS.

```
apiVersion: trident.netapp.io/v1
kind: TridentOrchestrator
metadata:
  name: trident
spec:
  debug: true
  namespace: trident
  cloudProvider: "AWS"
  cloudIdentity: "'eks.amazonaws.com/role-arn:
arn:aws:iam::123456:role/astratrident-role'"
```

Informazioni sul copyright

Copyright © 2024 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.