Aggiorna Astra Trident

Astra Trident

NetApp
August 18, 2026

This PDF was generated from https://docs.netapp.com/it-it/trident-2304/trident-managing-k8s/upgrade-trident.html on August 18, 2026. Always check docs.netapp.com for the latest.

Sommario

Aggiorna Astra Trident . . . 1
- Aggiorna Astra Trident . . . 1
  - Considerazioni prima dell'aggiornamento . . . 1
  - Fase 1: Selezionare una versione . . . 1
  - Fase 2: Determinare il metodo di installazione originale . . . 2
  - Fase 3: Selezionare un metodo di aggiornamento . . . 2
  - Modifiche all'operatore . . . 3
- Eseguire l'upgrade con l'operatore . . . 3
  - Eseguire l'aggiornamento utilizzando l'operatore Trident . . . 3
  - Aggiornare un'installazione di un operatore Trident con ambito cluster . . . 4
  - Aggiornare un'installazione dell'operatore con ambito namespace . . . 5
  - Aggiornare un'installazione basata su Helm . . . 8
  - Aggiornamento da un'installazione non eseguita dall'operatore . . . 9
- Upgrade con tridentctl . . . 11
  - Aggiorna Astra Trident usando `tridentctl` . . . 11
  - Aggiornare i volumi utilizzando `tridentctl` . . . 11

Aggiorna Astra Trident

Aggiorna Astra Trident

Astra Trident segue una cadenza di rilascio trimestrale, fornendo quattro release principali ogni anno di calendario. Ogni nuova release si basa sulle release precedenti, fornendo nuove funzionalità e miglioramenti delle performance, oltre a correzioni di bug e miglioramenti. Ti consigliamo di effettuare l'upgrade almeno una volta all'anno per sfruttare le nuove funzionalità di Astra Trident.

Considerazioni prima dell'aggiornamento

Quando si esegue l'aggiornamento all'ultima release di Astra Trident, considerare quanto segue:

- Dovrebbe essere installata una sola istanza di Astra Trident in tutti gli spazi dei nomi di un determinato cluster Kubernetes.
- A partire da Trident 20.01, solo la versione beta di "snapshot dei volumi" è supportato. Gli amministratori di Kubernetes devono fare attenzione a eseguire il backup o la conversione degli oggetti snapshot alfa in versione beta in modo sicuro per conservare le snapshot alfa legacy.
  - CSI Volume Snapshots è ora una funzionalità GA, a partire da Kubernetes 1.20. Prima di eseguire l'aggiornamento, è necessario rimuovere i CRD snapshot Alpha utilizzando `tridentctl obliviate alpha-snapshot-crd` Per eliminare i CRD per le specifiche di snapshot alfa.
  - La versione beta delle snapshot dei volumi introduce un set modificato di CRD e un controller di snapshot, entrambi da configurare prima di aggiornare Astra Trident.
  - Per ulteriori informazioni, fare riferimento a. "Cosa occorre sapere prima di aggiornare il cluster Kubernetes".
- Tutti gli aggiornamenti dalle versioni 19.04 e precedenti richiedono la migrazione dei metadati di Astra Trident `etcd` Agli oggetti CRD. Controllare "Documentazione specifica per la release di Astra Trident" per comprendere il funzionamento dell'aggiornamento.
- Durante l'aggiornamento, è importante fornire `parameter.fsType` poll `StorageClasses` Utilizzato da Astra Trident. È possibile eliminare e ricreare `StorageClasses` senza interrompere i volumi preesistenti.
  - Si tratta di un requisito ** per l'applicazione "contesti di sicurezza" Per volumi SAN.
  - La directory sample input contiene esempi, come `storage-class-basic.yaml.templ` e `storage-class-bronze-default.yaml`. Per ulteriori informazioni, vedere "Problemi noti".

Fase 1: Selezionare una versione

Le versioni di Astra Trident seguono una data-based `YY.MM` Convenzione di naming, dove "YY" è l'ultima cifra dell'anno e "MM" è il mese. I rilasci di punti seguono un `YY.MM.X` convention, dove "X" è il livello di patch. Selezionare la versione a cui eseguire l'aggiornamento in base alla versione da cui si sta eseguendo l'aggiornamento.

- È possibile eseguire un aggiornamento diretto a qualsiasi release di destinazione che si trova all'interno di una finestra di quattro release della versione installata. Ad esempio, è possibile eseguire direttamente l'aggiornamento alla versione 23.04 da 22.04 (incluse le release DOT, ad esempio 22.04.1).
- Se si dispone di una release precedente, è necessario eseguire un aggiornamento in più fasi utilizzando la documentazione della release corrispondente per istruzioni specifiche. Ciò richiede l'aggiornamento alla

versione più recente che si adatta alla finestra di quattro release. Ad esempio, se si utilizza 18.07 e si desidera eseguire l'aggiornamento alla versione 20.07, seguire il processo di aggiornamento in più fasi come indicato di seguito:

a. Primo aggiornamento da 18.07 a 19.07.
b. Eseguire l'aggiornamento da 19.07 a 20.07.

> Quando si esegue l'aggiornamento utilizzando l'operatore Trident su OpenShift Container Platform, è necessario eseguire l'aggiornamento a Trident 21.01.1 o versione successiva. L'operatore Trident rilasciato con 21.01.0 contiene un problema noto che è stato risolto nel 21.01.1. Per ulteriori informazioni, vedere "Dettagli del problema su GitHub".

Fase 2: Determinare il metodo di installazione originale

In genere, è necessario eseguire l'aggiornamento utilizzando lo stesso metodo utilizzato per l'installazione iniziale, tuttavia è possibile "passare da un metodo di installazione all'altro".

Per determinare la versione utilizzata per installare Astra Trident:

1. Utilizzare `kubectl get pods - trident` esaminare i pod.
   - Se non è disponibile un pod per l'operatore, Astra Trident è stato installato utilizzando `tridentctl`.
   - Se è presente un pod operatore, Astra Trident è stato installato utilizzando l'operatore Trident manualmente o utilizzando Helm.
2. Se è presente un pod per l'operatore, utilizzare `kubectl describe tproc trident` Per determinare se Astra Trident è stato installato utilizzando Helm.
   - Se è presente un'etichetta Helm, Astra Trident è stato installato utilizzando Helm.
   - Se non è presente alcuna etichetta Helm, Astra Trident è stato installato manualmente utilizzando l'operatore Trident.

Fase 3: Selezionare un metodo di aggiornamento

Esistono due metodi per aggiornare Astra Trident.

Quando eseguire l'aggiornamento utilizzando l'operatore

È possibile "Eseguire l'aggiornamento utilizzando l'operatore Trident" se:

- In origine Astra Trident è stato installato utilizzando l'operatore o `tridentctl`.
- CSI Trident è stato disinstallato e i metadati dell'installazione persistono.
- Si dispone di un'installazione Astra Trident basata su CSI. Tutte le release a partire dalla versione 19.07 sono basate su CSI. È possibile esaminare i pod nello spazio dei nomi Trident per verificare la versione.
  - La denominazione dei pod nelle versioni precedenti alla 23.01 utilizza: `trident-csi-*`
  - La naming dei pod nella versione 23.01 e successive utilizza:
    - `trident-controller-<generated id>` per il controller pod
    - `trident-node-<operating system>-<generated id>` per i pod di nodi
    - `trident-operator-<generated id>` per il pod operatore

Non utilizzare l'operatore per aggiornare Trident se si utilizza un `etcd`Versione Trident basata su (19.04 o precedente).

Quando eseguire l'aggiornamento utilizzando `tridentctl`

È possibile Se in origine è stato installato Astra Trident usando `tridenctll'.

`tridentctl` è il metodo convenzionale di installazione Astra Trident e offre il maggior numero di opzioni per chi necessita di personalizzazioni complesse. Per maggiori dettagli, consulta "Scegliere il metodo di installazione desiderato".

Modifiche all'operatore

La release 21.01 di Astra Trident ha introdotto modifiche architetturali per l'operatore:

- L'operatore è ora **con ambito cluster**. Le istanze precedenti dell'operatore Trident (versioni da 20.04 a 20.10) erano **namespace-scope**. Un operatore con ambito cluster è vantaggioso per i seguenti motivi:
  - Responsabilità delle risorse: L'operatore gestisce ora le risorse associate a un'installazione di Astra Trident a livello di cluster. Nell'ambito dell'installazione di Astra Trident, l'operatore crea e gestisce diverse risorse utilizzando `ownerReferences`. Manutenzione `ownerReferences` Su risorse con ambito cluster possono generare errori su alcuni distributori Kubernetes come OpenShift. Questo è mitigato da un operatore con ambito cluster. Per la riparazione automatica e l'applicazione di patch alle risorse Trident, questo è un requisito essenziale.
  - Pulizia durante la disinstallazione: Una rimozione completa di Astra Trident richiederebbe l'eliminazione di tutte le risorse associate. Un operatore con ambito spazio dei nomi potrebbe riscontrare problemi con la rimozione delle risorse con ambito del cluster (come ClusterRole, ClusterRoleBinding e PodSecurityPolicy) e portare a una pulizia incompleta. Un operatore con ambito cluster elimina questo problema. Gli utenti possono disinstallare completamente Astra Trident e installare di nuovo, se necessario.
- `TridentProvisioner` viene ora sostituito con `TridentOrchestrator` Come risorsa personalizzata utilizzata per installare e gestire Astra Trident. Inoltre, viene introdotto un nuovo campo in `TridentOrchestrator` spec. Gli utenti possono specificare che lo spazio dei nomi Trident deve essere installato/aggiornato utilizzando `spec.namespace` campo. Puoi dare un'occhiata a un esempio "qui".

Eseguire l'upgrade con l'operatore

È possibile aggiornare facilmente un'installazione Astra Trident utilizzando l'operatore manualmente o Helm.

Eseguire l'aggiornamento utilizzando l'operatore Trident

In genere, dovresti aggiornare Astra Trident usando lo stesso metodo che hai usato per installarlo la prima volta. Consulta "Selezionare un metodo di aggiornamento" prima di provare ad aggiornare con l'operatore Trident.

Quando si esegue l'aggiornamento da un'istanza di Astra Trident installata utilizzando l'operatore namespace-scoped (versioni da 20.07 a 20.10), l'operatore Trident:

- Migra `tridentProvisioner` a un `tridentOrchestrator` oggetto con lo stesso nome,
- Elimina `TridentProvisioner` oggetti e il `tridentprovisioner` CRD
- Aggiorna Astra Trident alla versione dell'operatore cluster-scoped in uso
- Installare Astra Trident nello stesso namespace in cui era stato originariamente installato

Aggiornare un'installazione di un operatore Trident con ambito cluster

È possibile aggiornare un'installazione di un operatore Trident con ambito cluster. Tutte le versioni di Astra Trident 21.01 e successive utilizzano un operatore con ambito cluster.

Prima di iniziare

Assicurarsi di utilizzare un cluster Kubernetes in esecuzione "Una versione di Kubernetes supportata".

Fasi

1. Verificare la versione di Astra Trident:

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

2. Eliminare l'operatore Trident utilizzato per installare l'istanza corrente di Astra Trident. Ad esempio, se si esegue l'aggiornamento da 22.01, eseguire il seguente comando:

```
kubectl delete -f 22.01/trident-installer/deploy/bundle.yaml -n trident
```

3. Se l'installazione iniziale è stata personalizzata utilizzando `TridentOrchestrator` è possibile modificare `TridentOrchestrator` oggetto per modificare i parametri di installazione. Ciò potrebbe includere le modifiche apportate per specificare i registri di immagini Trident e CSI mirrorati per la modalità offline, abilitare i registri di debug o specificare i segreti di pull delle immagini.
4. Installare Astra Trident utilizzando il file YAML del bundle corretto per il proprio ambiente e la versione di Astra Trident. Ad esempio, se si sta installando Astra Trident 23.04 per Kubernetes 1.27, eseguire il seguente comando:

```
kubectl create -f 23.04.0/trident-installer/deploy/bundle_post_1_25.yaml
-n trident
```

Trident fornisce un file bundle che può essere utilizzato per installare l'operatore e creare oggetti associati per la versione di Kubernetes.

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

Risultati

L'operatore Trident identificherà un'installazione Astra Trident esistente e la aggiornerà alla stessa versione dell'operatore.

Aggiornare un'installazione dell'operatore con ambito namespace

È possibile eseguire l'aggiornamento da un'istanza di Astra Trident installata utilizzando l'operatore namespace-scoped (versioni da 20.07 a 20.10) a un'installazione dell'operatore con ambito cluster.

Prima di iniziare

È necessario il file YAML bundle utilizzato per implementare l'operatore con ambito dello spazio dei nomi da `https://github.com/NetApp/trident/tree/stable/vXX.XX/deploy/BUNDLE.YAML` dove *`vXX.XX`* è il numero di versione e. *`BUNDLE.YAML`* È il nome del file YAML del bundle.

Fasi

1. Verificare `TridentProvisioner` Lo stato dell'installazione Trident esistente è `Installed`.

```
kubectl describe tprov trident -n trident | grep Message: -A 3

Message:  Trident installed
Status:   Installed
Version:  v20.10.1
```

> Se lo stato mostra `Updating`, assicurati di risolverlo prima di procedere. Per un elenco dei possibili valori di stato, consulta "qui".

2. Creare il `TridentOrchestrator` CRD utilizzando il manifesto fornito con il programma di installazione di Trident.

```
# Download the release required [23.04.0]
mkdir 23.04.0
cd 23.04.0
wget
https://github.com/NetApp/trident/releases/download/v23.04.0/trident-
installer-23.04.0.tar.gz
tar -xf trident-installer-23.04.0.tar.gz
cd trident-installer
kubectl create -f
deploy/crds/trident.netapp.io_tridentorchestrators_crd_post1.16.yaml
```

3. Eliminare l'operatore con ambito dello spazio dei nomi utilizzando il relativo manifesto.
   a. Assicurarsi di essere nella directory corretta.

```
pwd
/root/20.10.1/trident-installer
```

b. Eliminare l'operatore con ambito dello spazio dei nomi.

```
kubectl delete -f deploy/<BUNDLE.YAML> -n trident

serviceaccount "trident-operator" deleted
clusterrole.rbac.authorization.k8s.io "trident-operator" deleted
clusterrolebinding.rbac.authorization.k8s.io "trident-operator"
deleted
deployment.apps "trident-operator" deleted
podsecuritypolicy.policy "tridentoperatorpods" deleted
```

c. Verificare che l'operatore Trident sia stato rimosso.

```
kubectl get all -n trident

NAME                                READY   STATUS    RESTARTS   AGE
pod/trident-csi-68d979fb85-dsrmn    6/6     Running   12         99d
pod/trident-csi-8jfhf               2/2     Running   6          105d
pod/trident-csi-jtnjz               2/2     Running   6          105d
pod/trident-csi-lcxvh               2/2     Running   8          105d

NAME                  TYPE        CLUSTER-IP       EXTERNAL-IP
PORT(S)              AGE
service/trident-csi   ClusterIP   10.108.174.125   <none>
34571/TCP,9220/TCP   105d

NAME                           DESIRED   CURRENT   READY   UP-TO-DATE
AVAILABLE   NODE SELECTOR                                        AGE
daemonset.apps/trident-csi   3         3         3       3
3           kubernetes.io/arch=amd64,kubernetes.io/os=linux   105d

NAME                           READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/trident-csi    1/1     1            1           105d

NAME                                      DESIRED   CURRENT   READY
AGE
replicaset.apps/trident-csi-68d979fb85    1         1         1
105d
```

4. (Opzionale) Se è necessario modificare i parametri di installazione, aggiorna la specifica

`TridentProvisioner`. Ciò può includere modifiche come la variazione di: i valori per `tridentImage`, `autosupportImage`, repository di immagini privato e la fornitura di `imagePullSecrets`) dopo aver eliminato l'operatore con ambito namespace e prima di installare l'operatore con ambito cluster. Per un elenco completo dei parametri che possono essere aggiornati, consulta "opzioni di configurazione".

```
kubectl patch tprov <trident-provisioner-name> -n <trident-namespace>
--type=merge -p '{"spec":{"debug":true}}'
```

5. Installare l'operatore con ambito cluster Trident.
   a. Assicurarsi di essere nella directory corretta.

```
pwd
/root/23.04.0/trident-installer
```

   b. Installare l'operatore con ambito cluster nello stesso namespace.

> Trident fornisce un file bundle che può essere utilizzato per installare l'operatore e creare oggetti associati per la versione di Kubernetes.
>
> - Per i cluster che eseguono Kubernetes 1.24 o versioni precedenti, utilizzare "bundle_pre_1_25.yaml".
> - Per i cluster che eseguono Kubernetes 1.25 o versioni successive, utilizzare "bundle_post_1_25.yaml".

```
kubectl create -f deploy/<BUNDLE.YAML>

serviceaccount/trident-operator created
clusterrole.rbac.authorization.k8s.io/trident-operator created
clusterrolebinding.rbac.authorization.k8s.io/trident-operator created
deployment.apps/trident-operator created
podsecuritypolicy.policy/tridentoperatorpods created

#All tridentProvisioners will be removed, including the CRD itself
kubectl get tprov -n trident
Error from server (NotFound): Unable to list "trident.netapp.io/v1,
Resource=tridentprovisioners": the server could not find the
requested resource (get tridentprovisioners.trident.netapp.io)

#tridentProvisioners are replaced by tridentOrchestrator
kubectl get torc
NAME      AGE
trident   13s
```

   c. Esaminare i pod Trident nello spazio dei nomi. Il `trident-controller` e i nomi dei pod riflettono la

convenzione di naming introdotta nel 23.01.

```
kubectl get pods -n trident

NAME                                   READY   STATUS    RESTARTS
AGE
trident-controller-79df798bdc-m79dc    6/6     Running   0
1m41s
trident-node-linux-xrst8               2/2     Running   0
1m41s
trident-operator-5574dbbc68-nthjv      1/1     Running   0
1m52s
```

d. Verificare che Trident sia stato aggiornato alla versione prevista.

```
kubectl describe torc trident | grep Message -A 3
Message:                Trident installed
Namespace:              trident
Status:                 Installed
Version:                v23.04.0
```

Aggiornare un'installazione basata su Helm

Per aggiornare un'installazione basata su Helm, procedere come segue.

> Quando si aggiorna un cluster Kubernetes dalla versione 1.24 alla 1.25 o successiva su cui è installato Astra Trident, è necessario aggiornare values.yaml per impostarlo `excludePodSecurityPolicy` a. `true` oppure aggiungi `--set excludePodSecurityPolicy=true` al `helm upgrade` prima di aggiornare il cluster.

Fasi

1. Scarica l'ultima release di Astra Trident.
2. Utilizzare `helm upgrade` comando dove `trident-operator-23.04.0.tgz` indica la versione alla quale si desidera eseguire l'aggiornamento.

```
helm upgrade <name> trident-operator-23.04.0.tgz
```

Se si impostano opzioni non predefinite durante l'installazione iniziale (ad esempio, se si specificano registri privati mirrorati per le immagini Trident e CSI), utilizzare `--set` per assicurarsi che tali opzioni siano incluse nel comando upgrade, altrimenti i valori torneranno ai valori predefiniti.

Ad esempio, per modificare il valore predefinito di `tridentDebug`, eseguire il seguente comando:

```
helm upgrade <name> trident-operator-23.04.0-custom.tgz --set
tridentDebug=true
```

3. Eseguire `helm list` per verificare che la versione del grafico e dell'applicazione sia stata aggiornata. Eseguire `tridentctl logs` per esaminare eventuali messaggi di debug.

Risultati

L'operatore Trident identificherà un'installazione Astra Trident esistente e la aggiornerà alla stessa versione dell'operatore.

Aggiornamento da un'installazione non eseguita dall'operatore

È possibile eseguire l'aggiornamento all'ultima versione dell'operatore Trident da un `tridentctl` installazione.

Fasi

1. Scarica l'ultima release di Astra Trident.

```
# Download the release required [23.04.0]
mkdir 23.04.0
cd 23.04.0
wget
https://github.com/NetApp/trident/releases/download/v22.01.1/trident-
installer-23.04.0.tar.gz
tar -xf trident-installer-23.04.0.tar.gz
cd trident-installer
```

2. Creare il `tridentorchestrator` CRD dal manifesto.

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

3. Implementare l'operatore con ambito cluster nello stesso namespace.

```
kubectl create -f deploy/<BUNDLE.YAML>

serviceaccount/trident-operator created
clusterrole.rbac.authorization.k8s.io/trident-operator created
clusterrolebinding.rbac.authorization.k8s.io/trident-operator created
deployment.apps/trident-operator created
podsecuritypolicy.policy/tridentoperatorpods created

#Examine the pods in the Trident namespace
NAME                                  READY   STATUS    RESTARTS   AGE
trident-controller-79df798bdc-m79dc   6/6     Running   0          150d
trident-node-linux-xrst8              2/2     Running   0          150d
trident-operator-5574dbbc68-nthjv     1/1     Running   0          1m30s
```

4. Creare un `TridentOrchestrator` CR per l'installazione di Astra Trident.

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

kubectl create -f deploy/crds/tridentorchestrator_cr.yaml

#Examine the pods in the Trident namespace
NAME                                READY   STATUS    RESTARTS   AGE
trident-csi-79df798bdc-m79dc        6/6     Running   0          1m
trident-csi-xrst8                   2/2     Running   0          1m
trident-operator-5574dbbc68-nthjv   1/1     Running   0          5m41s
```

5. Confermare che Trident è stato aggiornato alla versione prevista.

```
kubectl describe torc trident | grep Message -A 3

Message:                Trident installed
Namespace:              trident
Status:                 Installed
Version:                v23.04.0
```

Risultati

I backend e i PVC esistenti sono automaticamente disponibili.

Upgrade con tridentctl

È possibile aggiornare facilmente un'installazione Astra Trident utilizzando `tridentctl`.

Aggiorna Astra Trident usando `tridentctl`

La disinstallazione e la reinstallazione di Astra Trident funge da aggiornamento. Quando si disinstalla Trident, i PVC (Persistent Volume Claim) e PV (Persistent Volume) utilizzati dall'implementazione di Astra Trident non vengono cancellati. I PVS già forniti resteranno disponibili mentre Astra Trident è offline e Astra Trident effettuerà il provisioning dei volumi per i PVC creati nel frattempo una volta tornati online.

Prima di iniziare

Rivedi "Selezionare un metodo di aggiornamento" prima di aggiornare usando `tridentctl`.

Fasi

1. Eseguire il comando di disinstallazione in `tridentctl` Rimuovere tutte le risorse associate ad Astra Trident, ad eccezione dei CRD e degli oggetti correlati.

```
./tridentctl uninstall -n <namespace>
```

2. Reinstalla Astra Trident. Consulta "Installare Astra Trident usando tridentctl".

> Non interrompere il processo di aggiornamento. Assicurarsi che il programma di installazione venga completato.

Aggiornare i volumi utilizzando `tridentctl`

Dopo l'aggiornamento, è possibile utilizzare il ricco set di funzionalità disponibili nelle versioni più recenti di Trident (ad esempio, on-Demand Volume Snapshots) e aggiornare i volumi utilizzando `tridentctl upgrade` comando.

Se sono presenti volumi legacy, è necessario aggiornarli da un tipo NFS o iSCSI al tipo CSI per utilizzare il set completo di nuove funzionalità di Astra Trident. Un PV legacy che è stato fornito da Trident supporta il set tradizionale di funzionalità.

Prima di iniziare

Prima di decidere di aggiornare i volumi al tipo CSI, considerare quanto segue:

- Potrebbe non essere necessario aggiornare tutti i volumi. I volumi creati in precedenza continueranno ad essere accessibili e funzioneranno normalmente.
- Un PV può essere montato come parte di un'implementazione/StatefulSet durante l'aggiornamento. Non è necessario mettere fuori servizio il deployment/StatefulSet.
- Non è possibile collegare un PV a un pod standalone durante l'aggiornamento. Chiudere il pod prima di aggiornare il volume.
- È possibile aggiornare solo un volume associato a un PVC. I volumi che non sono associati a PVC devono essere rimossi e importati prima dell'aggiornamento.

Fasi

1. Eseguire `kubectl get pv` Per elencare il PVS.

```
kubectl get pv
NAME                        CAPACITY     ACCESS MODES   RECLAIM POLICY
STATUS   CLAIM                   STORAGECLASS    REASON   AGE
default-pvc-1-a8475         1073741824   RWO            Delete
Bound    default/pvc-1          standard                19h
default-pvc-2-a8486         1073741824   RWO            Delete
Bound    default/pvc-2          standard                19h
default-pvc-3-a849e         1073741824   RWO            Delete
Bound    default/pvc-3          standard                19h
default-pvc-4-a84de         1073741824   RWO            Delete
Bound    default/pvc-4          standard                19h
trident                     2Gi          RWO            Retain
Bound    trident/trident                                19h
```

Attualmente sono disponibili quattro PVS creati da Trident 20.07, utilizzando `netapp.io/trident` provisioner.

2. Eseguire `kubectl describe pv` Per ottenere i dettagli del PV.

```
kubectl describe pv default-pvc-2-a8486

Name:            default-pvc-2-a8486
Labels:          <none>
Annotations:     pv.kubernetes.io/provisioned-by: netapp.io/trident
                 volume.beta.kubernetes.io/storage-class: standard
Finalizers:      [kubernetes.io/pv-protection]
StorageClass:    standard
Status:          Bound
Claim:           default/pvc-2
Reclaim Policy:  Delete
Access Modes:    RWO
VolumeMode:      Filesystem
Capacity:        1073741824
Node Affinity:   <none>
Message:
Source:
    Type:      NFS (an NFS mount that lasts the lifetime of a pod)
    Server:    10.xx.xx.xx
    Path:      /trid_1907_alpha_default_pvc_2_a8486
    ReadOnly:  false
```

Il PV è stato creato utilizzando `netapp.io/trident` provisioner ed è del tipo NFS. Per supportare tutte le nuove funzioni fornite da Astra Trident, questo PV deve essere aggiornato al tipo CSI.

3. Eseguire `tridentctl upgrade volume <name-of-trident-volume>` Comando per aggiornare un volume Astra Trident legacy alla specifica CSI.

```
./tridentctl get volumes -n trident
+---------------------+---------+---------------+----------
+--------------------------------------+--------+---------+
|            NAME     |  SIZE   | STORAGE CLASS | PROTOCOL |
BACKEND UUID             | STATE  | MANAGED |
+---------------------+---------+---------------+----------
+--------------------------------------+--------+---------+
| default-pvc-2-a8486 | 1.0 GiB | standard      | file     | c5a6f6a4-
b052-423b-80d4-8fb491a14a22 | online | true    |
| default-pvc-3-a849e | 1.0 GiB | standard      | file     | c5a6f6a4-
b052-423b-80d4-8fb491a14a22 | online | true    |
| default-pvc-1-a8475 | 1.0 GiB | standard      | file     | c5a6f6a4-
b052-423b-80d4-8fb491a14a22 | online | true    |
| default-pvc-4-a84de | 1.0 GiB | standard      | file     | c5a6f6a4-
b052-423b-80d4-8fb491a14a22 | online | true    |
+---------------------+---------+---------------+----------
+--------------------------------------+--------+---------+

./tridentctl upgrade volume default-pvc-2-a8486 -n trident
+---------------------+---------+---------------+----------
+--------------------------------------+--------+---------+
|            NAME     |  SIZE   | STORAGE CLASS | PROTOCOL |
BACKEND UUID             | STATE  | MANAGED |
+---------------------+---------+---------------+----------
+--------------------------------------+--------+---------+
| default-pvc-2-a8486 | 1.0 GiB | standard      | file     | c5a6f6a4-
b052-423b-80d4-8fb491a14a22 | online | true    |
+---------------------+---------+---------------+----------
+--------------------------------------+--------+---------+
```

4. Eseguire un `kubectl describe pv` Per verificare che il volume sia un volume CSI.

```
kubectl describe pv default-pvc-2-a8486
Name:            default-pvc-2-a8486
Labels:          <none>
Annotations:     pv.kubernetes.io/provisioned-by: csi.trident.netapp.io
                 volume.beta.kubernetes.io/storage-class: standard
Finalizers:      [kubernetes.io/pv-protection]
StorageClass:    standard
Status:          Bound
Claim:           default/pvc-2
Reclaim Policy:  Delete
Access Modes:    RWO
VolumeMode:      Filesystem
Capacity:        1073741824
Node Affinity:   <none>
Message:
Source:
    Type:              CSI (a Container Storage Interface (CSI) volume
source)
    Driver:            csi.trident.netapp.io
    VolumeHandle:      default-pvc-2-a8486
    ReadOnly:          false
    VolumeAttributes:      backendUUID=c5a6f6a4-b052-423b-80d4-
8fb491a14a22

internalName=trid_1907_alpha_default_pvc_2_a8486
                           name=default-pvc-2-a8486
                           protocol=file
Events:                <none>
```

Informazioni sul copyright

Copyright © 2026 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 COMMERCIABILITÀ 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 CONSEQUENZIALE (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.