



Riferimento

Trident

NetApp
January 15, 2026

This PDF was generated from <https://docs.netapp.com/it-it/trident-2506/trident-reference/ports.html> on January 15, 2026. Always check docs.netapp.com for the latest.

Sommario

Riferimento	1
Porte Trident	1
Porte Trident	1
API REST Trident	1
Quando utilizzare l'API REST	1
Utilizzo dell'API REST	1
Opzioni della riga di comando	2
Registrazione	2
Kubernetes	2
Docker	3
RIPOSO	3
Oggetti Kubernetes e Trident	3
Come interagiscono gli oggetti tra loro?	3
Kubernetes PersistentVolumeClaim oggetti	4
Kubernetes PersistentVolume oggetti	5
Kubernetes StorageClass oggetti	6
Kubernetes VolumeSnapshotClass oggetti	9
Kubernetes VolumeSnapshot oggetti	10
Kubernetes VolumeSnapshotContent oggetti	10
Kubernetes VolumeGroupSnapshotClass oggetti	11
Kubernetes VolumeGroupSnapshot oggetti	11
Kubernetes VolumeGroupSnapshotContent oggetti	11
Kubernetes CustomResourceDefinition oggetti	12
Trident StorageClass oggetti	12
Oggetti backend Trident	13
Trident StoragePool oggetti	13
Trident Volume oggetti	13
Trident Snapshot oggetti	14
Trident ResourceQuota oggetto	15
Standard di sicurezza dei pod (PSS) e vincoli di contesto di sicurezza (SCC)	16
Contesto di sicurezza Kubernetes obbligatorio e campi correlati	17
Standard di sicurezza del pod (PSS)	17
Criteri di sicurezza dei pod (PSP)	18
Vincoli di contesto di sicurezza (SCC)	19

Riferimento

Porte Trident

Scopri di più sulle porte utilizzate Trident per le comunicazioni.

Porte Trident

Trident utilizza le seguenti porte per la comunicazione all'interno di Kubernetes:

Porta	Scopo
8443	Backchannel HTTPS
8001	Endpoint delle metriche di Prometheus
8000	Server REST Trident
17546	Porta di sonda di attività/prontezza utilizzata dai pod del daemonset Trident



La porta della sonda di attività/prontezza può essere modificata durante l'installazione utilizzando `--probe-port` bandiera. È importante assicurarsi che questa porta non sia utilizzata da un altro processo sui nodi worker.

API REST Trident

Mentre "[comandi e opzioni tridentctl](#)" rappresentano il modo più semplice per interagire con l'API REST Trident ; se preferisci, puoi utilizzare direttamente l'endpoint REST.

Quando utilizzare l'API REST

L'API REST è utile per installazioni avanzate che utilizzano Trident come binario autonomo in distribuzioni non Kubernetes.

Per una maggiore sicurezza, il Trident REST API è limitato per impostazione predefinita a localhost quando viene eseguito all'interno di un pod. Per modificare questo comportamento, è necessario impostare Trident `-address` argomento nella sua configurazione pod.

Utilizzo dell'API REST

Per esempi di come vengono chiamate queste API, passare il debug(`-d`) bandiera. Per maggiori informazioni, fare riferimento a "[Gestisci Trident usando tridentctl](#)".

L'API funziona come segue:

OTTENERE

`GET <trident-address>/trident/v1/<object-type>`

Elenca tutti gli oggetti di quel tipo.

```
GET <trident-address>/trident/v1/<object-type>/<object-name>
```

Ottiene i dettagli dell'oggetto denominato.

INVIARE

```
POST <trident-address>/trident/v1/<object-type>
```

Crea un oggetto del tipo specificato.

- Richiede una configurazione JSON per l'oggetto da creare. Per la specifica di ciascun tipo di oggetto, fare riferimento a "[Gestisci Trident usando tridentctl](#)".
- Se l'oggetto esiste già, il comportamento varia: i backend aggiornano l'oggetto esistente, mentre tutti gli altri tipi di oggetti non riusciranno nell'operazione.

ELIMINARE

```
DELETE <trident-address>/trident/v1/<object-type>/<object-name>
```

Elimina la risorsa indicata.



I volumi associati ai backend o alle classi di archiviazione continueranno a esistere; dovranno essere eliminati separatamente. Per maggiori informazioni, fare riferimento a "[Gestisci Trident usando tridentctl](#)".

Opzioni della riga di comando

Trident espone diverse opzioni della riga di comando per l'orchestratore Trident . Puoi utilizzare queste opzioni per modificare la tua distribuzione.

Registrazione

-debug

Abilita l'output di debug.

-loglevel <level>

Imposta il livello di registrazione (debug, info, warn, error, fatal). Per impostazione predefinita è info.

Kubernetes

-k8s_pod

Utilizzare questa opzione o **-k8s_api_server** per abilitare il supporto Kubernetes. Impostando questa opzione, Trident utilizzerà le credenziali dell'account di servizio Kubernetes del pod contenente per contattare il server API. Funziona solo quando Trident viene eseguito come pod in un cluster Kubernetes con account di servizio abilitati.

-k8s_api_server <insecure-address:insecure-port>

Utilizzare questa opzione o **-k8s_pod** per abilitare il supporto Kubernetes. Se specificato, Trident si connette al server API Kubernetes utilizzando l'indirizzo e la porta non sicuri forniti. Ciò consente di distribuire Trident al di fuori di un pod; tuttavia, supporta solo connessioni non sicure al server API. Per connettersi in modo sicuro, distribuire Trident in un pod con **-k8s_pod** opzione.

Docker

-volume_driver <name>

Nome del driver utilizzato durante la registrazione del plugin Docker. Predefinito su `netapp` .

-driver_port <port-number>

Ascolta su questa porta anziché su un socket di dominio UNIX.

-config <file>

Obbligatorio; è necessario specificare questo percorso per un file di configurazione backend.

RIPOSO

-address <ip-or-host>

Specifica l'indirizzo su cui il server REST di Trident deve essere in ascolto. Il valore predefinito è `localhost`. Quando si ascolta su `localhost` ed è in esecuzione all'interno di un pod Kubernetes, l'interfaccia REST non è direttamente accessibile dall'esterno del pod. Utilizzo `-address " "` per rendere l'interfaccia REST accessibile dall'indirizzo IP del pod.



L'interfaccia REST Trident può essere configurata per ascoltare e servire solo su `127.0.0.1` (per IPv4) o `[:1]` (per IPv6).

-port <port-number>

Specifica la porta su cui il server REST di Trident deve essere in ascolto. Il valore predefinito è `8000`.

-rest

Abilita l'interfaccia REST. Il valore predefinito è `true`.

Oggetti Kubernetes e Trident

È possibile interagire con Kubernetes e Trident utilizzando le API REST leggendo e scrivendo oggetti risorsa. Esistono diversi oggetti risorsa che determinano la relazione tra Kubernetes e Trident, Trident e storage e Kubernetes e storage. Alcuni di questi oggetti sono gestiti tramite Kubernetes, mentre altri sono gestiti tramite Trident.

Come interagiscono gli oggetti tra loro?

Forse il modo più semplice per comprendere gli oggetti, a cosa servono e come interagiscono è seguire una singola richiesta di archiviazione da parte di un utente Kubernetes:

1. Un utente crea un `PersistentVolumeClaim` richiedendo un nuovo `PersistentVolume` di una dimensione particolare da un Kubernetes `StorageClass` che era stato precedentemente configurato dall'amministratore.
2. Il Kubernetes `StorageClass` identifica Trident come suo fornitore e include parametri che indicano a Trident come fornire un volume per la classe richiesta.
3. Trident guarda se stesso `StorageClass` con lo stesso nome che identifica la corrispondenza `Backends` E `StoragePools` che può utilizzare per fornire volumi per la classe.
4. Trident fornisce spazio di archiviazione su un backend corrispondente e crea due oggetti: un

`PersistentVolume` in Kubernetes che indica a Kubernetes come trovare, montare e trattare il volume, e un volume in Trident che mantiene la relazione tra `PersistentVolume` e l'archiviazione vera e propria.

5. Kubernetes lega il `PersistentVolumeClaim` al nuovo `PersistentVolume`. Baccelli che includono il `PersistentVolumeClaim` montare quel `PersistentVolume` su qualsiasi host su cui viene eseguito.
6. Un utente crea un `VolumeSnapshot` di un PVC esistente, utilizzando un `VolumeSnapshotClass` che punta a Trident.
7. Trident identifica il volume associato al PVC e crea uno snapshot del volume sul suo backend. Crea anche un `VolumeSnapshotContent` che indica a Kubernetes come identificare lo snapshot.
8. Un utente può creare un `PersistentVolumeClaim` usando `VolumeSnapshot` come fonte.
9. Trident identifica lo snapshot richiesto ed esegue la stessa serie di passaggi coinvolti nella creazione di un `PersistentVolume` e un `Volume`.



Per ulteriori approfondimenti sugli oggetti Kubernetes, ti consigliamo vivamente di leggere ["Volumi persistenti"](#) sezione della documentazione di Kubernetes.

Kubernetes `PersistentVolumeClaim` oggetti

Un Kubernetes `PersistentVolumeClaim` L'oggetto è una richiesta di archiviazione effettuata da un utente del cluster Kubernetes.

Oltre alle specifiche standard, Trident consente agli utenti di specificare le seguenti annotazioni specifiche del volume se desiderano sovrascrivere le impostazioni predefinite impostate nella configurazione del backend:

Annotazione	Opzione volume	Driver supportati
<code>trident.netapp.io/fileSystem</code>	file System	ontap-san, solidfire-san, ontap-san-economy
<code>trident.netapp.io/cloneFromPVC</code>	<code>cloneSourceVolume</code>	ontap-nas, ontap-san, solidfire-san, azure-netapp-files, gcp-cvs, ontap-san-economy
<code>trident.netapp.io/splitOnClone</code>	<code>splitOnClone</code>	ontap-nas, ontap-san
<code>trident.netapp.io/protocollo</code>	<code>protocollo</code>	Qualunque
<code>trident.netapp.io/exportPolicy</code>	Politica di esportazione	ontap-nas, ontap-nas-economy, ontap-nas-flexgroup
<code>trident.netapp.io/snapshotPolicy</code>	<code>snapshotPolicy</code>	ontap-nas, ontap-nas-economy, ontap-nas-flexgroup, ontap-san
<code>trident.netapp.io/snapshotReserve</code>	<code>snapshotReserve</code>	ontap-nas, ontap-nas-flexgroup, ontap-san, gcp-cvs
<code>trident.netapp.io/snapshotDirectory</code>	<code>directorystantanea</code>	ontap-nas, ontap-nas-economy, ontap-nas-flexgroup
<code>trident.netapp.io/unixPermissions</code>	Permessi unix	ontap-nas, ontap-nas-economy, ontap-nas-flexgroup
<code>trident.netapp.io/blockSize</code>	dimensione del blocco	solidfire-san

Se il PV creato ha il `Delete` politica di recupero, Trident elimina sia il PV che il volume di supporto quando il

PV viene rilasciato (ovvero quando l'utente elimina il PVC). Se l'operazione di eliminazione fallisce, Trident contrassegna il PV come tale e riprova periodicamente finché non riesce o finché il PV non viene eliminato manualmente. Se il PV utilizza il Retain policy, Trident lo ignora e presume che l'amministratore lo ripulirà da Kubernetes e dal backend, consentendo il backup o l'ispezione del volume prima della sua rimozione. Si noti che l'eliminazione del PV non comporta l'eliminazione del volume di supporto Trident. Dovresti rimuoverlo usando l'API REST(`tridentctl`).

Trident supporta la creazione di snapshot del volume utilizzando la specifica CSI: è possibile creare uno snapshot del volume e utilizzarlo come origine dati per clonare PVC esistenti. In questo modo, le copie point-in-time dei PV possono essere esposte a Kubernetes sotto forma di snapshot. Gli snapshot possono quindi essere utilizzati per creare nuovi PV. Dai un'occhiata a [On-Demand Volume Snapshots](#) per vedere come funzionerebbe.

Trident fornisce anche il `cloneFromPVC` E `splitOnClone` annotazioni per la creazione di cloni. È possibile utilizzare queste annotazioni per clonare un PVC senza dover ricorrere all'implementazione CSI.

Ecco un esempio: se un utente ha già un PVC chiamato `mysql` , l'utente può creare un nuovo PVC chiamato `mysqlclone` utilizzando l'annotazione, come ad esempio `trident.netapp.io/cloneFromPVC: mysql` . Con questo set di annotazioni, Trident clona il volume corrispondente al PVC `mysql`, anziché effettuare il provisioning di un volume da zero.

Considera i seguenti punti:

- NetApp consiglia di clonare un volume inattivo.
- Un PVC e il suo clone devono trovarsi nello stesso namespace Kubernetes e avere la stessa classe di archiviazione.
- Con il `ontap-nas` E `ontap-san` driver, potrebbe essere desiderabile impostare l'annotazione PVC `trident.netapp.io/splitOnClone` in collaborazione con `trident.netapp.io/cloneFromPVC` . Con `trident.netapp.io/splitOnClone` impostato su `true` Trident divide il volume clonato dal volume padre e, quindi, disaccoppia completamente il ciclo di vita del volume clonato dal suo padre, a scapito della perdita di efficienza di archiviazione. Non impostare `trident.netapp.io/splitOnClone` o impostandolo su `false` comporta una riduzione del consumo di spazio sul backend a scapito della creazione di dipendenze tra i volumi padre e clone, in modo che il volume padre non possa essere eliminato a meno che non venga eliminato prima il clone. Uno scenario in cui la suddivisione del clone ha senso è la clonazione di un volume di database vuoto in cui ci si aspetta che il volume e il suo clone divergano notevolmente e non traggano vantaggio dalle efficienze di archiviazione offerte da ONTAP.

IL `sample-input` La directory contiene esempi di definizioni PVC da utilizzare con Trident. Fare riferimento a per una descrizione completa dei parametri e delle impostazioni associati ai volumi Trident .

Kubernetes PersistentVolume oggetti

Un Kubernetes PersistentVolume L'oggetto rappresenta un elemento di storage reso disponibile al cluster Kubernetes. Ha un ciclo di vita indipendente dal pod che lo utilizza.



Trident crea PersistentVolume oggetti e li registra automaticamente nel cluster Kubernetes in base ai volumi che fornisce. Non ci si aspetta che tu li gestisca da solo.

Quando si crea un PVC che fa riferimento a un Trident-based StorageClass Trident predisponde un nuovo volume utilizzando la classe di archiviazione corrispondente e registra un nuovo PV per tale volume. Nella configurazione del volume fornito e del PV corrispondente, Trident segue le seguenti regole:

- Trident genera un nome PV per Kubernetes e un nome interno che utilizza per il provisioning dello storage. In entrambi i casi, si garantisce che i nomi siano univoci nel loro ambito.
- La dimensione del volume corrisponde il più possibile alla dimensione richiesta nel PVC, anche se potrebbe essere arrotondata alla quantità allocabile più vicina, a seconda della piattaforma.

Kubernetes StorageClass oggetti

Kubernetes StorageClass gli oggetti sono specificati per nome in PersistentVolumeClaims per fornire storage con un set di proprietà. La classe di archiviazione stessa identifica il provisioner da utilizzare e definisce il set di proprietà in termini comprensibili al provisioner.

È uno dei due oggetti di base che devono essere creati e gestiti dall'amministratore. L'altro è l'oggetto backend Trident.

Un Kubernetes StorageClass L'oggetto che utilizza Trident si presenta così:

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: <Name>
provisioner: csi.trident.netapp.io
mountOptions: <Mount Options>
parameters: <Trident Parameters>
allowVolumeExpansion: true
volumeBindingMode: Immediate
```

Questi parametri sono specifici di Trident e indicano a Trident come effettuare il provisioning dei volumi per la classe.

I parametri della classe di archiviazione sono:

Attributo	Tipo	Necessario	Descrizione
attributi	mappa[stringa]stringa	NO	Vedi la sezione attributi qui sotto
pool di stoccaggio	map[string]StringList	NO	Mappa dei nomi backend agli elenchi dei pool di archiviazione all'interno
pool di archiviazione aggiuntivi	map[string]StringList	NO	Mappa dei nomi backend agli elenchi dei pool di archiviazione all'interno
Escludi pool di archiviazione	map[string]StringList	NO	Mappa dei nomi backend agli elenchi dei pool di archiviazione all'interno

Gli attributi di archiviazione e i loro possibili valori possono essere classificati in attributi di selezione del pool di archiviazione e attributi Kubernetes.

Attributi di selezione del pool di archiviazione

Questi parametri determinano quali pool di archiviazione gestiti da Trident devono essere utilizzati per fornire volumi di un determinato tipo.

Attributo	Tipo	Valori	Offerta	Richiesta	Supportato da
media ¹	corda	hdd, ibrido, ssd	Il pool contiene supporti di questo tipo; ibrido significa entrambi	Tipo di supporto specificato	ontap-nas, ontap-nas-economy, ontap-nas-flexgroup, ontap-san, solidfire-san
provisioningType	corda	sottile, spesso	Il pool supporta questo metodo di provisioning	Metodo di provisioning specificato	spesso: tutto ontap; sottile: tutto ontap e solidfire-san
tipo backend	corda	ontap-nas, ontap-nas-economy, ontap-nas-flexgroup, ontap-san, solidfire-san, gcp-cvs, azure-netapp-files, ontap-san-economy	Pool appartiene a questo tipo di backend	Backend specificato	Tutti i conducenti
istantanee	bool	vero, falso	Il pool supporta volumi con snapshot	Volume con snapshot abilitati	ontap-nas, ontap-san, solidfire-san, gcp-cvs
cloni	bool	vero, falso	Il pool supporta la clonazione dei volumi	Volume con cloni abilitati	ontap-nas, ontap-san, solidfire-san, gcp-cvs
crittografia	bool	vero, falso	Il pool supporta volumi crittografati	Volume con crittografia abilitata	ontap-nas, ontap-nas-economy, ontap-nas-flexgroups, ontap-san
IOPS	interno	intero positivo	Pool è in grado di garantire IOPS in questo intervallo	Volume garantito per questi IOPS	solidfire-san

¹: Non supportato dai sistemi ONTAP Select

Nella maggior parte dei casi, i valori richiesti influenzano direttamente il provisioning; ad esempio, la richiesta di provisioning spesso produce un volume con provisioning spesso. Tuttavia, un pool di archiviazione Element utilizza i valori IOPS minimi e massimi offerti per impostare i valori QoS, anziché il valore richiesto. In questo

caso, il valore richiesto viene utilizzato solo per selezionare il pool di archiviazione.

Idealmente, puoi usare `attributes` da solo per modellare le qualità dello spazio di archiviazione necessario a soddisfare le esigenze di una particolare classe. Trident rileva e seleziona automaticamente i pool di archiviazione che corrispondono a *tutti* i `attributes` che specifichi.

Se ti accorgi di non essere in grado di utilizzare `attributes` per selezionare automaticamente le piscine giuste per una classe, puoi usare `storagePools` E `additionalStoragePools` parametri per perfezionare ulteriormente i pool o addirittura per selezionare un set specifico di pool.

Puoi usare il `storagePools` parametro per limitare ulteriormente l'insieme di pool che corrispondono a qualsiasi specificato `attributes`. In altre parole, Trident utilizza l'intersezione dei pool identificati dal `attributes` E `storagePools` parametri per il provisioning. È possibile utilizzare uno dei due parametri da solo oppure entrambi insieme.

Puoi usare il `additionalStoragePools` parametro per estendere il set di pool che Trident utilizza per il provisioning, indipendentemente dai pool selezionati da `attributes` E `storagePools` parametri.

Puoi usare il `excludeStoragePools` parametro per filtrare l'insieme di pool che Trident utilizza per il provisioning. Utilizzando questo parametro vengono rimossi tutti i pool corrispondenti.

Nel `storagePools` E `additionalStoragePools` parametri, ogni voce assume la forma

`<backend>:<storagePoolList>`, Dove `<storagePoolList>` è un elenco separato da virgole di pool di archiviazione per il backend specificato. Ad esempio, un valore per `additionalStoragePools` potrebbe sembrare `ontapnas_192.168.1.100:aggr1,aggr2;solidfire_192.168.1.101:bronze`. Questi elenchi accettano valori regex sia per i valori backend che per quelli dell'elenco. Puoi usare `tridentctl get backend` per ottenere l'elenco dei backend e dei loro pool.

Attributi di Kubernetes

Questi attributi non hanno alcun impatto sulla selezione dei pool di archiviazione/backend da parte di Trident durante il provisioning dinamico. Questi attributi, invece, forniscono semplicemente parametri supportati dai volumi persistenti di Kubernetes. I nodi worker sono responsabili delle operazioni di creazione del file system e potrebbero richiedere utilità del file system, come `xfsprogs`.

Attributo	Tipo	Valori	Descrizione	Fattori rilevanti	Versione di Kubernetes
<code>fsType</code>	corda	<code>ext4, ext3, xfs</code>	Il tipo di file system per i volumi a blocchi	<code>solidfire-san, ontap-nas, ontap-nas-economy, ontap-nas-flexgroup, ontap-san, ontap-san-economy</code>	Tutto

Attributo	Tipo	Valori	Descrizione	Fattori rilevanti	Versione di Kubernetes
consentiEspansioneVolume	booleano	vero, falso	Abilita o disabilita il supporto per l'aumento delle dimensioni del PVC	ontap-nas, ontap-nas-economy, ontap-nas-flexgroup, ontap-san, ontap-san-economy, solidfire-san, gcp-cvs, azure-netapp-files	1.11+
volumeBindingMode	corda	Immediato, WaitForFirstConsumer	Scegli quando si verifica il binding del volume e il provisioning dinamico	Tutto	1,19 - 1,26

- IL `fsType` Il parametro viene utilizzato per controllare il tipo di file system desiderato per le LUN SAN. Inoltre, Kubernetes sfrutta anche la presenza di `fsType` in una classe di archiviazione per indicare l'esistenza di un file system. La proprietà del volume può essere controllata utilizzando `fsGroup` contesto di sicurezza di un pod solo se `fsType` è impostato. Fare riferimento a "[Kubernetes: configurare un contesto di sicurezza per un pod o un contenitore](#)" per una panoramica sull'impostazione della proprietà del volume utilizzando `fsGroup` contesto. Kubernetes applicherà il `fsGroup` valore solo se:
 - `fsType` è impostato nella classe di archiviazione.
 - La modalità di accesso al PVC è RWO.



Per i driver di archiviazione NFS, esiste già un file system come parte dell'esportazione NFS. Per poter utilizzare `fsGroup` la classe di archiviazione deve ancora specificare un `fsType` Puoi impostarlo su `nfs` o qualsiasi valore non nullo.

- Fare riferimento a "[Espandi i volumi](#)" per ulteriori dettagli sull'espansione del volume.
- Il pacchetto di installazione Trident fornisce diverse definizioni di classi di archiviazione di esempio da utilizzare con Trident `insample-input/storage-class-*.yaml`. L'eliminazione di una classe di archiviazione Kubernetes comporta l'eliminazione anche della classe di archiviazione Trident corrispondente.

Kubernetes VolumeSnapshotClass oggetti

Kubernetes `VolumeSnapshotClass` gli oggetti sono analoghi a `StorageClasses`. Consentono di definire più classi di archiviazione e sono referenziati dagli snapshot del volume per associare lo snapshot alla classe di snapshot richiesta. Ogni snapshot del volume è associato a una singola classe di snapshot del volume.

UN `VolumeSnapshotClass` deve essere definito da un amministratore per poter creare snapshot. Viene creata una classe snapshot del volume con la seguente definizione:

```
apiVersion: snapshot.storage.k8s.io/v1
kind: VolumeSnapshotClass
metadata:
  name: csi-snapclass
  driver: csi.trident.netapp.io
  deletionPolicy: Delete
```

IL `driver` specifica a Kubernetes che le richieste per gli snapshot del volume di `csi-snapclass` la classe è gestita da Trident. IL `deletionPolicy` specifica l'azione da intraprendere quando è necessario eliminare uno snapshot. Quando `deletionPolicy` è impostato su `Delete`, gli oggetti snapshot del volume e lo snapshot sottostante sul cluster di archiviazione vengono rimossi quando uno snapshot viene eliminato. In alternativa, impostandolo su `Retain` significa che `VolumeSnapshotContent` e l'istantanea fisica vengono mantenute.

Kubernetes VolumeSnapshot oggetti

Un Kubernetes `VolumeSnapshot` object è una richiesta per creare uno snapshot di un volume. Proprio come un PVC rappresenta una richiesta effettuata da un utente per un volume, uno snapshot del volume è una richiesta effettuata da un utente per creare uno snapshot di un PVC esistente.

Quando arriva una richiesta di snapshot del volume, Trident gestisce automaticamente la creazione dello snapshot per il volume sul backend ed espone lo snapshot creando un'immagine univoca `VolumeSnapshotContent` oggetto. È possibile creare snapshot da PVC esistenti e utilizzarli come `DataSource` durante la creazione di nuovi PVC.

 Il ciclo di vita di un `VolumeSnapshot` è indipendente dal PVC di origine: uno snapshot persiste anche dopo l'eliminazione del PVC di origine. Quando si elimina un PVC a cui sono associati snapshot, Trident contrassegna il volume di supporto per questo PVC in uno stato **Eliminazione**, ma non lo rimuove completamente. Il volume viene rimosso quando vengono eliminati tutti gli snapshot associati.

Kubernetes VolumeSnapshotContent oggetti

Un Kubernetes `VolumeSnapshotContent` L'oggetto rappresenta uno snapshot preso da un volume già sottoposto a provisioning. È analogo a un `PersistentVolume` e indica uno snapshot fornito sul cluster di archiviazione. Simile a `PersistentVolumeClaim` E `PersistentVolume` oggetti, quando viene creato uno snapshot, il `VolumeSnapshotContent` l'oggetto mantiene una mappatura uno a uno con `VolumeSnapshot` oggetto che aveva richiesto la creazione dello snapshot.

IL `VolumeSnapshotContent` l'oggetto contiene dettagli che identificano in modo univoco lo snapshot, come ad esempio `snapshotHandle`. Questo `snapshotHandle` è una combinazione unica del nome del PV e del nome del `VolumeSnapshotContent` oggetto.

Quando arriva una richiesta di snapshot, Trident crea lo snapshot nel backend. Dopo aver creato lo snapshot, Trident configura un `VolumeSnapshotContent` oggetto e quindi espone lo snapshot all'API Kubernetes.

 In genere, non è necessario gestire il `VolumeSnapshotContent` oggetto. Un'eccezione a questo è quando vuoi "["importare uno snapshot del volume"](#) creato al di fuori di Trident.

Kubernetes VolumeGroupSnapshotClass oggetti

Kubernetes VolumeGroupSnapshotClass gli oggetti sono analoghi a VolumeSnapshotClass . Consentono di definire più classi di archiviazione e sono referenziati dagli snapshot del gruppo di volumi per associare lo snapshot alla classe di snapshot richiesta. Ogni snapshot del gruppo di volumi è associato a una singola classe di snapshot del gruppo di volumi.

UN VolumeGroupSnapshotClass dovrebbe essere definito da un amministratore per creare un gruppo di snapshot. Viene creata una classe snapshot del gruppo di volumi con la seguente definizione:

```
apiVersion: groupsnapshot.storage.k8s.io/v1beta1
kind: VolumeGroupSnapshotClass
metadata:
  name: csi-group-snap-class
  annotations:
    kubernetes.io/description: "Trident group snapshot class"
  driver: csi.trident.netapp.io
  deletionPolicy: Delete
```

IL driver specifica a Kubernetes che le richieste per gli snapshot del gruppo di volumi del csi-group-snap-class la classe è gestita da Trident. IL deletionPolicy specifica l'azione da intraprendere quando è necessario eliminare uno snapshot di gruppo. Quando deletionPolicy è impostato su Delete , gli oggetti snapshot del gruppo di volumi e lo snapshot sottostante sul cluster di archiviazione vengono rimossi quando uno snapshot viene eliminato. In alternativa, impostandolo su Retain significa che VolumeGroupSnapshotContent e l'istantanea fisica vengono mantenute.

Kubernetes VolumeGroupSnapshot oggetti

Un Kubernetes VolumeGroupSnapshot object è una richiesta per creare uno snapshot di più volumi. Proprio come un PVC rappresenta una richiesta effettuata da un utente per un volume, uno snapshot del gruppo di volumi è una richiesta effettuata da un utente per creare uno snapshot di un PVC esistente.

Quando arriva una richiesta di snapshot del gruppo di volumi, Trident gestisce automaticamente la creazione dello snapshot del gruppo per i volumi sul backend ed espone lo snapshot creando un'immagine univoca VolumeGroupSnapshotContent oggetto. È possibile creare snapshot da PVC esistenti e utilizzarli come DataSource durante la creazione di nuovi PVC.

 Il ciclo di vita di un VolumeGroupSnapshot è indipendente dal PVC di origine: uno snapshot persiste anche dopo l'eliminazione del PVC di origine. Quando si elimina un PVC a cui sono associati snapshot, Trident contrassegna il volume di supporto per questo PVC in uno stato **Eliminazione**, ma non lo rimuove completamente. Lo snapshot del gruppo di volumi viene rimosso quando vengono eliminati tutti gli snapshot associati.

Kubernetes VolumeGroupSnapshotContent oggetti

Un Kubernetes VolumeGroupSnapshotContent L'oggetto rappresenta uno snapshot di gruppo preso da un volume già sottoposto a provisioning. È analogo a un PersistentVolume e indica uno snapshot fornito sul cluster di archiviazione. Simile a PersistentVolumeClaim E PersistentVolume oggetti, quando viene creato uno snapshot, il VolumeSnapshotContent l'oggetto mantiene una mappatura uno a uno con

VolumeSnapshot oggetto che aveva richiesto la creazione dello snapshot.

IL VolumeGroupSnapshotContent l'oggetto contiene dettagli che identificano il gruppo di snapshot, come ad esempio volumeGroupSnapshotHandle e singoli volumeSnapshotHandles esistenti sul sistema di archiviazione.

Quando arriva una richiesta di snapshot, Trident crea lo snapshot del gruppo di volumi sul backend. Dopo aver creato lo snapshot del gruppo di volumi, Trident configura un VolumeGroupSnapshotContent oggetto e quindi espone lo snapshot all'API Kubernetes.

Kubernetes CustomResourceDefinition oggetti

Le risorse personalizzate di Kubernetes sono endpoint nell'API di Kubernetes definiti dall'amministratore e utilizzati per raggruppare oggetti simili. Kubernetes supporta la creazione di risorse personalizzate per l'archiviazione di una raccolta di oggetti. È possibile ottenere queste definizioni di risorse eseguendo `kubectl get crds`.

Le definizioni di risorse personalizzate (CRD) e i metadati degli oggetti associati vengono archiviati da Kubernetes nel suo archivio metadati. In questo modo si elimina la necessità di un archivio separato per Trident.

Usa Trident CustomResourceDefinition oggetti per preservare l'identità degli oggetti Trident, come i backend Trident, le classi di archiviazione Trident e i volumi Trident. Questi oggetti sono gestiti da Trident. Inoltre, il framework degli snapshot del volume CSI introduce alcuni CRD necessari per definire gli snapshot del volume.

I CRD sono una struttura di Kubernetes. Gli oggetti delle risorse definite sopra vengono creati da Trident. Come semplice esempio, quando un backend viene creato utilizzando `tridentctl`, un corrispondente `tridentbackends` L'oggetto CRD viene creato per essere utilizzato da Kubernetes.

Ecco alcuni punti da tenere a mente sui CRD di Trident:

- Quando Trident viene installato, viene creato un set di CRD che può essere utilizzato come qualsiasi altro tipo di risorsa.
- Quando si disinstalla Trident utilizzando `tridentctl uninstall` comando, i pod Trident vengono eliminati ma i CRD creati non vengono ripuliti. Fare riferimento a "[Disinstallare Trident](#)" per capire come Trident può essere completamente rimosso e riconfigurato da zero.

Trident StorageClass oggetti

Trident crea classi di archiviazione corrispondenti per Kubernetes StorageClass oggetti che specificano `csi.trident.netapp.io` nel loro campo di fornitura. Il nome della classe di archiviazione corrisponde a quello di Kubernetes StorageClass oggetto che rappresenta.



Con Kubernetes, questi oggetti vengono creati automaticamente quando un Kubernetes StorageClass che utilizza Trident come fornitore è registrato.

Le classi di archiviazione comprendono un insieme di requisiti per i volumi. Trident confronta questi requisiti con gli attributi presenti in ciascun pool di archiviazione; se corrispondono, quel pool di archiviazione è una destinazione valida per il provisioning dei volumi utilizzando quella classe di archiviazione.

È possibile creare configurazioni di classi di archiviazione per definire direttamente le classi di archiviazione

utilizzando l'API REST. Tuttavia, per le distribuzioni Kubernetes, ci aspettiamo che vengano create durante la registrazione di nuovi Kubernetes StorageClass oggetti.

Oggetti backend Trident

I backend rappresentano i provider di storage su cui Trident effettua il provisioning dei volumi; una singola istanza Trident può gestire un numero qualsiasi di backend.



Questo è uno dei due tipi di oggetti che puoi creare e gestire autonomamente. L'altro è Kubernetes StorageClass oggetto.

Per maggiori informazioni su come costruire questi oggetti, fare riferimento a ["configurazione dei backend"](#).

Trident StoragePool oggetti

I pool di archiviazione rappresentano le diverse posizioni disponibili per il provisioning su ciascun backend. Per ONTAP, questi corrispondono agli aggregati nelle SVM. Per NetApp HCI/ SolidFire, questi corrispondono alle bande QoS specificate dall'amministratore. Per il Cloud Volumes Service, questi corrispondono alle regioni del provider cloud. Ogni pool di archiviazione ha un set di attributi di archiviazione distinti, che ne definiscono le caratteristiche prestazionali e di protezione dei dati.

A differenza degli altri oggetti qui, i candidati al pool di archiviazione vengono sempre rilevati e gestiti automaticamente.

Trident Volume oggetti

I volumi sono l'unità di base del provisioning e comprendono endpoint back-end, come condivisioni NFS e LUN iSCSI e FC. In Kubernetes, questi corrispondono direttamente a PersistentVolumes. Quando si crea un volume, assicurarsi che disponga di una classe di archiviazione, che determina dove può essere eseguito il provisioning del volume, insieme a una dimensione.



- In Kubernetes, questi oggetti vengono gestiti automaticamente. Puoi visualizzarli per vedere cosa ha fornito Trident .
- Quando si elimina un PV con snapshot associati, il volume Trident corrispondente viene aggiornato allo stato **Eliminazione**. Per eliminare il volume Trident , è necessario rimuovere gli snapshot del volume.

Una configurazione del volume definisce le proprietà che un volume fornito dovrebbe avere.

Attributo	Tipo	Necessario	Descrizione
versione	corda	NO	Versione dell'API Trident ("1")
nome	corda	Sì	Nome del volume da creare
classe di archiviazione	corda	Sì	Classe di archiviazione da utilizzare durante il provisioning del volume
misurare	corda	Sì	Dimensione del volume da fornire in byte

Attributo	Tipo	Necessario	Descrizione
protocollo	corda	NO	Tipo di protocollo da utilizzare; "file" o "blocco"
Nome interno	corda	NO	Nome dell'oggetto sul sistema di archiviazione; generato da Trident
cloneSourceVolume	corda	NO	ontap (nas, san) e solidfire-*: Nome del volume da cui clonare
splitOnClone	corda	NO	ontap (nas, san): divide il clone dal suo genitore
snapshotPolicy	corda	NO	ontap-*: criterio di snapshot da utilizzare
snapshotReserve	corda	NO	ontap-*: percentuale di volume riservata per gli snapshot
Politica di esportazione	corda	NO	ontap-nas*: politica di esportazione da utilizzare
directorystantanea	bool	NO	ontap-nas*: se la directory snapshot è visibile
Permessi unix	corda	NO	ontap-nas*: permessi UNIX iniziali
dimensione del blocco	corda	NO	solidfire-*: dimensione del blocco/settore
file System	corda	NO	Tipo di file system

Trident genera `internalName` durante la creazione del volume. Si compone di due fasi. Innanzitutto, antepone il prefisso di archiviazione (predefinito `trident` o il prefisso nella configurazione del backend) al nome del volume, risultando in un nome del tipo `<prefix>-<volume-name>`. Procede quindi a ripulire il nome, sostituendo i caratteri non consentiti nel backend. Per i backend ONTAP, sostituisce i trattini con i caratteri di sottolineatura (quindi, il nome interno diventa `<prefix>_<volume-name>`). Per i backend Element, sostituisce i trattini bassi con i trattini.

È possibile utilizzare le configurazioni del volume per effettuare il provisioning diretto dei volumi tramite l'API REST, ma nelle distribuzioni Kubernetes ci aspettiamo che la maggior parte degli utenti utilizzi la configurazione standard di Kubernetes `PersistentVolumeClaim` metodo. Trident crea automaticamente questo oggetto volume come parte del processo di provisioning.

Trident Snapshot oggetti

Gli snapshot sono copie di volumi effettuate in un dato momento, che possono essere utilizzate per predisporre nuovi volumi o ripristinarne lo stato. In Kubernetes, questi corrispondono direttamente a `VolumeSnapshotContent` oggetti. Ogni snapshot è associato a un volume, che è l'origine dei dati per lo snapshot.

Ogni Snapshot l'oggetto include le proprietà elencate di seguito:

Attributo	Tipo	Necessario	Descrizione
versione	Corda	Sì	Versione dell'API Trident ("1")
nome	Corda	Sì	Nome dell'oggetto snapshot Trident
Nome interno	Corda	Sì	Nome dell'oggetto snapshot Trident sul sistema di archiviazione
NomeVolume	Corda	Sì	Nome del volume persistente per il quale viene creato lo snapshot
volumeInternalName	Corda	Sì	Nome dell'oggetto volume Trident associato sul sistema di archiviazione



In Kubernetes, questi oggetti vengono gestiti automaticamente. Puoi visualizzarli per vedere cosa ha fornito Trident .

Quando un Kubernetes `VolumeSnapshot` Una volta creata la richiesta dell'oggetto, Trident funziona creando un oggetto snapshot sul sistema di archiviazione di supporto. Il `internalName` di questo oggetto snapshot viene generato combinando il prefisso `snapshot-` con il `UID` del `VolumeSnapshot` oggetto (ad esempio, `snapshot-e8d8a0ca-9826-11e9-9807-525400f3f660`). `volumeName` E `volumeInternalName` vengono popolati ottenendo i dettagli del volume di supporto.

Trident ResourceQuota oggetto

Il set di demoni Trident consuma un `system-node-critical` Classe di priorità: la classe di priorità più elevata disponibile in Kubernetes, per garantire che Trident possa identificare e ripulire i volumi durante l'arresto regolare dei nodi e consentire ai pod daemonset Trident di anticipare i carichi di lavoro con una priorità inferiore nei cluster in cui vi è un'elevata pressione sulle risorse.

Per raggiungere questo obiettivo, Trident impiega un `ResourceQuota` oggetto per garantire che venga soddisfatta una classe di priorità "system-node-critical" sul daemonset Trident . Prima della distribuzione e della creazione del daemonset, Trident cerca `ResourceQuota` oggetto e, se non viene scoperto, lo applica.

Se hai bisogno di un maggiore controllo sulla quota di risorse predefinita e sulla classe di priorità, puoi generare un `custom.yaml` o configurare il `ResourceQuota` oggetto utilizzando il grafico Helm.

Di seguito è riportato un esempio di un oggetto `ResourceQuota` che assegna la priorità al daemonset Trident

```
apiVersion: <version>
kind: ResourceQuota
metadata:
  name: trident-csi
  labels:
    app: node.csi.trident.netapp.io
spec:
  scopeSelector:
    matchExpressions:
      - operator: In
        scopeName: PriorityClass
        values:
          - system-node-critical
```

Per ulteriori informazioni sulle quote di risorse, fare riferimento a "[Kubernetes: quote di risorse](#)" .

Ripulire ResourceQuota se l'installazione fallisce

Nel raro caso in cui l'installazione fallisca dopo l' ResourceQuota l'oggetto è stato creato, primo tentativo "[disinstallazione](#)" e poi reinstallarlo.

Se ciò non funziona, rimuovere manualmente il ResourceQuota oggetto.

Rimuovere ResourceQuota

Se preferisci controllare la tua allocazione delle risorse, puoi rimuovere il Trident ResourceQuota oggetto utilizzando il comando:

```
kubectl delete quota trident-csi -n trident
```

Standard di sicurezza dei pod (PSS) e vincoli di contesto di sicurezza (SCC)

Gli standard di sicurezza dei pod (PSS) e le policy di sicurezza dei pod (PSP) di Kubernetes definiscono i livelli di autorizzazione e limitano il comportamento dei pod. Allo stesso modo, i vincoli di contesto di sicurezza (SCC) di OpenShift definiscono le restrizioni dei pod specifiche per OpenShift Kubernetes Engine. Per fornire questa personalizzazione, Trident abilita determinate autorizzazioni durante l'installazione. Le sezioni seguenti descrivono in dettaglio le autorizzazioni impostate da Trident.



PSS sostituisce Pod Security Policies (PSP). PSP è stato deprecato in Kubernetes v1.21 e verrà rimosso nella v1.25. Per maggiori informazioni, fare riferimento a "[Kubernetes: sicurezza](#)" .

Contesto di sicurezza Kubernetes obbligatorio e campi correlati

Permesso	Descrizione
Privilegiato	CSI richiede che i punti di montaggio siano bidirezionali, il che significa che il pod del nodo Trident deve eseguire un contenitore privilegiato. Per maggiori informazioni, fare riferimento a " Kubernetes: propagazione del mount " .
Rete di host	Necessario per il demone iSCSI. <code>iscsiadm</code> gestisce i mount iSCSI e utilizza la rete host per comunicare con il demone iSCSI.
IPC host	NFS utilizza la comunicazione interprocesso (IPC) per comunicare con NFSD.
PID host	Necessario per iniziare <code>rpc-statd</code> per NFS. Trident interroga i processi host per determinare se <code>rpc-statd</code> è in esecuzione prima di montare i volumi NFS.
Capacità	IL <code>SYS_ADMIN</code> funzionalità è fornita come parte delle funzionalità predefinite per i contenitori privilegiati. Ad esempio, Docker imposta queste funzionalità per i contenitori privilegiati: <code>CapPrm: 0000003ffffffffffff</code> <code>CapEff: 0000003ffffffffffff</code>
Seccomp	Il profilo Seccomp è sempre "Non vincolato" nei contenitori privilegiati; pertanto, non può essere abilitato in Trident.
SELinux	Su OpenShift, i contenitori privilegiati vengono eseguiti in <code>spc_t</code> ("Super Privileged Container") dominio e i contenitori non privilegiati vengono eseguiti nel <code>container_t</code> dominio. SU <code>containerd</code> , con <code>container-selinux</code> installato, tutti i contenitori vengono eseguiti in <code>spc_t</code> dominio, che di fatto disabilita SELinux. Pertanto, Trident non aggiunge <code>seLinuxOptions</code> ai contenitori.
DAC	I contenitori privilegiati devono essere eseguiti come root. I contenitori non privilegiati vengono eseguiti come root per accedere ai socket Unix richiesti da CSI.

Standard di sicurezza del pod (PSS)

Etichetta	Descrizione	Predefinito
<code>pod-security.kubernetes.io/enforce</code> <code>pod-security.kubernetes.io/enforce-version</code>	Consente l'ammissione del controller Trident e dei nodi nello spazio dei nomi di installazione. Non modificare l'etichetta dello spazio dei nomi.	<code>enforce: privileged</code> <code>enforce-version: <version of the current cluster or highest version of PSS tested.></code>



La modifica delle etichette dello spazio dei nomi può comportare la mancata pianificazione dei pod, un messaggio di errore "Errore durante la creazione: ..." o "Avviso: trident-csi-...". Se ciò accade, controllare se l'etichetta dello spazio dei nomi per `privileged` è stato cambiato. In tal caso, reinstallare Trident.

Criteri di sicurezza dei pod (PSP)

Campo	Descrizione	Predefinito
<code>allowPrivilegeEscalation</code>	I contenitori privilegiati devono consentire l'escalation dei privilegi.	<code>true</code>
<code>allowedCSIDrivers</code>	Trident non utilizza volumi effimeri CSI in linea.	Vuoto
<code>allowedCapabilities</code>	I contenitori Trident non privilegiati non richiedono più capacità rispetto al set predefinito e ai contenitori privilegiati vengono concesse tutte le capacità possibili.	Vuoto
<code>allowedFlexVolumes</code>	Trident non utilizza un " Driver FlexVolume " , pertanto non sono inclusi nell'elenco dei volumi consentiti.	Vuoto
<code>allowedHostPaths</code>	Il pod del nodo Trident monta il file system radice del nodo, pertanto non vi è alcun vantaggio nell'impostare questo elenco.	Vuoto
<code>allowedProcMountTypes</code>	Trident non usa nessuno <code>ProcMountTypes</code> .	Vuoto
<code>allowedUnsafeSysctls</code>	Trident non richiede alcuna sicurezza <code>sysctls</code> .	Vuoto
<code>defaultAddCapabilities</code>	Non è necessario aggiungere alcuna funzionalità ai contenitori privilegiati.	Vuoto
<code>defaultAllowPrivilegeEscalation</code>	L'autorizzazione all'escalation dei privilegi viene gestita in ogni pod Trident .	<code>false</code>
<code>forbiddenSysctls</code>	NO <code>sysctls</code> sono consentiti.	Vuoto
<code>fsGroup</code>	I contenitori Trident vengono eseguiti come root.	<code>RunAsAny</code>
<code>hostIPC</code>	Il montaggio dei volumi NFS richiede che l'host IPC comunichi con <code>nfsd</code>	<code>true</code>
<code>hostNetwork</code>	<code>iscsiadm</code> richiede che la rete host comunichi con il demone iSCSI.	<code>true</code>

Campo	Descrizione	Predefinito
hostPID	È necessario il PID host per verificare se <code>rpc-statd</code> è in esecuzione sul nodo.	true
hostPorts	Trident non utilizza alcuna porta host.	Vuoto
privileged	I pod del nodo Trident devono eseguire un contenitore privilegiato per poter montare i volumi.	true
readOnlyRootFilesystem	I pod del nodo Trident devono scrivere sul file system del nodo.	false
requiredDropCapabilities	I pod del nodo Trident eseguono un contenitore privilegiato e non possono rilasciare capacità.	none
runAsGroup	I contenitori Trident vengono eseguiti come root.	RunAsAny
runAsUser	I contenitori Trident vengono eseguiti come root.	runAsAny
runtimeClass	Trident non usa RuntimeClasses .	Vuoto
seLinux	Il Trident non tramonta <code>seLinuxOptions</code> perché attualmente ci sono differenze nel modo in cui i runtime dei container e le distribuzioni di Kubernetes gestiscono SELinux.	Vuoto
supplementalGroups	I contenitori Trident vengono eseguiti come root.	RunAsAny
volumes	I pod Trident richiedono questi plugin di volume.	<code>hostPath</code> , <code>projected</code> , <code>emptyDir</code>

Vincoli di contesto di sicurezza (SCC)

Etichette	Descrizione	Predefinito
allowHostDirVolumePlugin	I pod del nodo Trident montano il file system radice del nodo.	true
allowHostIPC	Il montaggio dei volumi NFS richiede che l'host IPC comunichi con <code>nfsd</code> .	true
allowHostNetwork	<code>iscsiadm</code> richiede che la rete host comunichi con il demone iSCSI.	true
allowHostPID	È necessario il PID host per verificare se <code>rpc-statd</code> è in esecuzione sul nodo.	true

Etichette	Descrizione	Predefinito
allowHostPorts	Trident non utilizza alcuna porta host.	false
allowPrivilegeEscalation	I contenitori privilegiati devono consentire l'escalation dei privilegi.	true
allowPrivilegedContainer	I pod del nodo Trident devono eseguire un contenitore privilegiato per poter montare i volumi.	true
allowedUnsafeSysctls	Trident non richiede alcuna sicurezza sysctls .	none
allowedCapabilities	I contenitori Trident non privilegiati non richiedono più capacità rispetto al set predefinito e ai contenitori privilegiati vengono concesse tutte le capacità possibili.	Vuoto
defaultAddCapabilities	Non è necessario aggiungere alcuna funzionalità ai contenitori privilegiati.	Vuoto
fsGroup	I contenitori Trident vengono eseguiti come root.	RunAsAny
groups	Il presente SCC è specifico per Trident ed è vincolante per il suo utente.	Vuoto
readOnlyRootFilesystem	I pod del nodo Trident devono scrivere sul file system del nodo.	false
requiredDropCapabilities	I pod del nodo Trident eseguono un contenitore privilegiato e non possono rilasciare capacità.	none
runAsUser	I contenitori Trident vengono eseguiti come root.	RunAsAny
seLinuxContext	Il Trident non tramonta seLinuxOptions perché attualmente ci sono differenze nel modo in cui i runtime dei container e le distribuzioni di Kubernetes gestiscono SELinux.	Vuoto
seccompProfiles	I contenitori privilegiati vengono sempre eseguiti in modalità "Unconfined".	Vuoto
supplementalGroups	I contenitori Trident vengono eseguiti come root.	RunAsAny
users	Viene fornita una voce per associare questo SCC all'utente Trident nello spazio dei nomi Trident .	n / a

Etichette	Descrizione	Predefinito
volumes	I pod Trident richiedono questi plugin di volume.	hostPath, downwardAPI, projected, emptyDir

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 COMMERCIALITÀ 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.