Usa Trident

Trident

NetApp
October 29, 2026

This PDF was generated from https://docs.netapp.com/it-it/trident-2606/trident-use/worker-node-prep.html on October 29, 2026. Always check docs.netapp.com for the latest.

Sommario

Usa Trident . . . 1
- Prepara il nodo worker di Trident . . . 1
  - Selezionare gli strumenti giusti . . . 1
  - Rilevamento del servizio nodo . . . 1
  - volumi NFS . . . 2
  - volumi iSCSI . . . 2
  - Volumi NVMe/TCP . . . 6
  - SCSI over FC volumi . . . 7
  - Prepararsi al provisioning dei volumi SMB . . . 9
- Configura e gestisci i backend . . . 10
  - Configurare i backend di Trident . . . 10
  - Azure NetApp Files . . . 11
  - Google Cloud NetApp Volumes . . . 27
  - Configura un backend NetApp HCI o SolidFire . . . 55
  - Driver SAN ONTAP . . . 60
  - Driver NAS ONTAP . . . 86
  - Usa un posizionamento bilanciato con i backend ONTAP . . . 120
  - Amazon FSx for NetApp ONTAP . . . 122
  - Crea backend con kubectl . . . 156
  - Gestisci i backend . . . 162
- Crea e gestisci classi di archiviazione . . . 171
  - Crea una classe di storage Trident . . . 171
  - Gestisci le classi di storage di Trident . . . 174
- Effettua il provisioning e gestisci i volumi . . . 175
  - Esegui il provisioning di un volume Trident . . . 175
  - Sposta un volume . . . 178
  - Espandi i volumi Trident . . . 182
  - Comprendere i limiti del sottosistema RWX NVMe . . . 192
  - Scalabilità del controller . . . 193
  - Espansione automatica del volume . . . 198
  - Gestisci le policy di Autogrow . . . 204
  - Importa i volumi Trident . . . 213
  - Personalizza i nomi e le etichette dei volumi di Trident . . . 225
  - Condividi un volume NFS Trident tra namespace . . . 227
  - Clona i volumi Trident tra namespace . . . 231
  - Replicare i volumi utilizzando SnapMirror . . . 233
  - Usa la topologia CSI . . . 240
  - Lavora con gli snapshot di Trident . . . 247
  - Lavora con gli snapshot dei gruppi di volumi di Trident . . . 254

Usa Trident

Prepara il nodo worker di Trident

Tutti i nodi worker nel cluster Kubernetes devono essere in grado di montare i volumi che hai fornito per i tuoi pod. Per preparare i nodi worker, è necessario installare gli strumenti NFS, iSCSI, NVMe/TCP o FC in base alla selezione del driver.

Selezionare gli strumenti giusti

Se si utilizza una combinazione di driver, è necessario installare tutti gli strumenti necessari per i driver. Le versioni recenti di Red Hat Enterprise Linux CoreOS (RHCOS) hanno gli strumenti installati di default.

Strumenti NFS

"Installa gli strumenti NFS" se stai utilizzando: `ontap-nas`, `ontap-nas-economy`, `ontap-nas-flexgroup`, o `azure-netapp-files`.

Strumenti iSCSI

"Installare gli strumenti iSCSI" se stai utilizzando: `ontap-san`, `ontap-san-economy`, `solidfire-san`.

Strumenti NVMe

"Installa gli strumenti NVMe" se si utilizza `ontap-san` per nonvolatile memory express (NVMe) su TCP (NVMe/TCP) protocollo.

> NetApp consiglia ONTAP 9.12 o versione successiva per NVMe/TCP.

Strumenti SCSI su FC

Fare riferimento a "Modalità per configurare gli host SAN FC e FC-NVMe" per ulteriori informazioni sulla configurazione degli host SAN FC e FC-NVMe.

"Installa gli strumenti FC" se si utilizza `ontap-san` con sanType `fcp` (SCSI su FC).

Punti da considerare: * SCSI su FC è supportato su OpenShift e KubeVirt ambienti. * SCSI su FC non è supportato su Docker. * L'auto-riparazione iSCSI non è applicabile a SCSI su FC.

Strumenti SMB

"Prepararsi al provisioning dei volumi SMB" se si utilizza: `ontap-nas` per fornire volumi SMB.

Rilevamento del servizio nodo

Trident tenta di rilevare automaticamente se il nodo può eseguire i servizi iSCSI o NFS.

> La rilevazione dei servizi del nodo identifica i servizi rilevati, ma non garantisce che siano configurati correttamente. Al contrario, l'assenza di un servizio rilevato non garantisce che il montaggio del volume fallirà.

Rivedi gli eventi

Trident crea eventi per il nodo per identificare i servizi rilevati. Per esaminare questi eventi, eseguire:

```
kubectl get event -A --field-selector involvedObject.name=<Kubernetes node name>
```

Esamina i servizi scoperti

Trident identifica i servizi abilitati per ciascun nodo sul Trident node CR. Per visualizzare i servizi rilevati, eseguire:

```
tridentctl get node -o wide -n <Trident namespace>
```

volumi NFS

Installa gli strumenti NFS utilizzando i comandi del tuo sistema operativo. Assicurati che il servizio NFS sia avviato durante il tempo di avvio.

RHEL 8+

```
sudo yum install -y nfs-utils
```

Ubuntu

```
sudo apt-get install -y nfs-common
```

Riavvia i nodi worker dopo aver installato gli strumenti NFS per prevenire errori durante il collegamento dei volumi ai container.

volumi iSCSI

Trident può stabilire automaticamente una sessione iSCSI, analizzare le LUN, rilevare dispositivi multipath, formattarli e montarli su un pod.

Capacità di auto-riparazione iSCSI

Per i sistemi ONTAP, Trident esegue l'auto-riparazione iSCSI ogni cinque minuti per:

1. **Identifica** lo stato della sessione iSCSI desiderato e lo stato della sessione iSCSI corrente.
2. **Confronta** lo stato desiderato con quello attuale per identificare le riparazioni necessarie. Trident determina le priorità di riparazione e quando preemptare le riparazioni.
3. **Eseguire le riparazioni** necessarie per riportare lo stato della sessione iSCSI corrente allo stato della sessione iSCSI desiderato.

I log delle attività di auto-riparazione si trovano nel `trident-main` container sul rispettivo pod Daemonset. Per visualizzare i log, è necessario aver impostato `debug` su "true" durante l'installazione di Trident.

Le funzionalità di auto-riparazione iSCSI di Trident possono aiutare a prevenire:

- Sessioni iSCSI obsolete o non funzionanti che potrebbero verificarsi dopo un problema di connettività di rete. In caso di sessione obsoleta, Trident attende sette minuti prima di disconnettersi per ristabilire la connessione con un portale.

> Ad esempio, se i segreti CHAP vengono ruotati sul storage controller e la rete perde la connettività, i vecchi (*obsoleti*) segreti CHAP potrebbero persistere. La funzione di auto-riparazione può riconoscere questo e ristabilire automaticamente la sessione per applicare i segreti CHAP aggiornati.

- Sessioni iSCSI mancanti
- LUN mancanti

Punti da considerare prima di aggiornare Trident

- Se vengono utilizzati solo igroup per nodo (introdotti nella versione 23.04+), la funzione di auto-riparazione iSCSI avvierà nuove scansioni SCSI per tutti i dispositivi nel bus SCSI.
- Se vengono utilizzati solo igroup con ambito backend (obsoleti a partire dalla versione 23.04), la funzione di auto-riparazione iSCSI avvierà nuove scansioni SCSI per gli ID LUN esatti nel bus SCSI.
- Se viene utilizzato un mix di igroup per nodo e igroup con ambito backend, la funzione di auto-riparazione iSCSI avvierà nuove scansioni SCSI per gli ID LUN esatti nel bus SCSI.

Installare gli strumenti iSCSI

Installa gli strumenti iSCSI utilizzando i comandi per il tuo sistema operativo.

Prima di iniziare

- Ogni nodo nel cluster Kubernetes deve avere un IQN univoco. **Questo è un prerequisito necessario**.
- Se si utilizza RHCOS versione 4.5 o successiva, o un'altra distribuzione Linux compatibile con RHEL, con il `solidfire-san` driver ed Element OS 12.5 o precedente, assicurarsi che l'algoritmo di autenticazione CHAP sia impostato su MD5 in `/etc/iscsi/iscsid.conf`. Gli algoritmi CHAP sicuri conformi a FIPS SHA1, SHA-256 e SHA3-256 sono disponibili con Element 12.7.

```
sudo sed -i 's/^\(node.session.auth.chap_algs\).*/\1 = MD5/'
/etc/iscsi/iscsid.conf
```

- Quando si utilizzano nodi worker che eseguono RHEL/Red Hat Enterprise Linux CoreOS (RHCOS) con iSCSI PV, specificare `discard` mountOption nella StorageClass per eseguire la space reclamation inline. Fai riferimento a "Documentazione Red Hat".
- Assicurati di aver effettuato l'aggiornamento alla versione più recente del `multipath-tools`.

RHEL 8+

1. Installare i seguenti pacchetti di sistema:

```
sudo yum install -y lsscsi iscsi-initiator-utils device-mapper-
```

```
multipath
```

2. Verificare che la versione di iscsi-initiator-utils sia 6.2.0.874-2.el7 o successiva:

```
rpm -q iscsi-initiator-utils
```

3. Imposta la scansione su manuale:

```
sudo sed -i 's/^\(node.session.scan\).*/\1 = manual/'
/etc/iscsi/iscsid.conf
```

4. Abilita il multipathing:

```
sudo mpathconf --enable --with_multipathd y --find_multipaths n
```

> Assicurarsi `/etc/multipath.conf` che contenga `find_multipaths no` sotto `defaults`.

5. Assicurarsi che `iscsid` e `multipathd` siano in funzione:

```
sudo systemctl enable --now iscsid multipathd
```

6. Abilita e avvia `iscsi`:

```
sudo systemctl enable --now iscsi
```

Ubuntu

1. Installare i seguenti pacchetti di sistema:

```
sudo apt-get install -y open-iscsi lsscsi sg3-utils multipath-tools
scsitools
```

2. Verificare che la versione di open-iscsi sia 2.0.874-5ubuntu2.10 o successiva (per bionic) o 2.0.874-7.1ubuntu6.1 o successiva (per focal):

```
dpkg -l open-iscsi
```

3. Imposta la scansione su manuale:

```
sudo sed -i 's/^\(node.session.scan\).*/\1 = manual/'
/etc/iscsi/iscsid.conf
```

4. Abilita il multipathing:

```
sudo tee /etc/multipath.conf <<-EOF
defaults {
    user_friendly_names yes
    find_multipaths no
}
EOF
sudo systemctl enable --now multipath-tools.service
sudo service multipath-tools restart
```

> Assicurarsi `/etc/multipath.conf` che contenga `find_multipaths no` sotto `defaults`.

5. Assicurarsi che `open-iscsi` e `multipath-tools` siano abilitati e funzionanti:

```
sudo systemctl status multipath-tools
sudo systemctl enable --now open-iscsi.service
sudo systemctl status open-iscsi
```

> Per Ubuntu 18.04, è necessario rilevare le porte di destinazione con `iscsiadm` prima di avviare `open-iscsi` affinché il demone iSCSI si avvii. In alternativa, è possibile modificare il servizio `iscsi` per avviare `iscsid` automaticamente.

Configura o disabilita l'auto-riparazione iSCSI

È possibile configurare le seguenti impostazioni di auto-riparazione Trident iSCSI per correggere le sessioni obsolete:

- **Intervallo di auto-riparazione iSCSI**: determina la frequenza con cui viene invocata l'auto-riparazione iSCSI (impostazione predefinita: 5 minuti). È possibile configurarlo per essere eseguito più frequentemente impostando un numero inferiore o meno frequentemente impostando un numero superiore.

> Impostare l'intervallo di auto-riparazione iSCSI su 0 interrompe completamente l'auto-riparazione iSCSI. Non consigliamo di disabilitare l'auto-riparazione iSCSI; dovrebbe essere disabilitata solo in determinati scenari quando l'auto-riparazione iSCSI non funziona come previsto o per scopi di debug.

- **iSCSI Self-Healing Wait Time**: determina la durata che la funzionalità di auto-riparazione iSCSI attende prima di disconnettersi da una sessione non integra e tentare di accedere nuovamente (impostazione predefinita: 7 minuti). Puoi configurarlo su un numero maggiore affinché le sessioni identificate come non

integre debbano attendere più a lungo prima di essere disconnesse e venga quindi effettuato un tentativo di nuovo accesso, oppure su un numero minore per disconnettersi e accedere prima.

Helm

Per configurare o modificare le impostazioni di auto-riparazione iSCSI, passare i parametri `iscsiSelfHealingInterval` e `iscsiSelfHealingWaitTime` durante l'installazione o l'aggiornamento di helm.

Il seguente esempio imposta l'intervallo di auto-riparazione iSCSI su 3 minuti e il tempo di attesa di auto-riparazione su 6 minuti:

```
helm install trident trident-operator-100.2606.0.tgz --set
iscsiSelfHealingInterval=3m0s --set iscsiSelfHealingWaitTime=6m0s -n
trident
```

tridentctl

Per configurare o modificare le impostazioni di auto-riparazione iSCSI, passare i parametri `iscsi-self-healing-interval` e `iscsi-self-healing-wait-time` durante l'installazione o l'aggiornamento di tridentctl.

Il seguente esempio imposta l'intervallo di auto-riparazione iSCSI su 3 minuti e il tempo di attesa di auto-riparazione su 6 minuti:

```
tridentctl install --iscsi-self-healing-interval=3m0s --iscsi-self
-healing-wait-time=6m0s -n trident
```

Volumi NVMe/TCP

Installare gli strumenti NVMe utilizzando i comandi per il proprio sistema operativo.

- NVMe richiede RHEL 9 o versioni successive.
- Se la versione del kernel del nodo Kubernetes è troppo vecchia o se il pacchetto NVMe non è disponibile per la versione del kernel, potrebbe essere necessario aggiornare la versione del kernel del nodo a una con il pacchetto NVMe.

RHEL 9

```
sudo yum install nvme-cli
sudo yum install linux-modules-extra-$(uname -r)
sudo modprobe nvme-tcp
```

Ubuntu

```
sudo apt install nvme-cli
```

```
sudo apt -y install linux-modules-extra-$(uname -r)
sudo modprobe nvme-tcp
```

Verifica installazione

Dopo l'installazione, verifica che ogni nodo nel cluster Kubernetes abbia un NQN univoco utilizzando il comando:

```
cat /etc/nvme/hostnqn
```

> Trident modifica il `ctrl_device_tmo` valore per garantire che NVMe non rinunci al percorso se va giù. Non modificare questa impostazione.

SCSI over FC volumi

Ora è possibile utilizzare il protocollo Fibre Channel (FC) con Trident per fornire e gestire risorse di storage su sistemi ONTAP.

Prerequisiti

Configurare le impostazioni di rete e dei nodi richieste per FC.

Impostazioni di rete

1. Ottieni il WWPN delle interfacce di destinazione. Consulta "network interface show" per maggiori informazioni.
2. Ottieni il WWPN per le interfacce su initiator (Host).

   Fare riferimento alle utilità del sistema operativo host corrispondenti.

3. Configurare la suddivisione in zone sullo switch FC utilizzando i WWPN di Host e target.

   Fare riferimento alla documentazione del rispettivo switch vendor per informazioni.

   Per maggiori dettagli, fare riferimento alla seguente documentazione ONTAP:

   - "Panoramica sulla zonizzazione Fibre Channel e FCoE"
   - "Modalità per configurare gli host SAN FC e FC-NVMe"

Installa gli strumenti FC

Installa gli strumenti FC utilizzando i comandi del tuo sistema operativo.

- Quando si utilizzano nodi worker che eseguono RHEL/Red Hat Enterprise Linux CoreOS (RHCOS) con PV FC, specificare `discard` mountOption nel StorageClass per eseguire la space reclamation in linea. Fai riferimento a "Documentazione Red Hat".

RHEL 8+

1. Installare i seguenti pacchetti di sistema:

```
sudo yum install -y lsscsi device-mapper-multipath
```

2. Abilita il multipathing:

```
sudo mpathconf --enable --with_multipathd y --find_multipaths n
```

> Assicurarsi `/etc/multipath.conf` che contenga `find_multipaths no` sotto `defaults`.

3. Assicurati che `multipathd` sia in esecuzione:

```
sudo systemctl enable --now multipathd
```

Ubuntu

1. Installare i seguenti pacchetti di sistema:

```
sudo apt-get install -y lsscsi sg3-utils multipath-tools scsitools
```

2. Abilita il multipathing:

```
sudo tee /etc/multipath.conf <<-EOF
defaults {
    user_friendly_names yes
    find_multipaths no
}
EOF
sudo systemctl enable --now multipath-tools.service
sudo service multipath-tools restart
```

> Assicurarsi `/etc/multipath.conf` che contenga `find_multipaths no` sotto `defaults`.

3. Assicurarsi che `multipath-tools` sia abilitato e in esecuzione:

```
sudo systemctl status multipath-tools
```

Prepararsi al provisioning dei volumi SMB

È possibile eseguire il provisioning dei volumi SMB utilizzando `ontap-nas` driver.

> È necessario configurare entrambi i protocolli NFS e SMB/CIFS sull'SVM per creare un volume SMB `ontap-nas-economy` per i cluster ONTAP on-premises. La mancata configurazione di uno di questi protocolli causerà il fallimento della creazione del volume SMB.

> `autoExportPolicy` non è supportato per i volumi SMB.

Prima di iniziare

Per poter eseguire il provisioning dei volumi SMB, è necessario disporre di quanto segue.

- Un cluster Kubernetes con un nodo controller Linux e almeno un nodo worker Windows che esegue Windows Server 2022. Trident supporta volumi SMB montati solo su pod in esecuzione su nodi Windows.
- Almeno un secret di Trident contenente le credenziali di Active Directory. Per generare il secret `smbcreds`:

```
kubectl create secret generic smbcreds --from-literal username=user
--from-literal password='password'
```

- Un CSI proxy configurato come servizio Windows. Per configurare un `csi-proxy`, fare riferimento a "GitHub: CSI Proxy" o "GitHub: CSI Proxy per Windows" per i nodi Kubernetes in esecuzione su Windows.

Passaggi

1. Per ONTAP on-premises, puoi facoltativamente creare una condivisione SMB oppure Trident può crearne una per te.

   > Le condivisioni SMB sono necessarie per Amazon FSx per ONTAP.

   È possibile creare le condivisioni SMB admin in due modi: utilizzando lo snap-in "Microsoft Management Console" Shared Folders o utilizzando la ONTAP CLI. Per creare le condivisioni SMB utilizzando la ONTAP CLI:

   a. Se necessario, crea la struttura del percorso della directory per la condivisione.

      Il `vserver cifs share create` comando controlla il percorso specificato nell'opzione -path durante la creazione della condivisione. Se il percorso specificato non esiste, il comando fallisce.

   b. Crea una condivisione SMB associata alla SVM specificata:

```
vserver cifs share create -vserver vserver_name -share-name
share_name -path path [-share-properties share_properties,...]
[other_attributes] [-comment text]
```

   c. Verificare che la condivisione sia stata creata:

```
vserver cifs share show -share-name share_name
```

Consultare "Creare una condivisione SMB" per tutti i dettagli.

2. Quando si crea il backend, è necessario configurare quanto segue per specificare i volumi SMB. Per tutte le opzioni di configurazione del backend FSx per ONTAP, fare riferimento a "Opzioni ed esempi di configurazione di FSx per ONTAP".

Parametro	Descrizione	Esempio
`smbShare`	È possibile specificare uno dei seguenti valori: il nome di una condivisione SMB creata tramite Microsoft Management Console o ONTAP CLI; un nome per consentire a Trident di creare la condivisione SMB; oppure è possibile lasciare il parametro vuoto per impedire l'accesso condiviso ai volumi. Questo parametro è facoltativo per ONTAP on-premises. Questo parametro è obbligatorio per Amazon FSx for ONTAP backends e non può essere vuoto.	`smb-share`
`nasType`	**Deve essere impostato su `smb`.** Se nullo, il valore predefinito è `nfs`.	`smb`
`securityStyle`	Stile di sicurezza per i nuovi volumi. **Deve essere impostato su `ntfs` o `mixed` per i volumi SMB.**	`ntfs` o `mixed` per i volumi SMB
`unixPermissions`	Modalità per nuovi volumi. **Deve essere lasciato vuoto per i volumi SMB.**	""

Configura e gestisci i backend

Configurare i backend di Trident

Un backend definisce la relazione tra Trident e un sistema storage. Indica a Trident come comunicare con quel sistema storage e come Trident deve eseguire il provisioning dei volumi da esso.

Trident offre automaticamente pool di storage da backend che soddisfano i requisiti definiti da una storage class. Scopri come configurare il backend per il tuo storage system.

- "Configura un backend Azure NetApp Files"
- "Configura un backend Google Cloud NetApp Volumes"
- "Configura un backend NetApp HCI o SolidFire"
- "Configurare un backend con driver NAS ONTAP o Cloud Volumes ONTAP"
- "Configurare un backend con driver SAN ONTAP o Cloud Volumes ONTAP"
- "Usa Trident con Amazon FSx for NetApp ONTAP"

Azure NetApp Files

Configura un backend Azure NetApp Files

Utilizza Azure NetApp Files come backend per Trident. Questo backend supporta volumi NFS e SMB. Trident supporta identità gestite e workload identity per i cluster Azure Kubernetes Service (AKS).

Ambienti cloud Azure supportati

Trident supporta backend Azure NetApp Files in più ambienti cloud di Azure.

I cloud Azure supportati includono:

- Azure Commercial
- Azure Government (Azure Government / MAG)
- Azure Cina

Quando distribuisci Trident o configuri un backend Azure NetApp Files, assicurati che Azure Resource Manager e gli endpoint di autenticazione corrispondano al tuo ambiente cloud di Azure.

Esaminare il supporto del driver Azure NetApp Files

Trident fornisce il seguente driver di storage Azure NetApp Files.

Le modalità di accesso supportate includono *ReadWriteOnce* (RWO), *ReadOnlyMany* (ROX), *ReadWriteMany* (RWX) e *ReadWriteOncePod* (RWOP).

Driver	Protocollo	volumeMode	Modalità di accesso supportate	File system supportati
`azure-netapp-files`	NFS SMB	Filesystem	RWO, ROX, RWX, RWOP	`nfs`, `smb`

Considerazioni sulla revisione

- Azure NetApp Files non supporta volumi inferiori a 50 GiB. Trident crea un volume da 50 GiB quando viene richiesto un volume più piccolo.
- Trident supporta volumi SMB montati solo su pod in esecuzione su nodi Windows.
- Le distribuzioni di Azure NetApp Files nei cloud Azure non commerciali richiedono endpoint di Azure Resource Manager e di autenticazione specifici per il cloud. Assicurarsi che Trident e qualsiasi configurazione di backend utilizzino gli endpoint appropriati per l'ambiente cloud Azure.

Usa identità gestite per AKS

Trident supporta "identità gestite" per i cluster AKS.

Se si utilizza `tridentctl` per creare o gestire backend di Azure NetApp Files, assicurarsi che sia configurato per il corretto ambiente cloud di Azure.

Per utilizzare le identità gestite, è necessario avere:

- Un cluster Kubernetes distribuito utilizzando AKS

- Identità gestite configurate sul cluster Kubernetes AKS
- Trident installato con `cloudProvider` impostato su `"Azure"`

Operatore Trident

Modifica `tridentorchestrator_cr.yaml` e imposta `cloudProvider` su `"Azure"`.

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

Helm

Il seguente esempio installa Trident e imposta `cloudProvider` utilizzando la variabile di ambiente `$CP`:

```
helm install trident trident-operator-100.2606.0.tgz --create-namespace
--namespace <trident-namespace> --set cloudProvider=$CP
```

<code>tridentctl</code>

Il seguente esempio installa Trident e imposta il flag `cloud-provider` su `Azure`:

```
tridentctl install --cloud-provider="Azure" -n trident
```

Usa l'identità del carico di lavoro per AKS

Workload identity consente ai pod Kubernetes di accedere alle risorse di Azure autenticandosi come workload identity.

Se si utilizza `tridentctl` per creare o gestire backend di Azure NetApp Files, assicurarsi che sia configurato per il corretto ambiente cloud di Azure.

Per utilizzare l'identità del carico di lavoro, è necessario disporre di:

- Un cluster Kubernetes distribuito utilizzando AKS
- Identità del carico di lavoro e oidc-issuer configurati sul cluster AKS Kubernetes
- Trident installato con `cloudProvider` impostato su `"Azure"` e `cloudIdentity` impostato sul valore dell'identità del carico di lavoro

Operatore Trident

Modifica `tridentorchestrator_cr.yaml` e imposta `cloudProvider` su `"Azure"`. Imposta `cloudIdentity` su `azure.workload.identity/client-id: xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxx`.

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

Helm

Imposta i valori per i flag **cloud-provider (CP)** e **cloud-identity (CI)** utilizzando le seguenti variabili di ambiente:

```
export CP="Azure"
export CI="'azure.workload.identity/client-id: xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxx'"
```

Il seguente esempio installa Trident e imposta `cloudProvider` usando `$CP` e imposta `cloudIdentity` usando `$CI`:

```
helm install trident trident-operator-100.6.0.tgz --set
cloudProvider=$CP --set cloudIdentity="$CI"
```

<code>tridentctl</code>

Imposta i valori per i flag **cloud provider** e **cloud identity** utilizzando le seguenti variabili di ambiente:

```
export CP="Azure"
export CI="azure.workload.identity/client-id: xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxx"
```

Il seguente esempio installa Trident e imposta `cloud-provider` a `$CP` e `cloud-identity` a `$CI`:

```
tridentctl install --cloud-provider=$CP --cloud-identity="$CI" -n
trident
```

Prepararsi a configurare un backend Azure NetApp Files

Prima di poter configurare il backend Azure NetApp Files, è necessario assicurarsi che siano soddisfatti i seguenti requisiti.

Ambienti cloud Azure supportati

Trident supporta backend Azure NetApp Files in più ambienti cloud di Azure.

I cloud Azure supportati includono:

- Azure Commercial
- Azure Government (Azure Government / MAG)
- Azure Cina

Durante la preparazione dell'ambiente, assicurati che la sottoscrizione di Azure, la configurazione dell'identità e le risorse di Azure NetApp Files vengano create nell'ambiente cloud di Azure appropriato.

Prerequisiti per volumi NFS e SMB

Se si utilizza Azure NetApp Files per la prima volta o in una nuova posizione, è necessaria una configurazione iniziale per impostare Azure NetApp Files e creare un volume NFS. Fai riferimento a "Azure: Configurare Azure NetApp Files e creare un volume NFS".

Per configurare e utilizzare un "Azure NetApp Files" backend, è necessario quanto segue:

> - `subscriptionID`, `tenantID`, `clientID`, `location` e `clientSecret` sono opzionali quando si utilizzano identità gestite su un cluster AKS.
> - `tenantID`, `clientID` e `clientSecret` sono opzionali quando si utilizza un'identità cloud su un cluster AKS.
> - Le distribuzioni di Azure NetApp Files nei cloud Azure non commerciali richiedono endpoint di Azure Resource Manager e di autenticazione specifici per il cloud. Assicurarsi che Trident e qualsiasi configurazione di backend utilizzino gli endpoint appropriati per l'ambiente cloud Azure.

- Un pool di capacità. Fai riferimento a "Microsoft: Crea un pool di capacità per Azure NetApp Files".
- Una sottorete delegata ad Azure NetApp Files. Fai riferimento a "Microsoft: Delegare una subnet ad Azure NetApp Files".
- `subscriptionID` da un abbonamento Azure con Azure NetApp Files abilitato.
- `tenantID`, `clientID` e `clientSecret` da un "Registrazione dell'app" in Azure Active Directory con permessi sufficienti per il servizio Azure NetApp Files. La registrazione dell'app deve utilizzare uno dei seguenti metodi:
  - Il ruolo di Owner o Contributor "predefinito da Azure".
  - Un "ruolo Contributor personalizzato" al livello di sottoscrizione (`assignableScopes`) con i seguenti permessi che sono limitati solo a ciò che Trident richiede. Dopo aver creato il ruolo personalizzato, "assegna il ruolo utilizzando il portale Azure".

    ▼ Ruolo di collaboratore personalizzato

```
{
  "id": "/subscriptions/<subscription-
id>/providers/Microsoft.Authorization/roleDefinitions/<role-
definition-id>",
  "properties": {
    "roleName": "custom-role-with-limited-perms",
    "description": "custom role providing limited permissions",
    "assignableScopes": [
      "/subscriptions/<subscription-id>"
    ],
    "permissions": [
      {
        "actions": [
          "Microsoft.NetApp/netAppAccounts/capacityPools/read",
          "Microsoft.NetApp/netAppAccounts/capacityPools/write",

"Microsoft.NetApp/netAppAccounts/capacityPools/volumes/read",

"Microsoft.NetApp/netAppAccounts/capacityPools/volumes/write",

"Microsoft.NetApp/netAppAccounts/capacityPools/volumes/delete",

"Microsoft.NetApp/netAppAccounts/capacityPools/volumes/snapshots/re
ad",

"Microsoft.NetApp/netAppAccounts/capacityPools/volumes/snapshots/wr
ite",

"Microsoft.NetApp/netAppAccounts/capacityPools/volumes/snapshots/de
lete",

"Microsoft.NetApp/netAppAccounts/capacityPools/volumes/MountTargets
/read",
          "Microsoft.Network/virtualNetworks/read",
          "Microsoft.Network/virtualNetworks/subnets/read",

"Microsoft.Features/featureProviders/subscriptionFeatureRegistratio
ns/read",

"Microsoft.Features/featureProviders/subscriptionFeatureRegistratio
ns/write",

"Microsoft.Features/featureProviders/subscriptionFeatureRegistratio
ns/delete",
          "Microsoft.Features/features/read",
```

```
            "Microsoft.Features/operations/read",
            "Microsoft.Features/providers/features/read",
            "Microsoft.Features/providers/features/register/action",

"Microsoft.Features/providers/features/unregister/action",

"Microsoft.Features/subscriptionFeatureRegistrations/read"
          ],
          "notActions": [],
          "dataActions": [],
          "notDataActions": []
        }
      ]
    }
}
```

- L' `location`Azure che contiene almeno un "sottorete delegata". A partire da Trident 22.01, il parametro `location` è un campo obbligatorio al livello superiore del file di configurazione del backend. I valori di posizione specificati nei pool virtuali vengono ignorati.
- Per utilizzare `Cloud Identity`, ottenere il `client ID` da un "identità gestita assegnata all'utente" e specificare quell'ID in `azure.workload.identity/client-id: xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxx`.

Requisiti aggiuntivi per i volumi SMB

Per creare un volume SMB, è necessario disporre di:

- Active Directory configurato e connesso ad Azure NetApp Files. Fai riferimento a "Microsoft: Crea e gestisci le connessioni Active Directory per Azure NetApp Files".
- Un cluster Kubernetes con un nodo controller Linux e almeno un nodo worker Windows che esegue Windows Server 2022. Trident supporta volumi SMB montati solo su pod in esecuzione su nodi Windows.
- Almeno un segreto Trident contenente le credenziali di Active Directory, così che Azure NetApp Files possa autenticarsi ad Active Directory. Per generare il secret `smbcreds`:

```
kubectl create secret generic smbcreds --from-literal username=user
--from-literal password='password'
```

- Un CSI proxy configurato come servizio Windows. Per configurare un `csi-proxy`, fare riferimento a "GitHub: CSI Proxy" o "GitHub: CSI Proxy per Windows" per i nodi Kubernetes in esecuzione su Windows.

Opzioni di configurazione del backend di Azure NetApp Files ed esempi

Scopri le opzioni di configurazione del backend NFS e SMB per Azure NetApp Files e rivedi esempi di configurazione.

Opzioni di configurazione del backend

Trident utilizza la configurazione del backend (subnet, rete virtuale, livello di servizio e posizione) per creare volumi Azure NetApp Files su pool di capacità disponibili nella posizione richiesta e che corrispondono al livello di servizio e alla subnet richiesti.

I backend di Azure NetApp Files forniscono queste opzioni di configurazione.

Parametro	Descrizione	Predefinito
`version`	Versione della configurazione backend.	Sempre 1
`storageDriverName`	Nome del driver di archiviazione	"azure-netapp-files"
`backendName`	Nome personalizzato per il backend di storage	Driver name + "_" + caratteri casuali
`subscriptionID`	L'ID della sottoscrizione dalla tua sottoscrizione Azure. Facoltativo quando le managed identities sono abilitate su un cluster AKS.	
`tenantID`	L'ID tenant da una registrazione app. Facoltativo quando le identità gestite o l'identità cloud vengono utilizzate su un cluster AKS.	
`clientID`	L'ID client da una registrazione app. Facoltativo quando vengono utilizzate identità gestite o identità cloud su un cluster AKS.	
`clientSecret`	Il client secret da una App Registration è facoltativo quando vengono utilizzate managed identities o cloud identity su un cluster AKS.	
`serviceLevel`	Uno di `Standard`, `Premium` o `Ultra`	"" (casuale)
`location`	Nome della posizione di Azure in cui verranno creati i nuovi volumi Facoltativo quando le identità gestite sono abilitate su un cluster AKS.	
`cloudConfiguration.cloudName`	Identifica l'ambiente cloud di Azure per il backend. Imposta su `AzureGovernment` quando utilizzi Azure Government.	
`resourceGroups`	Elenco dei gruppi di risorse per filtrare le risorse scoperte	"[]" (nessun filtro)
`netappAccounts`	Elenco degli account NetApp per filtrare le risorse scoperte	"[]" (nessun filtro)

Parametro	Descrizione	Predefinito
`capacityPools`	Elenco dei pool di capacità per filtrare le risorse scoperte	"[]" (nessun filtro, casuale)
`virtualNetwork`	Nome di una rete virtuale con una subnet delegata	""
`subnet`	Nome di una subnet delegata a `Microsoft.Netapp/volumes`	""
`networkFeatures`	Insieme di funzionalità VNet per un volume, può essere `Basic` o `Standard`. Network Features non è disponibile in tutte le regioni e potrebbe dover essere abilitato in un abbonamento. Specificare `networkFeatures` quando la funzionalità non è abilitata causa il fallimento del provisioning del volume.	""
`nfsMountOptions`	Controllo dettagliato delle opzioni di montaggio NFS. Ignorato per i volumi SMB. Per montare volumi utilizzando NFS versione 4.1, includere `nfsvers=4` nell'elenco delle opzioni di montaggio separate da virgole per scegliere NFS v4.1. Le opzioni di montaggio impostate in una definizione di storage class sovrascrivono le opzioni di montaggio impostate nella configurazione del backend.	"nfsvers=3"
`limitVolumeSize`	Non eseguire il provisioning se la dimensione del volume richiesto è superiore a questo valore	"" (non applicato per impostazione predefinita)
`debugTraceFlags`	Flag di debug da usare per la risoluzione dei problemi. Esempio, `\{"api": false, "method": true, "discovery": true}`. Non usare questo a meno che non si stia eseguendo una risoluzione dei problemi e sia necessario un dump dettagliato del registro.	null
`nasType`	Configura la creazione di volumi NFS o SMB. Le opzioni sono `nfs`, `smb` o null. Impostando su null, vengono creati di default volumi NFS.	`nfs`

Parametro	Descrizione	Predefinito
`supportedTopologies`	Rappresenta un elenco di regioni e zone supportate da questo backend. Per ulteriori informazioni, consulta "Usa la topologia CSI".	
`qosType`	Rappresenta il tipo di QoS: Auto o Manual.	Auto
`maxThroughput`	Imposta il throughput massimo consentito in MiB/sec. Supportato solo per i pool di capacità QoS manuali.	`4 MiB/sec`

Per ulteriori informazioni sulle Network Features, consultare "Configura le funzionalità di rete per un volume Azure NetApp Files".

Considera gli ambienti cloud di Azure (26.06)

A partire dalla versione 26.06, Trident supporta la creazione e la gestione di backend Azure NetApp Files in più ambienti cloud di Azure.

I cloud Azure supportati includono:

- Azure Commercial
- Azure Government (Azure Government / MAG)
- Azure Cina

Quando distribuisci Trident o crei un backend di Azure NetApp Files, assicurati che gli endpoint di Azure Resource Manager e di autenticazione corrispondano al tuo ambiente cloud di Azure. Se gli endpoint non corrispondono, `tridentctl` non può autenticare e la creazione del backend fallisce.

Per un backend di Azure Government, imposta `cloudConfiguration.cloudName` nella configurazione del backend:

```
cloudConfiguration:
  cloudName: AzureGovernment
```

Autorizzazioni e risorse necessarie

Se durante la creazione di un PVC viene visualizzato l'errore "Nessun pool di capacità trovato", è probabile che la registrazione dell'app non disponga delle autorizzazioni e delle risorse necessarie (subnet, rete virtuale, pool di capacità) associate. Se il debug è abilitato, Trident registra le risorse di Azure rilevate durante la creazione del backend. Verificare che venga utilizzato un ruolo appropriato.

I valori per `resourceGroups`, `netappAccounts`, `capacityPools`, `virtualNetwork` e `subnet` possono essere specificati usando nomi brevi o completamente qualificati. I nomi completamente qualificati sono consigliati nella maggior parte delle situazioni perché i nomi brevi possono corrispondere a più risorse con lo stesso nome.

Se la vNet si trova in un gruppo di risorse diverso dall'account di archiviazione Azure NetApp Files (ANF), specificare il gruppo di risorse per la rete virtuale durante la configurazione dell'elenco resourceGroups per il backend.

I valori `resourceGroups`, `netappAccounts` e `capacityPools` sono filtri che limitano l'insieme delle risorse scoperte a quelle disponibili per questo storage backend e possono essere specificati in qualsiasi combinazione. I nomi completamente qualificati seguono questo formato:

Tipo	Formato
Gruppo di risorse	<resource group>
Account NetApp	<resource group>/<netapp account>
Pool di capacità	<resource group>/<netapp account>/<capacity pool>
Rete virtuale	<resource group>/<virtual network>
Sottorete	<resource group>/<virtual network>/<subnet>

Provisioning dei volumi

È possibile controllare il provisioning predefinito dei volumi specificando le seguenti opzioni in una sezione speciale del file di configurazione. Per i dettagli, fai riferimento a Esempi di configurazione.

Parametro	Descrizione	Predefinito
`exportRule`	Regole di esportazione per i nuovi volumi. `exportRule` deve essere un elenco separato da virgole di qualsiasi combinazione di indirizzi IPv4 o sottoreti IPv4 in notazione CIDR. Ignorato per i volumi SMB.	"0.0.0.0/0"
`snapshotDir`	Accesso alla `.snapshot` directory	`true`, `false` (Impostato esplicitamente).
`size`	La dimensione predefinita dei nuovi volumi	"100G"
`unixPermissions`	I permessi unix dei nuovi volumi (4 cifre ottali). Ignorato per i volumi SMB.	"" (funzione in anteprima, richiede whitelisting nell'abbonamento)

Esempi di configurazione

Gli esempi seguenti mostrano configurazioni di base che lasciano la maggior parte dei parametri ai valori predefiniti. Questo è il modo più semplice per definire un backend.

▼ **Configurazione minima**

Questa è la configurazione minima assoluta del backend. Con questa configurazione, Trident rileva tutti i tuoi account NetApp, pool di capacità e subnet delegate ad Azure NetApp Files nella posizione configurata e posiziona i nuovi volumi su uno di questi pool e subnet in modo casuale. Poiché `nasType` è omesso, `nfs` si applica l'impostazione predefinita e il backend esegue il provisioning dei volumi NFS.

Questa configurazione è ideale quando si sta iniziando a usare Azure NetApp Files e a fare delle prove, ma in pratica si vorrà fornire un ambito aggiuntivo per i volumi che si forniscono.

```
---
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-anf-1
  namespace: trident
spec:
  version: 1
  storageDriverName: azure-netapp-files
  subscriptionID: 9f87c765-4774-fake-ae98-a721add45451
  tenantID: 68e4f836-edc1-fake-bff9-b2d865ee56cf
  clientID: dd043f63-bf8e-fake-8076-8de91e5713aa
  clientSecret: SECRET
  location: eastus
```

▼ **Identità gestite per AKS**

Questa configurazione del backend omette `subscriptionID`, `tenantID`, `clientID` e `clientSecret`, che sono opzionali quando si usano le identità gestite.

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-anf-1
  namespace: trident
spec:
  version: 1
  storageDriverName: azure-netapp-files
  capacityPools:
    - resource-group-1/netapp-account-1/ultra-pool
  resourceGroups:
    - resource-group-1
  netappAccounts:
    - resource-group-1/netapp-account-1
  virtualNetwork: resource-group-1/eastus-prod-vnet
  subnet: resource-group-1/eastus-prod-vnet/eastus-anf-subnet
```

▼ **Identità cloud per AKS**

Questa configurazione del backend omette `tenantID`, `clientID` e `clientSecret`, che sono opzionali quando si utilizza un'identità cloud.

```
apiVersion: trident.netapp.io/v1
```

```
kind: TridentBackendConfig
metadata:
  name: backend-tbc-anf-1
  namespace: trident
spec:
  version: 1
  storageDriverName: azure-netapp-files
  capacityPools:
    - ultra-pool
  resourceGroups:
    - aks-ami-eastus-rg
  netappAccounts:
    - smb-na
  virtualNetwork: eastus-prod-vnet
  subnet: eastus-anf-subnet
  location: eastus
  subscriptionID: 9f87c765-4774-fake-ae98-a721add45451
```

▼ **Configurazione specifica del livello di servizio con filtri del capacity pool**

Questa configurazione di backend colloca i volumi nella posizione di Azure `eastus` in un `Ultra` pool di capacità. Trident scopre automaticamente tutte le sottoreti delegate ad Azure NetApp Files in quella posizione e colloca un nuovo volume su una di esse in modo casuale.

```
---
version: 1
storageDriverName: azure-netapp-files
subscriptionID: 9f87c765-4774-fake-ae98-a721add45451
tenantID: 68e4f836-edc1-fake-bff9-b2d865ee56cf
clientID: dd043f63-bf8e-fake-8076-8de91e5713aa
clientSecret: SECRET
location: eastus
serviceLevel: Ultra
capacityPools:
  - application-group-1/account-1/ultra-1
  - application-group-1/account-1/ultra-2
```

▼ **Esempio di backend con pool di capacità QoS manuali**

Questa configurazione di backend colloca i volumi nella posizione di Azure `eastus` con pool di capacità QoS manuali.

```
---
version: 1
storageDriverName: azure-netapp-files
backendName: anf1
location: eastus
```

```
labels:
  clusterName: test-cluster-1
  cloud: anf
  nasType: nfs
defaults:
  qosType: Manual
storage:
  - serviceLevel: Ultra
    labels:
      performance: gold
    defaults:
      maxThroughput: 10
  - serviceLevel: Premium
    labels:
      performance: silver
    defaults:
      maxThroughput: 5
  - serviceLevel: Standard
    labels:
      performance: bronze
    defaults:
      maxThroughput: 3
```

▼ **Configurazione avanzata**

Questa configurazione del backend riduce ulteriormente la portata del posizionamento dei volumi a una singola subnet e modifica anche alcune impostazioni predefinite del provisioning dei volumi.

```
---
version: 1
storageDriverName: azure-netapp-files
subscriptionID: 9f87c765-4774-fake-ae98-a721add45451
tenantID: 68e4f836-edc1-fake-bff9-b2d865ee56cf
clientID: dd043f63-bf8e-fake-8076-8de91e5713aa
clientSecret: SECRET
location: eastus
serviceLevel: Ultra
capacityPools:
  - application-group-1/account-1/ultra-1
  - application-group-1/account-1/ultra-2
virtualNetwork: application-group-1/eastus-prod-vnet
subnet: application-group-1/eastus-prod-vnet/my-subnet
networkFeatures: Standard
nfsMountOptions: vers=3,proto=tcp,timeo=600
limitVolumeSize: 500Gi
defaults:
```

```
  exportRule: 10.0.0.0/24,10.0.1.0/24,10.0.2.100
  snapshotDir: "true"
  size: 200Gi
  unixPermissions: "0777"
```

▼ **Configurazione del pool virtuale**

Questa configurazione di backend definisce più pool di storage in un unico file. Questo è utile quando si dispone di più pool di capacità che supportano diversi livelli di servizio e si desidera creare classi di storage in Kubernetes che li rappresentano. Le etichette dei pool virtuali sono state utilizzate per differenziare i pool in base a `performance`.

```
---
version: 1
storageDriverName: azure-netapp-files
subscriptionID: 9f87c765-4774-fake-ae98-a721add45451
tenantID: 68e4f836-edc1-fake-bff9-b2d865ee56cf
clientID: dd043f63-bf8e-fake-8076-8de91e5713aa
clientSecret: SECRET
location: eastus
resourceGroups:
  - application-group-1
networkFeatures: Basic
nfsMountOptions: vers=3,proto=tcp,timeo=600
labels:
  cloud: azure
storage:
  - labels:
      performance: gold
    serviceLevel: Ultra
    capacityPools:
      - application-group-1/netapp-account-1/ultra-1
      - application-group-1/netapp-account-1/ultra-2
    networkFeatures: Standard
  - labels:
      performance: silver
    serviceLevel: Premium
    capacityPools:
      - application-group-1/netapp-account-1/premium-1
  - labels:
      performance: bronze
    serviceLevel: Standard
    capacityPools:
      - application-group-1/netapp-account-1/standard-1
      - application-group-1/netapp-account-1/standard-2
```

▼ **Configurazione delle topologie supportate**

Trident facilita il provisioning dei volumi per i carichi di lavoro in base alle regioni e alle zone di disponibilità. Il `supportedTopologies` blocco in questa configurazione di backend viene utilizzato per fornire un elenco di regioni e zone per backend. I valori di regione e zona specificati qui devono corrispondere ai valori di regione e zona delle etichette su ciascun nodo del cluster Kubernetes. Queste regioni e zone rappresentano l'elenco dei valori consentiti che possono essere forniti in una classe di storage. Per le classi di storage che contengono un sottoinsieme delle regioni e delle zone fornite in un backend, Trident crea i volumi nella regione e nella zona menzionate. Per ulteriori informazioni, consulta "Usa la topologia CSI".

```
---
version: 1
storageDriverName: azure-netapp-files
subscriptionID: 9f87c765-4774-fake-ae98-a721add45451
tenantID: 68e4f836-edc1-fake-bff9-b2d865ee56cf
clientID: dd043f63-bf8e-fake-8076-8de91e5713aa
clientSecret: SECRET
location: eastus
serviceLevel: Ultra
capacityPools:
  - application-group-1/account-1/ultra-1
  - application-group-1/account-1/ultra-2
supportedTopologies:
  - topology.kubernetes.io/region: eastus
    topology.kubernetes.io/zone: eastus-1
  - topology.kubernetes.io/region: eastus
    topology.kubernetes.io/zone: eastus-2
```

Definizioni delle classi di storage

Le seguenti `StorageClass` definizioni si riferiscono ai pool di archiviazione sopra.

Esempi di definizioni che utilizzano `parameter.selector` campo

Utilizzando `parameter.selector` puoi specificare per ogni `StorageClass` il pool virtuale che viene utilizzato per ospitare un volume. Il volume avrà gli aspetti definiti nel pool scelto.

```
---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: gold
provisioner: csi.trident.netapp.io
parameters:
  selector: performance=gold
allowVolumeExpansion: true
```

```
---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: silver
provisioner: csi.trident.netapp.io
parameters:
  selector: performance=silver
allowVolumeExpansion: true

---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: bronze
provisioner: csi.trident.netapp.io
parameters:
  selector: performance=bronze
allowVolumeExpansion: true
```

Esempi di definizioni per volumi SMB

Utilizzando `nasType`, `node-stage-secret-name`e `node-stage-secret-namespace`, è possibile specificare un volume SMB e fornire le credenziali Active Directory richieste.

▼ **Configurazione di base sul namespace predefinito**

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: anf-sc-smb
provisioner: csi.trident.netapp.io
parameters:
  backendType: "azure-netapp-files"
  trident.netapp.io/nasType: "smb"
  csi.storage.k8s.io/node-stage-secret-name: "smbcreds"
  csi.storage.k8s.io/node-stage-secret-namespace: "default"
```

▼ **Utilizzo di segreti diversi per ogni namespace**

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: anf-sc-smb
provisioner: csi.trident.netapp.io
parameters:
```

```
  backendType: "azure-netapp-files"
  trident.netapp.io/nasType: "smb"
  csi.storage.k8s.io/node-stage-secret-name: "smbcreds"
  csi.storage.k8s.io/node-stage-secret-namespace: ${pvc.namespace}
```

▼ **Utilizzo di segreti diversi per volume**

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: anf-sc-smb
provisioner: csi.trident.netapp.io
parameters:
  backendType: "azure-netapp-files"
  trident.netapp.io/nasType: "smb"
  csi.storage.k8s.io/node-stage-secret-name: ${pvc.name}
  csi.storage.k8s.io/node-stage-secret-namespace: ${pvc.namespace}
```

`nasType: smb` filtri per i pool che supportano i volumi SMB.
`nasType: nfs` o `nasType: null` filtri per i pool NFS.

Crea il backend

Dopo aver creato il file di configurazione del backend, eseguire il seguente comando:

```
tridentctl create backend -f <backend-file>
```

Se utilizzi un cloud Azure non commerciale, assicurati che `tridentctl` sia configurato per utilizzare Azure Resource Manager e gli endpoint di autenticazione per il tuo ambiente cloud Azure. Se la creazione del backend non riesce, controlla la configurazione del backend e visualizza i log per determinarne la causa:

```
tridentctl logs
```

Dopo aver identificato e corretto il problema con il file di configurazione, è possibile eseguire nuovamente il comando create.

Google Cloud NetApp Volumes

Configurare Google Cloud NetApp Volumes

È possibile configurare Google Cloud NetApp Volumes come backend per Trident al fine di predisporre lo storage per i carichi di lavoro di Kubernetes.

Panoramica

Trident supporta Google Cloud NetApp Volumes sia per carichi di lavoro NAS (NFS e SMB) che a blocchi (iSCSI).

- I carichi di lavoro NAS utilizzano il `google-cloud-netapp-volumes` backend
- I carichi di lavoro a blocchi (iSCSI) utilizzano il `google-cloud-netapp-volumes-san` backend

I volumi NAS offrono storage basato su file e vi si accede tramite i protocolli NFS o SMB. Questi volumi supportano l'accesso condiviso tra più pod o nodi.

I volumi a blocchi forniscono storage a blocchi raw e sono accessibili come dispositivi iSCSI collegati ai nodi Kubernetes. Questi volumi vengono utilizzati quando le applicazioni richiedono l'accesso a livello di blocco.

Ciò si applica ai seguenti ambienti:

- Trident 26.02 e versioni successive
- Google Kubernetes Engine (GKE) o Red Hat OpenShift
- Google Cloud NetApp Volumes pool di storage

Per configurare lo storage a blocchi (iSCSI), vedere "Configurare storage a blocchi (iSCSI)".

Preparati alla configurazione

L'identità cloud consente ai carichi di lavoro Kubernetes di accedere alle risorse di Google Cloud autenticandosi come identità del carico di lavoro anziché utilizzare credenziali statiche.

Per utilizzare cloud identity con Google Cloud NetApp Volumes, è necessario disporre di:

- Un cluster Kubernetes distribuito utilizzando Google Kubernetes Engine (GKE)
- Identità del carico di lavoro abilitata sul cluster GKE e server dei metadati abilitato sui pool di nodi
- Un account di servizio Google Cloud con il ruolo di amministratore di Google Cloud NetApp Volumes (`roles/netapp.admin`) o un ruolo personalizzato equivalente
- Trident installato con il cloud provider impostato su `GCP` e l'annotazione dell'identità cloud configurata

Operatore Trident

Per installare Trident utilizzando l'operatore Trident, modificare `tridentorchestrator_cr.yaml`:

```
apiVersion: trident.netapp.io/v1
kind: TridentOrchestrator
metadata:
  name: trident
spec:
  namespace: trident
  cloudProvider: "GCP"
  cloudIdentity: "iam.gke.io/gcp-service-account: cloudvolumes-admin-
sa@mygcpproject.iam.gserviceaccount.com"
```

Helm

Imposta il cloud provider e l'identità cloud durante l'installazione di Trident con Helm:

```
helm install trident trident-operator-100.6.0.tgz \
  --set cloudProvider=GCP \
  --set cloudIdentity="iam.gke.io/gcp-service-account: cloudvolumes-
admin-sa@mygcpproject.iam.gserviceaccount.com"
```

tridentctl

Installa Trident specificando il cloud provider e l'identità cloud:

```
tridentctl install \
  --cloud-provider=GCP \
  --cloud-identity="iam.gke.io/gcp-service-account: cloudvolumes-admin-
sa@mygcpproject.iam.gserviceaccount.com" \
  -n trident
```

Configurare l'archiviazione NAS

Per i pool di storage UNIFIED di Google Cloud NetApp Volumes, Trident applica regole di denominazione e convalida specifiche di UNIFIED durante le operazioni sui volumi.

Durante la ricerca di un volume, Trident può valutare diverse varianti compatibili del nome del volume (ad esempio, formati con trattino e underscore) per migliorare l'affidabilità dell'importazione e del rilevamento.

Dettagli del driver

Trident fornisce il `google-cloud-netapp-volumes` driver per il provisioning dello storage NAS da Google Cloud NetApp Volumes.

Il driver supporta le seguenti modalità di accesso:

- ReadWriteOnce (RWO)
- ReadOnlyMany (ROX)
- ReadWriteMany (RWX)
- ReadWriteOncePod (RWOP)

Driver	Protocollo	volumeMode	Modalità di accesso supportate	File system supportati
`google-cloud-netapp-volumes`	NFS SMB	Filesystem	RWO, ROX, RWX, RWOP	`nfs`, `smb`

Configurare un backend NAS Trident

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: gcnv-nas
  namespace: trident
spec:
  version: 1
  storageDriverName: google-cloud-netapp-volumes
  projectNumber: "<project-number>"
  location: "<region>"
  sdkTimeout: "600"
  storage:
  - labels:
      cloud: gcp
    network: "<vpc-network>"
```

Effettuare il provisioning dei volumi NAS

I volumi NAS vengono forniti tramite il backend `google-cloud-netapp-volumes` e supportano i protocolli NFS e SMB.

StorageClass for volumi NFS

Per eseguire il provisioning dei volumi NFS, impostare `nasType` su `nfs`.

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: gcnv-nfs
provisioner: csi.trident.netapp.io
parameters:
  backendType: "google-cloud-netapp-volumes"
  trident.netapp.io/nasType: "nfs"
allowVolumeExpansion: true
```

StorageClass for volumi SMB

Per effettuare il provisioning dei volumi SMB, impostare `nasType` su `smb` e fornire le credenziali.

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: gcnv-smb
provisioner: csi.trident.netapp.io
```

```
parameters:
  backendType: "google-cloud-netapp-volumes"
  trident.netapp.io/nasType: "smb"
  csi.storage.k8s.io/node-stage-secret-name: "smbcreds"
  csi.storage.k8s.io/node-stage-secret-namespace: "default"
allowVolumeExpansion: true
```

Esempio di PersistentVolumeClaim (RWX)

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: gcnv-nas-rwx
spec:
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 100Gi
  storageClassName: gcnv-nfs
```

Esempio di PersistentVolumeClaim (RWO)

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: gcnv-nas-rwo
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 100Gi
  storageClassName: gcnv-nfs
```

I volumi NAS utilizzano `volumeMode: Filesystem`.

Configura Google Cloud NetApp Volumes per carichi di lavoro SAN

È possibile configurare Trident per eseguire il provisioning di volumi di storage a blocchi utilizzando il protocollo iSCSI da Google Cloud NetApp Volumes. I volumi SAN vengono creati dai pool di storage Flex Unified utilizzando il `google-cloud-netapp-volumes-san` driver di storage.

Questo driver è dedicato ai carichi di lavoro a blocchi e non supporta i protocolli NAS.

Il `google-cloud-netapp-volumes-san` backend è necessario per il provisioning di volumi a blocchi iSCSI. Il `google-cloud-netapp-volumes` backend supporta solo protocolli NAS e non può essere utilizzato per carichi di lavoro SAN.

Panoramica

Trident supporta i carichi di lavoro SAN (iSCSI) di Google Cloud NetApp Volumes utilizzando il driver `google-cloud-netapp-volumes-san`.

I volumi SAN vengono forniti dai pool di storage Flex Unified e presentati ai nodi Kubernetes come dispositivi a blocchi iSCSI.

Ciò si applica ai seguenti ambienti:

- Trident 26.02 e versioni successive
- Google Kubernetes Engine (GKE) o Red Hat OpenShift
- Google Cloud NetApp Volumes Flex pool di storage unificati
- Carichi di lavoro basati su iSCSI

Pool di storage unificati Flex

I pool di storage Flex Unified forniscono storage a blocchi utilizzando il protocollo iSCSI e sono necessari per il provisioning SAN:

- Sono supportati i pool regionali Flex Unified.
- I pool Flex Unified ZONAL sono supportati a partire da Trident 26.02.1.
- Per i carichi di lavoro SAN è supportato esclusivamente il livello di servizio **Flex**.

Configurare un backend SAN Trident

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: gcnv-san
  namespace: trident
spec:
  version: 1
  storageDriverName: google-cloud-netapp-volumes-san
  projectNumber: "<project-number>"
  location: "<region>"
  sdkTimeout: "600"
  storage:
  - labels:
      cloud: gcp
      performance: flex
```

```
    network: "<vpc-network>"
    serviceLevel: Flex
```

Crea un StorageClass

Dopo aver configurato il backend SAN, crea una StorageClass che fa riferimento al driver `google-cloud-netapp-volumes-san`.

Il tipo di file system è definito nella StorageClass, non nel backend.

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: gcnv-san
provisioner: csi.trident.netapp.io
parameters:
  backendType: "google-cloud-netapp-volumes-san"
  fsType: "ext4"
allowVolumeExpansion: true
```

Tipi di filesystem supportati:

- `ext4` (predefinito)
- `ext3`
- `xfs`

> Il driver SAN supporta solo il livello di servizio Flex e non utilizza parametri backend specifici del NAS come `exportRule`, `unixPermissions`, `nasType`, `snapshotDir`, `nfsMountOptions` o impostazioni relative al tiering.

Effettuare il provisioning dei volumi a blocchi

ReadWriteOnce (RWO)

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: gcnv-san-rwo
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 100Gi
```

```
  storageClassName: gcnv-san
```

ReadWriteOncePod (RWOP)

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: gcnv-san-rwop
spec:
  accessModes:
    - ReadWriteOncePod
  resources:
    requests:
      storage: 100Gi
  storageClassName: gcnv-san
```

ReadOnlyMany (ROX)

Un modello comune per ROX è clonare un volume ReadWriteOnce esistente e montare il clone come di sola lettura.

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: gcnv-san-rox
spec:
  accessModes:
    - ReadOnlyMany
  resources:
    requests:
      storage: 100Gi
  storageClassName: gcnv-san
  dataSource:
    kind: PersistentVolumeClaim
    name: gcnv-san-rwo
```

ReadWriteMany (RWX) — solo raw block

ReadWriteMany è supportato solo quando `volumeMode: Block`.

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: gcnv-san-raw-rwx
```

```
spec:
  accessModes:
    - ReadWriteMany
  volumeMode: Block
  resources:
    requests:
      storage: 100Gi
  storageClassName: gcnv-san
```

Comportamento del volume a blocchi

I volumi a blocchi vengono forniti come LUN iSCSI e presentati ai nodi Kubernetes come dispositivi a blocchi.

Volumi a blocchi:

- Utilizzare il protocollo iSCSI
- Supporta la presentazione sia del file system che dei blocchi raw
- Sono allegati e gestiti da Trident
- Supporta più modalità di accesso a Kubernetes

Modalità di accesso

I volumi a blocchi forniti da Trident supportano le seguenti modalità di accesso:

- `ReadWriteOnce` (RWO)
- `ReadOnlyMany` (ROX)
- `ReadWriteOncePod` (RWOP)
- `ReadWriteMany` (RWX), supportato solo quando `volumeMode: Block`

Comportamento di volumeMode

Il `volumeMode` campo controlla come viene esposto un volume di blocco:

- `Filesystem` Trident formatta e monta il volume.
- `Block` Trident collega il dispositivo e lo espone come raw block device.

Operazioni supportate

Volumi a blocchi forniti utilizzando il `google-cloud-netapp-volumes-san` driver support:

- Crea
- Elimina
- Clona
- Snapshot
- Ridimensiona
- Importare

Comportamento di overprovisioning extra GiB

I volumi a blocchi di Google Cloud NetApp Volumes includono un overhead interno di metadati. Questo overhead riduce le dimensioni del dispositivo visibile al kernel rispetto alla capacità fornita.

I test dimostrano:

- Circa 300 KiB di overhead sulla creazione iniziale
- Fino a circa 107 MiB di overhead dopo un ridimensionamento

Poiché Google Cloud NetApp Volumes accetta solo allocazioni di interi GiB, Trident garantisce che la dimensione utilizzabile del dispositivo soddisfi o superi sempre la richiesta PVC:

- Arrotondamento della dimensione richiesta al GiB intero successivo
- Aggiunta di un buffer aggiuntivo di 1 GiB

Esempio:

- Richiesta PVC: 100 GiB
- Dimensioni fornite in Google Cloud NetApp Volumes: 101 GiB
- Spazio utilizzabile visibile all'applicazione: almeno 100 GiB

Esempi di pod

Volume a blocchi montato sul filesystem (RWO)

```
apiVersion: v1
kind: Pod
metadata:
  name: app-rwo
spec:
  containers:
  - name: app
    image: ubuntu:22.04
    command: ["sleep", "infinity"]
    volumeMounts:
    - name: data
      mountPath: /mnt/data
  volumes:
  - name: data
    persistentVolumeClaim:
      claimName: gcnv-san-rwo
```

Dispositivo a blocchi raw (RWX)

```
apiVersion: v1
kind: Pod
```

```
metadata:
  name: app-raw-rwx
spec:
  containers:
  - name: app
    image: ubuntu:22.04
    command: ["sleep", "infinity"]
    volumeDevices:
    - name: data
      devicePath: /dev/xda
  volumes:
  - name: data
    persistentVolumeClaim:
      claimName: gcnv-san-raw-rwx
```

Comportamento di attach e mount

Per i volumi SAN forniti da Google Cloud NetApp Volumes:

- Trident crea un Logical Unit Number (LUN) in un pool di storage Flex Unified.
- Durante la pubblicazione, Trident mappa la LUN su un gruppo host per nodo.
- Durante la messa in scena del nodo, Trident:
  - Accede all'iSCSI target
  - Scopre il LUN
  - Configura multipath
- Se `volumeMode: Filesystem`, Trident formatta il dispositivo se necessario e lo monta.
- Se `volumeMode: Block`, Trident collega il dispositivo e lo espone direttamente al pod senza formattarlo o montarlo.

> I volumi a blocchi SAN non forniscono blocco distribuito o coordinamento delle scritture. Quando un volume a blocchi è accessibile da più nodi (ReadWriteMany con `volumeMode: Block`), l'applicazione o il file system devono gestire la concorrenza.

Prepararsi a configurare un backend Google Cloud NetApp Volumes

Prima di poter configurare il backend Google Cloud NetApp Volumes, è necessario assicurarsi che siano soddisfatti i seguenti requisiti.

Prerequisiti per i volumi NFS o SMB

Se utilizzi Google Cloud NetApp Volumes per la prima volta o in una nuova posizione, è necessaria una configurazione iniziale per impostare Google Cloud NetApp Volumes e creare un volume NFS o SMB. Fai riferimento a "Prima di iniziare".

Assicurarsi di disporre di quanto segue prima di configurare il backend Google Cloud NetApp Volumes:

- Un account Google Cloud configurato con il servizio Google Cloud NetApp Volumes. Fai riferimento a

"Google Cloud NetApp Volumes".

- Numero di progetto del tuo account Google Cloud. Fai riferimento a "Identificazione dei progetti".
- Un account del servizio Google Cloud con il ruolo NetApp Volumes Admin (`roles/netapp.admin`). Fai riferimento a "Ruoli e autorizzazioni di Identity and Access Management".
- File chiave API per il tuo account GCNV. Fai riferimento a "Crea una chiave dell'account di servizio"
- Un pool di storage. Fai riferimento a "Panoramica dei pool di storage".

Per ulteriori informazioni su come configurare l'accesso a Google Cloud NetApp Volumes, fare riferimento a "Configura l'accesso a Google Cloud NetApp Volumes".

Opzioni di configurazione del backend di Google Cloud NetApp Volumes ed esempi

Scoprite le opzioni di configurazione del backend per Google Cloud NetApp Volumes e consultate gli esempi di configurazione.

Opzioni di configurazione del backend

Ogni backend fornisce volumi in una singola regione Google Cloud. Per creare volumi in altre regioni, puoi definire backend aggiuntivi.

Parametro	Descrizione	Predefinito
`version`		Sempre 1
`storageDriverName`	Nome del driver di archiviazione	Il valore di `storageDriverName` deve essere specificato come "google-cloud-netapp-volumes".
`backendName`	(Facoltativo) Nome personalizzato dello storage backend	Nome driver + "_" + parte della chiave API
`storagePools`	Parametro opzionale usato per specificare i pool di storage per la creazione dei volumi.	
`projectNumber`	Numero di progetto dell'account Google Cloud. Il valore si trova nella home page del portale Google Cloud.	
`location`	La posizione di Google Cloud in cui Trident crea i volumi GCNV. Quando si creano cluster Kubernetes cross-region, i volumi creati in un `location` possono essere utilizzati nei carichi di lavoro pianificati sui nodi di più regioni di Google Cloud. Il traffico cross-region comporta un costo aggiuntivo.	

Parametro	Descrizione	Predefinito
`apiKey`	Chiave API per l'account del servizio Google Cloud con il `netapp.admin` ruolo. Include il contenuto in formato JSON del file della chiave privata di un account del servizio Google Cloud (copiato testualmente nel file di configurazione del backend). Il `apiKey` deve includere coppie chiave-valore per le seguenti chiavi: `type`, `project_id`, `client_email`, `client_id`, `auth_uri`, `token_uri`, `auth_provider_x509_cert_url` e `client_x509_cert_url`.	
`nfsMountOptions`	Controllo dettagliato delle opzioni di montaggio NFS.	"nfsvers=3"
`limitVolumeSize`	Il provisioning fallisce se la dimensione del volume richiesta è superiore a questo valore.	"" (non applicato per impostazione predefinita)
`serviceLevel`	Il livello di servizio di un pool di storage e dei suoi volumi. I valori sono `flex`, `standard`, `premium`, o `extreme`.	
`labels`	Set di etichette arbitrarie in formato JSON da applicare ai volumi	""
`network`	Rete Google Cloud utilizzata per i volumi GCNV.	
`debugTraceFlags`	Flag di debug da usare per la risoluzione dei problemi. Esempio, `{"api":false, "method":true}`. Non usare questo a meno che non si stia eseguendo una risoluzione dei problemi e sia necessario un dump dettagliato del registro.	null
`nasType`	Configura la creazione di volumi NFS o SMB. Le opzioni sono `nfs`, `smb` o null. Impostando su null, vengono creati di default volumi NFS.	`nfs`
`supportedTopologies`	Rappresenta un elenco di regioni e zone supportate da questo backend. Per ulteriori informazioni, consulta "Usa la topologia CSI". Ad esempio: `supportedTopologies: - topology.kubernetes.io/region: asia-east1 topology.kubernetes.io/zone: asia-east1-a`	

Opzioni di provisioning del volume

È possibile controllare il provisioning predefinito dei volumi nella sezione `defaults` del file di configurazione.

Parametro	Descrizione	Predefinito
`exportRule`	Le regole di esportazione per i nuovi volumi. Deve essere un elenco separato da virgole di qualsiasi combinazione di indirizzi IPv4.	"0.0.0.0/0"

Parametro	Descrizione	Predefinito
`snapshotDir`	Accesso alla `.snapshot` directory	`true`, `false` (Il comportamento predefinito potrebbe variare. Impostare esplicitamente) "false" per NFSv3
`snapshotReserve`	Percentuale di volume riservata alle snapshot	"" (accetta il valore predefinito di 0)
`unixPermissions`	I permessi unix dei nuovi volumi (4 cifre ottali).	""

Esempi di configurazione

Gli esempi seguenti mostrano configurazioni di base che lasciano la maggior parte dei parametri ai valori predefiniti. Questo è il modo più semplice per definire un backend.

▼ **Configurazione minima**

Questa è la configurazione minima assoluta del backend. Con questa configurazione, Trident scopre tutti i pool di storage delegati a Google Cloud NetApp Volumes nella posizione configurata e posiziona i nuovi volumi su uno di questi pool in modo casuale. Poiché `nasType` è omesso, `nfs` si applica l'impostazione predefinita e il backend esegue il provisioning dei volumi NFS.

Questa configurazione è ideale quando si inizia a utilizzare Google Cloud NetApp Volumes e si stanno facendo delle prove, ma in pratica potrebbe essere necessario fornire un ambito aggiuntivo per i volumi di cui si effettua il provisioning.

Sostituisci `<id_value>` e `<key_value>` con le credenziali del tuo account di servizio.

```
---
apiVersion: v1
kind: Secret
metadata:
  name: backend-tbc-gcnv-secret
type: Opaque
stringData:
  private_key_id: "<id_value>"
  private_key: |
    -----BEGIN PRIVATE KEY-----
    <key_value>
    -----END PRIVATE KEY-----
---
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-gcnv
spec:
  version: 1
  storageDriverName: google-cloud-netapp-volumes
```

```
  projectNumber: "123455380079"
  location: europe-west6
  serviceLevel: premium
  apiKey:
    type: service_account
    project_id: my-gcnv-project
    client_email: myproject-prod@my-gcnv-project.iam.gserviceaccount.com
    client_id: "103346282737811234567"
    auth_uri: https://accounts.google.com/o/oauth2/auth
    token_uri: https://oauth2.googleapis.com/token
    auth_provider_x509_cert_url:
https://www.googleapis.com/oauth2/v1/certs
    client_x509_cert_url:
https://www.googleapis.com/robot/v1/metadata/x509/myproject-prod%40my-
gcnv-project.iam.gserviceaccount.com
  credentials:
    name: backend-tbc-gcnv-secret
```

▼ **Configurazione per i volumi SMB**

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-gcnv1
  namespace: trident
spec:
  version: 1
  storageDriverName: google-cloud-netapp-volumes
  projectNumber: "123456789"
  location: asia-east1
  serviceLevel: flex
  nasType: smb
  apiKey:
    type: service_account
    project_id: cloud-native-data
    client_email: trident-sample@cloud-native-
data.iam.gserviceaccount.com
    client_id: "123456789737813416734"
    auth_uri: https://accounts.google.com/o/oauth2/auth
    token_uri: https://oauth2.googleapis.com/token
    auth_provider_x509_cert_url:
https://www.googleapis.com/oauth2/v1/certs
    client_x509_cert_url:
https://www.googleapis.com/robot/v1/metadata/x509/trident-
sample%40cloud-native-data.iam.gserviceaccount.com
  credentials:
```

```
    name: backend-tbc-gcnv-secret
```

▼ **Configurazione con filtro StoragePools**

```
---
apiVersion: v1
kind: Secret
metadata:
  name: backend-tbc-gcnv-secret
type: Opaque
stringData:
  private_key_id: "<id_value>"
  private_key: |
    -----BEGIN PRIVATE KEY-----
    <key_value>
    -----END PRIVATE KEY-----
---

---
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-gcnv
spec:
  version: 1
  storageDriverName: google-cloud-netapp-volumes
  projectNumber: "123455380079"
  location: europe-west6
  serviceLevel: premium
  storagePools:
    - premium-pool1-europe-west6
    - premium-pool2-europe-west6
  apiKey:
    type: service_account
    project_id: my-gcnv-project
    client_email: myproject-prod@my-gcnv-project.iam.gserviceaccount.com
    client_id: "103346282737811234567"
    auth_uri: https://accounts.google.com/o/oauth2/auth
    token_uri: https://oauth2.googleapis.com/token
    auth_provider_x509_cert_url:
https://www.googleapis.com/oauth2/v1/certs
    client_x509_cert_url:
https://www.googleapis.com/robot/v1/metadata/x509/myproject-prod%40my-
gcnv-project.iam.gserviceaccount.com
  credentials:
```

```
    name: backend-tbc-gcnv-secret
```

▼ **Configurazione del pool virtuale**

Questa configurazione di backend definisce più pool virtuali in un unico file. I pool virtuali sono definiti nella `storage` sezione. Sono utili quando si dispone di più pool di storage che supportano diversi livelli di servizio e si desidera creare classi di storage in Kubernetes che li rappresentano. Le etichette dei pool virtuali sono utilizzate per differenziare i pool. Ad esempio, nell'esempio seguente `performance` label e `serviceLevel` type sono utilizzati per differenziare i pool virtuali.

È anche possibile impostare alcuni valori predefiniti da applicare a tutti i virtual pool e sovrascrivere i valori predefiniti per i singoli virtual pool. Nell'esempio seguente, `snapshotReserve` e `exportRule` servono come valori predefiniti per tutti i virtual pool.

Per ulteriori informazioni, consulta "Pool virtuali".

```
---
apiVersion: v1
kind: Secret
metadata:
  name: backend-tbc-gcnv-secret
type: Opaque
stringData:
  private_key_id: "<id_value>"
  private_key: |
    -----BEGIN PRIVATE KEY-----
    <key_value>
    -----END PRIVATE KEY-----

---
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-gcnv
spec:
  version: 1
  storageDriverName: google-cloud-netapp-volumes
  projectNumber: "123455380079"
  location: europe-west6
  apiKey:
    type: service_account
    project_id: my-gcnv-project
    client_email: myproject-prod@my-gcnv-project.iam.gserviceaccount.com
    client_id: "103346282737811234567"
    auth_uri: https://accounts.google.com/o/oauth2/auth
    token_uri: https://oauth2.googleapis.com/token
    auth_provider_x509_cert_url:
https://www.googleapis.com/oauth2/v1/certs
```

```
    client_x509_cert_url:
https://www.googleapis.com/robot/v1/metadata/x509/myproject-prod%40my-
gcnv-project.iam.gserviceaccount.com
  credentials:
    name: backend-tbc-gcnv-secret
  defaults:
    snapshotReserve: "10"
    exportRule: 10.0.0.0/24
  storage:
    - labels:
        performance: extreme
      serviceLevel: extreme
      defaults:
        snapshotReserve: "5"
        exportRule: 0.0.0.0/0
    - labels:
        performance: premium
      serviceLevel: premium
    - labels:
        performance: standard
      serviceLevel: standard
```

▼ **Identità cloud per GKE**

Quando installi Trident con una cloud identity GKE, i backend si autenticano come workload identity e omettono il `apiKey` blocco. La cloud identity viene configurata allo stesso modo per entrambi i driver Google Cloud NetApp Volumes. L'unica differenza è il `storageDriverName` valore: usa `google-cloud-netapp-volumes` per i volumi NAS (NFS o SMB) e `google-cloud-netapp-volumes-san` per i volumi SAN (iSCSI block).

> Il supporto per l'identità cloud per il `google-cloud-netapp-volumes-san` driver richiede Trident 26.06 o versioni successive. I volumi SAN vengono forniti dai pool di storage Flex Unified, quindi imposta `serviceLevel` su `flex`.

Driver NAS (<code>google-cloud-netapp-volumes</code>)

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-gcp-gcnv
spec:
  version: 1
  storageDriverName: google-cloud-netapp-volumes
  projectNumber: '012345678901'
  network: gcnv-network
  location: us-west2
  serviceLevel: Premium
```

```
  storagePool: pool-premium1
```

Driver SAN (<code>google-cloud-netapp-volumes-san</code>)

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-gcp-gcnv-san
spec:
  version: 1
  storageDriverName: google-cloud-netapp-volumes-san
  projectNumber: '012345678901'
  network: gcnv-network
  location: us-west2
  serviceLevel: flex
  storagePool: pool-flex1
```

▼ **Configurazione delle topologie supportate**

Trident facilita il provisioning dei volumi per i carichi di lavoro in base alle regioni e alle zone di disponibilità. Il `supportedTopologies` blocco in questa configurazione di backend viene utilizzato per fornire un elenco di regioni e zone per backend. I valori di regione e zona specificati qui devono corrispondere ai valori di regione e zona delle etichette su ciascun nodo del cluster Kubernetes. Queste regioni e zone rappresentano l'elenco dei valori consentiti che possono essere forniti in una classe di storage. Per le classi di storage che contengono un sottoinsieme delle regioni e delle zone fornite in un backend, Trident crea i volumi nella regione e nella zona menzionate. Per ulteriori informazioni, consulta "Usa la topologia CSI".

```
---
version: 1
storageDriverName: google-cloud-netapp-volumes
subscriptionID: 9f87c765-4774-fake-ae98-a721add45451
tenantID: 68e4f836-edc1-fake-bff9-b2d865ee56cf
clientID: dd043f63-bf8e-fake-8076-8de91e5713aa
clientSecret: SECRET
location: asia-east1
serviceLevel: flex
supportedTopologies:
  - topology.kubernetes.io/region: asia-east1
    topology.kubernetes.io/zone: asia-east1-a
  - topology.kubernetes.io/region: asia-east1
    topology.kubernetes.io/zone: asia-east1-b
```

E ora?

Dopo aver creato il file di configurazione del backend, eseguire il seguente comando:

```
kubectl create -f <backend-file>
```

Per verificare che il backend sia stato creato correttamente, eseguire il seguente comando:

```
kubectl get tridentbackendconfig

NAME                BACKEND NAME        BACKEND UUID
PHASE   STATUS
backend-tbc-gcnv    backend-tbc-gcnv    b2fd1ff9-b234-477e-88fd-713913294f65
Bound   Success
```

Se la creazione del backend fallisce, c'è qualcosa di sbagliato nella configurazione del backend. Puoi descrivere il backend usando il `kubectl get tridentbackendconfig <backend-name>` comando o visualizzare i log per determinare la causa eseguendo il seguente comando:

```
tridentctl logs
```

Dopo aver identificato e corretto il problema con il file di configurazione, è possibile eliminare il backend ed eseguire nuovamente il comando create.

Definizioni delle classi di storage

Di seguito è riportata una definizione di base `StorageClass` che fa riferimento al backend sopra.

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: gcnv-nfs-sc
provisioner: csi.trident.netapp.io
parameters:
  backendType: "google-cloud-netapp-volumes"
```

Esempi di definizioni che utilizzano il `parameter.selector` campo:

Utilizzando `parameter.selector` puoi specificare per ogni `StorageClass` il "pool virtuale" che viene utilizzato per ospitare un volume. Il volume avrà gli aspetti definiti nel pool scelto.

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: extreme-sc
provisioner: csi.trident.netapp.io
parameters:
```

```
  selector: performance=extreme
  backendType: google-cloud-netapp-volumes

---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: premium-sc
provisioner: csi.trident.netapp.io
parameters:
  selector: performance=premium
  backendType: google-cloud-netapp-volumes

---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: standard-sc
provisioner: csi.trident.netapp.io
parameters:
  selector: performance=standard
  backendType: google-cloud-netapp-volumes
```

Per ulteriori dettagli sulle storage class, consultare "Creare una storage class".

Esempi di definizioni per volumi SMB

Utilizzando `nasType`, `node-stage-secret-name` e `node-stage-secret-namespace`, è possibile specificare un volume SMB e fornire le credenziali Active Directory richieste. Per il segreto dello stage del nodo è possibile utilizzare qualsiasi utente/password Active Directory con qualsiasi o nessuna autorizzazione.

▼ Configurazione di base sul namespace predefinito

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: gcnv-sc-smb
provisioner: csi.trident.netapp.io
parameters:
  backendType: "google-cloud-netapp-volumes"
  trident.netapp.io/nasType: "smb"
  csi.storage.k8s.io/node-stage-secret-name: "smbcreds"
  csi.storage.k8s.io/node-stage-secret-namespace: "default"
```

▼ Utilizzo di segreti diversi per ogni namespace

```
apiVersion: storage.k8s.io/v1
```

```
kind: StorageClass
metadata:
  name: gcnv-sc-smb
provisioner: csi.trident.netapp.io
parameters:
  backendType: "google-cloud-netapp-volumes"
  trident.netapp.io/nasType: "smb"
  csi.storage.k8s.io/node-stage-secret-name: "smbcreds"
  csi.storage.k8s.io/node-stage-secret-namespace: ${pvc.namespace}
```

▼ **Utilizzo di segreti diversi per volume**

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: gcnv-sc-smb
provisioner: csi.trident.netapp.io
parameters:
  backendType: "google-cloud-netapp-volumes"
  trident.netapp.io/nasType: "smb"
  csi.storage.k8s.io/node-stage-secret-name: ${pvc.name}
  csi.storage.k8s.io/node-stage-secret-namespace: ${pvc.namespace}
```

`nasType: smb` filtri per i pool che supportano i volumi SMB. `nasType: nfs` o `nasType: null` filtri per i pool NFS.

Esempio di definizione PVC

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: gcnv-nfs-pvc
spec:
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 100Gi
  storageClassName: gcnv-nfs-sc
```

Per verificare se il PVC è vincolato, eseguire il seguente comando:

```
kubectl get pvc gcnv-nfs-pvc
```

```
NAME          STATUS   VOLUME                                     CAPACITY
ACCESS MODES   STORAGECLASS AGE
gcnv-nfs-pvc  Bound    pvc-b00f2414-e229-40e6-9b16-ee03eb79a213  100Gi
RWX            gcnv-nfs-sc  1m
```

Configura il tiering automatico per Google Cloud NetApp Volumes

Il tiering automatico viene configurato tramite i parametri del backend di Trident e le annotazioni PersistentVolumeClaim durante il provisioning dei volumi. È possibile configurare il tiering automatico per Google Cloud NetApp Volumes utilizzando Trident.

Panoramica

L'auto-tiering consente a Trident di effettuare il provisioning di volumi che spostano automaticamente i dati inattivi da un livello di prestazioni a un livello di capacità. Questo riduce i costi di storage preservando le prestazioni per i dati a cui si accede di frequente.

Trident applica le impostazioni di auto-tiering solo al momento della creazione del volume. Le modifiche successive al provisioning non sono supportate in Trident 26.06.

Concetti

Auto-tiering

L'auto-tiering sposta i dati a cui si accede raramente da un livello di prestazioni a un livello di capacità in base ai modelli di accesso. Lo spostamento dei dati avviene in modo asincrono e non è immediato.

Policy di tiering

Il criterio di tiering determina se il tiering automatico è abilitato per un volume.

Sono supportate le seguenti policy: * `auto`: Abilita la suddivisione in livelli automatica in base ai modelli di accesso * `none`: Disabilita la suddivisione in livelli automatica

Giorni di raffreddamento

I giorni di raffreddamento specificano il numero minimo di giorni per cui un blocco di dati deve rimanere inattivo prima di diventare idoneo per il tiering. I giorni di raffreddamento si applicano solo quando la tiering policy è impostata su `auto`.

Modello di configurazione

Ambiti di configurazione

L'auto-tiering può essere configurato in più ambiti:

- **Ambito del pool di archiviazione** Si applica a tutti i volumi provisioned dal pool.
- **Ambito del volume** Si applica a un singolo volume tramite annotazioni PersistentVolumeClaim.

Trident determina la configurazione effettiva in base a dove è definita ciascuna impostazione.

Precedenza di configurazione

Quando la stessa impostazione è definita in più ambiti, Trident applica il seguente ordine di precedenza:

1. Annotazioni PersistentVolumeClaim
2. Configurazione del backend Trident
3. Valori predefiniti del pool di archiviazione

Le impostazioni definite a un livello di precedenza superiore sovrascrivono i valori di livello inferiore.

Funzionalità supportate in Trident 26.06

Trident 26.06 supporta le seguenti funzionalità di auto-tiering per Google Cloud NetApp Volumes:

- Abilitazione o disabilitazione del tiering automatico durante il provisioning del volume
- Definizione di una politica di tiering nella configurazione del backend Trident
- Sovrascrivere la policy di tiering e i giorni di raffreddamento per volume utilizzando le annotazioni PVC
- Configurazione dei giorni di raffreddamento per i volumi con auto-tiering abilitato

Funzionalità non supportata in Trident 26.06

Le seguenti operazioni non sono supportate:

- Modifica delle impostazioni di auto-tiering dopo la creazione del volume
- Modifica delle policy di tiering sui volumi esistenti tramite gli aggiornamenti di Kubernetes
- Applicazione delle impostazioni di auto-tiering al di fuori dei flussi di lavoro di provisioning gestiti da Trident

Parametri di configurazione del backend

I seguenti parametri controllano il comportamento dell'auto-tiering quando definiti nella configurazione del backend Trident:

Parametro	Richiesto	Descrizione
tieringPolicy	No	Criterio di suddivisione in livelli per volumi (`auto` or `none`)
tieringMinimumCoolingDays	No	Numero di giorni di inattività prima che i dati vengano tiered (intervallo: 2–183, predefinito: 31)

Sovrascritture a livello di volume tramite annotazioni PersistentVolumeClaim

Annotazioni supportate

Le annotazioni PersistentVolumeClaim consentono di ignorare le impostazioni di auto-tiering per volume.

Annotazione	Descrizione
`trident.netapp.io/tieringPolicy`	Sostituisce la policy di tiering per il volume

Annotazione	Descrizione
`trident.netapp.io/tieringMinimumCoolingDays`	Sostituisce il valore dei giorni di raffreddamento per il volume

Esempio: PersistentVolumeClaim con override di auto-tiering

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: auto-tiering-pvc
  annotations:
    trident.netapp.io/tieringPolicy: auto
    trident.netapp.io/tieringMinimumCoolingDays: "45"
spec:
  accessModes:
    - ReadWriteOnce
  storageClassName: google-cloud-netapp-volumes-auto-tiering
  resources:
    requests:
      storage: 500Gi
```

Comportamento e limitazioni

Comportamento di provisioning

- Le impostazioni di auto-tiering vengono valutate e applicate solo al momento della creazione del volume.
- Trident non riconcilia la configurazione del tiering dopo il provisioning.
- I giorni di raffreddamento vengono ignorati quando la policy di tiering è impostata su `none`.

Limitazioni della piattaforma

- Il tiering automatico è supportato solo per i volumi NAS (NFS e SMB).
- I volumi a blocchi (iSCSI) non supportano l'auto-tiering.
- Il pool di storage Google Cloud NetApp Volumes deve avere la suddivisione automatica in livelli abilitata in Google Cloud.

Valori supportati

- Intervallo valido per `tieringMinimumCoolingDays`: 2 a 183
- Valore predefinito: 31

Configura i pool di storage in modalità ONTAP di Google Cloud NetApp Volumes

A partire da Trident 26.06, puoi configurare backend esistenti `ontap-san` e `ontap-nas` backend per effettuare il provisioning dello storage su Google Cloud NetApp Volumes

(GCNV) pool di storage in modalità ONTAP. Questa funzionalità è una Tech Preview.

NOTA	Questa funzionalità è una Tech Preview in Trident 26.06. Le funzionalità Tech Preview non sono supportate per l'utilizzo in produzione. La funzionalità, i campi di configurazione e la matrice supportata possono cambiare prima della disponibilità generale (GA).

Come funziona

Trident riutilizza la logica del driver esistente `ontap-san` e `ontap-nas` e instrada le chiamate REST ONTAP attraverso un endpoint proxy GCNV. Questo approccio ti permette di eseguire il provisioning e gestire lo storage su cluster ONTAP ospitati da GCNV mantenendo gli stessi flussi di lavoro operativi che usi per un backend ONTAP diretto. La modalità ONTAP non introduce un nuovo driver di storage. La abiliti per ogni backend aggiungendo un blocco di configurazione `gcnv` a un backend `ontap-san` o `ontap-nas`.

Il percorso di provisioning è il seguente:

`PersistentVolumeClaim` → Trident (`ontap-san` o `ontap-nas`) → client REST ONTAP → proxy GCNV → cluster ONTAP nel pool GCNV

Ambito supportato e non supportato

Questa anteprima tecnica supporta le seguenti funzionalità:

- Driver: `ontap-san` (iSCSI) e `ontap-nas` (NFS o SMB).
- Operazioni del ciclo di vita tramite proxy: inizializzazione del backend, creazione ed eliminazione dei volumi e relativi flussi REST ONTAP.
- Modelli di autenticazione: Workload Identity Pool (WIP), chiave dell'account di servizio e fallback alle Application Default Credentials (ADC).

I seguenti elementi sono esclusi dall'ambito di questa Tech Preview:

- I `ontap-san-economy`, `ontap-nas-economy` e `ontap-nas-flexgroup` driver.
- Il percorso di personalità ASA r2.
- Ripiego ZAPI. La modalità ONTAP usa solo ONTAP REST.

Prerequisiti

Prima di configurare un backend in modalità ONTAP, assicurati di avere quanto segue:

- Trident 26.06 o versioni successive.
- Un pool di storage GCNV ONTAP-mode nel tuo progetto e nella posizione di destinazione.
- Autorizzazioni IAM di GCP per le operazioni proxy, con ambito appropriato per il tuo ambiente.
- Una configurazione di backend che include `proxyURL`, `projectNumber`, `location` e `poolID`.
- Per i flussi di lavoro SAN iSCSI, i prerequisiti iSCSI e multipath lato nodo per la tua piattaforma. Fai riferimento a "Prepara il nodo worker".

Configura il backend

A `TridentBackendConfig` richiede `credentials`. Per la modalità ONTAP, quel secret contiene le credenziali del proxy GCP, non ONTAP `managementLIF`/ `username`/ `password`. `svm` è facoltativo: ogni pool

GCNV in modalità ONTAP attualmente ha un SVM e Trident lo ricava dal pool quando omesso.

Usa lo stesso modello di account di servizio come per il GCNV nativo: campi non sensibili in `gcnv.apiKey`, `private_key` e `private_key_id` nel secret. Vedi "Esempi di configurazione di Google Cloud NetApp Volumes".

backend ontap-san

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: gcnv-ontap-san
  namespace: trident
spec:
  version: 1
  storageDriverName: ontap-san
  backendName: gcnv-ontap-san
  credentials:
    name: gcnv-sa-secret
    type: secret
  gcnv:
    proxyURL: "https://netapp.googleapis.com"
    projectNumber: "<project-number>"
    location: "<region-or-zone>"
    poolID: "<pool-id>"
  svm: "<svm-name>"
```

backend ontap-nas

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: gcnv-ontap-nas
  namespace: trident
spec:
  version: 1
  storageDriverName: ontap-nas
  backendName: gcnv-ontap-nas
  credentials:
    name: gcnv-sa-secret
    type: secret
  gcnv:
    proxyURL: "https://netapp.googleapis.com"
    projectNumber: "<project-number>"
    location: "<region-or-zone>"
    poolID: "<pool-id>"
    apiKey:
```

```
      type: service_account
      project_id: "<project-id>"
      client_email: "<service-account-email>"
      client_id: "<client-id>"
      auth_uri: "https://accounts.google.com/o/oauth2/auth"
      token_uri: "https://oauth2.googleapis.com/token"
      auth_provider_x509_cert_url:
"https://www.googleapis.com/oauth2/v1/certs"
      client_x509_cert_url: "<client-x509-cert-url>"
```

Segreto

```
apiVersion: v1
kind: Secret
metadata:
  name: gcnv-sa-secret
  namespace: trident
type: Opaque
stringData:
  private_key_id: "<private-key-id>"
  private_key: |
    -----BEGIN PRIVATE KEY-----
    <private-key>
    -----END PRIVATE KEY-----
```

NOTA Non mettere `private_key` o `private_key_id` nella `TridentBackendConfig` spec: il CRD li rifiuta. Usa `ontap-san` con lo stesso `gcnv` e lo stesso modello di credenziali per storage a blocchi.

```
svm: "<svm-name>"
```

```
== Authentication

Trident resolves credentials for proxy access in the following order:

. Workload Identity Pool (WIP)
. Service account key
. Application Default Credentials (ADC)

For this Tech Preview, use the following practices:

* Prefer Workload Identity Pool where it is available.
* Do not embed raw private keys in version-controlled backend files. Store
```

```
credentials in a Kubernetes secret.
* Apply least-privilege IAM scoping to the service account.

== Validation and initialization behavior

Trident validates an ONTAP-mode backend during initialization and fails
fast rather than partially initializing.

Backend initialization fails if a required `gcnv` field is missing, the
driver is unsupported for ONTAP-mode, or credential resolution fails.

== Known limitations

* This feature is a Tech Preview. Behavior and the supported matrix can
change before GA.
* ZAPI fallback is not used in ONTAP-mode.
* Protocol and node-readiness requirements still apply. For example, iSCSI
workflows require the node-side prerequisites described in
xref:{relative_path}../trident-use/worker-node-prep.html[Prepare the
worker node].
* Existing ONTAP and GCNV environment constraints still apply.

== Upgrade and compatibility

ONTAP-mode is opt-in per backend through the `gcnv` configuration block:

* Existing non-GCNV ONTAP backends are unaffected.
* Mixed deployments that combine direct ONTAP backends and ONTAP-mode
proxy backends are supported through backend-level configuration. Each
backend is configured independently.
* To stop using ONTAP-mode, remove or replace the affected backends. No
global switch is required.

== What's next?

Apply secret before TBC; use `kubectl apply` and `-n trident` to match
examples.
```

Configura un backend NetApp HCI o SolidFire

Scoprite come creare e utilizzare un backend Element con la vostra installazione di Trident.

Dettagli del driver Element

Trident fornisce il `solidfire-san` driver di storage per comunicare con il cluster. Le modalità di accesso

supportate sono: *ReadWriteOnce* (RWO), *ReadOnlyMany* (ROX), *ReadWriteMany* (RWX), *ReadWriteOncePod* (RWOP).

Il `solidfire-san` driver di storage supporta le modalità di volume *file* e *block*. Per la `Filesystem` volumeMode, Trident crea un volume e crea un filesystem. Il tipo di filesystem è specificato da StorageClass.

Driver	Protocollo	VolumeMode	Modalità di accesso supportate	File system supportati
`solidfire-san`	iSCSI	Blocco	RWO, ROX, RWX, RWOP	Nessun filesystem. Dispositivo a blocchi raw.
`solidfire-san`	iSCSI	Filesystem	RWO, RWOP	`xfs`, `ext3`, `ext4`

Prima di iniziare

Avrai bisogno dei seguenti elementi prima di creare un backend Element.

- Un sistema storage supportato che esegue il software Element.
- Credenziali per un utente admin o tenant di un cluster NetApp HCI/SolidFire che può gestire i volumi.
- Tutti i nodi worker di Kubernetes devono avere installati gli strumenti iSCSI appropriati. Fai riferimento a "informazioni sulla preparazione del nodo worker".

Opzioni di configurazione del backend

Consulta la tabella seguente per le opzioni di configurazione del backend:

Parametro	Descrizione	Predefinito
`version`		Sempre 1
`storageDriverName`	Nome del driver di archiviazione	Sempre "solidfire-san"
`backendName`	Nome personalizzato o lo storage backend	"solidfire_" + indirizzo IP storage (iSCSI)
`Endpoint`	MVIP per il SolidFire cluster con credenziali tenant	
`SVIP`	Storage (iSCSI) indirizzo IP e porta	
`labels`	Set di etichette arbitrarie in formato JSON da applicare ai volumi.	""
`TenantName`	Nome del tenant da utilizzare (creato se non trovato)	
`InitiatorIFace`	Limitare il traffico iSCSI a una specifica interfaccia verso gli host	"default"
`UseCHAP`	Usa CHAP per autenticare iSCSI. Trident usa CHAP.	true
`AccessGroups`	Elenco degli ID dei gruppi di accesso da utilizzare	Trova l'ID di un gruppo di accesso chiamato "trident"

Parametro	Descrizione	Predefinito
`Types`	Specifiche QoS	
`limitVolumeSize`	Il provisioning non riesce se la dimensione del volume richiesta è superiore a questo valore	"" (non applicato per impostazione predefinita)
`debugTraceFlags`	Flag di debug da usare per la risoluzione dei problemi. Esempio, {"api":false, "method":true}	null

ATTENZIONE Non utilizzare `debugTraceFlags` a meno che tu non stia effettuando una ricerca guasti e richieda un dump dettagliato dei registri.

Esempio 1: configurazione del backend per `solidfire-san` driver con tre tipi di volume

Questo esempio mostra un file di backend che utilizza l'autenticazione CHAP e modella tre tipi di volume con garanzie QoS specifiche. È molto probabile che si definiscano classi di storage per consumare ciascuna di queste utilizzando il parametro `IOPS` storage class.

```
---
version: 1
storageDriverName: solidfire-san
Endpoint: https://<user>:<password>@<mvip>/json-rpc/8.0
SVIP: <svip>:3260
TenantName: <tenant>
labels:
  k8scluster: dev1
  backend: dev1-element-cluster
UseCHAP: true
Types:
  - Type: Bronze
    Qos:
      minIOPS: 1000
      maxIOPS: 2000
      burstIOPS: 4000
  - Type: Silver
    Qos:
      minIOPS: 4000
      maxIOPS: 6000
      burstIOPS: 8000
  - Type: Gold
    Qos:
      minIOPS: 6000
      maxIOPS: 8000
      burstIOPS: 10000
```

Esempio 2: Configurazione del backend e della storage class per `solidfire-san` driver con pool virtuali

Questo esempio mostra il file di definizione del backend configurato con pool virtuali insieme a StorageClasses che fanno riferimento ad essi.

Trident copia le etichette presenti su un pool di storage sul LUN di storage backend al momento del provisioning. Per comodità, gli amministratori dello storage possono definire etichette per pool virtuale e raggruppare i volumi in base all'etichetta.

Nel file di definizione del backend di esempio mostrato di seguito, vengono impostati valori predefiniti specifici per tutti i pool di storage, che impostano `type` a Silver. I pool virtuali sono definiti nella sezione `storage`. In questo esempio, alcuni dei pool di storage impostano il proprio tipo e alcuni pool sovrascrivono i valori predefiniti impostati sopra.

```
---
version: 1
storageDriverName: solidfire-san
Endpoint: https://<user>:<password>@<mvip>/json-rpc/8.0
SVIP: <svip>:3260
TenantName: <tenant>
UseCHAP: true
Types:
  - Type: Bronze
    Qos:
      minIOPS: 1000
      maxIOPS: 2000
      burstIOPS: 4000
  - Type: Silver
    Qos:
      minIOPS: 4000
      maxIOPS: 6000
      burstIOPS: 8000
  - Type: Gold
    Qos:
      minIOPS: 6000
      maxIOPS: 8000
      burstIOPS: 10000
type: Silver
labels:
  store: solidfire
  k8scluster: dev-1-cluster
region: us-east-1
storage:
  - labels:
      performance: gold
      cost: "4"
    zone: us-east-1a
```

```
    type: Gold
  - labels:
      performance: silver
      cost: "3"
    zone: us-east-1b
    type: Silver
  - labels:
      performance: bronze
      cost: "2"
    zone: us-east-1c
    type: Bronze
  - labels:
      performance: silver
      cost: "1"
    zone: us-east-1d
```

Le seguenti definizioni di StorageClass si riferiscono ai pool virtuali di cui sopra. Utilizzando il campo `parameters.selector`, ogni StorageClass indica quali pool virtuali possono essere utilizzati per ospitare un volume. Il volume avrà gli aspetti definiti nel pool virtuale scelto.

Il primo StorageClass (`solidfire-gold-four`) si riferisce al primo pool virtuale. Questo è l'unico pool che offre prestazioni gold con un `Volume Type QoS` di Gold. L'ultimo StorageClass (`solidfire-silver`) richiama qualsiasi pool di storage che offre prestazioni silver. Trident deciderà quale pool virtuale selezionare e garantirà che il requisito di storage sia soddisfatto.

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: solidfire-gold-four
provisioner: csi.trident.netapp.io
parameters:
  selector: performance=gold; cost=4
  fsType: ext4

---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: solidfire-silver-three
provisioner: csi.trident.netapp.io
parameters:
  selector: performance=silver; cost=3
  fsType: ext4

---
apiVersion: storage.k8s.io/v1
```

```
kind: StorageClass
metadata:
  name: solidfire-bronze-two
provisioner: csi.trident.netapp.io
parameters:
  selector: performance=bronze; cost=2
  fsType: ext4

---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: solidfire-silver-one
provisioner: csi.trident.netapp.io
parameters:
  selector: performance=silver; cost=1
  fsType: ext4

---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: solidfire-silver
provisioner: csi.trident.netapp.io
parameters:
  selector: performance=silver
  fsType: ext4
```

Trova ulteriori informazioni

- "Gruppi di accesso al volume"

Driver SAN ONTAP

Panoramica del driver ONTAP SAN

Scopri come configurare un backend ONTAP con i driver ONTAP e Cloud Volumes ONTAP SAN.

Dettagli del driver ONTAP SAN

Trident fornisce i seguenti driver di storage SAN per comunicare con il cluster ONTAP. Le modalità di accesso supportate sono: *ReadWriteOnce* (RWO), *ReadOnlyMany* (ROX), *ReadWriteMany* (RWX), *ReadWriteOncePod* (RWOP).

Driver	Protocollo	volumeMode	Modalità di accesso supportate	File system supportati
`ontap-san`	iSCSI SCSI su FC	Blocco	RWO, ROX, RWX, RWOP	Nessun file system; dispositivo a blocchi raw
`ontap-san`	iSCSI SCSI su FC	Filesystem	RWO, RWOP ROX e RWX non sono disponibili nella modalità volume file system.	`xfs`, `ext3`, `ext4`
`ontap-san`	NVMe/TCP Fai riferimento a Considerazioni aggiuntive per NVMe/TCP.	Blocco	RWO, ROX, RWX, RWOP	Nessun file system; dispositivo a blocchi raw
`ontap-san`	NVMe/TCP Fai riferimento a Considerazioni aggiuntive per NVMe/TCP.	Filesystem	RWO, RWOP ROX e RWX non sono disponibili nella modalità volume file system.	`xfs`, `ext3`, `ext4`
`ontap-san-economy`	iSCSI	Blocco	RWO, ROX, RWX, RWOP	Nessun file system; dispositivo a blocchi raw
`ontap-san-economy`	iSCSI	Filesystem	RWO, RWOP ROX e RWX non sono disponibili nella modalità volume file system.	`xfs`, `ext3`, `ext4`

ATTENZIONE

- Utilizzare `ontap-san-economy` solo se si prevede che il conteggio dell'utilizzo del volume persistente sia superiore a "limiti di volume ONTAP supportati".
- Utilizzare `ontap-nas-economy` solo se si prevede che il conteggio dell'utilizzo del volume persistente sia superiore a "limiti di volume ONTAP supportati" e il `ontap-san-economy` driver non può essere utilizzato.
- Non utilizzare `ontap-nas-economy` se si prevede la necessità di protezione dei dati, disaster recovery o mobilità.
- NetApp non consiglia di utilizzare l'autogrow di FlexVol in tutti i driver ONTAP, ad eccezione di ontap-san. Come soluzione alternativa, Trident supporta l'utilizzo della riserva di snapshot e ridimensiona di conseguenza i volumi FlexVol.

Permessi utente

Trident prevede di essere eseguito come amministratore ONTAP o SVM, in genere utilizzando l'

`admin`utente del cluster o un `vsadmin`utente SVM, oppure un utente con un nome diverso che ha lo stesso ruolo. Per le distribuzioni Amazon FSx for NetApp ONTAP, Trident prevede di essere eseguito come amministratore ONTAP o SVM, utilizzando l'utente del cluster `fsxadmin` o un `vsadmin`utente SVM, oppure un utente con un nome diverso che ha lo stesso ruolo. L' `fsxadmin`utente è un sostituto limitato per l'utente amministratore del cluster.

NOTA	Se si utilizza il `limitAggregateUsage` parametro, sono richieste le autorizzazioni di amministratore del cluster. Quando si utilizza Amazon FSx for NetApp ONTAP con Trident, il `limitAggregateUsage` parametro non funzionerà con gli account utente `vsadmin` e `fsxadmin`. L'operazione di configurazione non andrà a buon fine se si specifica questo parametro.

Sebbene sia possibile creare un ruolo più restrittivo all'interno di ONTAP che un driver Trident può utilizzare, lo sconsigliamo. La maggior parte delle nuove versioni di Trident richiamerà API aggiuntive di cui bisognerebbe tenere conto, rendendo gli aggiornamenti difficili e soggetti a errori.

Considerazioni aggiuntive per NVMe/TCP

Trident supporta il protocollo non-volatile memory express (NVMe) utilizzando il `ontap-san` driver, incluso:

- IPv6
- Snapshot e cloni di volumi NVMe
- Ridimensionamento di un volume NVMe
- Importazione di un volume NVMe creato al di fuori di Trident in modo che il suo ciclo di vita possa essere gestito da Trident
- Multipathing nativo NVMe
- Arresto regolare o non regolare dei nodi K8s (24.06)

Trident non supporta:

- DH-HMAC-CHAP che è supportato nativamente da NVMe
- Multipathing del device mapper (DM)
- Crittografia LUKS

NOTA	NVMe è supportato solo con le API REST di ONTAP e non è supportato con ONTAPI (ZAPI).

Prepararsi a configurare il backend con i driver ONTAP SAN

Comprendere i requisiti e le opzioni di autenticazione per la configurazione di un backend ONTAP con driver ONTAP SAN.

Requisiti

Per tutti i backend ONTAP, Trident richiede che almeno un aggregato sia assegnato all'SVM.

NOTA	"Sistemi ASA r2" differiscono dagli altri sistemi ONTAP (ASA, AFF e FAS) nell'implementazione del loro livello di storage. Nei sistemi ASA r2, vengono utilizzate zone di disponibilità dello storage al posto degli aggregati. Fare riferimento all'"questo"articolo della Knowledge Base su come assegnare gli aggregati alle SVM nei sistemi ASA r2.

Ricorda che puoi anche eseguire più di un driver e creare classi di archiviazione che puntano all'uno o all'altro. Ad esempio, puoi configurare una `san-dev` classe che utilizza il `ontap-san` driver e una `san-default` classe che utilizza il `ontap-san-economy` driver.

Tutti i nodi worker di Kubernetes devono avere installati gli strumenti iSCSI appropriati. Per i dettagli, fai riferimento a "Prepara il nodo worker".

Autenticare il backend ONTAP

Trident offre due modalità di autenticazione per un backend ONTAP.

- Basato su credenziali: il nome utente e la password di un utente ONTAP con le autorizzazioni richieste. Si consiglia di utilizzare un ruolo di login di sicurezza predefinito, come `admin` o `vsadmin` per garantire la massima compatibilità con le versioni di ONTAP.
- Basato su certificato: Trident può anche comunicare con un cluster ONTAP utilizzando un certificato installato sul backend. In questo caso, la definizione del backend deve contenere i valori codificati in Base64 del certificato client, della chiave e del certificato CA attendibile, se utilizzato (consigliato).

È possibile aggiornare i backend esistenti per passare tra metodi basati su credenziali e metodi basati su certificati. Tuttavia, è supportato solo un metodo di autenticazione alla volta. Per passare a un diverso metodo di autenticazione, è necessario rimuovere il metodo esistente dalla configurazione del backend.

ATTENZIONE	Se si tenta di fornire **sia le credenziali che i certificati**, la creazione del backend fallirà con un errore che indica che è stato fornito più di un metodo di autenticazione nel file di configurazione.

Abilitare l'autenticazione basata su credenziali

Trident richiede le credenziali di un amministratore SVM-scoped/cluster-scoped per comunicare con il backend ONTAP. Si consiglia di utilizzare ruoli standard, predefiniti come `admin` o `vsadmin`. Questo garantisce la compatibilità futura con le versioni ONTAP che potrebbero esporre API di funzionalità da utilizzare nelle future versioni di Trident. È possibile creare e utilizzare un ruolo di accesso di sicurezza personalizzato con Trident, ma non è consigliato.

Un esempio di definizione di backend sarà simile al seguente:

YAML

```yaml
---
version: 1
backendName: ExampleBackend
storageDriverName: ontap-san
managementLIF: 10.0.0.1
svm: svm_nfs
username: vsadmin
password: password
```

JSON

```
{
```

```
    "version": 1,
    "backendName": "ExampleBackend",
    "storageDriverName": "ontap-san",
    "managementLIF": "10.0.0.1",
    "svm": "svm_nfs",
    "username": "vsadmin",
    "password": "password"
}
```

Tenete presente che la definizione del backend è l'unico posto in cui le credenziali vengono archiviate in testo normale. Dopo la creazione del backend, nomi utente e password vengono codificati in Base64 e archiviati come segreti Kubernetes. La creazione o l'aggiornamento di un backend è l'unico passaggio che richiede la conoscenza delle credenziali. Pertanto, si tratta di un'operazione riservata all'amministratore, da eseguire dall'amministratore di Kubernetes/storage.

Abilita l'autenticazione basata sul certificato

I backend nuovi ed esistenti possono utilizzare un certificato e comunicare con il backend ONTAP. Sono richiesti tre parametri nella definizione del backend.

- clientCertificate: Valore codificato in Base64 del certificato client.
- clientPrivateKey: Valore codificato in Base64 della chiave privata associata.
- trustedCACertificate: Valore codificato in Base64 del certificato della CA fidata. Se si utilizza una CA fidata, questo parametro deve essere fornito. Questo può essere ignorato se non si utilizza una CA fidata.

Un tipico flusso di lavoro prevede i seguenti passaggi.

Passaggi

1. Generare un certificato e una chiave client. Durante la generazione, impostare Common Name (CN) sull'utente ONTAP con cui autenticarsi.

```
openssl req -x509 -nodes -days 1095 -newkey rsa:2048 -keyout k8senv.key
-out k8senv.pem -subj "/C=US/ST=NC/L=RTP/O=NetApp/CN=admin"
```

2. Aggiungere un certificato CA attendibile al cluster ONTAP. Questa operazione potrebbe essere già gestita dall'amministratore dello storage. Ignorare se non viene utilizzata una CA attendibile.

```
security certificate install -type server -cert-name <trusted-ca-cert-
name> -vserver <vserver-name>
ssl modify -vserver <vserver-name> -server-enabled true -client-enabled
true -common-name <common-name> -serial <SN-from-trusted-CA-cert> -ca
<cert-authority>
```

3. Installare il certificato e la chiave del client (dal punto 1) sul cluster ONTAP.

```
security certificate install -type client-ca -cert-name <certificate-
name> -vserver <vserver-name>
security ssl modify -vserver <vserver-name> -client-enabled true
```

NOTA Dopo aver eseguito questo comando, ONTAP richiede l'inserimento del certificato. Incolla il contenuto del `k8senv.pem` file generato nel passaggio 1, quindi inserisci `END` per completare l'installazione.

4. Confermare che il ruolo di login di sicurezza ONTAP supporta `cert` il metodo di autenticazione.

```
security login create -user-or-group-name admin -application ontapi
-authentication-method cert
security login create -user-or-group-name admin -application http
-authentication-method cert
```

5. Verifica l'autenticazione utilizzando il certificato generato. Sostituisci <ONTAP Management LIF> e <vserver name> con l'IP LIF di gestione e il nome SVM.

```
curl -X POST -Lk https://<ONTAP-Management-
LIF>/servlets/netapp.servlets.admin.XMLrequest_filer --key k8senv.key
--cert ~/k8senv.pem -d '<?xml version="1.0" encoding="UTF-8"?><netapp
xmlns="http://www.netapp.com/filer/admin" version="1.21"
vfiler="<vserver-name>"><vserver-get></vserver-get></netapp>'
```

6. Codifica il certificato, la chiave e il certificato CA affidabile con Base64.

```
base64 -w 0 k8senv.pem >> cert_base64
base64 -w 0 k8senv.key >> key_base64
base64 -w 0 trustedca.pem >> trustedca_base64
```

7. Crea il backend utilizzando i valori ottenuti nel passaggio precedente.

```
cat cert-backend.json
{
"version": 1,
"storageDriverName": "ontap-san",
"backendName": "SanBackend",
"managementLIF": "1.2.3.4",
"svm": "vserver_test",
"clientCertificate": "Faaaakkkkeeee...Vaaalllluuuueeee",
"clientPrivateKey": "LS0tFaKE...0VaLuES0tLS0K",
"trustedCACertificate": "QNFinfO...SiqOyN",
```

```
"storagePrefix": "myPrefix_"
}

tridentctl create backend -f cert-backend.json -n trident
+------------+----------------+--------------------------------------
+--------+---------+
|    NAME    | STORAGE DRIVER |                 UUID                 |
STATE  | VOLUMES |
+------------+----------------+--------------------------------------
+--------+---------+
| SanBackend | ontap-san      | 586b1cd5-8cf8-428d-a76c-2872713612c1 |
online |       0 |
+------------+----------------+--------------------------------------
+--------+---------+
```

Aggiorna i metodi di autenticazione o ruota le credenziali

È possibile aggiornare un backend esistente per utilizzare un metodo di autenticazione diverso o per ruotare le credenziali. Questo funziona in entrambi i modi: i backend che utilizzano nome utente/password possono essere aggiornati per utilizzare certificati; i backend che utilizzano certificati possono essere aggiornati per utilizzare nome utente/password. Per fare ciò, è necessario rimuovere il metodo di autenticazione esistente e aggiungere il nuovo metodo di autenticazione. Quindi utilizzare il file backend.json aggiornato contenente i parametri richiesti per eseguire `tridentctl backend update`.

```
cat cert-backend-updated.json
{
"version": 1,
"storageDriverName": "ontap-san",
"backendName": "SanBackend",
"managementLIF": "1.2.3.4",
"svm": "vserver_test",
"username": "vsadmin",
"password": "password",
"storagePrefix": "myPrefix_"
}

#Update backend with tridentctl
tridentctl update backend SanBackend -f cert-backend-updated.json -n
trident
+------------+----------------+--------------------------------------
+--------+---------+
|    NAME    | STORAGE DRIVER |                 UUID                 |
STATE  | VOLUMES |
+------------+----------------+--------------------------------------
+--------+---------+
| SanBackend | ontap-san      | 586b1cd5-8cf8-428d-a76c-2872713612c1 |
```

```
online |       9 |
+------------+---------------+-------------------------------------
+--------+---------+
```

NOTA Quando si ruotano le password, l'amministratore dello storage deve prima aggiornare la password dell'utente su ONTAP. Questo è seguito da un aggiornamento del backend. Quando si ruotano i certificati, è possibile aggiungere più certificati all'utente. Il backend viene quindi aggiornato per utilizzare il nuovo certificato, dopo di che il vecchio certificato può essere eliminato dal cluster ONTAP.

L'aggiornamento di un backend non interrompe l'accesso ai volumi già creati, né influisce sulle connessioni al volume effettuate successivamente. Un aggiornamento del backend riuscito indica che Trident può comunicare con il backend ONTAP e gestire le future operazioni sui volumi.

Crea un ruolo personalizzato ONTAP per Trident

È possibile creare un ruolo di cluster ONTAP con privilegi minimi in modo da non dover utilizzare il ruolo di amministratore ONTAP per eseguire operazioni in Trident. Quando si include il nome utente in una configurazione backend di Trident, Trident utilizza il ruolo di cluster ONTAP creato per eseguire le operazioni.

Fare riferimento a "Generatore di ruoli personalizzati Trident" per ulteriori informazioni sulla creazione di ruoli personalizzati Trident.

Utilizzo di ONTAP CLI

1. Crea un nuovo ruolo utilizzando il seguente comando:

```
security login role create <role_name\> -cmddirname "command" -access all
–vserver <svm_name\>
```

2. Crea un nome utente per l'utente Trident:

```
security login create -username <user_name\> -application ontapi
-authmethod <password\> -role <name_of_role_in_step_1\> –vserver
<svm_name\> -comment "user_description"
```

3. Assegna il ruolo all'utente:

```
security login modify username <user_name\> –vserver <svm_name\> -role
<role_name\> -application ontapi -application console -authmethod
<password\>
```

Utilizzo di System Manager

Eseguire i seguenti passaggi in ONTAP System Manager:

1. **Crea un ruolo personalizzato**:
   a. Per creare un ruolo personalizzato a livello di cluster, selezionare **Cluster > Settings**.

      (Oppure) Per creare un ruolo personalizzato a livello SVM, selezionare **Archiviazione > Storage VM > `required SVM`> Impostazioni > Utenti e ruoli**.

b. Selezionare l'icona della freccia (→) accanto a **Users and Roles**.
c. Seleziona **+Add** in **Roles**.
d. Definisci le regole per il ruolo e fai clic su **Save**.

2. **Mappa il ruolo all'utente Trident**: + Esegui i seguenti passaggi nella pagina **Utenti e ruoli**:
a. Selezionare l'icona Aggiungi **+** sotto **Utenti**.
b. Selezionare il nome utente richiesto e selezionare un ruolo nel menu a discesa per **Role**.
c. Fare clic su **Save**.

Per maggiori informazioni, consultare le seguenti pagine:

- "Ruoli personalizzati per l'amministrazione di ONTAP" o "Definisci ruoli personalizzati"
- "Lavorare con ruoli e utenti"

Autenticare le connessioni con CHAP bidirezionale

Trident può autenticare le sessioni iSCSI con CHAP bidirezionale per i `ontap-san` e `ontap-san-economy` driver. Ciò richiede l'abilitazione dell'opzione `useCHAP` nella definizione del backend. Quando impostato su `true`, Trident configura la sicurezza dell'initiator predefinito dell'SVM su CHAP bidirezionale e imposta il nome utente e i segreti dal file di backend. NetApp consiglia di utilizzare CHAP bidirezionale per autenticare le connessioni. Vedere la seguente configurazione di esempio:

```
---
version: 1
storageDriverName: ontap-san
backendName: ontap_san_chap
managementLIF: 192.168.0.135
svm: ontap_iscsi_svm
useCHAP: true
username: vsadmin
password: password
chapInitiatorSecret: cl9qxIm36DKyawxy
chapTargetInitiatorSecret: rqxigXgkesIpwxyz
chapTargetUsername: iJF4heBRT0TCwxyz
chapUsername: uh2aNCLSd6cNwxyz
```

ATTENZIONE	Il `useCHAP` parametro è un'opzione booleana che può essere configurata una sola volta. Per impostazione predefinita, è impostato su false. Dopo averlo impostato su true, non è possibile impostarlo su false.

Oltre a `useCHAP=true`, i campi `chapInitiatorSecret`, `chapTargetInitiatorSecret`, `chapTargetUsername` e `chapUsername`devono essere inclusi nella definizione del backend. I segreti possono essere modificati dopo la creazione di un backend eseguendo `tridentctl update`.

Come funziona

Impostando `useCHAP` su true, l'amministratore dello storage indica a Trident di configurare CHAP sul backend dello storage. Ciò include quanto segue:

- Configurazione di CHAP sull'SVM:
  - Se il tipo di sicurezza predefinito dell'iniziatore SVM è none (impostato per impostazione predefinita) **e** non sono presenti LUN preesistenti nel volume, Trident imposterà il tipo di sicurezza predefinito su `CHAP` e procederà alla configurazione del nome utente e dei segreti dell'iniziatore e del target CHAP.
  - Se l'SVM contiene LUN, Trident non abiliterà CHAP sull'SVM. Questo garantisce che l'accesso alle LUN già presenti sull'SVM non sia limitato.
- Configurazione del nome utente e dei segreti dell'iniziatore e del target CHAP; queste opzioni devono essere specificate nella configurazione del backend (come mostrato sopra).

Dopo la creazione del backend, Trident crea un corrispondente `tridentbackend` CRD e memorizza i segreti CHAP e i nomi utente come segreti Kubernetes. Tutti i PV creati da Trident su questo backend verranno montati e collegati tramite CHAP.

Ruota le credenziali e aggiorna i backend

È possibile aggiornare le credenziali CHAP aggiornando i parametri CHAP nel `backend.json` file. Ciò richiederà l'aggiornamento dei segreti CHAP e l'utilizzo del `tridentctl update` comando per riflettere queste modifiche.

ATTENZIONE	Quando si aggiornano i segreti CHAP per un backend, è necessario utilizzare `tridentctl` per aggiornare il backend. Non aggiornare le credenziali sul cluster di storage utilizzando ONTAP CLI o ONTAP System Manager, poiché Trident non sarà in grado di rilevare queste modifiche.

```
cat backend-san.json
{
    "version": 1,
    "storageDriverName": "ontap-san",
    "backendName": "ontap_san_chap",
    "managementLIF": "192.168.0.135",
    "svm": "ontap_iscsi_svm",
    "useCHAP": true,
    "username": "vsadmin",
    "password": "password",
    "chapInitiatorSecret": "cl9qxUpDaTeD",
    "chapTargetInitiatorSecret": "rqxigXgkeUpDaTeD",
    "chapTargetUsername": "iJF4heBRT0TCwxyz",
    "chapUsername": "uh2aNCLSd6cNwxyz",
}

./tridentctl update backend ontap_san_chap -f backend-san.json -n trident
+----------------+----------------+--------------------------------------
+--------+---------+
```

```
|   NAME          | STORAGE DRIVER |                 UUID                 |
STATE   | VOLUMES |
+----------------+----------------+--------------------------------------
+--------+---------+
| ontap_san_chap | ontap-san      | aa458f3b-ad2d-4378-8a33-1a472ffbeb5c |
online |       7 |
+----------------+----------------+--------------------------------------
+--------+---------+
```

Le connessioni esistenti non saranno interessate; continueranno a rimanere attive se le credenziali vengono aggiornate da Trident sulla SVM. Le nuove connessioni utilizzano le credenziali aggiornate e le connessioni esistenti continuano a rimanere attive. La disconnessione e la riconnessione dei vecchi PV comporterà l'utilizzo delle credenziali aggiornate.

Opzioni ed esempi di configurazione SAN ONTAP

Crea e usa i driver SAN ONTAP con la tua installazione di Trident. Questa sezione fornisce esempi di configurazione del backend e dettagli per la mappatura dei backend a StorageClasses. "Sistemi ASA r2" differiscono dagli altri sistemi ONTAP (ASA, AFF e FAS) nell'implementazione del loro livello di storage. Queste variazioni influiscono sull'uso di alcuni parametri come indicato. "Scopri di più sulle differenze tra i sistemi ASA r2 e gli altri sistemi ONTAP". Nella configurazione del backend di Trident, non devi specificare che il tuo sistema è ASA r2. Quando selezioni `ontap-san` come `storageDriverName`, Trident rileva automaticamente ASA r2 o altri sistemi ONTAP. Alcuni parametri di configurazione del backend non sono applicabili ai sistemi ASA r2 come indicato nella tabella seguente.

NOTA	Solo il `ontap-san` driver (con protocolli iSCSI, NVMe/TCP e FC) è supportato per i sistemi ASA r2.

Opzioni di configurazione del backend

A partire dalla versione 26.06, puoi abilitare il posizionamento bilanciato di ONTAP così che ONTAP selezioni l'aggregato per i nuovi volumi. Consulta "Usa un posizionamento bilanciato con i backend ONTAP" per i requisiti e gli esempi.

Consulta la tabella seguente per le opzioni di configurazione del backend:

Parametro	Descrizione	Predefinito
`version`		Sempre 1
`storageDriverName`	Nome del driver di archiviazione. Non puoi modificarlo dopo la creazione. Per aggiornare questo parametro, devi creare un nuovo backend.	`ontap-san` o `ontap-san-economy`
`backendName`	Nome personalizzato o lo storage backend	Nome driver + "_" + dataLIF

Parametro	Descrizione	Predefinito
`managementLIF`	Indirizzo IP di un cluster o di una LIF di gestione SVM. È possibile specificare un domain name pienamente qualificato (FQDN). Può essere impostato per usare indirizzi IPv6 se hai installato Trident usando il flag IPv6. Gli indirizzi IPv6 devono essere definiti tra parentesi quadre, ad esempio `[28e8:d9fb:a825:b7bf:69a8:d02f:9e7b:3555]`. Per un passaggio senza interruzioni di MetroCluster, vedere Esempio di MetroCluster. **NOTA** Se si utilizzano le credenziali "vsadmin", `managementLIF` deve essere quella dell'SVM; se si utilizzano le credenziali "admin", `managementLIF` deve essere quella del cluster.	"10.0.0.1", "[2001:1234:abcd::fefe]"
`dataLIF`	Indirizzo IP del protocollo LIF. Può essere impostato per usare indirizzi IPv6 se hai installato Trident usando il flag IPv6. Gli indirizzi IPv6 devono essere definiti tra parentesi quadre, ad esempio `[28e8:d9fb:a825:b7bf:69a8:d02f:9e7b:3555]`. **Non specificare per iSCSI.** Trident utilizza "ONTAP Selective LUN Map" per rilevare i LIF iSCSI necessari per stabilire una sessione multipath. Viene generato un avviso se `dataLIF` è definito esplicitamente. **Omettere per MetroCluster.** Vedi il Esempio di MetroCluster.	Derivato dall'SVM
`svm`	Macchina virtuale di storage da utilizzare **Ometti per MetroCluster.** Vedi il Esempio di MetroCluster.	Derivato se viene specificato un SVM `managementLIF`
`useCHAP`	Utilizzare CHAP per autenticare iSCSI per i driver SAN ONTAP [Booleano]. Impostare su `true` per consentire a Trident di configurare e utilizzare CHAP bidirezionale come autenticazione predefinita per l'SVM specificato nel backend. Per i dettagli, fai riferimento a "Prepararsi a configurare il backend con i driver ONTAP SAN". **Non supportato per FCP o NVMe/TCP.**	`false`
`chapInitiatorSecret`	Segreto dell'iniziatore CHAP. Obbligatorio se `useCHAP=true`	""
`labels`	Set di etichette arbitrarie in formato JSON da applicare ai volumi	""

Parametro	Descrizione	Predefinito
`chapTargetInitiatorSecret`	Segreto dell'iniziatore di destinazione CHAP. Obbligatorio se `useCHAP=true`	""
`chapUsername`	Nome utente in entrata. Obbligatorio se `useCHAP=true`	""
`chapTargetUsername`	Nome utente di destinazione. Obbligatorio se `useCHAP=true`	""
`clientCertificate`	Valore codificato in Base64 del certificato client. Utilizzato per l'autenticazione basata su certificato	""
`clientPrivateKey`	Valore codificato in Base64 della chiave privata del client. Utilizzato per l'autenticazione basata su certificato	""
`trustedCACertificate`	Valore codificato in Base64 del certificato CA attendibile. Opzionale. Utilizzato per l'autenticazione basata su certificato.	""
`username`	Nome utente necessario per comunicare con il cluster ONTAP. Utilizzata per l'autenticazione basata sulle credenziali. Per l'autenticazione Active Directory, vedere "Autenticare Trident a un backend SVM utilizzando le credenziali di Active Directory".	""
`password`	Password necessaria per comunicare con il cluster ONTAP. Utilizzata per l'autenticazione basata sulle credenziali. Per l'autenticazione Active Directory, vedere "Autenticare Trident a un backend SVM utilizzando le credenziali di Active Directory".	""
`svm`	Macchina virtuale di storage da utilizzare	Derivato se viene specificato un SVM `managementLIF`
`storagePrefix`	Prefisso utilizzato durante il provisioning di nuovi volumi nell'SVM. Non può essere modificato in seguito. Per aggiornare questo parametro, devi creare un nuovo backend.	`trident`

Parametro	Descrizione	Predefinito
`aggregate`	Aggregato per il provisioning (facoltativo; se impostato, deve essere assegnato all'SVM). Per il `ontap-nas-flexgroup` driver, questa opzione viene ignorata. Se non assegnato, è possibile utilizzare uno qualsiasi degli aggregati disponibili per il provisioning di un volume FlexGroup. **NOTA** Quando l'aggregato viene aggiornato in SVM, viene aggiornato automaticamente anche in Trident tramite polling di SVM, senza dover riavviare il Trident Controller. Quando hai configurato un aggregato specifico in Trident per il provisioning dei volumi, se l'aggregato viene rinominato o spostato fuori da SVM, il backend passa allo stato di errore in Trident durante il polling dell'aggregato SVM. Devi cambiare l'aggregato con uno presente su SVM oppure rimuoverlo del tutto per riportare online il backend. **Non specificare per i sistemi ASA r2**. **Non specificare quando `useBalancedPlacement` è impostato su `true`.**	""
`useBalancedPlacement`	Delega la selezione dell'aggregato per i nuovi volumi a ONTAP [Boolean]. Richiede ONTAP 9.19.1 o versioni successive e `useREST` impostato su `true`. Non può essere combinato con `aggregate`. Consulta "Usa un posizionamento bilanciato con i backend ONTAP" per i requisiti e gli esempi.	`false`
`limitAggregateUsage`	Il provisioning fallisce se l'utilizzo supera questa percentuale. Se si utilizza un Amazon FSx for NetApp ONTAP backend, non specificare `limitAggregateUsage`. I `fsxadmin` e `vsadmin` forniti non contengono le autorizzazioni necessarie per recuperare l'utilizzo aggregato e limitarlo tramite Trident. **Non specificare per i sistemi ASA r2**.	"" (non applicato per impostazione predefinita)
`limitVolumeSize`	Il provisioning fallisce se la dimensione del volume richiesto supera questo valore. Limita anche la dimensione massima dei volumi che gestisce per LUN.	"" (non applicato per impostazione predefinita)
`lunsPerFlexvol`	Numero massimo di LUN per FlexVol, deve essere nell'intervallo [50, 200]	`100`

Parametro	Descrizione	Predefinito
`debugTraceFlags`	Flag di debug da usare per la risoluzione dei problemi. Esempio, {"api":false, "method":true} non usare a meno che non si stia eseguendo una risoluzione dei problemi e sia necessario un dump dettagliato del registro.	`null`
`useREST`	Parametro booleano per utilizzare le API REST di ONTAP. `useREST` Quando impostato su `true`, Trident utilizza le ONTAP REST APIs per comunicare con il backend; quando impostato su `false`, Trident utilizza le chiamate ONTAPI (ZAPI) per comunicare con il backend. Questa funzione richiede ONTAP 9.11.1 e versioni successive. Inoltre, il ruolo di login ONTAP utilizzato deve avere accesso all'applicazione `ontapi`. Questo è soddisfatto dai ruoli predefiniti `vsadmin` e `cluster-admin`. A partire dalla release Trident 24.06 e ONTAP 9.15.1 o versioni successive, `useREST` è impostato su `true` per impostazione predefinita; modificare `useREST` su `false` per utilizzare le chiamate ONTAPI (ZAPI). `useREST` è pienamente qualificato per NVMe/TCP. **NOTA** NVMe è supportato solo con le API REST di ONTAP e non è supportato con ONTAPI (ZAPI). **Se specificato, impostare sempre su `true` per i sistemi ASA r2**.	`true` per ONTAP 9.15.1 o versioni successive, altrimenti `false`.
`sanType`	Utilizzare per selezionare `iscsi` per iSCSI, `nvme` per NVMe/TCP o `fcp` per SCSI over Fibre Channel (FC).	`iscsi` se vuoto

Parametro	Descrizione	Predefinito
`formatOptions`	Usa `formatOptions` per specificare gli argomenti della riga di comando per il comando `mkfs`, che viene applicato ogni volta che un volume viene formattato. Questo ti permette di formattare il volume secondo le tue preferenze. Assicurati di specificare le formatOptions in modo simile alle opzioni del comando mkfs, escludendo il percorso del dispositivo. Esempio: "-E nodiscard" **Supportato per `ontap-san` e `ontap-san-economy` driver con protocollo iSCSI. Inoltre, supportato per sistemi ASA r2 quando si utilizzano i protocolli iSCSI e NVMe/TCP.**	
`limitVolumePoolSize`	Dimensione massima richiedibile FlexVol quando si usano LUN nel backend ontap-san-economy.	"" (non applicato per impostazione predefinita)
`denyNewVolumePools`	Restringe i backend dal `ontap-san-economy` creare nuovi volumi FlexVol per contenere le loro LUN. Solo i FlexVol preesistenti vengono utilizzati per il provisioning di nuovi PV.	

Raccomandazioni per l'utilizzo di formatOptions

Trident raccomanda le seguenti opzioni per accelerare il processo di formattazione:

- **-E nodiscard (ext3, ext4):** Non tentare di scartare i blocchi al momento di mkfs (scartare i blocchi inizialmente è utile sui dispositivi a stato solido e sullo storage sparse / con thin provisioning). Questo sostituisce l'opzione deprecata "-K" ed è applicabile ai file system ext3, ext4.
- **-K (xfs):** Non tentare di scartare i blocchi al momento di mkfs. Questa opzione è applicabile al file system xfs.

Autenticare Trident a un backend SVM utilizzando le credenziali di Active Directory

È possibile configurare Trident per autenticarsi a un backend SVM utilizzando le credenziali di Active Directory (AD). Prima che un account AD possa accedere all'SVM, è necessario configurare l'accesso del domain controller AD al cluster o all'SVM. Per l'amministrazione del cluster con un account AD, è necessario creare un domain tunnel. Per i dettagli, fai riferimento a "Configurare l'accesso del domain controller Active Directory in ONTAP".

passi

1. Configurare le impostazioni del Domain Name System (DNS) per una SVM di backend:

```
vserver services dns create -vserver <svm_name> -dns-servers
<dns_server_ip1>,<dns_server_ip2>
```

2. Eseguire il seguente comando per creare un account computer per la SVM in Active Directory:

```
vserver active-directory create -vserver DataSVM -account-name ADSERVER1
-domain demo.netapp.com
```

3. Utilizzare questo comando per creare un utente o un gruppo AD per gestire il cluster o SVM

```
security login create -vserver <svm_name> -user-or-group-name
<ad_user_or_group> -application <application> -authentication-method domain
-role vsadmin
```

4. Nel file di configurazione del backend Trident, impostare i parametri `username` e `password` rispettivamente sul nome utente o del gruppo AD e sulla password.

Opzioni di configurazione del backend per il provisioning dei volumi

È possibile controllare il provisioning predefinito utilizzando queste opzioni nella `defaults` sezione della configurazione. Per un esempio, vedere gli esempi di configurazione riportati di seguito.

Parametro	Descrizione	Predefinito
`spaceAllocation`	Allocazione dello spazio per le LUN	"true" **Se specificato, impostare su `true` per sistemi ASA r2**.
`spaceReserve`	Modalità di prenotazione dello spazio; "none" (con thin provisioning) o "volume" (con thick provisioning). **Impostare su `none` per sistemi ASA r2**.	"none"
`snapshotPolicy`	Policy di Snapshot da utilizzare. **Impostare su `none` per sistemi ASA r2**.	"none"
`qosPolicy`	Gruppo di policy QoS da assegnare ai volumi creati. Scegliere uno tra qosPolicy o adaptiveQosPolicy per pool di storage/backend. L'utilizzo di gruppi di policy QoS con Trident richiede ONTAP 9.8 o versioni successive. È consigliabile utilizzare un gruppo di policy QoS non condiviso e assicurarsi che il gruppo di policy venga applicato a ciascun componente singolarmente. Un gruppo di policy QoS condiviso impone il limite massimo per il throughput di tutti i carichi di lavoro.	""
`adaptiveQosPolicy`	Gruppo di policy QoS adattivo da assegnare ai volumi creati. Scegliere uno tra qosPolicy o adaptiveQosPolicy per storage pool/backend	""
`snapshotReserve`	Percentuale di volume riservata alle snapshot. **Non specificare per i sistemi ASA r2**.	"0" se `snapshotPolicy` è "none", altrimenti ""
`splitOnClone`	Dividere un clone dal suo genitore al momento della creazione	"false"
`encryption`	Abilita NetApp Volume Encryption (NVE) sul nuovo volume; il valore predefinito è `false`. NVE deve essere concesso in licenza e abilitato sul cluster per usare questa opzione. Se NAE è abilitato sul backend, qualsiasi volume effettuato il provisioning in Trident è abilitato per NAE. Per ulteriori informazioni, consulta: "Come funziona Trident con NVE e NAE".	"false" **Se specificato, impostare su `true` per i sistemi ASA r2**.
`luksEncryption`	Abilita la crittografia LUKS. Fai riferimento a "Usa Linux Unified Key Setup (LUKS)".	"" **Impostato su `false` per i sistemi ASA r2**.

Parametro	Descrizione	Predefinito
`tieringPolicy`	Criterio di tiering da utilizzare "none" **Non specificare per i sistemi ASA r2**.	
`nameTemplate`	Modello per creare nomi di volume personalizzati.	""

Esempi di provisioning del volume

Ecco un esempio con i valori predefiniti:

```
---
version: 1
storageDriverName: ontap-san
managementLIF: 10.0.0.1
svm: trident_svm
username: admin
password: <password>
labels:
  k8scluster: dev2
  backend: dev2-sanbackend
storagePrefix: alternate-trident
debugTraceFlags:
  api: false
  method: true
defaults:
  spaceReserve: volume
  qosPolicy: standard
  spaceAllocation: 'false'
  snapshotPolicy: default
  snapshotReserve: '10'
```

NOTA Per tutti i volumi creati utilizzando il `ontap-san` driver, Trident aggiunge un ulteriore 10% di capacità al FlexVol per ospitare i metadati del LUN. Il LUN viene predisposto con la dimensione esatta richiesta dall'utente nel PVC. Trident aggiunge il 10% al FlexVol (mostrato come Dimensione disponibile in ONTAP). Ora ottieni la quantità di capacità utilizzabile che hai richiesto. Questa modifica impedisce anche che i LUN diventino di sola lettura, a meno che lo spazio disponibile non sia completamente utilizzato. Questo non si applica a ontap-san-economy.

Per i backend che definiscono `snapshotReserve`, Trident calcola la dimensione dei volumi come segue:

```
Total volume size = [(PVC requested size) / (1 - (snapshotReserve
percentage) / 100)] * 1.1
```

L'1,1 è il 10 percento in più che Trident aggiunge a FlexVol per ospitare i metadati della LUN. Per `snapshotReserve` = 5% e PVC request = 5 GiB, la dimensione totale del volume è 5,79 GiB e la dimensione

disponibile è 5,5 GiB. Il comando `volume show` dovrebbe mostrare risultati simili a questo esempio:

Attualmente, il ridimensionamento è l'unico modo per utilizzare il nuovo calcolo per un volume esistente.

Esempi di configurazione minima

Gli esempi seguenti mostrano configurazioni di base che lasciano la maggior parte dei parametri ai valori predefiniti. Questo è il modo più semplice per definire un backend.

NOTA	Se si utilizza Amazon FSx su NetApp ONTAP con Trident, NetApp consiglia di specificare i nomi DNS per i LIF invece degli indirizzi IP.

▼ **Esempio SAN ONTAP**

Questa è una configurazione di base che utilizza il `ontap-san` driver.

```
---
version: 1
storageDriverName: ontap-san
managementLIF: 10.0.0.1
svm: svm_iscsi
labels:
  k8scluster: test-cluster-1
  backend: testcluster1-sanbackend
username: vsadmin
password: <password>
```

▼ **Esempio di MetroCluster**

È possibile configurare il backend per evitare di dover aggiornare manualmente la definizione del backend dopo lo switchover e lo switchback durante "Replica e recovery SVM".

Per un passaggio e un ritorno senza interruzioni, specificare l'SVM utilizzando `managementLIF` e omettere i `svm` parametri. Ad esempio:

```
version: 1
storageDriverName: ontap-san
managementLIF: 192.168.1.66
username: vsadmin
password: password
```

▼ **Esempio di economia SAN ONTAP**

```
version: 1
storageDriverName: ontap-san-economy
managementLIF: 10.0.0.1
svm: svm_iscsi_eco
username: vsadmin
password: <password>
```

▼ **Esempio di autenticazione basata su certificato**

In questo esempio di configurazione di base `clientCertificate`, `clientPrivateKey`, e `trustedCACertificate` (facoltativo, se si utilizza una CA attendibile) sono popolati in `backend.json` e accettano rispettivamente i valori codificati in base64 del certificato client, della chiave privata e del certificato CA attendibile.

```
---
version: 1
storageDriverName: ontap-san
backendName: DefaultSANBackend
managementLIF: 10.0.0.1
svm: svm_iscsi
useCHAP: true
chapInitiatorSecret: cl9qxIm36DKyawxy
chapTargetInitiatorSecret: rqxigXgkesIpwxyz
chapTargetUsername: iJF4heBRT0TCwxyz
chapUsername: uh2aNCLSd6cNwxyz
clientCertificate: ZXR0ZXJwYXB...ICMgJ3BhcGVyc2
clientPrivateKey: vciwKIyAgZG...0cnksIGRlc2NyaX
trustedCACertificate: zcyBbaG...b3Igb3duIGNsYXNz
```

▼ **Esempi di CHAP bidirezionale**

Questi esempi creano un backend con `useCHAP` impostato su `true`.

Esempio ONTAP SAN CHAP

```
---
version: 1
storageDriverName: ontap-san
managementLIF: 10.0.0.1
svm: svm_iscsi
labels:
  k8scluster: test-cluster-1
  backend: testcluster1-sanbackend
useCHAP: true
chapInitiatorSecret: cl9qxIm36DKyawxy
chapTargetInitiatorSecret: rqxigXgkesIpwxyz
```

```
chapTargetUsername: iJF4heBRT0TCwxyz
chapUsername: uh2aNCLSd6cNwxyz
username: vsadmin
password: <password>
```

Esempio di CHAP economy ONTAP SAN

```
---
version: 1
storageDriverName: ontap-san-economy
managementLIF: 10.0.0.1
svm: svm_iscsi_eco
useCHAP: true
chapInitiatorSecret: cl9qxIm36DKyawxy
chapTargetInitiatorSecret: rqxigXgkesIpwxyz
chapTargetUsername: iJF4heBRT0TCwxyz
chapUsername: uh2aNCLSd6cNwxyz
username: vsadmin
password: <password>
```

▼ **Esempio di posizionamento bilanciato**

Questo esempio abilita il posizionamento bilanciato di ONTAP, così che ONTAP selezioni l'aggregato per i nuovi volumi. Il posizionamento bilanciato richiede ONTAP 9.19.1 o versioni successive. Per i dettagli, fai riferimento a "Usa un posizionamento bilanciato con i backend ONTAP".

```
---
version: 1
storageDriverName: ontap-san
backendName: san-balanced-placement
managementLIF: 10.0.0.1
svm: svm_iscsi
username: vsadmin
password: <password>
useBalancedPlacement: true
```

▼ **Esempio NVMe/TCP**

È necessario disporre di una SVM configurata con NVMe sul backend ONTAP. Questa è una configurazione di base del backend per NVMe/TCP.

```
---
version: 1
backendName: NVMeBackend
storageDriverName: ontap-san
managementLIF: 10.0.0.1
```

```
svm: svm_nvme
username: vsadmin
password: password
sanType: nvme
useREST: true
```

▼ **Esempio di SCSI su FC (FCP)**

È necessario disporre di una SVM configurata con FC sul backend ONTAP. Questa è una configurazione di base del backend per FC.

```
---
version: 1
backendName: fcp-backend
storageDriverName: ontap-san
managementLIF: 10.0.0.1
svm: svm_fc
username: vsadmin
password: password
sanType: fcp
useREST: true
```

▼ **Esempio di configurazione del backend con nameTemplate**

```
---
version: 1
storageDriverName: ontap-san
backendName: ontap-san-backend
managementLIF: <ip address>
svm: svm0
username: <admin>
password: <password>
defaults:
  nameTemplate:
"{{.volume.Name}}_{{.labels.cluster}}_{{.volume.Namespace}}_{{.vo\
    lume.RequestName}}"
labels:
  cluster: ClusterA
  PVC: "{{.volume.Namespace}}_{{.volume.RequestName}}"
```

▼ **formatOptions example per il driver ontap-san-economy**

```
---
version: 1
storageDriverName: ontap-san-economy
managementLIF: ""
```

```
svm: svm1
username: ""
password: "!"
storagePrefix: whelk_
debugTraceFlags:
  method: true
  api: true
defaults:
  formatOptions: -E nodiscard
```

Esempi di backend con pool virtuali

In questi file di definizione backend di esempio, vengono impostati valori predefiniti specifici per tutti i pool di storage, come `spaceReserve` a none, `spaceAllocation` a false e `encryption` a false. I pool virtuali sono definiti nella sezione storage.

Trident imposta le etichette di provisioning nel campo "Commenti". I commenti vengono impostati sul FlexVol volume. Trident copia tutte le etichette presenti su un pool virtuale sul volume di storage al momento del provisioning. Per comodità, gli amministratori dello storage possono definire etichette per pool virtuale e raggruppare i volumi in base all'etichetta.

In questi esempi, alcuni pool di storage impostano i propri `spaceReserve`, `spaceAllocation` e `encryption` valori, mentre alcuni pool sovrascrivono i valori predefiniti.

▼ **Esempio SAN ONTAP**

```
---
version: 1
storageDriverName: ontap-san
managementLIF: 10.0.0.1
svm: svm_iscsi
useCHAP: true
chapInitiatorSecret: cl9qxIm36DKyawxy
chapTargetInitiatorSecret: rqxigXgkesIpwxyz
chapTargetUsername: iJF4heBRT0TCwxyz
chapUsername: uh2aNCLSd6cNwxyz
username: vsadmin
password: <password>
defaults:
  spaceAllocation: "false"
  encryption: "false"
  qosPolicy: standard
labels:
  store: san_store
  kubernetes-cluster: prod-cluster-1
region: us_east_1
storage:
  - labels:
```

```
      protection: gold
      creditpoints: "40000"
    zone: us_east_1a
    defaults:
      spaceAllocation: "true"
      encryption: "true"
      adaptiveQosPolicy: adaptive-extreme
  - labels:
      protection: silver
      creditpoints: "20000"
    zone: us_east_1b
    defaults:
      spaceAllocation: "false"
      encryption: "true"
      qosPolicy: premium
  - labels:
      protection: bronze
      creditpoints: "5000"
    zone: us_east_1c
    defaults:
      spaceAllocation: "true"
      encryption: "false"
```

▼ **Esempio di economia SAN ONTAP**

```
---
version: 1
storageDriverName: ontap-san-economy
managementLIF: 10.0.0.1
svm: svm_iscsi_eco
useCHAP: true
chapInitiatorSecret: cl9qxIm36DKyawxy
chapTargetInitiatorSecret: rqxigXgkesIpwxyz
chapTargetUsername: iJF4heBRT0TCwxyz
chapUsername: uh2aNCLSd6cNwxyz
username: vsadmin
password: <password>
defaults:
  spaceAllocation: "false"
  encryption: "false"
labels:
  store: san_economy_store
region: us_east_1
storage:
  - labels:
      app: oracledb
```

```
      cost: "30"
    zone: us_east_1a
    defaults:
      spaceAllocation: "true"
      encryption: "true"
  - labels:
      app: postgresdb
      cost: "20"
    zone: us_east_1b
    defaults:
      spaceAllocation: "false"
      encryption: "true"
  - labels:
      app: mysqldb
      cost: "10"
    zone: us_east_1c
    defaults:
      spaceAllocation: "true"
      encryption: "false"
  - labels:
      department: legal
      creditpoints: "5000"
    zone: us_east_1c
    defaults:
      spaceAllocation: "true"
      encryption: "false"
```

▼ **Esempio NVMe/TCP**

```
---
version: 1
storageDriverName: ontap-san
sanType: nvme
managementLIF: 10.0.0.1
svm: nvme_svm
username: vsadmin
password: <password>
useREST: true
defaults:
  spaceAllocation: "false"
  encryption: "true"
storage:
  - labels:
      app: testApp
      cost: "20"
    defaults:
```

```
      spaceAllocation: "false"
      encryption: "false"
```

Mappa i backend a StorageClasses

Le seguenti StorageClass definizioni si riferiscono a Esempi di backend con pool virtuali. Utilizzando il campo `parameters.selector`, ciascuna StorageClass indica quali pool virtuali puoi usare per ospitare un volume. Ora il volume ha le caratteristiche definite nel pool virtuale scelto.

- Il `protection-gold` StorageClass mappa il primo pool virtuale nel `ontap-san` backend. Questo è l'unico pool che offre protezione di livello oro.

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: protection-gold
provisioner: csi.trident.netapp.io
parameters:
  selector: "protection=gold"
  fsType: "ext4"
```

- La `protection-not-gold` StorageClass corrisponde al secondo e al terzo pool virtuale nel `ontap-san` backend. Questi sono gli unici pool che offrono un livello di protezione diverso da gold.

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: protection-not-gold
provisioner: csi.trident.netapp.io
parameters:
  selector: "protection!=gold"
  fsType: "ext4"
```

- La `app-mysqldb` StorageClass corrisponde al terzo pool virtuale nel `ontap-san-economy` backend. Questo è l'unico pool che offre la configurazione del pool di storage per le app di tipo mysqldb.

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: app-mysqldb
provisioner: csi.trident.netapp.io
parameters:
  selector: "app=mysqldb"
  fsType: "ext4"
```

- La `protection-silver-creditpoints-20k` StorageClass corrisponde al secondo pool virtuale nel `ontap-san` backend. Questo è l'unico pool che offre protezione di livello argento e 20000 punti credito.

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: protection-silver-creditpoints-20k
provisioner: csi.trident.netapp.io
parameters:
  selector: "protection=silver; creditpoints=20000"
  fsType: "ext4"
```

- La `creditpoints-5k` StorageClass corrisponde al terzo pool virtuale nel `ontap-san` backend e al quarto pool virtuale nel `ontap-san-economy` backend. Queste sono le uniche offerte di pool con 5000 creditpoints.

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: creditpoints-5k
provisioner: csi.trident.netapp.io
parameters:
  selector: "creditpoints=5000"
  fsType: "ext4"
```

- La `my-test-app-sc` StorageClass corrisponde al `testAPP` pool virtuale nel `ontap-san` driver con `sanType: nvme`. Questo è l'unico pool che offre `testApp`.

```
---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: my-test-app-sc
provisioner: csi.trident.netapp.io
parameters:
  selector: "app=testApp"
  fsType: "ext4"
```

Trident decide quale pool virtuale viene selezionato e garantisce che il requisito di storage sia soddisfatto.

Driver NAS ONTAP

Panoramica del driver ONTAP NAS

Scopri come configurare un backend ONTAP con i driver ONTAP e Cloud Volumes ONTAP NAS.

Dettagli del driver ONTAP NAS

Trident fornisce i seguenti driver di storage NAS per comunicare con il cluster ONTAP. Le modalità di accesso supportate sono: *ReadWriteOnce* (RWO), *ReadOnlyMany* (ROX), *ReadWriteMany* (RWX), *ReadWriteOncePod* (RWOP).

Driver	Protocollo	volumeMode	Modalità di accesso supportate	File system supportati
`ontap-nas`	NFS SMB	Filesystem	RWO, ROX, RWX, RWOP	"", `nfs`, `smb`
`ontap-nas-economy`	NFS SMB	Filesystem	RWO, ROX, RWX, RWOP	"", `nfs`, `smb`
`ontap-nas-flexgroup`	NFS SMB	Filesystem	RWO, ROX, RWX, RWOP	"", `nfs`, `smb`

ATTENZIONE

- Utilizzare `ontap-san-economy` solo se si prevede che il conteggio dell'utilizzo del volume persistente sia superiore a "limiti di volume ONTAP supportati".
- Utilizzare `ontap-nas-economy` solo se si prevede che il conteggio dell'utilizzo del volume persistente sia superiore a "limiti di volume ONTAP supportati" e il `ontap-san-economy` driver non può essere utilizzato.
- Non utilizzare `ontap-nas-economy` se si prevede la necessità di protezione dei dati, disaster recovery o mobilità.
- NetApp non consiglia di utilizzare l'autogrow di FlexVol in tutti i driver ONTAP, ad eccezione di ontap-san. Come soluzione alternativa, Trident supporta l'utilizzo della riserva di snapshot e ridimensiona di conseguenza i volumi FlexVol.

Permessi utente

Trident prevede di essere eseguito come amministratore ONTAP o SVM, in genere utilizzando l' `admin`utente del cluster o un `vsadmin`utente SVM, oppure un utente con un nome diverso che ha lo stesso ruolo.

Per le distribuzioni Amazon FSx for NetApp ONTAP, Trident prevede di essere eseguito come amministratore ONTAP o SVM, utilizzando l'utente del cluster `fsxadmin` o un `vsadmin`utente SVM, oppure un utente con un nome diverso che ha lo stesso ruolo. L' `fsxadmin`utente è un sostituto limitato per l'utente amministratore del cluster.

NOTA

Se si utilizza il `limitAggregateUsage` parametro, sono richieste le autorizzazioni di amministratore del cluster. Quando si utilizza Amazon FSx for NetApp ONTAP con Trident, il `limitAggregateUsage` parametro non funzionerà con gli account utente `vsadmin` e `fsxadmin`. L'operazione di configurazione non andrà a buon fine se si specifica questo parametro.

Sebbene sia possibile creare un ruolo più restrittivo all'interno di ONTAP che un driver Trident può utilizzare, lo sconsigliamo. La maggior parte delle nuove versioni di Trident richiamerà API aggiuntive di cui bisognerebbe tenere conto, rendendo gli aggiornamenti difficili e soggetti a errori.

Prepararsi a configurare un backend con i driver ONTAP NAS

Comprendere i requisiti, le opzioni di autenticazione e le policy di esportazione per la configurazione di un backend ONTAP con driver ONTAP NAS. A partire dalla versione 25.10, NetApp Trident supporta "NetApp AFX sistema storage". NetApp AFX storage systems differiscono dagli altri sistemi ONTAP (ASA, AFF e FAS) nell'implementazione del loro livello di storage. Nella configurazione del backend Trident non è necessario specificare che il sistema è AFX. Quando si seleziona `ontap-nas` come `storageDriverName`, Trident rileva automaticamente i sistemi AFX.

NOTA	Solo il `ontap-nas` driver (con protocollo NFS) è supportato per i sistemi AFX; il protocollo SMB non è supportato.

Requisiti

- Per tutti i backend ONTAP, Trident richiede che almeno un aggregato sia assegnato all'SVM.
- È possibile eseguire più di un driver e creare classi di archiviazione che puntano all'uno o all'altro. Ad esempio, si può configurare una classe Gold che utilizza il `ontap-nas` driver e una classe Bronze che utilizza il `ontap-nas-economy` driver.
- Su tutti i nodi worker di Kubernetes devono essere installati gli strumenti NFS appropriati. Per ulteriori dettagli, consulta "qui".
- Trident supporta volumi SMB montati solo su pod in esecuzione su nodi Windows. Per i dettagli, fai riferimento a Prepararsi al provisioning dei volumi SMB.

Autenticare il backend ONTAP

Trident offre due modalità di autenticazione per un backend ONTAP.

- Basato su credenziali: Questa modalità richiede autorizzazioni sufficienti al backend ONTAP. Si consiglia di utilizzare un account associato a un ruolo di login di sicurezza predefinito, come `admin` o `vsadmin` per garantire la massima compatibilità con le versioni ONTAP.
- Basato su certificato: Questa modalità richiede un certificato installato sul backend per Trident per comunicare con un cluster ONTAP. In questo caso, la definizione del backend deve contenere i valori codificati in Base64 del certificato client, della chiave e del certificato CA attendibile, se utilizzato (consigliato).

È possibile aggiornare i backend esistenti per passare tra metodi basati su credenziali e metodi basati su certificati. Tuttavia, è supportato solo un metodo di autenticazione alla volta. Per passare a un diverso metodo di autenticazione, è necessario rimuovere il metodo esistente dalla configurazione del backend.

ATTENZIONE	Se si tenta di fornire **sia le credenziali che i certificati**, la creazione del backend fallirà con un errore che indica che è stato fornito più di un metodo di autenticazione nel file di configurazione.

Abilitare l'autenticazione basata su credenziali

Trident richiede le credenziali di un amministratore SVM-scoped/cluster-scoped per comunicare con il backend ONTAP. Si consiglia di utilizzare ruoli standard, predefiniti come `admin` o `vsadmin`. Questo garantisce la compatibilità futura con le versioni ONTAP che potrebbero esporre API di funzionalità da utilizzare nelle future versioni di Trident. È possibile creare e utilizzare un ruolo di accesso di sicurezza personalizzato con Trident,

ma non è consigliato.

Un esempio di definizione di backend sarà simile al seguente:

YAML

```yaml
---
version: 1
backendName: ExampleBackend
storageDriverName: ontap-nas
managementLIF: 10.0.0.1
dataLIF: 10.0.0.2
svm: svm_nfs
credentials:
  name: secret-backend-creds
```

JSON

```json
{
  "version": 1,
  "backendName": "ExampleBackend",
  "storageDriverName": "ontap-nas",
  "managementLIF": "10.0.0.1",
  "dataLIF": "10.0.0.2",
  "svm": "svm_nfs",
  "credentials": {
        "name": "secret-backend-creds"
    }
}
```

Tenete presente che la definizione del backend è l'unico posto in cui le credenziali vengono archiviate in testo normale. Dopo la creazione del backend, nomi utente e password vengono codificati in Base64 e archiviati come segreti Kubernetes. La creazione/aggiornamento di un backend è l'unico passaggio che richiede la conoscenza delle credenziali. Pertanto, si tratta di un'operazione riservata all'amministratore, da eseguire dall'amministratore di Kubernetes/storage.

Abilita l'autenticazione basata su certificati

I backend nuovi ed esistenti possono utilizzare un certificato e comunicare con il backend ONTAP. Sono richiesti tre parametri nella definizione del backend.

- clientCertificate: Valore codificato in Base64 del certificato client.
- clientPrivateKey: Valore codificato in Base64 della chiave privata associata.
- trustedCACertificate: Valore codificato in Base64 del certificato della CA fidata. Se si utilizza una CA fidata, questo parametro deve essere fornito. Questo può essere ignorato se non si utilizza una CA fidata.

Un tipico flusso di lavoro prevede i seguenti passaggi.

Passaggi

1. Generare un certificato e una chiave client. Durante la generazione, impostare Common Name (CN) sull'utente ONTAP con cui autenticarsi.

```
openssl req -x509 -nodes -days 1095 -newkey rsa:2048 -keyout k8senv.key
-out k8senv.pem -subj "/C=US/ST=NC/L=RTP/O=NetApp/CN=vsadmin"
```

2. Aggiungere un certificato CA attendibile al cluster ONTAP. Questa operazione potrebbe essere già gestita dall'amministratore dello storage. Ignorare se non viene utilizzata una CA attendibile.

```
security certificate install -type server -cert-name <trusted-ca-cert-
name> -vserver <vserver-name>
ssl modify -vserver <vserver-name> -server-enabled true -client-enabled
true -common-name <common-name> -serial <SN-from-trusted-CA-cert> -ca
<cert-authority>
```

3. Installare il certificato e la chiave del client (dal punto 1) sul cluster ONTAP.

```
security certificate install -type client-ca -cert-name <certificate-
name> -vserver <vserver-name>
security ssl modify -vserver <vserver-name> -client-enabled true
```

4. Confermare che il ruolo di login di sicurezza ONTAP supporta `cert` il metodo di autenticazione.

```
security login create -user-or-group-name vsadmin -application ontapi
-authentication-method cert -vserver <vserver-name>
security login create -user-or-group-name vsadmin -application http
-authentication-method cert -vserver <vserver-name>
```

5. Verifica l'autenticazione utilizzando il certificato generato. Sostituisci <ONTAP Management LIF> e <vserver name> con l'IP LIF di gestione e il nome SVM. È necessario assicurarsi che il LIF abbia la sua service policy impostata su `default-data-management`.

```
curl -X POST -Lk https://<ONTAP-Management-
LIF>/servlets/netapp.servlets.admin.XMLrequest_filer --key k8senv.key
--cert ~/k8senv.pem -d '<?xml version="1.0" encoding="UTF-8"?><netapp
xmlns="http://www.netapp.com/filer/admin" version="1.21"
vfiler="<vserver-name>"><vserver-get></vserver-get></netapp>'
```

6. Codifica il certificato, la chiave e il certificato CA affidabile con Base64.

```
base64 -w 0 k8senv.pem >> cert_base64
base64 -w 0 k8senv.key >> key_base64
base64 -w 0 trustedca.pem >> trustedca_base64
```

7. Crea il backend utilizzando i valori ottenuti nel passaggio precedente.

```
cat cert-backend-updated.json
{
"version": 1,
"storageDriverName": "ontap-nas",
"backendName": "NasBackend",
"managementLIF": "1.2.3.4",
"dataLIF": "1.2.3.8",
"svm": "vserver_test",
"clientCertificate": "Faaaakkkkeeee...Vaaalllluuuueeee",
"clientPrivateKey": "LS0tFaKE...0VaLuES0tLS0K",
"storagePrefix": "myPrefix_"
}

#Update backend with tridentctl
tridentctl update backend NasBackend -f cert-backend-updated.json -n
trident
+------------+----------------+--------------------------------------
+--------+---------+
|    NAME    | STORAGE DRIVER |                 UUID                 |
STATE  | VOLUMES |
+------------+----------------+--------------------------------------
+--------+---------+
| NasBackend | ontap-nas      | 98e19b74-aec7-4a3d-8dcf-128e5033b214 |
online |       9 |
+------------+----------------+--------------------------------------
+--------+---------+
```

Aggiorna i metodi di autenticazione o ruota le credenziali

È possibile aggiornare un backend esistente per utilizzare un metodo di autenticazione diverso o per ruotare le credenziali. Questo funziona in entrambi i modi: i backend che utilizzano nome utente/password possono essere aggiornati per utilizzare certificati; i backend che utilizzano certificati possono essere aggiornati per utilizzare nome utente/password. Per fare ciò, è necessario rimuovere il metodo di autenticazione esistente e aggiungere il nuovo metodo di autenticazione. Quindi utilizzare il file backend.json aggiornato contenente i parametri richiesti per eseguire `tridentctl update backend`.

```
cat cert-backend-updated.json
```

```
{
"version": 1,
"storageDriverName": "ontap-nas",
"backendName": "NasBackend",
"managementLIF": "1.2.3.4",
"dataLIF": "1.2.3.8",
"svm": "vserver_test",
"username": "vsadmin",
"password": "password",
"storagePrefix": "myPrefix_"
}
```

```
#Update backend with tridentctl
tridentctl update backend NasBackend -f cert-backend-updated.json -n
trident
+------------+----------------+--------------------------------------
+--------+---------+
|    NAME    | STORAGE DRIVER |                 UUID                 |
STATE   | VOLUMES |
+------------+----------------+--------------------------------------
+--------+---------+
| NasBackend | ontap-nas      | 98e19b74-aec7-4a3d-8dcf-128e5033b214 |
online |       9 |
+------------+----------------+--------------------------------------
+--------+---------+
```

NOTA Quando si ruotano le password, l'amministratore dello storage deve prima aggiornare la password dell'utente su ONTAP. Questo è seguito da un aggiornamento del backend. Quando si ruotano i certificati, è possibile aggiungere più certificati all'utente. Il backend viene quindi aggiornato per utilizzare il nuovo certificato, dopo di che il vecchio certificato può essere eliminato dal cluster ONTAP.

L'aggiornamento di un backend non interrompe l'accesso ai volumi già creati, né influisce sulle connessioni al volume effettuate successivamente. Un aggiornamento del backend riuscito indica che Trident può comunicare con il backend ONTAP e gestire le future operazioni sui volumi.

Crea un ruolo personalizzato ONTAP per Trident

È possibile creare un ruolo di cluster ONTAP con privilegi minimi in modo da non dover utilizzare il ruolo di amministratore ONTAP per eseguire operazioni in Trident. Quando si include il nome utente in una configurazione backend di Trident, Trident utilizza il ruolo di cluster ONTAP creato per eseguire le operazioni.

Fare riferimento a "Generatore di ruoli personalizzati Trident" per ulteriori informazioni sulla creazione di ruoli personalizzati Trident.

Utilizzo di ONTAP CLI

1. Crea un nuovo ruolo utilizzando il seguente comando:

```
security login role create <role_name\> -cmddirname "command" -access all
–vserver <svm_name\>
```

2. Crea un nome utente per l'utente Trident:

```
security login create -username <user_name\> -application ontapi
-authmethod <password\> -role <name_of_role_in_step_1\> –vserver
<svm_name\> -comment "user_description"
```

3. Assegna il ruolo all'utente:

```
security login modify username <user_name\> –vserver <svm_name\> -role
<role_name\> -application ontapi -application console -authmethod
<password\>
```

Utilizzo di System Manager

Eseguire i seguenti passaggi in ONTAP System Manager:

1. **Crea un ruolo personalizzato**:
   a. Per creare un ruolo personalizzato a livello di cluster, selezionare **Cluster > Settings**.

      (Oppure) Per creare un ruolo personalizzato a livello SVM, selezionare **Archiviazione > Storage VM > `required SVM`> Impostazioni > Utenti e ruoli**.

   b. Selezionare l'icona della freccia (→) accanto a **Users and Roles**.
   c. Seleziona **+Add** in **Roles**.
   d. Definisci le regole per il ruolo e fai clic su **Save**.
2. **Mappa il ruolo all'utente Trident**: + Esegui i seguenti passaggi nella pagina **Utenti e ruoli**:
   a. Selezionare l'icona Aggiungi **+** sotto **Utenti**.
   b. Selezionare il nome utente richiesto e selezionare un ruolo nel menu a discesa per **Role**.
   c. Fare clic su **Save**.

Per maggiori informazioni, consultare le seguenti pagine:

- "Ruoli personalizzati per l'amministrazione di ONTAP" o "Definisci ruoli personalizzati"
- "Lavorare con ruoli e utenti"

Gestisci le policy di esportazione NFS

Trident utilizza le policy di esportazione NFS per controllare l'accesso ai volumi che fornisce.

Trident offre due opzioni quando si lavora con le policy di esportazione:

- Trident può gestire dinamicamente la policy di esportazione stessa; in questa modalità operativa, l'amministratore dello storage specifica un elenco di blocchi CIDR che rappresentano gli indirizzi IP ammissibili. Trident aggiunge automaticamente gli IP dei nodi che rientrano in questi intervalli alla policy di

esportazione al momento della pubblicazione. In alternativa, quando non vengono specificati CIDR, tutti gli IP unicast a livello globale trovati sul nodo a cui si sta pubblicando il volume verranno aggiunti alla policy di esportazione.

- Gli amministratori dello storage possono creare una policy di esportazione e aggiungere regole manualmente. Trident utilizza la policy di esportazione predefinita a meno che nella configurazione non venga specificato un nome diverso per la policy di esportazione.

Gestisci dinamicamente le policy di esportazione

Trident offre la possibilità di gestire dinamicamente le policy di esportazione per i backend ONTAP. Questo offre all'amministratore dello storage la possibilità di specificare uno spazio di indirizzi consentito per gli IP dei nodi worker, invece di definire manualmente regole esplicite. Questo semplifica notevolmente la gestione delle policy di esportazione; le modifiche alla policy di esportazione non richiedono più l'intervento manuale sul cluster di storage. Inoltre, questo aiuta a limitare l'accesso al cluster di storage solo ai nodi worker che stanno montando i volumi e hanno IP nell'intervallo specificato, supportando una gestione automatizzata e a grana fine.

NOTA	Non utilizzare il Network Address Translation (NAT) quando si usano le policy di esportazione dinamiche. Con il NAT, lo storage controller vede l'indirizzo NAT del frontend e non l'indirizzo IP effettivo dell'host, quindi l'accesso sarà negato quando non viene trovata alcuna corrispondenza nelle regole di esportazione.

Esempio

Ci sono due opzioni di configurazione che devono essere utilizzate. Ecco un esempio di definizione backend:

```
---
version: 1
storageDriverName: ontap-nas-economy
backendName: ontap_nas_auto_export
managementLIF: 192.168.0.135
svm: svm1
username: vsadmin
password: password
autoExportCIDRs:
  - 192.168.0.0/24
autoExportPolicy: true
```

NOTA	Quando si utilizza questa funzione, è necessario assicurarsi che la giunzione principale nell'SVM abbia una policy di esportazione precedentemente creata con una regola di esportazione che permetta il blocco CIDR del nodo (ad esempio la policy di esportazione predefinita). Seguire sempre la best practice raccomandata da NetApp di dedicare un SVM per Trident.

Ecco una spiegazione di come funziona questa funzione utilizzando l'esempio sopra:

- `autoExportPolicy` è impostato su `true`. Questo indica che Trident crea una policy di esportazione per ogni volume fornito con questo backend per la SVM `svm1` e gestisce l'aggiunta e l'eliminazione delle regole utilizzando blocchi di indirizzi `autoexportCIDRs`. Fino a quando un volume non è collegato a un

nodo, il volume utilizza una policy di esportazione vuota senza regole per impedire accessi indesiderati a quel volume. Quando un volume viene pubblicato su un nodo, Trident crea una policy di esportazione con lo stesso nome del qtree sottostante contenente l'IP del nodo all'interno del blocco CIDR specificato. Questi IP verranno anche aggiunti alla policy di esportazione utilizzata dal volume FlexVol padre

    - Ad esempio:
        - UUID backend 403b5326-8482-40db-96d0-d83fb3f4daec
        - `autoExportPolicy` impostato su `true`
        - prefisso storage `trident`
        - PVC UUID a79bcf5f-7b6d-4a40-9876-e2551f159c1c
        - Il qtree denominato trident_pvc_a79bcf5f_7b6d_4a40_9876_e2551f159c1c crea una policy di esportazione per il FlexVol denominato `trident-403b5326-8482-40db96d0-d83fb3f4daec`, una policy di esportazione per il qtree denominato `trident_pvc_a79bcf5f_7b6d_4a40_9876_e2551f159c1c` e una policy di esportazione vuota denominata `trident_empty` sull'SVM. Le regole per la policy di esportazione del FlexVol saranno un superset di tutte le regole contenute nelle policy di esportazione dei qtree. La policy di esportazione vuota sarà riutilizzata da tutti i volumi che non sono collegati.

- `autoExportCIDRs` contiene un elenco di blocchi di indirizzi. Questo campo è opzionale e il valore predefinito è ["0.0.0.0/0", "::/0"]. Se non è definito, Trident aggiunge tutti gli indirizzi unicast con ambito globale trovati sui nodi worker con pubblicazioni.

In questo esempio, viene fornito lo spazio di indirizzi `192.168.0.0/24`. Ciò indica che gli IP dei nodi Kubernetes che rientrano in questo intervallo di indirizzi con pubblicazioni saranno aggiunti alla export policy che Trident crea. Quando Trident registra un nodo su cui viene eseguito, recupera gli indirizzi IP del nodo e li controlla rispetto ai blocchi di indirizzi forniti in `autoExportCIDRs`. Al momento della pubblicazione, dopo aver filtrato gli IP, Trident crea le regole di export policy per gli IP client del nodo su cui sta pubblicando.

È possibile aggiornare `autoExportPolicy` e `autoExportCIDRs` per i backend dopo averli creati. È possibile aggiungere nuovi CIDR per un backend gestito automaticamente o eliminare i CIDR esistenti. Prestare attenzione quando si eliminano i CIDR per garantire che le connessioni esistenti non vengano interrotte. È anche possibile scegliere di disabilitare `autoExportPolicy` per un backend e tornare a una policy di esportazione creata manualmente. Questo richiederà l'impostazione del parametro `exportPolicy` nella configurazione del backend.

Dopo che Trident ha creato o aggiornato un backend, puoi verificare il backend utilizzando `tridentctl` o il corrispondente `tridentbackend` CRD:

```
./tridentctl get backends ontap_nas_auto_export -n trident -o yaml
items:
- backendUUID: 403b5326-8482-40db-96d0-d83fb3f4daec
  config:
    aggregate: ""
    autoExportCIDRs:
    - 192.168.0.0/24
    autoExportPolicy: true
    backendName: ontap_nas_auto_export
    chapInitiatorSecret: ""
    chapTargetInitiatorSecret: ""
```

```
    chapTargetUsername: ""
    chapUsername: ""
    dataLIF: 192.168.0.135
    debug: false
    debugTraceFlags: null
    defaults:
      encryption: "false"
      exportPolicy: <automatic>
      fileSystemType: ext4
```

Quando un nodo viene rimosso, Trident controlla tutte le policy di esportazione per rimuovere le regole di accesso corrispondenti al nodo. Rimuovendo questo IP del nodo dalle policy di esportazione dei backend gestiti, Trident impedisce i montaggi non autorizzati, a meno che questo IP non venga riutilizzato da un nuovo nodo nel cluster.

Per i backend già esistenti, l'aggiornamento del backend con `tridentctl update backend` garantisce che Trident gestisca automaticamente le policy di esportazione. Questo crea due nuove policy di esportazione denominate in base all'UUID del backend e al nome del qtree quando necessario. I volumi presenti sul backend utilizzeranno le nuove policy di esportazione dopo essere stati smontati e rimontati.

NOTA	L'eliminazione di un backend con policy di esportazione gestite automaticamente eliminerà la policy di esportazione creata dinamicamente. Se il backend viene ricreato, viene trattato come un nuovo backend e comporterà la creazione di una nuova policy di esportazione.

Se l'indirizzo IP di un nodo live viene aggiornato, è necessario riavviare il pod Trident sul nodo. Trident aggiornerà quindi la policy di esportazione per i backend che gestisce per riflettere questa modifica dell'IP.

Prepararsi al provisioning dei volumi SMB

Con una piccola preparazione aggiuntiva, è possibile eseguire il provisioning dei volumi SMB utilizzando `ontap-nas` driver.

ATTENZIONE	È necessario configurare entrambi i protocolli NFS e SMB/CIFS sull'SVM per creare un volume SMB `ontap-nas-economy` per i cluster ONTAP on-premises. La mancata configurazione di uno di questi protocolli causerà il fallimento della creazione del volume SMB.

NOTA	`autoExportPolicy` non è supportato per i volumi SMB.

Prima di iniziare

Per poter eseguire il provisioning dei volumi SMB, è necessario disporre di quanto segue.

- Un cluster Kubernetes con un nodo controller Linux e almeno un nodo worker Windows che esegue Windows Server 2022. Trident supporta volumi SMB montati solo su pod in esecuzione su nodi Windows.
- Almeno un secret di Trident contenente le credenziali di Active Directory. Per generare il secret `smbcreds`:

```
kubectl create secret generic smbcreds --from-literal username=user
```

```
--from-literal password='password'
```

- Un CSI proxy configurato come servizio Windows. Per configurare un `csi-proxy`, fare riferimento a "GitHub: CSI Proxy" o "GitHub: CSI Proxy per Windows" per i nodi Kubernetes in esecuzione su Windows.

Passaggi

1. Per ONTAP on-premises, puoi facoltativamente creare una condivisione SMB oppure Trident può crearne una per te.

   NOTA Le condivisioni SMB sono necessarie per Amazon FSx per ONTAP.

   È possibile creare le condivisioni SMB admin in due modi: utilizzando lo snap-in "Microsoft Management Console" Shared Folders o utilizzando la ONTAP CLI. Per creare le condivisioni SMB utilizzando la ONTAP CLI:

   a. Se necessario, crea la struttura del percorso della directory per la condivisione.

      Il `vserver cifs share create` comando controlla il percorso specificato nell'opzione -path durante la creazione della condivisione. Se il percorso specificato non esiste, il comando fallisce.

   b. Crea una condivisione SMB associata alla SVM specificata:

      ```
      vserver cifs share create -vserver vserver_name -share-name
      share_name -path path [-share-properties share_properties,...]
      [other_attributes] [-comment text]
      ```

   c. Verificare che la condivisione sia stata creata:

      ```
      vserver cifs share show -share-name share_name
      ```

      NOTA Consultare "Creare una condivisione SMB" per tutti i dettagli.

2. Quando si crea il backend, è necessario configurare quanto segue per specificare i volumi SMB. Per tutte le opzioni di configurazione del backend FSx per ONTAP, fare riferimento a "Opzioni ed esempi di configurazione di FSx per ONTAP".

Parametro	Descrizione	Esempio
`smbShare`	È possibile specificare uno dei seguenti valori: il nome di una condivisione SMB creata tramite Microsoft Management Console o ONTAP CLI; un nome per consentire a Trident di creare la condivisione SMB; oppure è possibile lasciare il parametro vuoto per impedire l'accesso condiviso ai volumi. Questo parametro è facoltativo per ONTAP on-premises. Questo parametro è obbligatorio per Amazon FSx for ONTAP backends e non può essere vuoto.	`smb-share`

Parametro	Descrizione	Esempio
`nasType`	**Deve essere impostato su `smb`.** Se nullo, il valore predefinito è `nfs`.	`smb`
`securityStyle`	Stile di sicurezza per i nuovi volumi. **Deve essere impostato su `ntfs` o `mixed` per i volumi SMB.**	`ntfs` o `mixed` per i volumi SMB
`unixPermissions`	Modalità per nuovi volumi. **Deve essere lasciato vuoto per i volumi SMB.**	""

Abilita SMB sicuro

A partire dalla release 25.06, NetApp Trident supporta il provisioning sicuro dei volumi SMB creati utilizzando `ontap-nas` e `ontap-nas-economy` backends. Quando è abilitato l'SMB sicuro, è possibile fornire un accesso controllato alle condivisioni SMB per gli utenti e i gruppi di utenti di Active Directory (AD) utilizzando le Access Control Lists (ACLs).

Punti da ricordare

- L'importazione `ontap-nas-economy` di volumi non è supportata.
- Sono supportati solo i cloni di sola lettura per i volumi `ontap-nas-economy`.
- Se Secure SMB è abilitato, Trident ignorerà la condivisione SMB menzionata nel backend.
- L'aggiornamento dell'annotazione PVC, dell'annotazione storage class e del campo backend non aggiorna l'ACL della condivisione SMB.
- Le ACL di condivisione SMB specificate nell'annotazione del PVC clone avranno la precedenza su quelle nel PVC di origine.
- Assicurati di fornire utenti AD validi quando abiliti SMB sicuro. Gli utenti non validi non verranno aggiunti all'ACL.
- Se si fornisce lo stesso utente AD nel backend, storage class e PVC con autorizzazioni diverse, la priorità delle autorizzazioni sarà: PVC, storage class e poi backend.
- Secure SMB è supportato per `ontap-nas` le importazioni di volumi gestiti e non è applicabile alle importazioni di volumi non gestiti.

Passaggi

1. Specificare adAdminUser in TridentBackendConfig come mostrato nel seguente esempio:

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-ontap
  namespace: trident
spec:
  version: 1
  storageDriverName: ontap-nas
  managementLIF: 10.193.176.x
  svm: svm0
  useREST: true
  defaults:
```

```
    adAdminUser: tridentADtest
  credentials:
    name: backend-tbc-ontap-invest-secret
```

2. Aggiungi l'annotazione nella storage class.

   Aggiungere l'annotazione `trident.netapp.io/smbShareAdUser` alla storage class per abilitare SMB sicuro senza errori. Il valore utente specificato per l'annotazione `trident.netapp.io/smbShareAdUser` deve essere lo stesso del nome utente specificato nel `smbcreds` secret. È possibile scegliere una delle seguenti opzioni per `smbShareAdUserPermission`: `full_control`, `change` o `read`. L'autorizzazione predefinita è `full_control`.

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontap-smb-sc
  annotations:
    trident.netapp.io/smbShareAdUserPermission: change
    trident.netapp.io/smbShareAdUser: tridentADuser
parameters:
  backendType: ontap-nas
  csi.storage.k8s.io/node-stage-secret-name: smbcreds
  csi.storage.k8s.io/node-stage-secret-namespace: trident
  trident.netapp.io/nasType: smb
provisioner: csi.trident.netapp.io
reclaimPolicy: Delete
volumeBindingMode: Immediate
```

1. Crea un PVC.

   Il seguente esempio crea un PVC:

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: my-pvc4
  namespace: trident
  annotations:
    trident.netapp.io/snapshotDirectory: "true"
    trident.netapp.io/smbShareAccessControl: |
      read:
        - tridentADtest
spec:
  accessModes:
    - ReadWriteOnce
```

```
  resources:
    requests:
      storage: 1Gi
  storageClassName: ontap-smb-sc
```

Opzioni ed esempi di configurazione NAS ONTAP

Crea e usa i driver NAS ONTAP con la tua installazione di Trident. Questa sezione fornisce esempi di configurazione del backend e dettagli per la mappatura dei backend a StorageClasses. A partire dalla versione 25.10, NetApp Trident supporta "Sistemi storage NetApp AFX". I sistemi storage NetApp AFX differiscono dagli altri sistemi basati su ONTAP (ASA, AFF e FAS) nell'implementazione del loro livello di storage.

NOTA Solo il `ontap-nas` driver (con protocollo NFS) è supportato per i sistemi NetApp AFX; il protocollo SMB non è supportato.

Opzioni di configurazione del backend

Nella configurazione del backend di Trident, non è necessario specificare che il sistema è un NetApp AFX storage system. Quando si seleziona `ontap-nas` come `storageDriverName`, Trident rileva automaticamente il sistema storage AFX. Alcuni parametri di configurazione del backend non sono applicabili ai sistemi storage AFX.

A partire dalla versione 26.06, puoi abilitare il posizionamento bilanciato di ONTAP così che ONTAP selezioni l'aggregato per i nuovi volumi. Il posizionamento bilanciato è richiesto per i sistemi storage AFX su ONTAP 9.19.1 o versioni successive. Consulta "Usa un posizionamento bilanciato con i backend ONTAP" per i requisiti e gli esempi.

La tabella seguente mostra le opzioni di configurazione del backend:

Parametro	Descrizione	Predefinito
`version`		Sempre 1
`storageDriverName`	Nome del driver di archiviazione. Non puoi modificarlo dopo la creazione. Per aggiornare questo parametro, devi creare un nuovo backend. **NOTA** Per i sistemi NetApp AFX, è supportato solo `ontap-nas`.	`ontap-nas`, `ontap-nas-economy`, o `ontap-nas-flexgroup`
`backendName`	Nome personalizzato o lo storage backend	Nome driver + "_" + dataLIF

Parametro	Descrizione	Predefinito
`managementLIF`	Indirizzo IP di un cluster o di una LIF di gestione SVM. È possibile specificare un fully-qualified domain name (FQDN). Può essere impostato per usare indirizzi IPv6 se hai installato Trident usando il flag IPv6. Gli indirizzi IPv6 devono essere definiti tra parentesi quadre, ad esempio `[28e8:d9fb:a825:b7bf:69a8:d02f:9e7b:3555]`. Per un passaggio senza interruzioni di MetroCluster, vedere Esempio di MetroCluster.	"10.0.0.1", "[2001:1234:abcd::fefe]"
`dataLIF`	Indirizzo IP del protocollo LIF. NetApp consiglia di specificare `dataLIF`. Se non fornito, Trident recupera i dataLIF dall'SVM. Puoi specificare un fully-qualified domain name (FQDN) da usare per le operazioni di mount NFS, permettendoti di creare un DNS round-robin per bilanciare il carico tra più dataLIF. Può essere modificato dopo l'impostazione iniziale. Fai riferimento a . Può essere impostato per usare indirizzi IPv6 se hai installato Trident usando il flag IPv6. Gli indirizzi IPv6 devono essere definiti tra parentesi quadre, ad esempio `[28e8:d9fb:a825:b7bf:69a8:d02f:9e7b:3555]`. **Omettere per MetroCluster.** Vedi il Esempio di MetroCluster.	Indirizzo specificato o derivato da SVM, se non specificato (non consigliato)
`svm`	Macchina virtuale di storage da utilizzare **Ometti per MetroCluster.** Vedi il Esempio di MetroCluster.	Derivato se viene specificato un SVM `managementLIF`
`autoExportPolicy`	Abilita la creazione e l'aggiornamento automatici delle policy di esportazione [Boolean]. Utilizzando le `autoExportPolicy` e `autoExportCIDRs` opzioni, Trident può gestire automaticamente le policy di esportazione.	falso
`autoExportCIDRs`	Elenco di CIDR in base ai quali filtrare gli IP dei nodi Kubernetes quando `autoExportPolicy` è abilitato. Utilizzando le `autoExportPolicy` e `autoExportCIDRs` opzioni, Trident può gestire automaticamente le policy di esportazione.	["0.0.0.0/0", "::/0"]`
`labels`	Set di etichette arbitrarie in formato JSON da applicare ai volumi	""
`clientCertificate`	Valore codificato in Base64 del certificato client. Utilizzato per l'autenticazione basata su certificato	""
`clientPrivateKey`	Valore codificato in Base64 della chiave privata del client. Utilizzato per l'autenticazione basata su certificato	""
`trustedCACertificate`	Valore codificato in Base64 del certificato CA attendibile. Opzionale. Utilizzato per l'autenticazione basata su certificato	""

Parametro	Descrizione	Predefinito
`username`	Nome utente per connettersi al cluster/SVM. Utilizzata per l'autenticazione basata su credenziali. Per l'autenticazione di Active Directory, vedere "Autenticare Trident a un backend SVM utilizzando le credenziali di Active Directory".	
`password`	Password per connettersi al cluster/SVM. Utilizzata per l'autenticazione basata su credenziali. Per l'autenticazione di Active Directory, vedere "Autenticare Trident a un backend SVM utilizzando le credenziali di Active Directory".	
`storagePrefix`	Prefisso utilizzato durante il provisioning di nuovi volumi nell'SVM. Non può essere aggiornato dopo averlo impostato **NOTA** Quando usi ontap-nas-economy e un storagePrefix di 24 o più caratteri, i qtree non hanno il prefisso di storage incluso, anche se è presente nel nome del volume.	"Trident"
`aggregate`	Aggregato per il provisioning (facoltativo; se impostato, deve essere assegnato all'SVM). Per il `ontap-nas-flexgroup` driver, questa opzione viene ignorata. Se non assegnato, è possibile utilizzare uno qualsiasi degli aggregati disponibili per il provisioning di un volume FlexGroup. **NOTA** Quando l'aggregato viene aggiornato in SVM, viene aggiornato automaticamente anche in Trident tramite polling di SVM, senza dover riavviare il Trident Controller. Quando hai configurato un aggregato specifico in Trident per il provisioning dei volumi, se l'aggregato viene rinominato o spostato fuori dalla SVM, il backend passa allo stato di errore in Trident durante il polling dell'aggregato SVM. Devi cambiare l'aggregato con uno presente su SVM oppure rimuoverlo del tutto per riportare online il backend. **Non specificare per i sistemi di storage AFX.** **Non specificare quando `useBalancedPlacement` è impostato su `true`.**	""

Parametro	Descrizione	Predefinito
`useBalancedPlacement`	Delega la selezione dell'aggregato per i nuovi volumi a ONTAP [Boolean]. Richiede ONTAP 9.19.1 o versioni successive e `useREST` impostato su `true`. Non può essere combinato con `aggregate`. Consulta "Usa un posizionamento bilanciato con i backend ONTAP" per i requisiti e gli esempi. **Se specificato, impostalo sempre su `true` per i sistemi storage AFX su ONTAP 9.19.1 o versioni successive.**	`false`; `true` per sistemi di storage AFX su ONTAP 9.19.1 o versioni successive
`limitAggregateUsage`	Il provisioning fallisce se l'utilizzo supera questa percentuale. **Non si applica ad Amazon FSx per ONTAP**. **Non specificare per i sistemi di storage AFX.**	"" (non applicato per impostazione predefinita)
flexgroupAggregateList	Elenco di aggregati per il provisioning (facoltativo; se impostato, deve essere assegnato alla SVM). Tutti gli aggregati assegnati alla SVM vengono utilizzati per il provisioning di un FlexGroup volume. Supportato per il driver di storage **ontap-nas-flexgroup**. **NOTA** Quando l'elenco degli aggregati viene aggiornato in SVM, l'elenco viene aggiornato automaticamente in Trident tramite polling di SVM, senza dover riavviare il Trident Controller. Quando hai configurato un elenco specifico di aggregati in Trident per il provisioning dei volumi, se l'elenco degli aggregati viene rinominato o spostato fuori da SVM, il backend passa allo stato di errore in Trident durante il polling dell'aggregato SVM. Devi cambiare l'elenco degli aggregati con uno presente su SVM oppure rimuoverlo del tutto per riportare online il backend.	""
`limitVolumeSize`	Il provisioning fallisce se la dimensione del volume richiesto supera questo valore.	"" (non applicato per impostazione predefinita)
`debugTraceFlags`	Flag di debug da usare per la risoluzione dei problemi. Ad esempio, {"api":false, "method":true} non utilizzare `debugTraceFlags` a meno che non si stia risolvendo un problema e si richieda un dump dettagliato del log.	null
`nasType`	Configura la creazione di volumi NFS o SMB. Le opzioni sono `nfs`, `smb` o null. Impostando su null, vengono creati di default volumi NFS. **Se specificato, impostare sempre su `nfs` per i sistemi di storage AFX**.	`nfs`

Parametro	Descrizione	Predefinito
`nfsMountOptions`	Elenco separato da virgole delle opzioni di montaggio NFS. Le opzioni di montaggio per i volumi persistenti di Kubernetes sono normalmente specificate nelle classi di storage, ma se non vengono specificate opzioni di montaggio in una classe di storage, Trident utilizza le opzioni di montaggio specificate nel file di configurazione del backend di storage. Se non vengono specificate opzioni di montaggio nella classe di storage o nel file di configurazione, Trident non imposta alcuna opzione di montaggio su un volume persistente associato.	""
`qtreesPerFlexvol`	Numero massimo di Qtree per FlexVol, deve essere nell'intervallo [50, 300]	"200"
`smbShare`	È possibile specificare uno dei seguenti valori: il nome di una condivisione SMB creata tramite Microsoft Management Console o ONTAP CLI; un nome per consentire a Trident di creare la condivisione SMB; oppure è possibile lasciare il parametro vuoto per impedire l'accesso condiviso ai volumi. Questo parametro è facoltativo per ONTAP on-premises. Questo parametro è obbligatorio per Amazon FSx for ONTAP backends e non può essere vuoto.	`smb-share`
`useREST`	Parametro booleano per utilizzare le API REST di ONTAP. `useREST` Quando impostato su `true`, Trident utilizza le ONTAP REST APIs per comunicare con il backend; quando impostato su `false`, Trident utilizza le chiamate ONTAPI (ZAPI) per comunicare con il backend. Questa funzione richiede ONTAP 9.11.1 e versioni successive. Inoltre, il ruolo di login ONTAP utilizzato deve avere accesso all'applicazione `ontapi`. Questo è soddisfatto dai ruoli predefiniti `vsadmin` e `cluster-admin`. A partire dalla release Trident 24.06 e ONTAP 9.15.1 o versioni successive, `useREST` è impostato su `true` per impostazione predefinita; modificare `useREST` su `false` per utilizzare le chiamate ONTAPI (ZAPI). **Se specificato, impostare sempre su `true` per i sistemi di storage AFX**.	`true` per ONTAP 9.15.1 o versioni successive, altrimenti `false`.
`limitVolumePoolSize`	Dimensione massima richiedibile FlexVol quando si utilizzano Qtrees nel backend ontap-nas-economy.	"" (non applicato per impostazione predefinita)
`denyNewVolumePools`	Restringe `ontap-nas-economy` i backend dal creare nuovi volumi FlexVol per contenere i loro Qtree. Solo i FlexVol preesistenti vengono utilizzati per il provisioning di nuovi PV.	

Parametro	Descrizione	Predefinito
`adAdminUser`	Utente o gruppo di utenti amministratore di Active Directory con accesso completo alle condivisioni SMB. Utilizzare questo parametro per fornire diritti di amministratore alla condivisione SMB con controllo completo.	

Opzioni di configurazione del backend per il provisioning dei volumi

È possibile controllare il provisioning predefinito utilizzando queste opzioni nella `defaults` sezione della configurazione. Per un esempio, vedere gli esempi di configurazione riportati di seguito.

Parametro	Descrizione	Predefinito
`spaceAllocation`	Allocazione dello spazio per Qtrees	"true"
`spaceReserve`	Modalità di prenotazione dello spazio; "none" (thin) o "volume" (thick)	"none"
`snapshotPolicy`	policy di Snapshot da utilizzare	"none"
`qosPolicy`	Gruppo di policy QoS da assegnare ai volumi creati. Scegliere uno tra qosPolicy o adaptiveQosPolicy per storage pool/backend	""
`adaptiveQosPolicy`	Gruppo di policy QoS adattivo da assegnare ai volumi creati. Scegliere uno tra qosPolicy o adaptiveQosPolicy per pool di storage/backend. Non supportato da ontap-nas-economy.	""
`snapshotReserve`	Percentuale di volume riservata alle snapshot	"0" se `snapshotPolicy` è "none", altrimenti ""
`splitOnClone`	Dividere un clone dal suo genitore al momento della creazione	"false"
`encryption`	Abilita NetApp Volume Encryption (NVE) sul nuovo volume; il valore predefinito è `false`. NVE deve essere concesso in licenza e abilitato sul cluster per usare questa opzione. Se NAE è abilitato sul backend, qualsiasi volume effettuato il provisioning in Trident è abilitato per NAE. Per ulteriori informazioni, consulta: "Come funziona Trident con NVE e NAE".	"false"
`tieringPolicy`	Policy di tiering da utilizzare "none"	
`unixPermissions`	Modalità per nuovi volumi	"777" per i volumi NFS; vuoto (non applicabile) per i volumi SMB
`snapshotDir`	Controlla l'accesso alla `.snapshot` directory	`true`, `false` (Impostato esplicitamente).
`exportPolicy`	Policy di esportazione da utilizzare	"default"

Parametro	Descrizione	Predefinito
`securityStyle`	Stile di sicurezza per i nuovi volumi. NFS supporta `mixed` e `unix` stili di sicurezza. SMB supporta `mixed` e `ntfs` stili di sicurezza.	L'impostazione predefinita di NFS è `unix`. L'impostazione predefinita di SMB è `ntfs`.
`nameTemplate`	Modello per creare nomi di volume personalizzati.	""

NOTA L'utilizzo di gruppi di policy QoS con Trident richiede ONTAP 9.8 o versioni successive. Dovresti utilizzare un gruppo di policy QoS non condiviso e assicurarti che il gruppo di policy venga applicato a ciascun componente individualmente. Un gruppo di policy QoS condiviso impone il limite massimo per il throughput di tutti i carichi di lavoro.

Esempi di provisioning del volume

Ecco un esempio con i valori predefiniti:

```
---
version: 1
storageDriverName: ontap-nas
backendName: customBackendName
managementLIF: 10.0.0.1
dataLIF: 10.0.0.2
labels:
  k8scluster: dev1
  backend: dev1-nasbackend
svm: trident_svm
username: cluster-admin
password: <password>
limitAggregateUsage: 80%
limitVolumeSize: 50Gi
nfsMountOptions: nfsvers=4
debugTraceFlags:
  api: false
  method: true
defaults:
  spaceReserve: volume
  qosPolicy: premium
  exportPolicy: myk8scluster
  snapshotPolicy: default
  snapshotReserve: "10"
```

Per `ontap-nas` e `ontap-nas-flexgroups`, Trident ora utilizza un nuovo calcolo per garantire che il FlexVol sia dimensionato correttamente con la percentuale di snapshotReserve e il PVC. Quando l'utente richiede un PVC, Trident crea il FlexVol originale con più spazio utilizzando il nuovo calcolo. Questo calcolo garantisce che l'utente riceva lo spazio scrivibile richiesto nel PVC e non meno spazio di quanto richiesto. Prima della v21.07, quando l'utente richiedeva un PVC (ad esempio, 5 GiB), con la snapshotReserve al 50 percento, otteneva solo 2,5 GiB di spazio scrivibile. Questo perché ciò che l'utente richiedeva era l'intero volume e

`snapshotReserve` era una percentuale di quello. Con Trident 21.07, ciò che l'utente richiede è lo spazio scrivibile e Trident definisce il numero di `snapshotReserve` come percentuale dell'intero volume. Questo non si applica a `ontap-nas-economy`. Vedi il seguente esempio per capire come funziona:

Il calcolo è il seguente:

```
Total volume size = <PVC requested size> / (1 - (<snapshotReserve
percentage> / 100))
```

Per snapshotReserve = 50% e richiesta PVC = 5 GiB, la dimensione totale del volume è 5/0,5 = 10 GiB e la dimensione disponibile è 5 GiB, che è quanto richiesto dall'utente nella richiesta PVC. Il comando `volume show` dovrebbe mostrare risultati simili a questo esempio:

Per i backend esistenti da installazioni precedenti, esegui il provisioning dei volumi come spiegato sopra durante l'aggiornamento di Trident. Per i volumi che hai creato prima dell'aggiornamento, dovresti ridimensionare i volumi affinché la modifica venga rilevata. Ad esempio, un PVC da 2 GiB con `snapshotReserve=50` in precedenza generava un volume che forniva 1 GiB di spazio scrivibile. Ridimensionando il volume a 3 GiB, ad esempio, l'applicazione avrà a disposizione 3 GiB di spazio scrivibile su un volume da 6 GiB.

Esempi di configurazione minima

Gli esempi seguenti mostrano configurazioni di base che lasciano la maggior parte dei parametri ai valori predefiniti. Questo è il modo più semplice per definire un backend.

NOTA	Se si utilizza Amazon FSx su NetApp ONTAP con Trident, si consiglia di specificare i nomi DNS per i LIF anziché gli indirizzi IP.

▼ Esempio di economia NAS ONTAP

```
---
version: 1
storageDriverName: ontap-nas-economy
managementLIF: 10.0.0.1
dataLIF: 10.0.0.2
svm: svm_nfs
username: vsadmin
password: password
```

▼ Esempio di ONTAP NAS FlexGroup

```
---
```

```
version: 1
storageDriverName: ontap-nas-flexgroup
managementLIF: 10.0.0.1
dataLIF: 10.0.0.2
svm: svm_nfs
username: vsadmin
password: password
```

▼ **Esempio di MetroCluster**

È possibile configurare il backend per evitare di dover aggiornare manualmente la definizione del backend dopo lo switchover e lo switchback durante "Replica e recovery SVM".

Per un passaggio e un ritorno senza interruzioni, specificare l'SVM utilizzando `managementLIF` e omettere i parametri `dataLIF` e `svm`. Ad esempio:

```
---
version: 1
storageDriverName: ontap-nas
managementLIF: 192.168.1.66
username: vsadmin
password: password
```

▼ **Esempio di volumi SMB**

```
---
version: 1
backendName: ExampleBackend
storageDriverName: ontap-nas
managementLIF: 10.0.0.1
nasType: smb
securityStyle: ntfs
unixPermissions: ""
dataLIF: 10.0.0.2
svm: svm_nfs
username: vsadmin
password: password
```

▼ **Esempio di autenticazione basata su certificato**

Questo è un esempio di configurazione minima del backend. `clientCertificate`, `clientPrivateKey`, e `trustedCACertificate` (facoltativo, se si utilizza una CA attendibile) vengono popolati in `backend.json` e accettano rispettivamente i valori codificati in base64 del certificato client, della chiave privata e del certificato CA attendibile.

```
---
version: 1
```

```
backendName: DefaultNASBackend
storageDriverName: ontap-nas
managementLIF: 10.0.0.1
dataLIF: 10.0.0.15
svm: nfs_svm
clientCertificate: ZXR0ZXJwYXB...ICMgJ3BhcGVyc2
clientPrivateKey: vciwKIyAgZG...0cnksIGRlc2NyaX
trustedCACertificate: zcyBbaG...b3Igb3duIGNsYXNz
storagePrefix: myPrefix_
```

▼ **Esempio di policy di esportazione automatica**

Questo esempio mostra come istruire Trident a utilizzare policy di esportazione dinamiche per creare e gestire automaticamente la policy di esportazione. Questo funziona allo stesso modo per i driver `ontap-nas-economy` e `ontap-nas-flexgroup`.

```
---
version: 1
storageDriverName: ontap-nas
managementLIF: 10.0.0.1
dataLIF: 10.0.0.2
svm: svm_nfs
labels:
  k8scluster: test-cluster-east-1a
  backend: test1-nasbackend
autoExportPolicy: true
autoExportCIDRs:
- 10.0.0.0/24
username: admin
password: password
nfsMountOptions: nfsvers=4
```

▼ **Esempio di posizionamento bilanciato**

Questo esempio abilita il posizionamento bilanciato di ONTAP, così che ONTAP selezioni l'aggregato per i nuovi volumi. Il posizionamento bilanciato richiede ONTAP 9.19.1 o versioni successive. Per i dettagli, fai riferimento a "Usa un posizionamento bilanciato con i backend ONTAP".

```
---
version: 1
storageDriverName: ontap-nas
backendName: nas-balanced-placement
managementLIF: 10.0.0.1
dataLIF: 10.0.0.2
svm: svm_nfs
username: vsadmin
password: password
```

```
useBalancedPlacement: true
```

▼ **Esempio di indirizzi IPv6**

Questo esempio mostra `managementLIF` l'utilizzo di un indirizzo IPv6.

```
---
version: 1
storageDriverName: ontap-nas
backendName: nas_ipv6_backend
managementLIF: "[5c5d:5edf:8f:7657:bef8:109b:1b41:d491]"
labels:
  k8scluster: test-cluster-east-1a
  backend: test1-ontap-ipv6
svm: nas_ipv6_svm
username: vsadmin
password: password
```

▼ **Esempio di Amazon FSx for ONTAP che utilizza volumi SMB**

Il `smbShare` parametro è obbligatorio per Amazon FSx for ONTAP che utilizza volumi SMB.

```
---
version: 1
backendName: SMBBackend
storageDriverName: ontap-nas
managementLIF: example.mgmt.fqdn.aws.com
nasType: smb
dataLIF: 10.0.0.15
svm: nfs_svm
smbShare: smb-share
clientCertificate: ZXR0ZXJwYXB...ICMgJ3BhcGVyc2
clientPrivateKey: vciwKIyAgZG...0cnksIGRlc2NyaX
trustedCACertificate: zcyBbaG...b3Igb3duIGNsYXNz
storagePrefix: myPrefix_
```

▼ **Esempio di configurazione del backend con nameTemplate**

```
---
version: 1
storageDriverName: ontap-nas
backendName: ontap-nas-backend
managementLIF: <ip address>
svm: svm0
username: <admin>
password: <password>
```

```
defaults:
  nameTemplate:
"{{.volume.Name}}_{{.labels.cluster}}_{{.volume.Namespace}}_{{.vo\
    lume.RequestName}}"
labels:
  cluster: ClusterA
  PVC: "{{.volume.Namespace}}_{{.volume.RequestName}}"
```

Esempi di backend con pool virtuali

Nei file di definizione del backend di esempio mostrati di seguito, vengono impostati valori predefiniti specifici per tutti i pool di storage, come `spaceReserve` a none, `spaceAllocation` a false e `encryption` a false. I pool virtuali sono definiti nella sezione storage.

Trident imposta le etichette di provisioning nel campo "Commenti". I commenti sono impostati su FlexVol per `ontap-nas` o FlexGroup per `ontap-nas-flexgroup`. Trident copia tutte le etichette presenti su un pool virtuale sul volume di storage al momento del provisioning. Per comodità, gli amministratori dello storage possono definire etichette per pool virtuale e raggruppare i volumi in base all'etichetta.

In questi esempi, alcuni pool di storage impostano i propri `spaceReserve`, `spaceAllocation` e `encryption` valori, mentre alcuni pool sovrascrivono i valori predefiniti.

▼ **Esempio NAS ONTAP**

```
---
version: 1
storageDriverName: ontap-nas
managementLIF: 10.0.0.1
svm: svm_nfs
username: admin
password: <password>
nfsMountOptions: nfsvers=4
defaults:
  spaceReserve: none
  encryption: "false"
  qosPolicy: standard
labels:
  store: nas_store
  k8scluster: prod-cluster-1
region: us_east_1
storage:
  - labels:
      app: msoffice
      cost: "100"
    zone: us_east_1a
    defaults:
      spaceReserve: volume
      encryption: "true"
```

```
      unixPermissions: "0755"
      adaptiveQosPolicy: adaptive-premium
  - labels:
      app: slack
      cost: "75"
    zone: us_east_1b
    defaults:
      spaceReserve: none
      encryption: "true"
      unixPermissions: "0755"
  - labels:
      department: legal
      creditpoints: "5000"
    zone: us_east_1b
    defaults:
      spaceReserve: none
      encryption: "true"
      unixPermissions: "0755"
  - labels:
      app: wordpress
      cost: "50"
    zone: us_east_1c
    defaults:
      spaceReserve: none
      encryption: "true"
      unixPermissions: "0775"
  - labels:
      app: mysqldb
      cost: "25"
    zone: us_east_1d
    defaults:
      spaceReserve: volume
      encryption: "false"
      unixPermissions: "0775"
```

▼ **Esempio di ONTAP NAS FlexGroup**

```
---
version: 1
storageDriverName: ontap-nas-flexgroup
managementLIF: 10.0.0.1
svm: svm_nfs
username: vsadmin
password: <password>
defaults:
  spaceReserve: none
```

```
  encryption: "false"
labels:
  store: flexgroup_store
  k8scluster: prod-cluster-1
region: us_east_1
storage:
  - labels:
      protection: gold
      creditpoints: "50000"
    zone: us_east_1a
    defaults:
      spaceReserve: volume
      encryption: "true"
      unixPermissions: "0755"
  - labels:
      protection: gold
      creditpoints: "30000"
    zone: us_east_1b
    defaults:
      spaceReserve: none
      encryption: "true"
      unixPermissions: "0755"
  - labels:
      protection: silver
      creditpoints: "20000"
    zone: us_east_1c
    defaults:
      spaceReserve: none
      encryption: "true"
      unixPermissions: "0775"
  - labels:
      protection: bronze
      creditpoints: "10000"
    zone: us_east_1d
    defaults:
      spaceReserve: volume
      encryption: "false"
      unixPermissions: "0775"
```

▼ **Esempio di economia NAS ONTAP**

```
---
version: 1
storageDriverName: ontap-nas-economy
managementLIF: 10.0.0.1
svm: svm_nfs
```

```
username: vsadmin
password: <password>
defaults:
  spaceReserve: none
  encryption: "false"
labels:
  store: nas_economy_store
region: us_east_1
storage:
  - labels:
      department: finance
      creditpoints: "6000"
    zone: us_east_1a
    defaults:
      spaceReserve: volume
      encryption: "true"
      unixPermissions: "0755"
  - labels:
      protection: bronze
      creditpoints: "5000"
    zone: us_east_1b
    defaults:
      spaceReserve: none
      encryption: "true"
      unixPermissions: "0755"
  - labels:
      department: engineering
      creditpoints: "3000"
    zone: us_east_1c
    defaults:
      spaceReserve: none
      encryption: "true"
      unixPermissions: "0775"
  - labels:
      department: humanresource
      creditpoints: "2000"
    zone: us_east_1d
    defaults:
      spaceReserve: volume
      encryption: "false"
      unixPermissions: "0775"
```

Mappa i backend a StorageClasses

Le seguenti definizioni di StorageClass si riferiscono a Esempi di backend con pool virtuali. Utilizzando il campo `parameters.selector`, ciascuna StorageClass indica quali pool virtuali puoi usare per ospitare un

volume. Il volume ha le caratteristiche definite nel pool virtuale scelto.

- La `protection-gold` StorageClass corrisponde al primo e al secondo pool virtuale nel `ontap-nas-flexgroup` backend. Questi sono gli unici pool che offrono la protezione di livello oro.

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: protection-gold
provisioner: csi.trident.netapp.io
parameters:
  selector: "protection=gold"
  fsType: "ext4"
```

- La `protection-not-gold` StorageClass corrisponde al terzo e al quarto pool virtuale nel `ontap-nas-flexgroup` backend. Questi sono gli unici pool che offrono un livello di protezione diverso da gold.

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: protection-not-gold
provisioner: csi.trident.netapp.io
parameters:
  selector: "protection!=gold"
  fsType: "ext4"
```

- La `app-mysqldb` StorageClass corrisponde al quarto pool virtuale nel `ontap-nas` backend. Questo è l'unico pool che offre la configurazione del pool di storage per app di tipo mysqldb.

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: app-mysqldb
provisioner: csi.trident.netapp.io
parameters:
  selector: "app=mysqldb"
  fsType: "ext4"
```

- La `protection-silver-creditpoints-20k` StorageClass corrisponde al terzo pool virtuale nel `ontap-nas-flexgroup` backend. Questo è l'unico pool che offre protezione di livello argento e 20000 punti credito.

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
```

```
metadata:
  name: protection-silver-creditpoints-20k
provisioner: csi.trident.netapp.io
parameters:
  selector: "protection=silver; creditpoints=20000"
  fsType: "ext4"
```

- La `creditpoints-5k` StorageClass corrisponde al terzo pool virtuale nel `ontap-nas` backend e al secondo pool virtuale nel `ontap-nas-economy` backend. Queste sono le uniche offerte di pool con 5000 creditpoints.

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: creditpoints-5k
provisioner: csi.trident.netapp.io
parameters:
  selector: "creditpoints=5000"
  fsType: "ext4"
```

Trident decide quale pool virtuale viene selezionato e garantisce che il requisito di storage sia soddisfatto.

Aggiorna `dataLIF` dopo la configurazione iniziale

È possibile modificare il dataLIF dopo la configurazione iniziale eseguendo il comando seguente per fornire il nuovo file JSON backend con il dataLIF aggiornato.

```
tridentctl update backend <backend-name> -f <path-to-backend-json-file-
with-updated-dataLIF>
```

NOTA Se i PVC sono collegati a uno o più pod, è necessario disattivare tutti i pod corrispondenti e quindi riattivarli affinché il nuovo dataLIF abbia effetto.

Esempi di SMB sicuro

Configurazione del backend con driver ontap-nas

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-ontap-nas
  namespace: trident
spec:
  version: 1
  storageDriverName: ontap-nas
```

```
  managementLIF: 10.0.0.1
  svm: svm2
  nasType: smb
  defaults:
    adAdminUser: tridentADtest
  credentials:
    name: backend-tbc-ontap-invest-secret
```

Configurazione backend con driver ontap-nas-economy

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-ontap-nas
  namespace: trident
spec:
  version: 1
  storageDriverName: ontap-nas-economy
  managementLIF: 10.0.0.1
  svm: svm2
  nasType: smb
  defaults:
    adAdminUser: tridentADtest
  credentials:
    name: backend-tbc-ontap-invest-secret
```

Configurazione backend con pool di storage

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-ontap-nas
  namespace: trident
spec:
  version: 1
  storageDriverName: ontap-nas
  managementLIF: 10.0.0.1
  svm: svm0
  useREST: false
  storage:
  - labels:
      app: msoffice
    defaults:
      adAdminUser: tridentADuser
```

```
  nasType: smb
  credentials:
    name: backend-tbc-ontap-invest-secret
```

Esempio di storage class con driver ontap-nas

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontap-smb-sc
  annotations:
    trident.netapp.io/smbShareAdUserPermission: change
    trident.netapp.io/smbShareAdUser: tridentADtest
parameters:
  backendType: ontap-nas
  csi.storage.k8s.io/node-stage-secret-name: smbcreds
  csi.storage.k8s.io/node-stage-secret-namespace: trident
  trident.netapp.io/nasType: smb
provisioner: csi.trident.netapp.io
reclaimPolicy: Delete
volumeBindingMode: Immediate
```

NOTA Assicurati di aggiungere `annotations` per abilitare SMB sicuro. SMB sicuro non funziona senza le annotazioni, indipendentemente dalle configurazioni impostate nel Backend o nel PVC.

Esempio di storage class con driver ontap-nas-economy

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontap-smb-sc
  annotations:
    trident.netapp.io/smbShareAdUserPermission: change
    trident.netapp.io/smbShareAdUser: tridentADuser3
parameters:
  backendType: ontap-nas-economy
  csi.storage.k8s.io/node-stage-secret-name: smbcreds
  csi.storage.k8s.io/node-stage-secret-namespace: trident
  trident.netapp.io/nasType: smb
provisioner: csi.trident.netapp.io
reclaimPolicy: Delete
volumeBindingMode: Immediate
```

Esempio di PVC con un singolo utente AD

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: my-pvc4
  namespace: trident
  annotations:
    trident.netapp.io/smbShareAccessControl: |
      change:
        - tridentADtest
      read:
        - tridentADuser
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: ontap-smb-sc
```

Esempio di PVC con più utenti AD

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: my-test-pvc
  annotations:
    trident.netapp.io/smbShareAccessControl: |
      full_control:
        - tridentTestuser
        - tridentuser
        - tridentTestuser1
        - tridentuser1
      change:
        - tridentADuser
        - tridentADuser1
        - tridentADuser4
        - tridentTestuser2
      read:
        - tridentTestuser2
        - tridentTestuser3
        - tridentADuser2
        - tridentADuser3
spec:
```

```
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
```

Usa un posizionamento bilanciato con i backend ONTAP

Puoi abilitare il posizionamento bilanciato ONTAP sui backend ONTAP. Con il posizionamento bilanciato, ONTAP seleziona l'aggregate ottimale per ogni nuovo volume, così non devi gestire l'assegnazione degli aggregate in Trident.

Come funziona il posizionamento bilanciato

Per impostazione predefinita, Trident rileva gli aggregati assegnati a un SVM, tratta ciascun aggregato come un pool di storage e seleziona un aggregato quando esegue il provisioning di un volume.

Quando abiliti il posizionamento bilanciato:

- Trident riporta l'intero SVM come un singolo pool di storage invece di riportare i singoli aggregati.
- ONTAP seleziona l'aggregato per ogni nuovo volume e bilancia i volumi tra gli aggregati disponibili.
- Trident crea volumi utilizzando l'ONTAP `/application/containers` REST API.

Il posizionamento bilanciato è disponibile per i sistemi ONTAP unificati (AFF, FAS e ASA) e per i sistemi di storage NetApp AFX.

Requisiti

- Il backend deve utilizzare ONTAP 9.19.1 o versioni successive.
- Trident deve comunicare con ONTAP tramite REST. Il posizionamento bilanciato non è disponibile con ZAPI.
- La configurazione del backend non deve specificare gli aggregati. Non puoi combinare il posizionamento bilanciato con la selezione esplicita degli aggregati.

Valori predefiniti

Il valore predefinito di `useBalancedPlacement` dipende dal sistema storage:

sistema storage	Predefinito	Note
ONTAP unificato (AFF, FAS, ASA)	`false`	Trident seleziona gli aggregati a meno che tu non abiliti il posizionamento bilanciato.
NetApp AFX (ONTAP 9.19.1 o versioni successive)	`true`	È richiesto un posizionamento bilanciato. Impostare `useBalancedPlacement` su `false` causa il fallimento della convalida del backend.
NetApp AFX (versioni precedenti a ONTAP 9.19.1)	`false`	Il posizionamento bilanciato non è disponibile.

Opzione di configurazione

Parametro	Descrizione	Predefinito
`useBalancedPlacement`	Delega la selezione aggregata per i nuovi volumi a ONTAP. Imposta questa opzione alla radice della configurazione del backend o sui singoli pool virtuali.	`false`; `true` per AFX su ONTAP 9.19.1 o versioni successive

Posizionamento bilanciato con pool virtuali

Il posizionamento bilanciato funziona con o senza pool virtuali:

- Senza pool virtuali, Trident segnala l'SVM come un singolo pool di storage e crea tutti i nuovi volumi utilizzando un posizionamento bilanciato.
- Con i pool virtuali, Trident segnala i pool virtuali e associa i volumi ad essi come di consueto. Se il pool virtuale corrispondente ha il posizionamento bilanciato abilitato, ONTAP seleziona l'aggregato per il volume.

Imposta `useBalancedPlacement` nella directory principale della configurazione del backend per applicarlo a tutti i pool virtuali, oppure impostalo su singoli pool virtuali per abilitarlo in modo selettivo.

Esempi

Il seguente esempio abilita il posizionamento bilanciato su un backend ONTAP unificato:

```
---
version: 1
storageDriverName: ontap-nas
backendName: nas-balanced-placement
managementLIF: 10.0.0.1
dataLIF: 10.0.0.2
svm: trident_svm
username: cluster-admin
password: <password>
useBalancedPlacement: true
```

L'esempio seguente consente un posizionamento bilanciato utilizzando `TridentBackendConfig`:

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-ontap-nas
spec:
  version: 1
  storageDriverName: ontap-nas
  backendName: nas-balanced-placement
  managementLIF: 10.0.0.1
  svm: trident_svm
```

```
  useBalancedPlacement: true
  credentials:
    name: backend-tbc-ontap-nas-secret
```

Il seguente esempio abilita il posizionamento bilanciato solo sul `gold` pool virtuale. Trident continua a selezionare gli aggregati per i volumi che corrispondono al `bronze` pool virtuale:

```
---
version: 1
storageDriverName: ontap-nas
backendName: nas-virtual-pools
managementLIF: 10.0.0.1
svm: trident_svm
username: cluster-admin
password: <password>
storage:
- labels:
    performance: gold
  useBalancedPlacement: true
- labels:
    performance: bronze
```

Errori di convalida del backend

La creazione o l'aggiornamento del backend non riesce nei seguenti casi:

Configurazione	Risultato
`useBalancedPlacement: true` su ONTAP precedente alla versione 9.19.1	La convalida non è riuscita. Aggiorna a ONTAP 9.19.1 o versione successiva.
`useBalancedPlacement: true` su un backend utilizzando ZAPI	La convalida non è riuscita. Configura il backend per usare REST.
`useBalancedPlacement: true` combinato con aggregati specifici	La convalida non è riuscita. Rimuovi la configurazione dell'aggregate o disabilita il posizionamento bilanciato.
`useBalancedPlacement: false`su AFX con ONTAP 9.19.1 o versioni successive	La convalida non è riuscita. Per AFX su ONTAP 9.19.1 o versioni successive è richiesto un posizionamento bilanciato.

Amazon FSx for NetApp ONTAP

Usa Trident con Amazon FSx for NetApp ONTAP

"Amazon FSx for NetApp ONTAP" è un servizio AWS completamente gestito che esegue file system basati sul sistema operativo per lo storage NetApp ONTAP. Fornisce funzionalità, prestazioni e amministrazione ONTAP con la scalabilità e la semplicità

operativa di AWS. Un file system è la risorsa primaria in Amazon FSx ed è analogo a un cluster ONTAP on-premises. Ogni file system contiene una o più macchine virtuali di storage (SVM) e ogni SVM contiene uno o più volumi che archiviano file e directory. Questa integrazione consente ai cluster Kubernetes in esecuzione su Amazon Elastic Kubernetes Service (EKS) di eseguire il provisioning di volumi persistenti basati su ONTAP per carichi di lavoro a blocchi e file.

Requisiti

Oltre a "Requisiti di Trident", per integrare FSx for ONTAP con Trident, è necessario:

- Un cluster Amazon EKS esistente o un cluster Kubernetes autogestito con `kubectl` installato.
- Un file system Amazon FSx for NetApp ONTAP esistente e una macchina virtuale di storage (SVM) che sia raggiungibile dai nodi worker del cluster.
- Nodi worker che sono preparati per "NFS o iSCSI".

> **NOTA** Assicurati di seguire i passaggi di preparazione del nodo richiesti per Amazon Linux e Ubuntu "Amazon Machine Images" (AMIs) a seconda del tipo di EKS AMI.

Considerazioni

- Volumi SMB:
  - I volumi SMB sono supportati utilizzando solo il `ontap-nas` driver.
  - I volumi SMB non sono supportati con il componente aggiuntivo Trident EKS.
  - Trident supporta volumi SMB montati solo su pod in esecuzione su nodi Windows. Per i dettagli, fai riferimento a "Prepararsi al provisioning dei volumi SMB".
- Prima di Trident 24.02, i volumi creati su Amazon FSx file system che hanno backup automatici abilitati non potevano essere eliminati da Trident. Per evitare questo problema in Trident 24.02 o versioni successive, specificare l' `fsxFilesystemID`, AWS `apiRegion`, AWS `apikey` e AWS `secretKey` nel file di configurazione backend per AWS FSx for ONTAP.

> **NOTA** Se si specifica un ruolo IAM per Trident, è possibile omettere di specificare i campi `apiRegion`, `apiKey` e `secretKey` a Trident esplicitamente. Per ulteriori informazioni, consulta "Opzioni ed esempi di configurazione di FSx per ONTAP".

Utilizzo simultaneo di Trident SAN/iSCSI e del driver EBS-CSI

Se prevedi di utilizzare driver ontap-san (ad esempio, iSCSI) con AWS (EKS, ROSA, EC2 o qualsiasi altra istanza), la configurazione multipath richiesta sui nodi potrebbe entrare in conflitto con il driver CSI di Amazon Elastic Block Store (EBS). Per garantire che il multipathing funzioni senza interferire con i dischi EBS sullo stesso nodo, è necessario escludere EBS dalla configurazione del multipathing. Questo esempio mostra un `multipath.conf` file che include le impostazioni Trident richieste, escludendo i dischi EBS dal multipathing:

```
defaults {
    find_multipaths no
}
blacklist {
```

```
    device {
        vendor "NVME"
        product "Amazon Elastic Block Store"
    }
}
```

Autenticazione

Trident offre due modalità di autenticazione.

- Basato su credenziali (consigliato): memorizza le credenziali in modo sicuro in AWS Secrets Manager. Puoi utilizzare l' `fsxadmin`utente per il tuo file system o l' `vsadmin`utente configurato per la tua SVM.

ATTENZIONE	Trident prevede di essere eseguito come `vsadmin` utente SVM o come utente con un nome diverso che abbia lo stesso ruolo. Amazon FSx for NetApp ONTAP ha un `fsxadmin` utente che è una sostituzione limitata dell'utente `admin` cluster di ONTAP. Si consiglia vivamente di utilizzare `vsadmin` con Trident.

- Basato su certificato: Trident comunicherà con l'SVM sul tuo file system FSx utilizzando un certificato installato sul tuo SVM.

Per i dettagli sull'abilitazione dell'autenticazione, fare riferimento all'autenticazione per il tipo di driver:

- "Autenticazione NAS ONTAP"
- "Autenticazione SAN ONTAP"

Amazon Machine Images (AMI) testate

EKS cluster supporta vari sistemi operativi, ma AWS ha ottimizzato alcune Amazon Machine Images (AMIs) per container ed EKS. Le seguenti AMI sono state testate con NetApp Trident 25.02.

AMI	NAS	NAS-economy	iSCSI	iSCSI-economy
AL2023_x86_64_STANDARD	Sì	Sì	Sì	Sì
AL2_x86_64	Sì	Sì	Sì*	Sì*
BOTTLEROCKET_x86_64	Sì**	Sì	N/A	N/A
AL2023_ARM_64_STANDARD	Sì	Sì	Sì	Sì
AL2_ARM_64	Sì	Sì	Sì*	Sì*
BOTTLEROCKET_ARM_64	Sì**	Sì	N/A	N/A

- * Impossibile eliminare il PV senza riavviare il nodo
- ** Non funziona con NFSv3 con Trident versione 25.02.

NOTA	Se l'AMI desiderata non è elencata qui, non significa che non sia supportata; significa

semplicemente che non è stata testata. Questo elenco serve come guida per le AMI di cui è noto il funzionamento.

Test eseguiti con:

- EKS versione: 1.32
- Metodo di installazione: Helm 25.06 e come AWS add-On 25.06
- Per NAS sono stati testati sia NFSv3 che NFSv4.1.
- Per SAN è stato testato solo iSCSI, non NVMe-oF.

Test eseguiti:

- Crea: Storage Class, pvc, pod
- Elimina: pod, pvc (normale, qtree/lun – economy, NAS con backup AWS)

Trova ulteriori informazioni

- "Documentazione di Amazon FSx for NetApp ONTAP"
- "Post del blog su Amazon FSx for NetApp ONTAP"

Crea un ruolo IAM e un AWS Secret

È possibile configurare i pod Kubernetes per accedere alle risorse AWS autenticandosi come ruolo AWS IAM invece di fornire credenziali AWS esplicite.

NOTA Per eseguire l'autenticazione tramite un ruolo AWS IAM, è necessario disporre di un cluster Kubernetes distribuito tramite EKS.

Crea un secret di AWS Secrets Manager

Poiché Trident emetterà API contro un FSx vserver per gestire lo storage per te, avrà bisogno di credenziali per farlo. Il modo sicuro per trasmettere tali credenziali è tramite un segreto AWS Secrets Manager. Pertanto, se non ne hai già uno, dovrai creare un segreto AWS Secrets Manager che contenga le credenziali per l'account vsadmin.

Questo esempio crea un segreto AWS Secrets Manager per archiviare le credenziali Trident CSI:

```
aws secretsmanager create-secret --name trident-secret --description
"Trident CSI credentials"\
  --secret-string
"{\"username\":\"vsadmin\",\"password\":\"<svmpassword>\"}"
```

Crea policy IAM

Anche Trident necessita delle autorizzazioni AWS per funzionare correttamente. Pertanto, è necessario creare una policy che dia a Trident le autorizzazioni di cui ha bisogno.

I seguenti esempi creano una policy IAM utilizzando l'AWS CLI:

```
aws iam create-policy --policy-name AmazonFSxNCSIDriverPolicy --policy
-document file://policy.json
  --description "This policy grants access to Trident CSI to FSxN and
Secrets manager"
```

Esempio di policy JSON:

```
{
  "Statement": [
    {
      "Action": [
        "fsx:DescribeFileSystems",
        "fsx:DescribeVolumes",
        "fsx:CreateVolume",
        "fsx:RestoreVolumeFromSnapshot",
        "fsx:DescribeStorageVirtualMachines",
        "fsx:UntagResource",
        "fsx:UpdateVolume",
        "fsx:TagResource",
        "fsx:DeleteVolume"
      ],
      "Effect": "Allow",
      "Resource": "*"
    },
    {
      "Action": "secretsmanager:GetSecretValue",
      "Effect": "Allow",
      "Resource": "arn:aws:secretsmanager:<aws-region>:<aws-account-
id>:secret:<aws-secret-manager-name>*"
    }
  ],
  "Version": "2012-10-17"
}
```

Crea Pod Identity o ruolo IAM per l'associazione dell'account di servizio (IRSA)

È possibile configurare un account di servizio Kubernetes per assumere un ruolo AWS Identity and Access Management (IAM) con EKS Pod Identity o IAM role for Service account association (IRSA). Tutti i Pod configurati per utilizzare l'account di servizio possono quindi accedere a qualsiasi servizio AWS a cui il ruolo ha permessi di accesso.

Identità del pod

Le associazioni Amazon EKS Pod Identity consentono di gestire le credenziali per le tue applicazioni, in modo simile a come i profili delle istanze Amazon EC2 forniscono credenziali alle istanze Amazon EC2.

Installa Pod Identity sul tuo cluster EKS:

Puoi creare l'identità del Pod tramite la console AWS o utilizzando il seguente comando AWS CLI:

```
aws eks create-addon --cluster-name <EKS_CLUSTER_NAME> --addon-name
eks-pod-identity-agent
```

Per ulteriori informazioni, fare riferimento a "Configura l'agente di identità del pod Amazon EKS".

Crea trust-relationship.json:

Crea trust-relationship.json per consentire al Service Principal EKS di assumere questo ruolo per Pod Identity. Quindi crea un ruolo con questa trust policy:

```
aws iam create-role \
  --role-name fsxn-csi-role --assume-role-policy-document file://trust-
relationship.json \
  --description "fsxn csi pod identity role"
```

file trust-relationship.json:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "pods.eks.amazonaws.com"
      },
      "Action": [
        "sts:AssumeRole",
        "sts:TagSession"
      ]
    }
  ]
}
```

Associare la policy del ruolo al ruolo IAM:

Associa il criterio di ruolo del passaggio precedente al ruolo IAM che è stato creato:

```
aws iam attach-role-policy \
  --policy-arn arn:aws:iam::aws:111122223333:policy/fsxn-csi-policy \
  --role-name fsxn-csi-role
```

Crea un'associazione di identità pod:

Crea un'associazione di identità pod tra il ruolo IAM e il service account Trident (trident-controller)

```
aws eks create-pod-identity-association \
  --cluster-name <EKS_CLUSTER_NAME> \
  --role-arn arn:aws:iam::111122223333:role/fsxn-csi-role \
  --namespace trident --service-account trident-controller
```

Ruolo IAM per l'associazione dell'account di servizio (IRSA)
Utilizzando l'AWS CLI:

```
aws iam create-role --role-name AmazonEKS_FSxN_CSI_DriverRole \
  --assume-role-policy-document file://trust-relationship.json
```

file trust-relationship.json:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Federated": "arn:aws:iam::<account_id>:oidc-
provider/<oidc_provider>"
      },
      "Action": "sts:AssumeRoleWithWebIdentity",
      "Condition": {
        "StringEquals": {
          "<oidc_provider>:aud": "sts.amazonaws.com",
          "<oidc_provider>:sub":
"system:serviceaccount:trident:trident-controller"
        }
      }
    }
  ]
}
```

Aggiorna i seguenti valori nel file `trust-relationship.json`:

- **<account_id>** - ID del tuo account AWS
- **<oidc_provider>** - L'OIDC del tuo cluster EKS. Puoi ottenere l'oidc_provider eseguendo:

```
aws eks describe-cluster --name my-cluster --query
"cluster.identity.oidc.issuer"\
  --output text | sed -e "s/^https:\/\///"
```

Associare il ruolo IAM alla policy IAM:

Una volta creato il ruolo, associare il criterio (che è stato creato nel passaggio sopra) al ruolo utilizzando questo comando:

```
aws iam attach-role-policy --role-name my-role --policy-arn <IAM policy
ARN>
```

Verifica che il provider OICD sia associato:

Verifica che il tuo provider OIDC sia associato al tuo cluster. Puoi verificarlo utilizzando questo comando:

```
aws iam list-open-id-connect-providers | grep $oidc_id | cut -d "/" -f4
```

Se l'output è vuoto, utilizzare il seguente comando per associare IAM OIDC al cluster:

```
eksctl utils associate-iam-oidc-provider --cluster $cluster_name
--approve
```

Se si utilizza eksctl, utilizzare il seguente esempio per creare un ruolo IAM per account di servizio in EKS:

```
eksctl create iamserviceaccount --name trident-controller --namespace
trident \
  --cluster <my-cluster> --role-name AmazonEKS_FSxN_CSI_DriverRole
--role-only \
  --attach-policy-arn <IAM-Policy ARN> --approve
```

Installare Trident

Trident semplifica la gestione dello storage Amazon FSx for NetApp ONTAP in Kubernetes, consentendo a sviluppatori e amministratori di concentrarsi sulla distribuzione delle applicazioni. Puoi installare Trident utilizzando uno dei seguenti metodi:

- Helm
- Componente aggiuntivo EKS

Se desideri utilizzare la funzionalità di snapshot, installa il componente aggiuntivo CSI snapshot controller. Per ulteriori informazioni, consulta "Abilita la funzionalità snapshot per i volumi CSI".

Installa Trident tramite helm

Identità del pod

1. Aggiungi il repository Trident Helm:

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

2. Installa Trident utilizzando il seguente esempio:

```
helm install trident-operator netapp-trident/trident-operator --version 100.2502.1 --namespace trident --create-namespace
```

Puoi utilizzare il comando `helm list` per rivedere i dettagli dell'installazione come nome, namespace, chart, stato, versione dell'app e numero di revisione.

```
helm list -n trident
```

```
NAME                NAMESPACE   REVISION    UPDATED
STATUS       CHART                          APP VERSION

trident-operator    trident     1           2024-10-14
14:31:22.463122 +0300 IDT    deployed     trident-operator-
100.2502.0    25.02.0
```

Associazione account di servizio (IRSA)

1. Aggiungi il repository Trident Helm:

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

2. Imposta i valori per **cloud provider** e **cloud identity**:

```
helm install trident-operator netapp-trident/trident-operator
--version 100.2502.1  \
--set cloudProvider="AWS" \
--set cloudIdentity="'eks.amazonaws.com/role-arn:
arn:aws:iam::<accountID>:role/<AmazonEKS_FSxN_CSI_DriverRole>'" \
```

```
--namespace trident \
--create-namespace
```

Puoi utilizzare il comando `helm list` per rivedere i dettagli dell'installazione come nome, namespace, chart, stato, versione dell'app e numero di revisione.

```
helm list -n trident
```

```
NAME                NAMESPACE   REVISION    UPDATED
STATUS       CHART                          APP VERSION

trident-operator    trident     1           2024-10-14
14:31:22.463122 +0300 IDT    deployed     trident-operator-
100.2510.0    25.10.0
```

NOTA

Se si prevede di utilizzare iSCSI, assicurarsi che iSCSI sia abilitato sul computer client. Se si utilizza il sistema operativo AL2023 Worker node, è possibile automatizzare l'installazione del client iSCSI aggiungendo il parametro node prep nell'installazione di helm:

```
helm install trident-operator netapp-trident/trident-operator
--version 100.2502.1 --namespace trident --create-namespace --
set nodePrep={iscsi}
```

Installa Trident tramite il componente aggiuntivo EKS

Il componente aggiuntivo Trident EKS include le patch di sicurezza più recenti, le correzioni di bug ed è convalidato da AWS per funzionare con Amazon EKS. Il componente aggiuntivo EKS consente di garantire costantemente che i cluster Amazon EKS siano sicuri e stabili e di ridurre la quantità di lavoro necessaria per installare, configurare e aggiornare i componenti aggiuntivi.

Prerequisiti

Assicurati di avere quanto segue prima di configurare il componente aggiuntivo Trident per AWS EKS:

- Un account cluster Amazon EKS con abbonamento aggiuntivo
- Autorizzazioni AWS per l'AWS marketplace:
  `"aws-marketplace:ViewSubscriptions",`
  `"aws-marketplace:Subscribe",`
  `"aws-marketplace:Unsubscribe`
- Tipo AMI: Amazon Linux 2 (AL2_x86_64) o Amazon Linux 2 Arm(AL2_ARM_64)
- Tipo di nodo: AMD o ARM
- Un file system Amazon FSx per NetApp ONTAP esistente

Abilita il componente aggiuntivo Trident per AWS

Console di gestione

1. Apri la console Amazon EKS su https://console.aws.amazon.com/eks/home#/clusters.
2. Nel riquadro di navigazione a sinistra, seleziona **Clusters**.
3. Seleziona il nome del cluster per cui desideri configurare il componente aggiuntivo NetApp Trident CSI.
4. Seleziona **Componenti aggiuntivi** e poi seleziona **Ottieni altri componenti aggiuntivi**.
5. Seguire questi passaggi per selezionare il software add-on:
   a. Scorri verso il basso fino alla sezione **AWS Marketplace add-ons** e digita **"Trident"** nella casella di ricerca.
   b. Selezionare la check box nell'angolo in alto a destra della casella Trident by NetApp.
   c. Seleziona **Avanti**.
6. Nella pagina delle impostazioni **Configura i componenti aggiuntivi selezionati**, eseguire le seguenti operazioni:

   NOTA **Salta questi passaggi se utilizzi l'associazione Pod Identity.**

   a. Seleziona la **Version** che desideri utilizzare.
   b. Se si utilizza l'autenticazione IRSA, assicurarsi di impostare i valori di configurazione disponibili nelle impostazioni di configurazione opzionali:
      - Seleziona la **Version** che desideri utilizzare.
      - Seguire lo **Schema di configurazione aggiuntivo** e impostare il parametro **configurationValues** nella sezione **Valori di configurazione** sul role-arn creato nel passaggio precedente (il valore deve essere nel formato seguente):

```
{

  "cloudIdentity": "'eks.amazonaws.com/role-arn: <role ARN>'",
  "cloudProvider": "AWS"

}
```

   +
   Se si seleziona Override per il metodo di risoluzione dei conflitti, una o più impostazioni del componente aggiuntivo esistente possono essere sovrascritte con le impostazioni dell'add-on Amazon EKS. Se non si abilita questa opzione e si verifica un conflitto con le impostazioni esistenti, l'operazione non riesce. È possibile utilizzare il messaggio di errore risultante per risolvere il conflitto. Prima di selezionare questa opzione, assicurarsi che l'add-on Amazon EKS non gestisca impostazioni che è necessario autogestire.
7. Scegli **Next**.
8. Nella pagina **Revisione e aggiunta**, scegliere **Crea**.

   Al termine dell'installazione del software add-on, viene visualizzato il software add-on installato.

AWS CLI

1. Crea il `add-on.json` file:

Per Pod Identity, utilizzare il seguente formato:

NOTA Utilizzare il

```
{
  "clusterName": "<eks-cluster>",
  "addonName": "netapp_trident-operator",
  "addonVersion": "v25.6.0-eksbuild.1",
}
```

Per l'autenticazione IRSA, utilizzare il seguente formato:

```
{
  "clusterName": "<eks-cluster>",
  "addonName": "netapp_trident-operator",
  "addonVersion": "v25.6.0-eksbuild.1",
  "serviceAccountRoleArn": "<role ARN>",
  "configurationValues": {
    "cloudIdentity": "'eks.amazonaws.com/role-arn: <role ARN>'",
    "cloudProvider": "AWS"
  }
}
```

NOTA Sostituisci `<role ARN>` con l'ARN del ruolo che è stato creato nel passaggio precedente.

2. Installare il Trident EKS add-on.

```
aws eks create-addon --cli-input-json file://add-on.json
```

eksctl

Il seguente esempio di comando installa il Trident EKS add-on:

```
eksctl create addon --name netapp_trident-operator --cluster
<cluster_name> --force
```

Aggiornare il software add-on Trident EKS

Console di gestione

1. Apri la console Amazon EKS https://console.aws.amazon.com/eks/home#/clusters.
2. Nel riquadro di navigazione a sinistra, seleziona **Clusters**.
3. Selezionare il nome del cluster per cui si desidera aggiornare il software add-on NetApp Trident CSI.
4. Selezionare la scheda **Add-ons**.
5. Seleziona **Trident by NetApp** e poi seleziona **Modifica**.
6. Nella pagina **Configura Trident by NetApp**, procedere come segue:
   a. Seleziona la **Version** che desideri utilizzare.
   b. Espandi le **Impostazioni di configurazione opzionali** e modificale secondo necessità.
   c. Seleziona **Salva modifiche**.

AWS CLI

Il seguente esempio aggiorna l'add-on EKS:

```
aws eks update-addon --cluster-name <eks_cluster_name> --addon-name
netapp_trident-operator --addon-version v25.6.0-eksbuild.1 \
  --service-account-role-arn <role-ARN> --resolve-conflict preserve \
  --configuration-values "{\"cloudIdentity\":
\"'eks.amazonaws.com/role-arn: <role ARN>'\"}"
```

eksctl

- Controlla la versione corrente del tuo software add-on FSxN Trident CSI. Sostituisci `my-cluster` con il nome del tuo cluster.

  ```
  eksctl get addon --name netapp_trident-operator --cluster my-cluster
  ```

  Esempio di output:

```
NAME                          VERSION              STATUS    ISSUES
IAMROLE    UPDATE AVAILABLE    CONFIGURATION VALUES
netapp_trident-operator    v25.6.0-eksbuild.1    ACTIVE    0
{"cloudIdentity":"'eks.amazonaws.com/role-arn:
arn:aws:iam::139763910815:role/AmazonEKS_FSXN_CSI_DriverRole'"}
```

- Aggiornare il software add-on alla versione riportata sotto UPDATE AVAILABLE nell'output del passaggio precedente.

  ```
  eksctl update addon --name netapp_trident-operator --version
  v25.6.0-eksbuild.1 --cluster my-cluster --force
  ```

Se si rimuove l' `--force`opzione e una qualsiasi delle impostazioni Amazon EKS add-on è in conflitto con le impostazioni esistenti, l'aggiornamento dell'Amazon EKS add-on non riesce; viene visualizzato un messaggio di errore per aiutarti a risolvere il conflitto. Prima di specificare questa opzione, assicurati che l'Amazon EKS add-on non gestisca impostazioni che devi gestire, perché tali impostazioni vengono sovrascritte con questa opzione. Per ulteriori informazioni su altre opzioni per questa impostazione, vedi "Componenti aggiuntivi". Per ulteriori informazioni sulla gestione dei campi Amazon EKS Kubernetes, vedi "Gestione dei campi Kubernetes".

Disinstallare/rimuovere il Trident EKS add-on

Hai due opzioni per rimuovere un add-on di Amazon EKS:

- **Conserva il software add-on sul tuo cluster** – Questa opzione rimuove la gestione di qualsiasi impostazione da parte di Amazon EKS. Rimuove anche la possibilità per Amazon EKS di notificarti gli aggiornamenti e di aggiornare automaticamente l'add-on Amazon EKS dopo che hai avviato un aggiornamento. Tuttavia, conserva il software add-on sul tuo cluster. Questa opzione rende l'add-on un'installazione autogestita, invece che un add-on Amazon EKS. Con questa opzione, non c'è alcun downtime per l'add-on. Mantieni l' `--preserve` opzione nel comando per conservare l'add-on.
- **Rimuovere il software add-on interamente dal cluster** – NetApp consiglia di rimuovere l'add-on Amazon EKS dal cluster solo se non ci sono risorse sul cluster che dipendono da esso. Rimuovere l'opzione `--preserve` dal comando `delete` per rimuovere l'add-on.

NOTA Se al software add-on è associato un account IAM, l'account IAM non viene rimosso.

Console di gestione

1. Apri la console Amazon EKS su https://console.aws.amazon.com/eks/home#/clusters.
2. Nel riquadro di navigazione sinistro, selezionare **Clusters**.
3. Selezionare il nome del cluster dal quale si desidera rimuovere il software add-on NetApp Trident CSI.
4. Selezionare la scheda **Add-ons** e poi scegliere **Trident by NetApp**.*
5. Seleziona **Rimuovi**.
6. Nella finestra di dialogo **Remove netapp_trident-operator confirmation**, procedere come segue:
   a. Se si desidera che Amazon EKS smetta di gestire le impostazioni del software add-on, selezionare **Preserva sul cluster**. Eseguire questa operazione se si desidera conservare il software add-on sul cluster in modo da poter gestire autonomamente tutte le impostazioni del software add-on.
   b. Immettere **netapp_trident-operator**.
   c. Seleziona **Rimuovi**.

AWS CLI

Sostituire `my-cluster` con il nome del cluster, quindi eseguire il seguente comando.

```
aws eks delete-addon --cluster-name my-cluster --addon-name
netapp_trident-operator --preserve
```

eksctl

Il seguente comando disinstalla il componente aggiuntivo Trident EKS:

```
eksctl delete addon --cluster K8s-arm --name netapp_trident-operator
```

Configurare una classe di storage

Il "Oggetto Kubernetes StorageClass" identifica un provisioner e istruisce il provisioner su come effettuare il provisioning dei volumi. Questa sezione mostra come configurare un oggetto Kubernetes StorageClass che specifica Trident come provisioner.

Crea un oggetto StorageClass

Quando si crea una StorageClass per FSx for ONTAP, Trident creerà automaticamente la configurazione del backend.

NOTA	Se desideri configurare manualmente il backend di storage, consulta la Crea un Kubernetes StorageClass senza configurazione automatica del backend sezione per sapere come creare separatamente il backend Trident e la storage class.

Specificare i parametri richiesti StorageClass

I seguenti tre parametri devono essere definiti durante la creazione di un StorageClass:

Parametro	Richiesto	Tipo	Descrizione
`fsxFilesystemID`	Sì	stringa	ID del file system FSx for NetApp ONTAP
`storageDriverName`	Sì	stringa	Driver di storage Trident (ad esempio, `ontap-nas` o `ontap-san`)
`credentialsName`	Sì	stringa	Nome del segreto Kubernetes che contiene le credenziali FSx for ONTAP

Specificare parametri facoltativi

È possibile passare parametri backend opzionali tramite la StorageClass. Definire tutti i valori opzionali come stringhe nella sezione StorageClass `parameters`. Per un elenco completo dei parametri backend, vedere: "Configurazione del backend FSx for NetApp ONTAP".

Esempi di file di configurazione StorageClass.

L'esempio seguente mostra un StorageClass che attiva la configurazione automatica del backend.

YAML

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
```

```
  name: ontap-fsx-demo
  annotations:
    description: "Demo StorageClass for FSx for NetApp ONTAP"
provisioner: csi.trident.netapp.io
parameters:
  fsxFilesystemID: "fs-0abc123"
  storageDriverName: "ontap-nas"
  credentialsName: trident-fsx-credentials
allowVolumeExpansion: true
reclaimPolicy: Delete
volumeBindingMode: Immediate
```

JSON

```
{
  "apiVersion": "storage.k8s.io/v1",
  "kind": "StorageClass",
  "metadata": {
    "name": "ontap-fsx-demo",
    "annotations": {
      "description": "Demo StorageClass for FSx for NetApp ONTAP"
    }
  },
  "provisioner": "csi.trident.netapp.io",
  "parameters": {
    "fsxFilesystemID": "fs-0abc123",
    "storageDriverName": "ontap-nas",
    "credentialsName": "trident-fsx-credentials"
  },
  "allowVolumeExpansion": true,
  "reclaimPolicy": "Delete",
  "volumeBindingMode": "Immediate"
}
```

Crea il StorageClass

Una volta creato il file di configurazione, esegui il seguente comando per creare la storage class.

```
kubectl create -f storage-class-ontapnas.yaml
```

Ora dovresti vedere una classe di storage **basic-csi** sia in Kubernetes che in Trident, e Trident dovrebbe aver rilevato i pool sul backend.

```
kubectl get sc basic-csi
```

```
NAME         PROVISIONER             AGE
basic-csi    csi.trident.netapp.io   15h
```

Dopo aver applicato il StorageClass, Trident crea automaticamente il backend. Puoi quindi creare PersistentVolumeClaims che fanno riferimento a questo StorageClass.

Verifica lo stato della configurazione del backend

Trident registra il risultato della creazione del backend nelle annotazioni di StorageClass.

Annotazione	Descrizione
`trident.netapp.io/configuratorStatus`	Risultato della configurazione (`Success` o `Failure`)
`trident.netapp.io/configuratorMessage`	Messaggio di stato o errore dettagliato
`trident.netapp.io/configuratorName`	Nome della risorsa del configuratore interno
`trident.netapp.io/managed`	Indica che la StorageClass è gestita da Trident
`trident.netapp.io/additionalStoragePools`	Pool di storage creati per questo backend

Per verificare lo stato, eseguire:

```
kubectl get storageclass ontap-fsx-demo -o yaml
```

Confermare che `trident.netapp.io/configuratorStatus` è impostato su `Success`. Se il valore è `Failure`, esaminare `trident.netapp.io/configuratorMessage` per l'errore.

Aggiungi file system FSxN aggiuntivi

Se hai bisogno di ulteriore capacità di archiviazione continuando a utilizzare lo stesso StorageClass, aggiungi ulteriori ID del file system FSxN.

Modifica la StorageClass e aggiungi la seguente annotazione:

```
metadata:
  annotations:
    trident.netapp.io/additionalFsxNFileSystemID: '["fs-
xxxxxxxxxxxxxxxxx"]'
```

Dopo aver applicato la modifica, Trident aggiorna la configurazione del backend e aggiorna le annotazioni StorageClass.

Considerazioni operative e limitazioni

- L'eliminazione di un StorageClass che ha la configurazione automatica del backend solitamente elimina il backend Trident associato. Ciò può interrompere la connettività di storage e interrompere i carichi di lavoro in esecuzione. Verifica l'impatto prima di eliminare un StorageClass gestito.
- La configurazione automatica del backend è supportata solo per AWS FSx per NetApp ONTAP.

Crea un Kubernetes StorageClass senza configurazione automatica del backend

Se desideri creare il backend Trident e la StorageClass separatamente, segui questi passaggi.

Scopri come funziona la configurazione automatica del backend

Trident deriva la configurazione del backend dalla definizione di StorageClass. Quando si applica la StorageClass, Trident convalida i parametri richiesti, crea il backend e annota la StorageClass con lo stato.

Trident crea il VolumeSnapshotClass solo una volta. Trident riutilizza lo stesso VolumeSnapshotClass per i successivi StorageClasses.

Crea il backend Trident

Per creare un backend Trident, è necessario creare un file di configurazione in formato JSON o YAML. Il file deve specificare il tipo di storage desiderato (NAS o SAN), il file system, la SVM da cui ottenerlo e come autenticarsi con essa. L'esempio seguente mostra come definire uno storage basato su NAS e utilizzare un segreto AWS per memorizzare le credenziali per la SVM che si desidera utilizzare:

YAML

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-ontap-nas
  namespace: trident
spec:
  version: 1
  storageDriverName: ontap-nas
  backendName: tbc-ontap-nas
  svm: svm-name
  aws:
    fsxFilesystemID: fs-xxxxxxxxxx
  credentials:
    name: "arn:aws:secretsmanager:us-west-2:xxxxxxxx:secret:secret-
name"
    type: awsarn
```

JSON

```
{
  "apiVersion": "trident.netapp.io/v1",
```

```
  "kind": "TridentBackendConfig",
  "metadata": {
    "name": "backend-tbc-ontap-nas"
    "namespace": "trident"
  },
  "spec": {
    "version": 1,
    "storageDriverName": "ontap-nas",
    "backendName": "tbc-ontap-nas",
    "svm": "svm-name",
    "aws": {
      "fsxFilesystemID": "fs-xxxxxxxxxx"
    },
    "managementLIF": null,
    "credentials": {
      "name": "arn:aws:secretsmanager:us-west-2:xxxxxxxx:secret:secret-
name",
      "type": "awsarn"
    }
  }
}
```

Dettagli del driver FSx per ONTAP

È possibile integrare Trident con Amazon FSx for NetApp ONTAP utilizzando i seguenti driver:

Nome driver	Descrizione
`ontap-san`	Ogni PV di cui viene effettuato il provisioning è una LUN all'interno del proprio volume Amazon FSx for NetApp ONTAP. Consigliato per storage a blocchi.
`ontap-nas`	Ogni PV fornito è un volume Amazon FSx for NetApp ONTAP completo. Consigliato per NFS e SMB.
`ontap-san-economy`	Ogni PV di cui viene effettuato il provisioning è un LUN con un numero configurabile di LUN per Amazon FSx for NetApp ONTAP volume.
`ontap-nas-economy`	Ogni PV di cui viene effettuato il provisioning è un qtree, con un numero configurabile di qtree per Amazon FSx for NetApp ONTAP volume.
`ontap-nas-flexgroup`	Ogni PV fornito è un volume Amazon FSx for NetApp ONTAP FlexGroup completo.

Per i dettagli sui driver, fare riferimento a "Driver NAS" e "Driver SAN".

Crea il backend

Dopo aver creato il file di configurazione, eseguire i seguenti comandi per creare e convalidare la Trident Backend Configuration (TBC):

- Crea la configurazione del backend Trident (TBC) dal file yaml ed esegui il seguente comando:

```
kubectl create -f backendconfig.yaml -n trident
```

```
tridentbackendconfig.trident.netapp.io/backend-tbc-ontap-nas created
```

- Verificare che la configurazione del backend Trident (TBC) sia stata creata correttamente:

```
Kubectl get tbc -n trident
```

```
NAME                           BACKEND NAME          BACKEND UUID
PHASE   STATUS

backend-tbc-ontap-nas          tbc-ontap-nas         933e0071-66ce-4324-
b9ff-f96d916ac5e9   Bound   Success
```

Per ulteriori informazioni su altre opzioni di configurazione, consultare la Configurazione avanzata del backend ed esempi sezione seguente.

Configura una Storage Class senza configurazione automatica del backend

Di seguito sono riportati alcuni esempi di configurazioni di Storage Class da utilizzare con Trident e FSx for ONTAP.

Classe di archiviazione per NFS

È possibile utilizzare questo esempio per configurare StorageClass per volumi tramite NFS (fare riferimento alla sezione Attributi Trident di seguito per l'elenco completo degli attributi):

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontap-gold
provisioner: csi.trident.netapp.io
parameters:
  backendType: "ontap-nas"
  provisioningType: "thin"
  snapshots: "true"
```

Classe di archiviazione per iSCSI

Utilizza questo esempio per configurare StorageClass per volumi tramite iSCSI:

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontap-gold
provisioner: csi.trident.netapp.io
parameters:
  backendType: "ontap-san"
  provisioningType: "thin"
  snapshots: "true"
```

Classe di archiviazione che utilizza NFSv3 e AWS Bottlerocket

Per eseguire il provisioning di volumi NFSv3 su AWS Bottlerocket, aggiungete il necessario `mountOptions` alla storage class:

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontap-gold
provisioner: csi.trident.netapp.io
parameters:
  backendType: "ontap-nas"
  media: "ssd"
  provisioningType: "thin"
  snapshots: "true"
mountOptions:
  - nfsvers=3
  - nolock
```

Attributi di Trident StorageClass

Questi parametri determinano quali pool di storage gestiti da Trident devono essere utilizzati per effettuare il provisioning dei volumi di un determinato tipo.

Attributo	Tipo	Valori	Offerta	Richiesta	Supportato da
media[1]	stringa	hdd, hybrid, ssd	Il pool contiene supporti di questo tipo; ibrido significa entrambi	Tipo di media specificato	ontap-nas, ontap-nas-economy, ontap-nas-flexgroup, ontap-san, solidfire-san
provisioningType	stringa	sottile, spesso	Il pool supporta questo metodo di provisioning	Metodo di provisioning specificato	spesso: tutti ontap; sottile: tutti ontap & solidfire-san

Attributo	Tipo	Valori	Offerta	Richiesta	Supportato da
backendType	stringa	ontap-nas, ontap-nas-economy, ontap-nas-flexgroup, ontap-san, solidfire-san, azure-netapp-files, ontap-san-economy	Il pool appartiene a questo tipo di backend	Backend specificato	Tutti i driver
istantanee	bool	vero, falso	Il pool supporta volumi con snapshot	Volume con snapshot abilitato	ontap-nas, ontap-san, solidfire-san
cloni	bool	vero, falso	Il pool supporta la clonazione dei volumi	Volume con cloni abilitati	ontap-nas, ontap-san, solidfire-san
crittografia	bool	vero, falso	Il pool supporta volumi criptati	Volume con crittografia abilitata	ontap-nas, ontap-nas-economy, ontap-nas-flexgroups, ontap-san
IOPS	int	intero positivo	Il pool è in grado di garantire IOPS in questo intervallo	Volume garantisce questi IOPS	solidfire-san

[1]: Non supportato da ONTAP Select o FSx for ONTAP

Fate riferimento a "Oggetti Kubernetes e Trident" per i dettagli su come le classi di storage interagiscono con `PersistentVolumeClaim` e sui parametri per controllare come Trident effettua il provisioning dei volumi.

Crea la classe di archiviazione

Una volta configurato il StorageClass, puoi crearlo in Kubernetes.

Passaggi

1. Si tratta di un oggetto Kubernetes, quindi usa `kubectl` per crearlo in Kubernetes.

```
kubectl create -f storage-class-ontapnas.yaml
```

2. Ora dovresti vedere una classe di storage **basic-csi** sia in Kubernetes che in Trident, e Trident dovrebbe aver rilevato i pool sul backend.

```
kubectl get sc basic-csi
```

```
NAME         PROVISIONER             AGE
```

```
basic-csi    csi.trident.netapp.io   15h
```

Effettuare il provisioning dei volumi SMB

È possibile eseguire il provisioning dei volumi SMB utilizzando il `ontap-nas` driver. Tuttavia, per farlo è necessario completare questi passaggi: "Prepararsi al provisioning dei volumi SMB".

Configurazione avanzata del backend ed esempi

Consulta la tabella seguente per le opzioni di configurazione del backend:

Parametro	Descrizione	Esempio
`version`		Sempre 1
`storageDriverName`	Nome del driver di archiviazione	`ontap-nas`, `ontap-nas-economy`, `ontap-nas-flexgroup`, `ontap-san`, `ontap-san-economy`
`backendName`	Nome personalizzato o lo storage backend	Nome driver + "_" + dataLIF
`managementLIF`	Indirizzo IP di un cluster o di una LIF di gestione SVM. È possibile specificare un fully-qualified domain name (FQDN). Può essere impostato per usare indirizzi IPv6 se hai installato Trident usando il flag IPv6. Gli indirizzi IPv6 devono essere definiti tra parentesi quadre, ad esempio [28e8:d9fb:a825:b7bf:69a8:d02f:9e7b:3555]. Se si fornisce `fsxFilesystemID` sotto il campo `aws`, non è necessario fornire `managementLIF` perché Trident recupera le informazioni SVM `managementLIF` da AWS. Quindi, è necessario fornire le credenziali per un utente sotto l'SVM (ad esempio: vsadmin) e l'utente deve avere il ruolo `vsadmin`.	"10.0.0.1", "[2001:1234:abcd::fefe]"

Parametro	Descrizione	Esempio
`dataLIF`	Indirizzo IP del protocollo LIF. **ONTAP NAS drivers**: NetApp consiglia di specificare dataLIF. Se non fornito, Trident recupera i dataLIF dall'SVM. Puoi specificare un fully-qualified domain name (FQDN) da usare per le operazioni di mount NFS, permettendoti di creare un DNS round-robin per bilanciare il carico tra più dataLIF. Può essere modificato dopo l'impostazione iniziale. **ONTAP SAN drivers**: non specificare per iSCSI. Trident usa ONTAP Selective LUN Map per individuare i LIF iSCSI necessari a stabilire una sessione multipath. Viene generato un avviso se dataLIF è definito esplicitamente. Può essere impostato per usare indirizzi IPv6 se hai installato Trident usando il flag IPv6. Gli indirizzi IPv6 devono essere definiti tra parentesi quadre, ad esempio [28e8:d9fb:a825:b7bf:69a8:d02f:9e7b:3555].	
`autoExportPolicy`	Abilita la creazione e l'aggiornamento automatici delle policy di esportazione [Boolean]. Utilizzando le `autoExportPolicy` e `autoExportCIDRs` opzioni, Trident può gestire automaticamente le policy di esportazione.	`false`
`autoExportCIDRs`	Elenco di CIDR in base ai quali filtrare gli IP dei nodi Kubernetes quando `autoExportPolicy` è abilitato. Utilizzando le `autoExportPolicy` e `autoExportCIDRs` opzioni, Trident può gestire automaticamente le policy di esportazione.	"["0.0.0.0/0", "::/0"]"
`labels`	Set di etichette arbitrarie in formato JSON da applicare ai volumi	""
`clientCertificate`	Valore codificato in Base64 del certificato client. Utilizzato per l'autenticazione basata su certificato	""

Parametro	Descrizione	Esempio
`clientPrivateKey`	Valore codificato in Base64 della chiave privata del client. Utilizzato per l'autenticazione basata su certificato	""
`trustedCACertificate`	Valore codificato in Base64 del certificato CA attendibile. Opzionale. Utilizzato per l'autenticazione basata su certificato.	""
`username`	Nome utente per connettersi al cluster o alla SVM. Utilizzata per l'autenticazione basata sulle credenziali. Ad esempio, vsadmin.	
`password`	Password per connettersi al cluster o alla SVM. Utilizzata per l'autenticazione basata sulle credenziali.	
`svm`	Macchina virtuale di storage da utilizzare	Derivato se viene specificato un SVM managementLIF.
`storagePrefix`	Prefisso utilizzato durante il provisioning di nuovi volumi nell'SVM. Non puoi modificarlo dopo la creazione. Per aggiornare questo parametro, sarà necessario creare un nuovo backend.	`trident`
`limitAggregateUsage`	**Non specificare per Amazon FSx per NetApp ONTAP.** I `fsxadmin` e `vsadmin` forniti non contengono le autorizzazioni necessarie per recuperare l'utilizzo aggregato e limitarlo tramite Trident.	Non utilizzare.
`limitVolumeSize`	Il provisioning fallisce se la dimensione del volume richiesto supera questo valore. Limita inoltre la dimensione massima dei volumi che gestisce per qtree e LUN, e l' `qtreesPerFlexvol`opzione consente di personalizzare il numero massimo di qtree per FlexVol volume	"" (non applicato per impostazione predefinita)
`lunsPerFlexvol`	Numero massimo di LUN per FlexVol volume, deve essere compreso nell'intervallo [50, 200]. Solo SAN.	"100"

Parametro	Descrizione	Esempio
`debugTraceFlags`	Flag di debug da usare per la risoluzione dei problemi. Ad esempio, {"api":false, "method":true} non utilizzare `debugTraceFlags` a meno che non si stia risolvendo un problema e si richieda un dump dettagliato del log.	null
`nfsMountOptions`	Elenco separato da virgole delle opzioni di montaggio NFS. Le opzioni di montaggio per i volumi persistenti Kubernetes sono normalmente specificate nelle classi di storage, ma se non vengono specificate opzioni di montaggio in una classe di storage, Trident utilizzerà le opzioni di montaggio specificate nel file di configurazione del backend di storage. Se non vengono specificate opzioni di montaggio nella classe di storage o nel file di configurazione, Trident non imposterà alcuna opzione di montaggio su un volume persistente associato.	""
`nasType`	Configura la creazione di volumi NFS o SMB. Le opzioni sono `nfs`, `smb` o null. **Deve essere impostato su `smb` per i volumi SMB.** Impostando su null, vengono creati di default volumi NFS.	`nfs`
`qtreesPerFlexvol`	Numero massimo di qtree per volume FlexVol, deve essere compreso nell'intervallo [50, 300]	`"200"`
`smbShare`	È possibile specificare uno dei seguenti parametri: il nome di una condivisione SMB creata tramite Microsoft Management Console o ONTAP CLI oppure un nome che consenta a Trident di creare la condivisione SMB. Questo parametro è obbligatorio per Amazon FSx for ONTAP backends.	`smb-share`

Parametro	Descrizione	Esempio
`useREST`	Parametro booleano per utilizzare le API REST di ONTAP. Se impostato su `true`, Trident utilizzerà le API REST di ONTAP per comunicare con il backend. Questa funzione richiede ONTAP 9.11.1 e versioni successive. Inoltre, il ruolo di login ONTAP utilizzato deve avere accesso all'applicazione `ontap`. Questo è soddisfatto dai ruoli predefiniti `vsadmin` e `cluster-admin`.	`false`
`aws`	È possibile specificare quanto segue nel file di configurazione per AWS FSx for ONTAP: - `fsxFilesystemID`: Specificare l'ID del file system AWS FSx. - `apiRegion`: Nome della regione API AWS. - `apikey`: Chiave API AWS. - `secretKey`: Chiave segreta AWS.	"" "" ""
`credentials`	Specificare le credenziali FSx SVM da archiviare in AWS Secrets Manager. - `name`: Amazon Resource Name (ARN) del segreto, che contiene le credenziali di SVM. - `type`: Impostare su `awsarn`. Per ulteriori informazioni, consulta "Crea un segreto AWS Secrets Manager".	

Opzioni di configurazione del backend per il provisioning dei volumi

È possibile controllare il provisioning predefinito utilizzando queste opzioni nella `defaults` sezione della configurazione. Per un esempio, vedere gli esempi di configurazione riportati di seguito.

Parametro	Descrizione	Predefinito
`spaceAllocation`	Allocazione dello spazio per le LUN	`true`
`spaceReserve`	Modalità di prenotazione dello spazio; "none" (thin) o "volume" (thick)	`none`
`snapshotPolicy`	policy di Snapshot da utilizzare	`none`

Parametro	Descrizione	Predefinito
`qosPolicy`	Gruppo di policy QoS da assegnare ai volumi creati. Scegli uno tra qosPolicy o adaptiveQosPolicy per ogni pool di storage o backend. L'utilizzo di gruppi di policy QoS con Trident richiede ONTAP 9.8 o versioni successive. È consigliabile utilizzare un gruppo di policy QoS non condiviso e assicurarsi che il gruppo di policy venga applicato a ciascun componente singolarmente. Un gruppo di policy QoS condiviso impone il limite massimo per il throughput di tutti i carichi di lavoro.	""
`adaptiveQosPolicy`	Gruppo di policy QoS adattivo da assegnare ai volumi creati. Scegli uno tra qosPolicy o adaptiveQosPolicy per ogni pool di storage o backend. Non supportato da ontap-nas-economy.	""
`snapshotReserve`	Percentuale di volume riservata per gli snapshot "0"	Se `snapshotPolicy` è `none`, `else` ""
`splitOnClone`	Dividere un clone dal suo genitore al momento della creazione	`false`
`encryption`	Abilita NetApp Volume Encryption (NVE) sul nuovo volume; il valore predefinito è `false`. NVE deve essere concesso in licenza e abilitato sul cluster per usare questa opzione. Se NAE è abilitato sul backend, qualsiasi volume fornito in Trident sarà abilitato per NAE. Per ulteriori informazioni, consulta: "Come funziona Trident con NVE e NAE".	`false`
`luksEncryption`	Abilita la crittografia LUKS. Fai riferimento a "Usa Linux Unified Key Setup (LUKS)". Solo SAN.	""
`tieringPolicy`	Criterio di tiering da utilizzare `none`	
`unixPermissions`	Modalità per nuovi volumi. **Lasciare vuoto per volumi SMB.**	""
`securityStyle`	Stile di sicurezza per i nuovi volumi. NFS supporta `mixed` e `unix` stili di sicurezza. SMB supporta `mixed` e `ntfs` stili di sicurezza.	L'impostazione predefinita di NFS è `unix`. L'impostazione predefinita di SMB è `ntfs`.

Configurare un PVC

Questa sezione include istruzioni su come creare un PersistentVolumeClaim (PVC) che utilizza la StorageClass di Kubernetes configurata per richiedere un PV. In caso di successo, è possibile montare il PV su un pod.

Crea il PVC

Un "*PersistentVolumeClaim*" (PVC) è una richiesta di accesso al PersistentVolume sul cluster. Il PVC può essere configurato per richiedere storage di una certa dimensione o modalità di accesso. Utilizzando il StorageClass associato, l'amministratore del cluster può controllare più della sola dimensione e modalità di accesso della PersistentVolume, come ad esempio le prestazioni o il livello di servizio.

Dopo aver creato il backend Trident e la StorageClass, è possibile creare un PVC. Una volta creato il PVC, è possibile montare il volume in un pod.

Esempi di manifest

I seguenti esempi mostrano le opzioni di configurazione di base del PVC.

PVC con accesso RWX

Questo esempio mostra un PVC di base con accesso RWX associato a un StorageClass denominato `basic-csi`.

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: pvc-storage
spec:
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 1Gi
  storageClassName: ontap-gold
```

Esempio di PVC utilizzando iSCSI

Questo esempio mostra un PVC di base per iSCSI con accesso RWO che è associato a un StorageClass denominato `protection-gold`.

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: pvc-san
spec:
  accessModes:
    - ReadWriteOnce
  resources:
```

```
    requests:
      storage: 1Gi
  storageClassName: protection-gold
```

Crea PVC

Passaggi

1. Crea il PVC.

```
kubectl create -f pvc.yaml
```

2. Verificare lo stato del PVC.

```
kubectl get pvc
```

```
NAME         STATUS VOLUME     CAPACITY ACCESS MODES STORAGECLASS AGE
pvc-storage Bound  pv-name 2Gi      RWO                          5m
```

Fate riferimento a "Oggetti Kubernetes e Trident" per i dettagli su come le classi di storage interagiscono con `PersistentVolumeClaim` e sui parametri per controllare come Trident effettua il provisioning dei volumi.

Distribuisci un'applicazione

Una volta creati la classe di storage e il PVC, è possibile montare il PV su un pod. Questa sezione elenca il comando di esempio e la configurazione per collegare il PV a un pod.

Distribuisci un'applicazione di esempio

Passaggi

1. Monta il volume in un pod.

```
kubectl create -f pv-pod.yaml
```

Questi esempi mostrano configurazioni di base per collegare il PVC a un pod: **Configurazione di base**:

```
kind: Pod
apiVersion: v1
metadata:
  name: pv-pod
spec:
  volumes:
    - name: pv-storage
```

```
      persistentVolumeClaim:
       claimName: basic
  containers:
    - name: pv-container
      image: nginx
      ports:
        - containerPort: 80
          name: "http-server"
      volumeMounts:
        - mountPath: "/my/mount/path"
          name: pv-storage
```

NOTA Puoi monitorare l'avanzamento usando `kubectl get pod --watch`.

2. Verificare che il volume sia montato su `/my/mount/path`.

```
kubectl exec -it pv-pod -- df -h /my/mount/path
```

```
Filesystem                                                      Size
Used  Avail Use%  Mounted on
192.168.188.78:/trident_pvc_ae45ed05_3ace_4e7c_9080_d2a83ae03d06  1.1G
320K  1.0G   1%   /my/mount/path
```

Ora puoi eliminare il Pod. L'applicazione Pod non esisterà più, ma il volume rimarrà.

```
kubectl delete pod pv-pod
```

Configura il componente aggiuntivo Trident EKS su un cluster EKS

NetApp Trident semplifica la gestione dello storage Amazon FSx for NetApp ONTAP in Kubernetes, consentendo a sviluppatori e amministratori di concentrarsi sulla distribuzione delle applicazioni. Il componente aggiuntivo NetApp Trident EKS include le patch di sicurezza più recenti, le correzioni di bug ed è convalidato da AWS per funzionare con Amazon EKS. Il componente aggiuntivo EKS consente di garantire costantemente che i cluster Amazon EKS siano sicuri e stabili e di ridurre la quantità di lavoro necessaria per installare, configurare e aggiornare i componenti aggiuntivi.

Prerequisiti

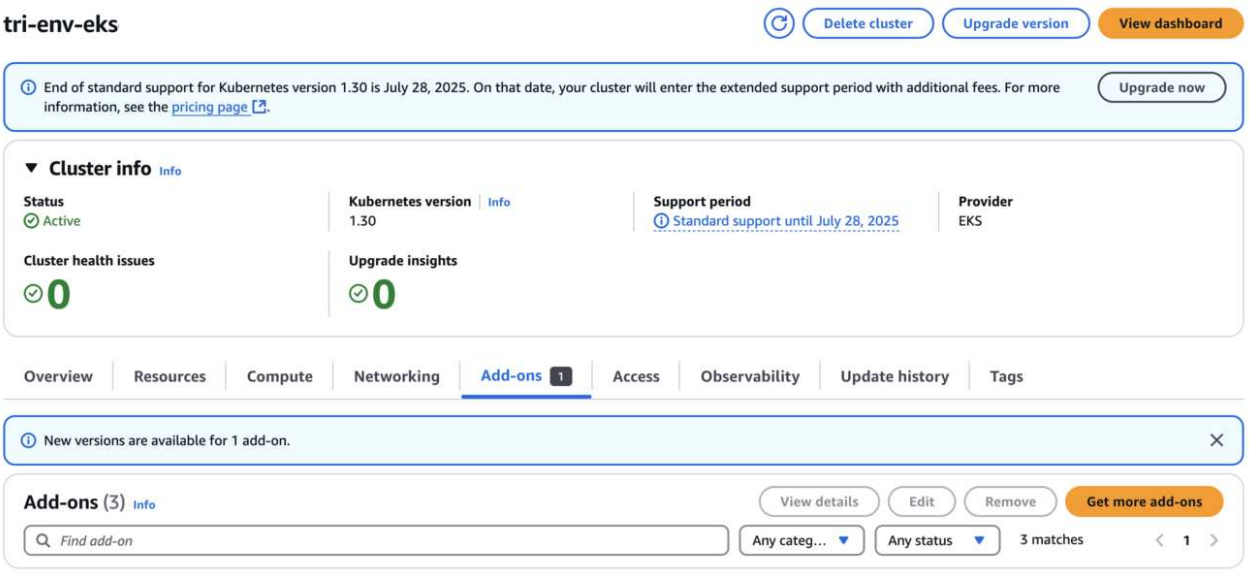
Assicurati di avere quanto segue prima di configurare il componente aggiuntivo Trident per AWS EKS:

- Un account cluster Amazon EKS con autorizzazioni per utilizzare i componenti aggiuntivi. Fai riferimento a "Componenti aggiuntivi Amazon EKS".

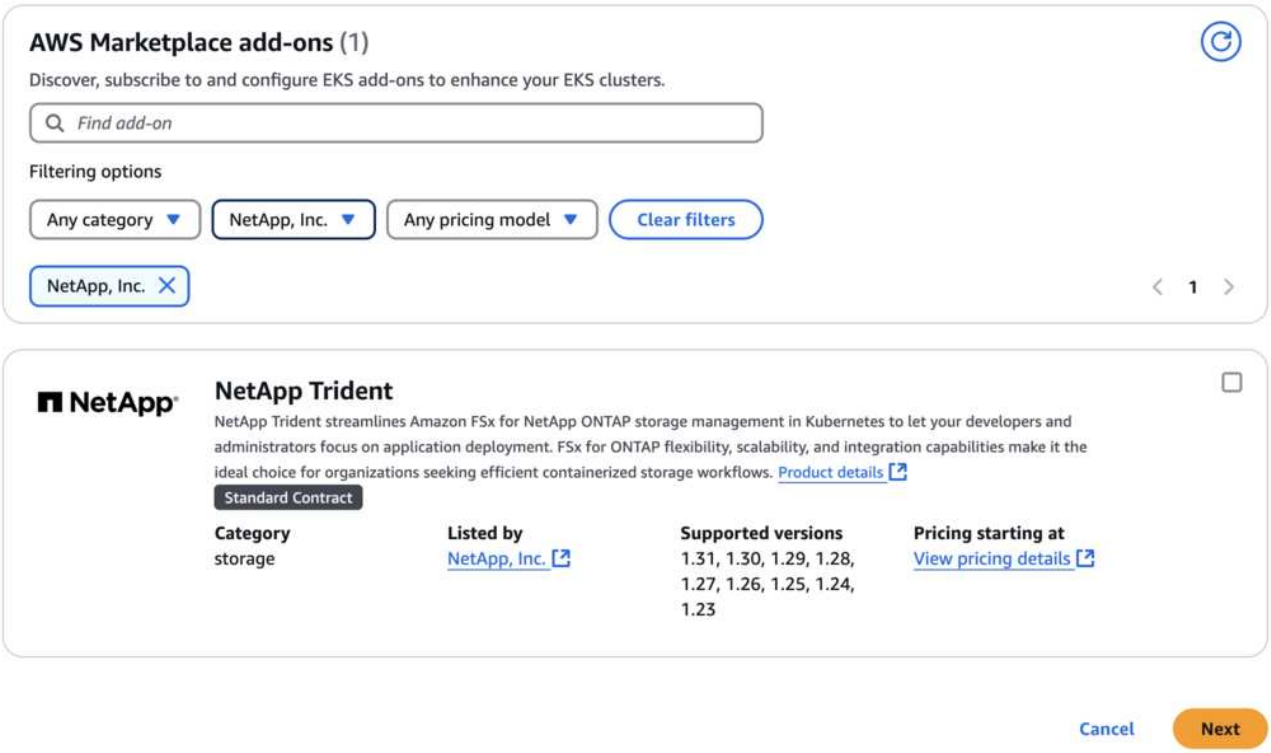
- Autorizzazioni AWS per l'AWS marketplace:
  `"aws-marketplace:ViewSubscriptions",`
  `"aws-marketplace:Subscribe",`
  `"aws-marketplace:Unsubscribe`
- Tipo AMI: Amazon Linux 2 (AL2_x86_64) o Amazon Linux 2 Arm(AL2_ARM_64)
- Tipo di nodo: AMD o ARM
- Un file system Amazon FSx per NetApp ONTAP esistente

Passaggi

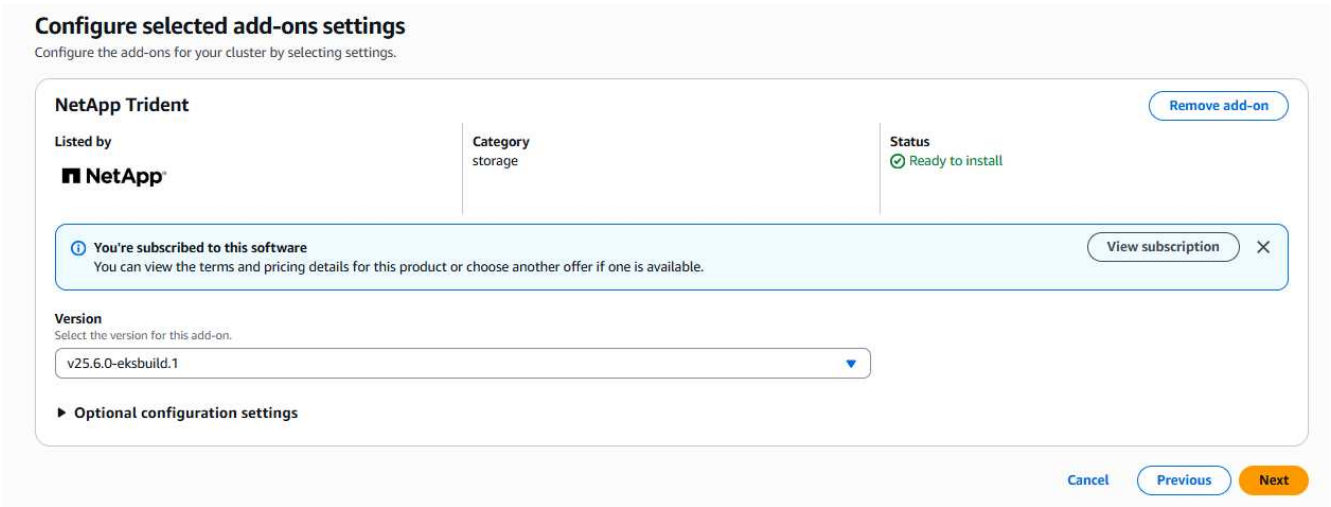
1. Assicurati di creare il ruolo IAM e il segreto AWS per consentire ai pod EKS di accedere alle risorse AWS. Per istruzioni, vedi "Crea un ruolo IAM e un AWS Secret".
2. Nel tuo cluster EKS Kubernetes, vai alla scheda **Componenti aggiuntivi**.



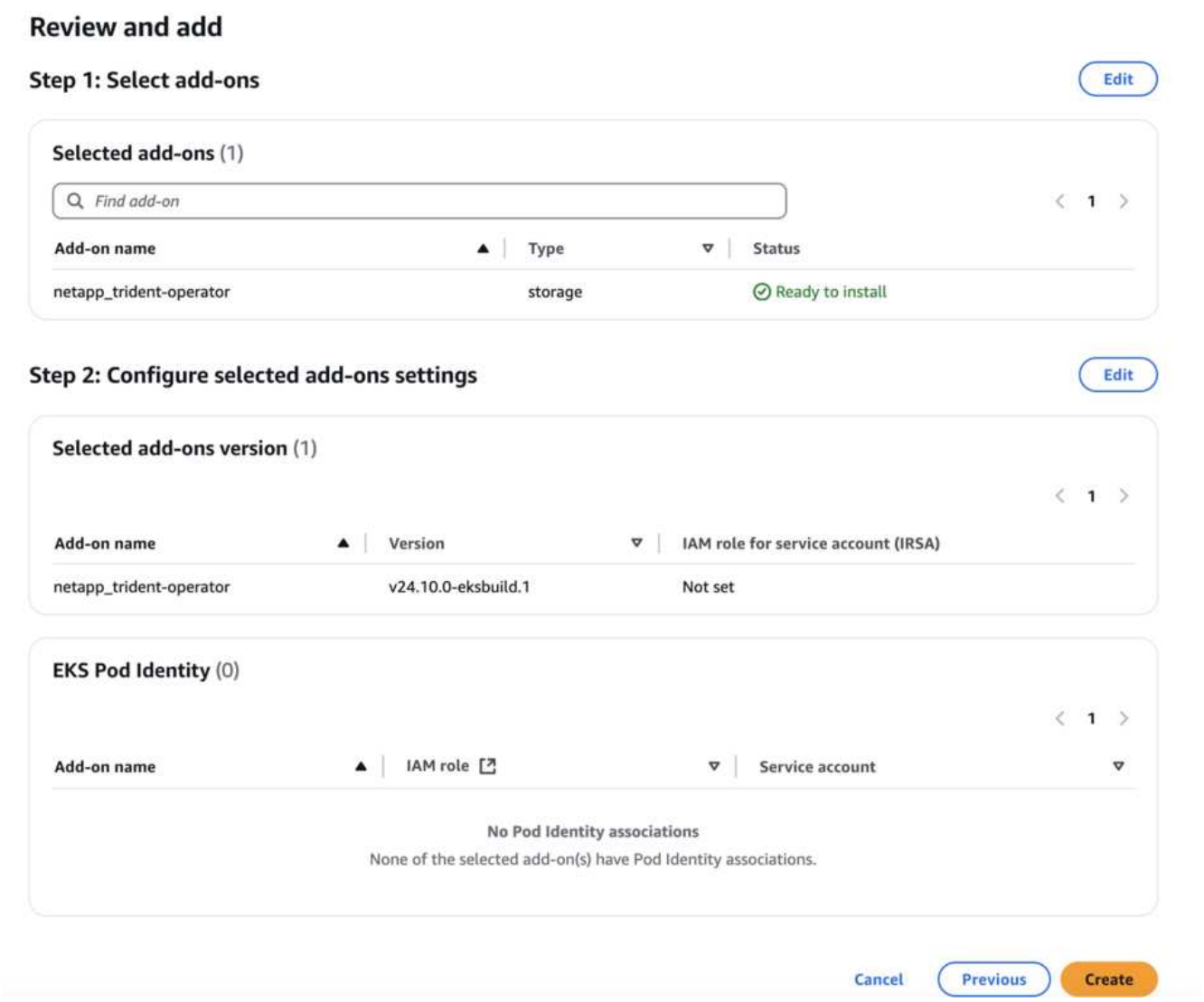
3. Vai su **AWS Marketplace add-ons** e scegli la categoria *storage*.



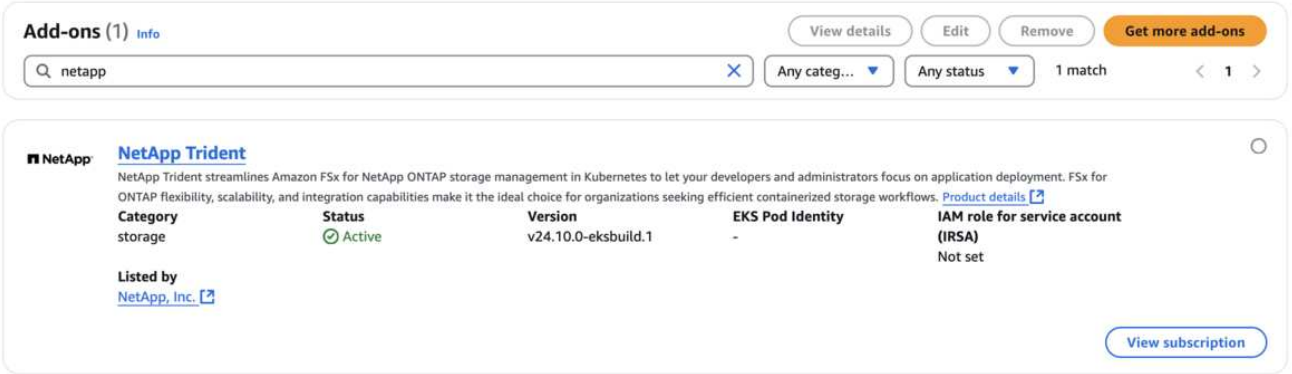
4. Individua **NetApp Trident** e seleziona la casella di controllo per il componente aggiuntivo Trident, quindi fai clic su **Avanti**.
5. Scegli la versione desiderata dell'add-on.



6. Configura le impostazioni aggiuntive richieste.



7. Se si utilizza IRSA (IAM roles for service account), fare riferimento ai passaggi di configurazione aggiuntivi "qui".
8. Seleziona **Crea**.
9. Verificare che lo stato del componente aggiuntivo sia *Active*.



10. Eseguire il seguente comando per verificare che Trident sia installato correttamente sul cluster:

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

11. Continua la configurazione e configura il backend di storage. Per informazioni, vedi "Configura il backend di storage".

Installa/disinstalla il componente aggiuntivo Trident EKS tramite CLI

Installa il componente aggiuntivo Trident EKS di NetApp utilizzando la CLI:

Il seguente comando di esempio installa il componente aggiuntivo Trident EKS: `eksctl create addon --cluster clusterName --name netapp_trident-operator --version v25.6.0-eksbuild.1` (con una versione dedicata)

Il seguente comando di esempio installa il componente aggiuntivo Trident EKS versione 25.6.1: `eksctl create addon --cluster clusterName --name netapp_trident-operator --version v25.6.1-eksbuild.1` (con una versione dedicata)

Il seguente comando di esempio installa il componente aggiuntivo Trident EKS versione 25.6.2: `eksctl create addon --cluster clusterName --name netapp_trident-operator --version v25.6.2-eksbuild.1` (con una versione dedicata)

Disinstalla il componente aggiuntivo NetApp Trident EKS utilizzando la CLI:

Il seguente comando disinstalla il componente aggiuntivo Trident EKS:

```
eksctl delete addon --cluster K8s-arm --name netapp_trident-operator
```

Crea backend con kubectl

Un backend definisce la relazione tra Trident e un sistema storage. Indica a Trident come comunicare con quel sistema storage e come Trident deve eseguire il provisioning dei volumi da esso. Dopo l'installazione di Trident, il passo successivo è creare un backend. La `TridentBackendConfig` Custom Resource Definition (CRD) consente di creare e gestire i backend di Trident direttamente attraverso l'interfaccia di Kubernetes. Puoi farlo utilizzando `kubectl` o lo strumento CLI equivalente per la tua distribuzione Kubernetes.

`TridentBackendConfig`

`TridentBackendConfig` (`tbc`, `tbconfig`, `tbackendconfig`) è un CRD frontend e namespaced che consente di gestire i backend Trident utilizzando `kubectl`. Gli amministratori di Kubernetes e dello storage possono ora creare e gestire i backend direttamente attraverso la CLI di Kubernetes senza richiedere un'utility a riga di comando dedicata (`tridentctl`).

Alla creazione di un `TridentBackendConfig` oggetto, avviene quanto segue:

- Un backend viene creato automaticamente da Trident in base alla configurazione che fornisci. Questo è rappresentato internamente come un `TridentBackend` (`tbe`, `tridentbackend`) CR.
- Il `TridentBackendConfig` è legato in modo univoco a un `TridentBackend` che è stato creato da Trident.

Ogni `TridentBackendConfig` mantiene una mappatura uno-a-uno con un `TridentBackend`. Il primo è l'interfaccia fornita all'utente per progettare e configurare i backend; il secondo è il modo in cui Trident rappresenta l'effettivo oggetto backend.

ATTENZIONE	`TridentBackend` I CR sono creati automaticamente da Trident. Non dovresti modificarli. Se vuoi apportare aggiornamenti ai backend, fallo modificando l'oggetto `TridentBackendConfig`.

Vedere il seguente esempio per il formato del `TridentBackendConfig` CR:

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-ontap-san
spec:
  version: 1
  backendName: ontap-san-backend
  storageDriverName: ontap-san
  managementLIF: 10.0.0.1
  dataLIF: 10.0.0.2
  svm: trident_svm
  credentials:
    name: backend-tbc-ontap-san-secret
```

È anche possibile dare un'occhiata agli esempi nella directory "trident-installer" per configurazioni di esempio per la piattaforma di storage/servizio desiderato.

Il `spec` accetta parametri di configurazione specifici del backend. In questo esempio, il backend utilizza il driver di storage `ontap-san` e utilizza i parametri di configurazione qui tabulati. Per l'elenco delle opzioni di configurazione per il driver di storage desiderato, fare riferimento al "informazioni sulla configurazione del backend per il tuo storage driver".

La `spec` sezione include anche `credentials` e `deletionPolicy` campi, che sono stati introdotti di recente nella `TridentBackendConfig` CR:

- `credentials`: Questo parametro è un campo obbligatorio e contiene le credenziali utilizzate per l'autenticazione con il sistema storage/servizio. Questo è impostato su un Kubernetes Secret creato dall'utente. Le credenziali non possono essere passate in testo normale e genereranno un errore.
- `deletionPolicy`: Questo campo definisce cosa dovrebbe accadere quando `TridentBackendConfig` viene eliminato. Può assumere uno dei due valori possibili:
  - `delete`: Ciò comporta l'eliminazione sia di `TridentBackendConfig` CR che del backend associato. Questo è il valore predefinito.
  - `retain`: Quando un `TridentBackendConfig` CR viene eliminato, la definizione del backend sarà ancora presente e potrà essere gestita con `tridentctl`. Impostando la policy di eliminazione su `retain`, gli utenti potranno effettuare il downgrade a una versione precedente (pre-21.04) e mantenere i backend creati. Il valore di questo campo può essere aggiornato dopo che un `TridentBackendConfig` è stato creato.

NOTA	Il nome di un backend viene impostato usando `spec.backendName`. Se non specificato, il nome del backend viene impostato sul nome dell' `TridentBackendConfig`oggetto (`metadata.name`). Si consiglia di impostare esplicitamente i nomi dei backend usando `spec.backendName`.
SUGGERIMENTO	I backend creati con `tridentctl` non hanno un oggetto `TridentBackendConfig` associato. Puoi scegliere di gestire tali backend con `kubectl` creando un `TridentBackendConfig` CR. È necessario prestare attenzione a specificare parametri di configurazione identici (come `spec.backendName`, `spec.storagePrefix`, `spec.storageDriverName` e così via). Trident assocerà automaticamente il nuovo `TridentBackendConfig` creato con il backend preesistente.

Panoramica dei passaggi

Per creare un nuovo backend utilizzando `kubectl`, dovresti fare quanto segue:

1. Crea un "Kubernetes Secret". Il secret contiene le credenziali di cui Trident ha bisogno per comunicare con il cluster/servizio di storage.
2. Crea un `TridentBackendConfig` oggetto. Questo contiene informazioni specifiche sul cluster/servizio di storage e fa riferimento al segreto creato nel passaggio precedente.

Dopo aver creato un backend, puoi osservarne lo stato utilizzando `kubectl get tbc <tbc-name> -n <trident-namespace>` e raccogliere ulteriori dettagli.

Passaggio 1: crea un Kubernetes Secret

Crea un Secret che contiene le credenziali di accesso per il backend. Questo è univoco per ogni servizio/piattaforma di storage. Ecco un esempio:

```
kubectl -n trident create -f backend-tbc-ontap-san-secret.yaml
```

```
apiVersion: v1
kind: Secret
metadata:
  name: backend-tbc-ontap-san-secret
type: Opaque
stringData:
  username: cluster-admin
  password: password
```

Questa tabella riassume i campi che devono essere inclusi nel Secret per ogni storage platform:

Descrizione dei campi segreti della storage platform	Segreto	Descrizione dei campi
Azure NetApp Files	clientID	L'ID client da una registrazione dell'app
Element (NetApp HCI/SolidFire)	Punto finale	MVIP per il SolidFire cluster con credenziali tenant
ONTAP	nome utente	Nome utente per connettersi al cluster/SVM. Utilizzata per l'autenticazione basata sulle credenziali
ONTAP	password	Password per connettersi al cluster/SVM. Utilizzata per l'autenticazione basata sulle credenziali
ONTAP	clientPrivateKey	Valore codificato in Base64 della chiave privata del client. Utilizzato per l'autenticazione basata su certificato
ONTAP	chapUsername	Nome utente in entrata. Obbligatorio se useCHAP=true. Per `ontap-san` e `ontap-san-economy`
ONTAP	chapInitiatorSecret	Segreto dell'iniziatore CHAP. Obbligatorio se useCHAP=true. Per `ontap-san` e `ontap-san-economy`
ONTAP	chapTargetUsername	Nome utente di destinazione. Obbligatorio se useCHAP=true. Per `ontap-san` e `ontap-san-economy`
ONTAP	chapTargetInitiatorSecret	Segreto dell'iniziatore di destinazione CHAP. Obbligatorio se useCHAP=true. Per `ontap-san` e `ontap-san-economy`

Il Secret creato in questo passaggio verrà referenziato nel campo `spec.credentials` dell'oggetto `TridentBackendConfig` che viene creato nel passaggio successivo.

Passaggio 2: crea la `TridentBackendConfig` CR

Ora sei pronto per creare il tuo `TridentBackendConfig` CR. In questo esempio, un backend che utilizza il `ontap-san` driver viene creato utilizzando l'oggetto `TridentBackendConfig` mostrato di seguito:

```
kubectl -n trident create -f backend-tbc-ontap-san.yaml
```

```
apiVersion: trident.netapp.io/v1
```

```
kind: TridentBackendConfig
metadata:
  name: backend-tbc-ontap-san
spec:
  version: 1
  backendName: ontap-san-backend
  storageDriverName: ontap-san
  managementLIF: 10.0.0.1
  dataLIF: 10.0.0.2
  svm: trident_svm
  credentials:
    name: backend-tbc-ontap-san-secret
```

Fase 3: verificare lo stato del `TridentBackendConfig` CR

Ora che hai creato la `TridentBackendConfig` CR, puoi verificarne lo stato. Vedere il seguente esempio:

```
kubectl -n trident get tbc backend-tbc-ontap-san
NAME                     BACKEND NAME          BACKEND UUID
PHASE   STATUS
backend-tbc-ontap-san    ontap-san-backend     8d24fce7-6f60-4d4a-8ef6-
bab2699e6ab8   Bound   Success
```

Un backend è stato creato correttamente e associato al `TridentBackendConfig` CR.

La fase può assumere uno dei seguenti valori:

- `Bound`: Il `TridentBackendConfig` CR è associato a un backend e tale backend contiene `configRef` impostato sull' `TridentBackendConfig` uid del CR.
- `Unbound`: Rappresentato usando `""`. L' `TridentBackendConfig`oggetto non è vincolato a un backend. Tutte le nuove `TridentBackendConfig` CR sono in questa fase per impostazione predefinita. Dopo il cambio di fase, non può tornare a Unbound.
- `Deleting`: Il `TridentBackendConfig` CR `deletionPolicy` è stato impostato per l'eliminazione. Quando il `TridentBackendConfig` CR viene eliminato, passa allo stato Eliminazione in corso.
  - Se non esistono persistent volume claims (PVC) sul backend, l'eliminazione del `TridentBackendConfig` comporterà che Trident elimini sia il backend sia il `TridentBackendConfig` CR.
  - Se uno o più PVC sono presenti sul backend, questo entra in uno stato di eliminazione. Il `TridentBackendConfig` CR successivamente entra anch'esso in fase di eliminazione. Il backend e `TridentBackendConfig` vengono eliminati solo dopo che tutti i PVC sono stati eliminati.
- `Lost`: Il backend associato al `TridentBackendConfig` CR è stato eliminato accidentalmente o deliberatamente e il `TridentBackendConfig` CR contiene ancora un riferimento al backend eliminato. Il `TridentBackendConfig` CR può comunque essere eliminato indipendentemente dal `deletionPolicy` valore.
- `Unknown`: Trident non è in grado di determinare lo stato o l'esistenza del backend associato al

`TridentBackendConfig` CR. Ad esempio, se il server API non risponde o se il `tridentbackends.trident.netapp.io` CRD è mancante. Ciò potrebbe richiedere un intervento.

A questo punto, un backend è stato creato con successo! Ci sono diverse operazioni che possono essere gestite in aggiunta, come "aggiornamenti del backend ed eliminazioni del backend".

(Facoltativo) Passaggio 4: Ottieni maggiori dettagli

Puoi eseguire il seguente comando per ottenere maggiori informazioni sul tuo backend:

```
kubectl -n trident get tbc backend-tbc-ontap-san -o wide
```

```
NAME                      BACKEND NAME         BACKEND UUID
PHASE   STATUS    STORAGE DRIVER   DELETION POLICY
backend-tbc-ontap-san   ontap-san-backend   8d24fce7-6f60-4d4a-8ef6-
bab2699e6ab8   Bound   Success   ontap-san        delete
```

Inoltre, è possibile ottenere anche un dump YAML/JSON di `TridentBackendConfig`.

```
kubectl -n trident get tbc backend-tbc-ontap-san -o yaml
```

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  creationTimestamp: 2021-04-21T20:45:11Z
  finalizers:
    - trident.netapp.io
  generation: 1
  name: backend-tbc-ontap-san
  namespace: trident
  resourceVersion: "947143"
  uid: 35b9d777-109f-43d5-8077-c74a4559d09c
spec:
  backendName: ontap-san-backend
  credentials:
    name: backend-tbc-ontap-san-secret
  managementLIF: 10.0.0.1
  dataLIF: 10.0.0.2
  storageDriverName: ontap-san
  svm: trident_svm
  version: 1
status:
  backendInfo:
```

```
    backendName: ontap-san-backend
    backendUUID: 8d24fce7-6f60-4d4a-8ef6-bab2699e6ab8
  deletionPolicy: delete
  lastOperationStatus: Success
  message: Backend 'ontap-san-backend' created
  phase: Bound
```

`backendInfo` contiene il `backendName` e il `backendUUID` del backend che è stato creato in risposta al `TridentBackendConfig` CR. Il `lastOperationStatus` campo rappresenta lo stato dell'ultima operazione del `TridentBackendConfig` CR, che può essere attivata dall'utente (ad esempio, l'utente ha modificato qualcosa in `spec`) o attivata da Trident (ad esempio, durante i riavvii di Trident). Può essere Success o Failed. `phase` rappresenta lo stato della relazione tra il `TridentBackendConfig` CR e il backend. Nell'esempio sopra, `phase` ha il valore Bound, il che significa che il `TridentBackendConfig` CR è associato al backend.

È possibile eseguire il comando `kubectl -n trident describe tbc <tbc-cr-name>` per ottenere i dettagli dei registri eventi.

ATTENZIONE	Non è possibile aggiornare o eliminare un backend che contiene un oggetto associato `TridentBackendConfig` utilizzando `tridentctl`. Per comprendere i passaggi necessari per passare tra `tridentctl` e `TridentBackendConfig`, "vedi qui".

Gestisci i backend

Gestisci i backend di Trident con kubectl

Scopri come eseguire operazioni di gestione del backend utilizzando `kubectl`.

Elimina un backend

Eliminando un `TridentBackendConfig`, si indica a Trident di eliminare/mantenere i backend (in base a `deletionPolicy`). Per eliminare un backend, assicurarsi che `deletionPolicy` sia impostato su delete. Per eliminare solo `TridentBackendConfig`, assicurarsi che `deletionPolicy` sia impostato su retain. Questo garantisce che il backend sia ancora presente e possa essere gestito utilizzando `tridentctl`.

Eseguire il seguente comando:

```
kubectl delete tbc <tbc-name> -n trident
```

Trident non elimina i Kubernetes Secrets che erano in uso da `TridentBackendConfig`. L'utente Kubernetes è responsabile della pulizia dei secrets. È necessario prestare attenzione quando si eliminano i secrets. Dovresti eliminare i secrets solo se non sono in uso dai backends.

Visualizza i backend esistenti

Eseguire il seguente comando:

```
kubectl get tbc -n trident
```

È anche possibile eseguire `tridentctl get backend -n trident` o `tridentctl get backend -o yaml -n trident` per ottenere un elenco di tutti i backend esistenti. Questo elenco includerà anche i backend creati con `tridentctl`.

Aggiorna un backend

Possono esserci molteplici motivi per aggiornare un backend:

- Le credenziali per il sistema storage sono cambiate. Per aggiornare le credenziali, il Kubernetes Secret utilizzato nell' `TridentBackendConfig`oggetto deve essere aggiornato. Trident aggiornerà automaticamente il backend con le credenziali più recenti fornite. Eseguire il seguente comando per aggiornare il Kubernetes Secret:

```
kubectl apply -f <updated-secret-file.yaml> -n trident
```

- È necessario aggiornare i parametri (ad esempio il nome dell'ONTAP SVM utilizzato).
  - È possibile aggiornare `TridentBackendConfig` gli oggetti direttamente tramite Kubernetes utilizzando il seguente comando:

```
kubectl apply -f <updated-backend-file.yaml>
```

  - In alternativa, puoi apportare modifiche al CR `TridentBackendConfig` esistente utilizzando il seguente comando:

```
kubectl edit tbc <tbc-name> -n trident
```

NOTA

- Se un aggiornamento del backend fallisce, il backend continua a mantenere l'ultima configurazione nota. Puoi visualizzare i log per determinarne la causa eseguendo `kubectl get tbc <tbc-name> -o yaml -n trident` o `kubectl describe tbc <tbc-name> -n trident`.
- Dopo aver identificato e corretto il problema con il file di configurazione, puoi rieseguire il comando di aggiornamento.

Gestisci i backend di Trident con tridentctl

Scopri come eseguire operazioni di gestione del backend utilizzando `tridentctl`.

Crea un backend

Dopo aver creato un "file di configurazione backend", esegui il seguente comando:

```
tridentctl create backend -f <backend-file> -n trident
```

Se la creazione del backend fallisce, si è verificato un errore nella configurazione del backend. Puoi visualizzare i log per determinarne la causa eseguendo il seguente comando:

```
tridentctl logs -n trident
```

Dopo aver identificato e corretto il problema con il file di configurazione, puoi semplicemente eseguire il `create` comando nuovamente.

Elimina un backend

Per eliminare un backend da Trident, procedere come segue:

1. Recupera il nome del backend:

```
tridentctl get backend -n trident
```

2. Elimina il backend:

```
tridentctl delete backend <backend-name> -n trident
```

NOTA Se Trident ha eseguito il provisioning di volumi e snapshot da questo backend che sono ancora esistenti, l'eliminazione del backend impedisce il provisioning di nuovi volumi da parte sua. Il backend continuerà a esistere in uno stato di "Eliminazione".

Visualizza i backend esistenti

Per visualizzare i backend di cui Trident è a conoscenza, procedere come segue:

- Per ottenere un riepilogo, eseguire il seguente comando:

```
tridentctl get backend -n trident
```

- Per ottenere tutti i dettagli, eseguire il seguente comando:

```
tridentctl get backend -o json -n trident
```

Aggiorna un backend

Dopo aver creato un nuovo file di configurazione backend, eseguire il seguente comando:

```
tridentctl update backend <backend-name> -f <backend-file> -n trident
```

Se l'aggiornamento del backend fallisce, si è verificato un errore nella configurazione del backend o hai tentato un aggiornamento non valido. Puoi visualizzare i log per determinarne la causa eseguendo il seguente comando:

```
tridentctl logs -n trident
```

Dopo aver identificato e corretto il problema con il file di configurazione, puoi semplicemente eseguire il `update` comando nuovamente.

Identificare le classi di storage che utilizzano un backend

Questo è un esempio del tipo di domande a cui è possibile rispondere con il JSON che `tridentctl` restituisce per gli oggetti backend. Questo utilizza l' `jq`utility, che è necessario installare.

```
tridentctl get backend -o json | jq '[.items[] | {backend: .name,
storageClasses: [.storage[].storageClasses]|unique}]'
```

Ciò vale anche per i backend creati utilizzando `TridentBackendConfig`.

Scegli un metodo di gestione del backend Trident

Scopri i diversi modi per gestire i backend in Trident.

Opzioni per la gestione dei backend

Con l'introduzione di `TridentBackendConfig`, gli amministratori hanno ora due modi unici per gestire i backend. Questo pone le seguenti domande:

- I backend creati utilizzando `tridentctl` possono essere gestiti con `TridentBackendConfig`?
- I backend creati utilizzando `TridentBackendConfig`possono essere gestiti utilizzando `tridentctl`?

Gestisci `tridentctl` backend utilizzando `TridentBackendConfig`

Questa sezione illustra i passaggi necessari per gestire i backend che sono stati creati utilizzando `tridentctl` direttamente tramite l'interfaccia Kubernetes creando `TridentBackendConfig` oggetti.

Questo si applicherà ai seguenti scenari:

- Backend preesistenti, che non hanno un `TridentBackendConfig` perché sono stati creati con `tridentctl`.
- Nuovi backend che sono stati creati con `tridentctl`, mentre esistono altri oggetti `TridentBackendConfig`.

In entrambi gli scenari, i backend continueranno a essere presenti, con Trident che pianifica i volumi e opera su di essi. Gli amministratori hanno due possibilità:

- Continua a utilizzare `tridentctl` per gestire i backend creati utilizzandolo.
- Associa i backend creati usando `tridentctl` a un nuovo `TridentBackendConfig` oggetto. Così facendo, i backend saranno gestiti usando `kubectl` e non `tridentctl`.

Per gestire un backend preesistente utilizzando `kubectl`, è necessario creare un `TridentBackendConfig`

che si colleghi al backend esistente. Ecco una panoramica di come funziona:

1. Crea un segreto Kubernetes. Il segreto contiene le credenziali Trident necessarie per comunicare con il cluster/servizio di storage.
2. Crea un `TridentBackendConfig` oggetto. Questo contiene informazioni specifiche sul cluster/servizio di storage e fa riferimento al segreto creato nel passaggio precedente. È necessario prestare attenzione a specificare parametri di configurazione identici (ad esempio, `spec.backendName`, `spec.storagePrefix`, `spec.storageDriverName` e così via). `spec.backendName`deve essere impostato sul nome del backend esistente.

Passaggio 0: Identificare il backend

Per creare un `TridentBackendConfig` che si leghi a un backend esistente, è necessario ottenere la configurazione del backend. In questo esempio, supponiamo che un backend sia stato creato utilizzando la seguente definizione JSON:

```
tridentctl get backend ontap-nas-backend -n trident
+---------------------+----------------
+--------------------------------------+--------+---------+
|          NAME       | STORAGE DRIVER |                 UUID
| STATE   | VOLUMES |
+---------------------+----------------
+--------------------------------------+--------+---------+
| ontap-nas-backend   | ontap-nas      | 52f2eb10-e4c6-4160-99fc-
96b3be5ab5d7 | online |      25 |
+---------------------+----------------
+--------------------------------------+--------+---------+
```

```
cat ontap-nas-backend.json
```

```
{
  "version": 1,
  "storageDriverName": "ontap-nas",
  "managementLIF": "10.10.10.1",
  "dataLIF": "10.10.10.2",
  "backendName": "ontap-nas-backend",
  "svm": "trident_svm",
  "username": "cluster-admin",
  "password": "admin-password",
  "defaults": {
    "spaceReserve": "none",
    "encryption": "false"
  },
  "labels": {
    "store": "nas_store"
```

```
  },
  "region": "us_east_1",
  "storage": [
    {
      "labels": {
        "app": "msoffice",
        "cost": "100"
      },
      "zone": "us_east_1a",
      "defaults": {
        "spaceReserve": "volume",
        "encryption": "true",
        "unixPermissions": "0755"
      }
    },
    {
      "labels": {
        "app": "mysqldb",
        "cost": "25"
      },
      "zone": "us_east_1d",
      "defaults": {
        "spaceReserve": "volume",
        "encryption": "false",
        "unixPermissions": "0775"
      }
    }
  ]
}
```

Passaggio 1: crea un Kubernetes Secret

Crea un Secret che contiene le credenziali per il backend, come mostrato in questo esempio:

```
cat tbc-ontap-nas-backend-secret.yaml
```

```
apiVersion: v1
kind: Secret
metadata:
  name: ontap-nas-backend-secret
type: Opaque
stringData:
  username: cluster-admin
```

```
  password: admin-password
```

```
kubectl create -f tbc-ontap-nas-backend-secret.yaml -n trident
secret/backend-tbc-ontap-san-secret created
```

Passaggio 2: crea un `TridentBackendConfig` CR

Il passaggio successivo consiste nel creare una `TridentBackendConfig` CR che si associ automaticamente a quella preesistente `ontap-nas-backend` (come in questo esempio). Assicurarsi che siano soddisfatti i seguenti requisiti:

- Lo stesso nome del backend è definito in `spec.backendName`.
- I parametri di configurazione sono identici al backend originale.
- I pool virtuali (se presenti) devono mantenere lo stesso ordine come nel backend originale.
- Le credenziali vengono fornite tramite un Kubernetes Secret e non in testo normale.

In questo caso, il `TridentBackendConfig` apparirà così:

```
cat backend-tbc-ontap-nas.yaml
```

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: tbc-ontap-nas-backend
spec:
  version: 1
  storageDriverName: ontap-nas
  managementLIF: 10.10.10.1
  dataLIF: 10.10.10.2
  backendName: ontap-nas-backend
  svm: trident_svm
  credentials:
    name: mysecret
  defaults:
    spaceReserve: none
    encryption: 'false'
  labels:
    store: nas_store
  region: us_east_1
  storage:
  - labels:
      app: msoffice
      cost: '100'
```

```
    zone: us_east_1a
    defaults:
      spaceReserve: volume
      encryption: 'true'
      unixPermissions: '0755'
  - labels:
      app: mysqldb
      cost: '25'
    zone: us_east_1d
    defaults:
      spaceReserve: volume
      encryption: 'false'
      unixPermissions: '0775'
```

```
kubectl create -f backend-tbc-ontap-nas.yaml -n trident
tridentbackendconfig.trident.netapp.io/tbc-ontap-nas-backend created
```

Fase 3: verificare lo stato del `TridentBackendConfig` CR

Dopo che la `TridentBackendConfig` è stata creata, la sua fase deve essere `Bound`. Dovrebbe inoltre riflettere lo stesso nome backend e UUID del backend esistente.

```
kubectl get tbc tbc-ontap-nas-backend -n trident
NAME                    BACKEND NAME          BACKEND UUID
PHASE   STATUS
tbc-ontap-nas-backend  ontap-nas-backend     52f2eb10-e4c6-4160-99fc-
96b3be5ab5d7   Bound   Success

#confirm that no new backends were created (i.e., TridentBackendConfig did
not end up creating a new backend)
tridentctl get backend -n trident
+---------------------+----------------
+--------------------------------------+--------+---------+
|          NAME       | STORAGE DRIVER |                 UUID
| STATE  | VOLUMES |
+---------------------+----------------
+--------------------------------------+--------+---------+
| ontap-nas-backend   | ontap-nas      | 52f2eb10-e4c6-4160-99fc-
96b3be5ab5d7 | online |      25 |
+---------------------+----------------
+--------------------------------------+--------+---------+
```

Il backend sarà ora completamente gestito utilizzando l' `tbc-ontap-nas-backend` `TridentBackendConfig`oggetto.

`tridentctl` può essere utilizzato per elencare i backend che sono stati creati usando `TridentBackendConfig`. Inoltre, gli amministratori possono anche scegliere di gestire completamente tali backend tramite `tridentctl` eliminando `TridentBackendConfig` e assicurandosi che `spec.deletionPolicy` sia impostato su `retain`.

Passaggio 0: Identificare il backend

Ad esempio, supponiamo che il seguente backend sia stato creato utilizzando `TridentBackendConfig`:

```
kubectl get tbc backend-tbc-ontap-san -n trident -o wide
NAME                    BACKEND NAME        BACKEND UUID
PHASE   STATUS    STORAGE DRIVER   DELETION POLICY
backend-tbc-ontap-san   ontap-san-backend   81abcb27-ea63-49bb-b606-
0a5315ac5f82   Bound   Success   ontap-san        delete

tridentctl get backend ontap-san-backend -n trident
+------------------+---------------
+--------------------------------------+--------+---------+
|       NAME        | STORAGE DRIVER |                 UUID
| STATE  | VOLUMES |
+------------------+---------------
+--------------------------------------+--------+---------+
| ontap-san-backend | ontap-san      | 81abcb27-ea63-49bb-b606-
0a5315ac5f82 | online |      33 |
+------------------+---------------
+--------------------------------------+--------+---------+
```

Dall'output si vede che `TridentBackendConfig` è stato creato correttamente ed è associato a un backend [osservare l'UUID del backend].

Passaggio 1: confermare `deletionPolicy` è impostato su `retain`

Diamo un'occhiata al valore di `deletionPolicy`. Questo deve essere impostato su `retain`. Questo garantisce che quando un `TridentBackendConfig` CR viene eliminato, la definizione del backend sarà ancora presente e potrà essere gestita con `tridentctl`.

```
kubectl get tbc backend-tbc-ontap-san -n trident -o wide
NAME                    BACKEND NAME        BACKEND UUID
PHASE   STATUS    STORAGE DRIVER   DELETION POLICY
backend-tbc-ontap-san   ontap-san-backend   81abcb27-ea63-49bb-b606-
0a5315ac5f82   Bound   Success   ontap-san        delete
```

```
# Patch value of deletionPolicy to retain
kubectl patch tbc backend-tbc-ontap-san --type=merge -p
'{"spec":{"deletionPolicy":"retain"}}' -n trident
tridentbackendconfig.trident.netapp.io/backend-tbc-ontap-san patched

#Confirm the value of deletionPolicy
kubectl get tbc backend-tbc-ontap-san -n trident -o wide
NAME                    BACKEND NAME        BACKEND UUID
PHASE   STATUS    STORAGE DRIVER   DELETION POLICY
backend-tbc-ontap-san   ontap-san-backend   81abcb27-ea63-49bb-b606-
0a5315ac5f82   Bound   Success   ontap-san        retain
```

NOTA Non procedere al passaggio successivo a meno che `deletionPolicy` non sia impostato su `retain`.

Passaggio 2: Eliminare il `TridentBackendConfig` CR

Il passaggio finale consiste nell'eliminare il `TridentBackendConfig` CR. Dopo aver verificato che `deletionPolicy` è impostato su `retain`, è possibile procedere con l'eliminazione:

```
kubectl delete tbc backend-tbc-ontap-san -n trident
tridentbackendconfig.trident.netapp.io "backend-tbc-ontap-san" deleted

tridentctl get backend ontap-san-backend -n trident
+-------------------+----------------
+--------------------------------------+--------+---------+
|       NAME        | STORAGE DRIVER |                 UUID
| STATE  | VOLUMES |
+-------------------+----------------
+--------------------------------------+--------+---------+
| ontap-san-backend | ontap-san      | 81abcb27-ea63-49bb-b606-
0a5315ac5f82 | online |      33 |
+-------------------+----------------
+--------------------------------------+--------+---------+
```

Dopo l'eliminazione dell'oggetto `TridentBackendConfig`, Trident lo rimuove semplicemente senza eliminare effettivamente il backend stesso.

Crea e gestisci classi di archiviazione

Crea una classe di storage Trident

Configura un oggetto Kubernetes StorageClass e crea la storage class per istruire Trident su come effettuare il provisioning dei volumi.

Configura un oggetto Kubernetes StorageClass

Il "Oggetto Kubernetes StorageClass" identifica Trident come il provisioner utilizzato per quella classe e istruisce Trident su come eseguire il provisioning di un volume. Ad esempio:

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontap-gold
provisioner: csi.trident.netapp.io
mountOptions:
  - nfsvers=3
  - nolock
parameters:
  backendType: "ontap-nas"
  media: "ssd"
allowVolumeExpansion: true
volumeBindingMode: Immediate
```

Fate riferimento a "Oggetti Kubernetes e Trident" per i dettagli su come le classi di storage interagiscono con `PersistentVolumeClaim` e sui parametri per controllare come Trident effettua il provisioning dei volumi.

Creare una storage class

Dopo aver creato l'oggetto StorageClass, puoi creare la storage class. Esempi di storage class fornisce alcuni esempi di base che puoi utilizzare o modificare.

Passaggi

1. Si tratta di un oggetto Kubernetes, quindi usa `kubectl` per crearlo in Kubernetes.

   ```
   kubectl create -f sample-input/storage-class-basic-csi.yaml
   ```

2. Ora dovresti vedere una classe di storage **basic-csi** sia in Kubernetes che in Trident, e Trident dovrebbe aver rilevato i pool sul backend.

   ```
   kubectl get sc basic-csi
   ```

```
NAME         PROVISIONER             AGE
basic-csi    csi.trident.netapp.io   15h
```

```
./tridentctl -n trident get storageclass basic-csi -o json
```

```
{
  "items": [
    {
      "Config": {
        "version": "1",
        "name": "basic-csi",
        "attributes": {
          "backendType": "ontap-nas"
        },
        "storagePools": null,
        "additionalStoragePools": null
      },
      "storage": {
        "ontapnas_10.0.0.1": [
          "aggr1",
          "aggr2",
          "aggr3",
          "aggr4"
        ]
      }
    }
  ]
}
```

Esempi di storage class

Trident fornisce "definizioni di storage class semplici per specifici backend".

In alternativa, puoi modificare `sample-input/storage-class-csi.yaml.templ` il file fornito con il programma di installazione e sostituire *`BACKEND_TYPE`* con il nome del driver di storage.

```
./tridentctl -n trident get backend
+-------------+----------------+--------------------------------------
+--------+---------+
|    NAME     | STORAGE DRIVER |                 UUID                 |
STATE  | VOLUMES |
+-------------+----------------+--------------------------------------
+--------+---------+
| nas-backend | ontap-nas      | 98e19b74-aec7-4a3d-8dcf-128e5033b214 |
online |       0 |
+-------------+----------------+--------------------------------------
+--------+---------+

cp sample-input/storage-class-csi.yaml.templ sample-input/storage-class-
basic-csi.yaml
```

```
# Modify __BACKEND_TYPE__ with the storage driver field above (e.g.,
ontap-nas)
vi sample-input/storage-class-basic-csi.yaml
```

Gestisci le classi di storage di Trident

È possibile visualizzare le classi di storage esistenti, impostare una classe di storage predefinita, identificare il backend della classe di storage ed eliminare le classi di storage.

Visualizza le classi di storage esistenti

- Per visualizzare le classi di storage Kubernetes esistenti, eseguire il seguente comando:

```
kubectl get storageclass
```

- Per visualizzare i dettagli della storage class Kubernetes, eseguire il seguente comando:

```
kubectl get storageclass <storage-class> -o json
```

- Per visualizzare le classi di storage sincronizzate di Trident, eseguire il seguente comando:

```
tridentctl get storageclass
```

- Per visualizzare i dettagli della classe di archiviazione sincronizzata di Trident, eseguire il seguente comando:

```
tridentctl get storageclass <storage-class> -o json
```

Imposta una storage class predefinita

Kubernetes 1.6 ha aggiunto la possibilità di impostare una classe di storage predefinita. Questa è la classe di storage che verrà utilizzata per il provisioning di un Persistent Volume se un utente non ne specifica una in una Persistent Volume Claim (PVC).

- Definire una classe di archiviazione predefinita impostando l'annotazione `storageclass.kubernetes.io/is-default-class` a true nella definizione della classe di archiviazione. Secondo la specifica, qualsiasi altro valore o l'assenza dell'annotazione è interpretato come false.
- È possibile configurare una classe di storage esistente come classe di storage predefinita utilizzando il seguente comando:

```
kubectl patch storageclass <storage-class-name> -p '{"metadata":
```

```
{"annotations":{"storageclass.kubernetes.io/is-default-class":"true"}}}'
```

- Allo stesso modo, puoi rimuovere l'annotazione della storage class predefinita utilizzando il seguente comando:

```
kubectl patch storageclass <storage-class-name> -p '{"metadata":
{"annotations":{"storageclass.kubernetes.io/is-default-class":"false"}}}'
```

Ci sono anche esempi nel bundle di installazione di Trident che includono questa annotazione.

NOTA Dovrebbe esserci solo una classe di storage predefinita nel tuo cluster alla volta. Kubernetes non ti impedisce tecnicamente di averne più di una, ma si comporterà come se non ci fosse affatto una classe di storage predefinita.

Identificare il backend per una storage class

Questo è un esempio del tipo di domande a cui è possibile rispondere con il JSON che `tridentctl` restituisce per gli oggetti backend di Trident. Questo utilizza l' `jq`utility, che potrebbe essere necessario installare prima.

```
tridentctl get storageclass -o json | jq  '[.items[] | {storageClass:
.Config.name, backends: [.storage]|unique}]'
```

Elimina una storage class

Per eliminare una storage class da Kubernetes, esegui il seguente comando:

```
kubectl delete storageclass <storage-class>
```

`<storage-class>` dovrebbe essere sostituito con la tua storage class.

Tutti i volumi persistenti che sono stati creati tramite questa classe di archiviazione rimarranno intatti e Trident continuerà a gestirli.

NOTA Trident applica un valore vuoto `fsType` per i volumi che crea. Per i backend iSCSI, si consiglia di applicare `parameters.fsType` nella StorageClass. È necessario eliminare le StorageClasses esistenti e ricrearle con `parameters.fsType` specificato.

Effettua il provisioning e gestisci i volumi

Esegui il provisioning di un volume Trident

Crea un PersistentVolumeClaim (PVC) che utilizza il StorageClass Kubernetes configurato per richiedere l'accesso al PV. Puoi quindi montare il PV su un pod.

Panoramica

Un "*PersistentVolumeClaim*" (PVC) è una richiesta di accesso al PersistentVolume sul cluster.

Il PVC può essere configurato per richiedere storage di una certa dimensione o modalità di accesso. Utilizzando il StorageClass associato, l'amministratore del cluster può controllare più della sola dimensione e modalità di accesso della PersistentVolume, come ad esempio le prestazioni o il livello di servizio.

Dopo aver creato il PVC puoi montare il volume in un pod.

Crea il PVC

Passaggi

1. Crea il PVC.

```
kubectl create -f pvc.yaml
```

2. Verificare lo stato del PVC.

```
kubectl get pvc
```

```
NAME         STATUS  VOLUME     CAPACITY   ACCESS MODES  STORAGECLASS AGE
pvc-storage Bound   pv-name    1Gi        RWO                   5m
```

1. Monta il volume in un pod.

```
kubectl create -f pv-pod.yaml
```

NOTA Puoi monitorare l'avanzamento usando `kubectl get pod --watch`.

2. Verificare che il volume sia montato su `/my/mount/path`.

```
kubectl exec -it task-pv-pod -- df -h /my/mount/path
```

3. Ora puoi eliminare il Pod. L'applicazione Pod non esisterà più, ma il volume rimarrà.

```
kubectl delete pod pv-pod
```

Esempi di manifest

▼ **PersistentVolumeClaim manifesti di esempio**

Questi esempi mostrano le opzioni di configurazione di base del PVC.

PVC con accesso RWO

Questo esempio mostra un PVC di base con accesso RWO associato a un StorageClass denominato `basic-csi`.

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: pvc-storage
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: basic-csi
```

PVC con NVMe/TCP

Questo esempio mostra un PVC di base per NVMe/TCP con accesso RWO associato a un StorageClass denominato `protection-gold`.

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
name: pvc-san-nvme
spec:
accessModes:
  - ReadWriteOnce
resources:
  requests:
    storage: 300Mi
storageClassName: protection-gold
```

▼ Esempi di manifest di pod

Questi esempi mostrano configurazioni di base per collegare il PVC a un pod.

Configurazione di base

```
kind: Pod
apiVersion: v1
metadata:
  name: pv-pod
spec:
  volumes:
    - name: storage
      persistentVolumeClaim:
```

```
        claimName: pvc-storage
  containers:
    - name: pv-container
      image: nginx
      ports:
        - containerPort: 80
          name: "http-server"
      volumeMounts:
        - mountPath: "/my/mount/path"
          name: storage
```

Configurazione di base NVMe/TCP

```
apiVersion: v1
kind: Pod
metadata:
  name: pod-nginx
spec:
  volumes:
    - name: basic-pvc
      persistentVolumeClaim:
        claimName: pvc-san-nvme
  containers:
    - name: task-pv-container
      image: nginx
      volumeMounts:
        - mountPath: "/my/mount/path"
          name: basic-pvc
```

Fate riferimento a "Oggetti Kubernetes e Trident" per i dettagli su come le classi di storage interagiscono con `PersistentVolumeClaim` e sui parametri per controllare come Trident effettua il provisioning dei volumi.

Sposta un volume

A partire da Trident 26.06, un amministratore dello storage può spostare in modo non distruttivo un volume iSCSI su un nodo ONTAP e un aggregato diversi all'interno dello stesso cluster ONTAP mentre il carico di lavoro che lo utilizza continua a funzionare. Richiedi lo spostamento in modo dichiarativo con una `TridentVolumeMove` custom resource o l' `tridentctl` equivalente, e Trident coordina ONTAP e l'host così che l'I/O non venga mai interrotto.

Requisiti

La funzione di spostamento del volume si applica al driver `ontap-san` per i volumi iSCSI solo. I protocolli NVMe e NAS non sono supportati.

Prima di spostare un volume, assicurati che siano soddisfatti i seguenti requisiti:

- Il backend utilizza il `ontap-san` driver con l'API REST ONTAP.
- L'ONTAP cluster ha almeno due nodi.
- Hai uno dei seguenti set di credenziali:
  - ONTAP 9.19.1 o versioni successive se usi il ruolo ONTAP personalizzato dedicato.
  - Credenziali di amministratore del cluster nelle versioni precedenti di ONTAP.

Come funziona

Trident sposta il volume (LUN) tra due posizioni all'interno dello stesso ONTAP cluster. Prima che i dati vengano spostati, Trident aggiunge nuovi percorsi iSCSI alla destinazione, affiancandoli a quelli esistenti, così il dispositivo multipath dell'host non perde mai l'accesso. Dopo che ONTAP ha completato lo spostamento, Trident elimina i percorsi di origine obsoleti.

Trident gestisce lo spostamento attraverso una serie ordinata di stati visibili nella `TridentVolumeMove` risorsa, così puoi seguire l'avanzamento dello spostamento.

Sposta un volume

Identifica il volume e le posizioni

1. Individua il nome del PV da spostare. Il nome del PV è anche il nome della risorsa di spostamento.
2. Dal cluster ONTAP, annota il nodo e l'aggregato correnti (di origine) per quel volume e scegli il nodo e l'aggregato di destinazione.

Inizia lo spostamento

Puoi usare entrambi i metodi. Entrambi creano la stessa `TridentVolumeMove`risorsa, il cui nome è il nome del PV.

Utilizzo di tridentctl

```
tridentctl create volume-move \
  --volume      pvc-8a814d62-bd58-4253-b0d1-82f2885db671 \
  --source-node ontap-node-01 \
  --source-pool aggr1_node01 \
  --target-node ontap-node-02 \
  --target-pool aggr1_node02 \
  -n trident
```

`tvm` è un alias per `volume-move`. Aggiungi `--delete-after-success 10m` per rimuovere automaticamente la risorsa dopo che lo spostamento ha avuto successo.

Utilizzando una risorsa TridentVolumeMove

```
apiVersion: trident.netapp.io/v1
kind: TridentVolumeMove
```

```
metadata:
  name: pvc-8a814d62-bd58-4253-b0d1-82f2885db671   # the PV to move
  namespace: trident
spec:
  sourceNode: ontap-node-01
  sourcePool: aggr1_node01
  targetNode: ontap-node-02
  targetPool: aggr1_node02
  # deleteAfterSuccess: 10m   # optional
```

Applica la risorsa:

```
kubectl apply -f tridentvolumemove.yaml
```

Campi di specifica

`sourceNode`/ `targetNode`

Nomi dei nodi ONTAP (non UUID).

`sourcePool`/ `targetPool`

Nomi degli aggregate ONTAP.

`deleteAfterSuccess`

Durata facoltativa, ad esempio `30s` o `10m`. Imposta su `0` per eliminare subito la risorsa, oppure ometti per conservarla.

Monitora lo spostamento

Monitora i progressi utilizzando il `tvm`nome breve o `tridentvolumemove`:

```
kubectl get tridentvolumemove -n trident
```

Per visualizzare lo stato attuale e i progressi per ciascun nodo, descrivi la risorsa:

```
kubectl describe tridentvolumemove <pv-name> -n trident
```

Il volume può avere collegamenti simultanei durante lo `Moving` stato, perché ONTAP può impiegare del tempo per completare lo spostamento di un volume di grandi dimensioni. Per garantire uno spostamento senza interruzioni, Trident coordina tra il Trident controller e il Trident node. Ogni stato esegue i seguenti passaggi in ordine, terminando con `Succeeded` o `Failed`.

Stato	Che cosa sta succedendo
`Pending`	Trident accetta e convalida la richiesta di spostamento e raccoglie le informazioni di cui ha bisogno prima che vengano apportate modifiche allo storage.
`ControllerStaging`	Trident prepara la posizione di destinazione. Mappa le interfacce di archiviazione del nodo di destinazione sulla LUN e aggiorna Kubernetes in modo che un futuro riavvio del nodo si riconnetta alla posizione corretta.
`NodeStaging`	Su ciascun host in cui è montato il volume, Trident aggiunge i nuovi percorsi di archiviazione dal nodo di destinazione accanto ai percorsi esistenti, in modo che l'I/O del pod continui su entrambi i set di percorsi.
`Moving`	ONTAP copia e trasferisce i dati del volume all'aggregato di destinazione. Questa è la fase più lunga e ONTAP la gestisce interamente, mentre Trident esegue il polling fino al completamento. Durante questa fase sono consentite nuove pubblicazioni.
`ControllerUnstaging`	Il trasferimento dei dati è completato. Trident aggiorna Kubernetes per rimuovere i percorsi di origine dallo stato desiderato e poi scollega le interfacce di storage del nodo di origine dalla LUN.
`NodeUnstaging`	Su ciascun host, Trident rimuove i percorsi sorgente obsoleti. All'host rimangono solo i percorsi di destinazione e il dispositivo multipath continua a gestire le operazioni di I/O senza interruzioni.
`Succeeded`	Il volume è completamente spostato e online sulla destinazione. Trident conserva la `TridentVolumeMove` risorsa oppure la elimina automaticamente se imposti `deleteAfterSuccess`.
`Failed`	Si è verificato un errore irreversibile che ha interrotto lo spostamento o ne ha impedito l'avvio. Trident ripristina lo stato del volume per consentire la ripresa delle normali operazioni e mantiene la risorsa con un messaggio di errore. Esamina il messaggio, risolvi la causa principale, elimina la risorsa e riprova.

L' `.status.attachments`elenco riporta lo stato di avanzamento di ciascun nodo collegato nel suo `state`campo:

Stato di collegamento	Significato
`Pending`	Nodo identificato come collegato; il lavoro non è ancora iniziato.
`Bridged`	Nuovi percorsi di destinazione aggiunti accanto a quelli esistenti.
`Migrated`	Il traffico ora scorre sui nuovi percorsi.
`Cleaned`	Percorsi sorgente obsoleti rimossi dal nodo.
`Detached`	Il nodo si è staccato durante lo spostamento ed è stato escluso da ulteriori valutazioni mentre Trident verifica se l'aggancio è pronto per procedere al passo successivo.
`Failed`	Errore del terminale su questo nodo. Guarda il messaggio.

Pulisci la risorsa di spostamento

Se imposti `deleteAfterSuccess`, Trident rimuove automaticamente la risorsa. Altrimenti, puoi eliminarla

manualmente. Prima di richiedere un altro spostamento sullo stesso volume, assicurati che non esista alcuna risorsa `TridentVolumeMove` per quel volume né in uno stato `Succeeded` né in uno stato `Failed`.

NOTA Trident non elimina mai automaticamente le operazioni non riuscite. Trident conserva una `Failed` risorsa come record di controllo. Dopo che hai risolto la causa principale, elimina manualmente la risorsa prima di riprovare.

Espandi i volumi Trident

Trident offre agli utenti Kubernetes la possibilità di espandere i propri volumi dopo che sono stati creati. Trova informazioni sulle configurazioni necessarie per espandere i volumi iSCSI, NFS, SMB, NVMe/TCP e FC.

Espandere un volume iSCSI

È possibile espandere un volume persistente iSCSI (PV) utilizzando il provisioner CSI.

NOTA L'espansione del volume iSCSI è supportata dai `ontap-san`, `ontap-san-economy`, `solidfire-san` driver e richiede Kubernetes 1.16 e versioni successive.

Passaggio 1: configurare la StorageClass per supportare l'espansione del volume

Modifica la definizione di StorageClass per impostare il campo `allowVolumeExpansion` su `true`.

```
cat storageclass-ontapsan.yaml
```

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontap-san
provisioner: csi.trident.netapp.io
parameters:
  backendType: "ontap-san"
allowVolumeExpansion: True
```

Per un StorageClass già esistente, modificarlo per includere il `allowVolumeExpansion` parametro.

Passaggio 2: crea un PVC con la StorageClass che hai creato

Modifica la definizione del PVC e aggiorna il `spec.resources.requests.storage` per riflettere la nuova dimensione desiderata, che deve essere maggiore della dimensione originale.

```
cat pvc-ontapsan.yaml
```

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: san-pvc
spec:
  accessModes:
  - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: ontap-san
```

Trident crea un Persistent Volume (PV) e lo associa a questo Persistent Volume Claim (PVC).

```
kubectl get pvc
NAME      STATUS   VOLUME                                     CAPACITY
ACCESS MODES   STORAGECLASS   AGE
san-pvc   Bound    pvc-8a814d62-bd58-4253-b0d1-82f2885db671   1Gi
RWO            ontap-san      8s

kubectl get pv
NAME                                       CAPACITY   ACCESS MODES
RECLAIM POLICY   STATUS   CLAIM              STORAGECLASS   REASON   AGE
pvc-8a814d62-bd58-4253-b0d1-82f2885db671   1Gi        RWO
Delete           Bound    default/san-pvc    ontap-san               10s
```

Fase 3: definire un pod che collega il PVC

Collega il PV a un pod per ridimensionarlo. Esistono due scenari quando si ridimensiona un PV iSCSI:

- Se il PV è collegato a un pod, Trident espande il volume sul backend di archiviazione, esegue una nuova scansione del dispositivo e ridimensiona il filesystem.
- Quando si tenta di ridimensionare un PV non collegato, Trident espande il volume sul backend di storage. Dopo che il PVC è stato associato a un pod, Trident esegue una nuova scansione del dispositivo e ridimensiona il filesystem. Kubernetes aggiorna la dimensione del PVC dopo che l'operazione di espansione è stata completata con successo.

In questo esempio, viene creato un pod che utilizza il `san-pvc`.

```
 kubectl get pod
NAME         READY   STATUS    RESTARTS   AGE
ubuntu-pod   1/1     Running   0          65s

 kubectl describe pvc san-pvc
Name:          san-pvc
```

```
Namespace:     default
StorageClass:  ontap-san
Status:        Bound
Volume:        pvc-8a814d62-bd58-4253-b0d1-82f2885db671
Labels:        <none>
Annotations:   pv.kubernetes.io/bind-completed: yes
               pv.kubernetes.io/bound-by-controller: yes
               volume.beta.kubernetes.io/storage-provisioner:
csi.trident.netapp.io
Finalizers:    [kubernetes.io/pvc-protection]
Capacity:      1Gi
Access Modes:  RWO
VolumeMode:    Filesystem
Mounted By:    ubuntu-pod
```

Fase 4: Espandere il PV

Per ridimensionare il PV creato da 1Gi a 2Gi, modificare la definizione del PVC e aggiornare la `spec.resources.requests.storage` a 2Gi.

```
kubectl edit pvc san-pvc
```

```
# Please edit the object below. Lines beginning with a '#' will be
ignored,
# and an empty file will abort the edit. If an error occurs while saving
this file will be
# reopened with the relevant failures.
#
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  annotations:
    pv.kubernetes.io/bind-completed: "yes"
    pv.kubernetes.io/bound-by-controller: "yes"
    volume.beta.kubernetes.io/storage-provisioner: csi.trident.netapp.io
  creationTimestamp: "2019-10-10T17:32:29Z"
  finalizers:
  - kubernetes.io/pvc-protection
  name: san-pvc
  namespace: default
  resourceVersion: "16609"
  selfLink: /api/v1/namespaces/default/persistentvolumeclaims/san-pvc
  uid: 8a814d62-bd58-4253-b0d1-82f2885db671
spec:
```

```
  accessModes:
  - ReadWriteOnce
  resources:
    requests:
      storage: 2Gi
# ...
```

Fase 5: convalida l'espansione

È possibile verificare che l'espansione abbia funzionato correttamente controllando le dimensioni del PVC, del PV e del Trident volume:

```
kubectl get pvc san-pvc
NAME      STATUS   VOLUME                                     CAPACITY
ACCESS MODES   STORAGECLASS   AGE
san-pvc   Bound    pvc-8a814d62-bd58-4253-b0d1-82f2885db671   2Gi
RWO            ontap-san      11m
kubectl get pv
NAME                                       CAPACITY   ACCESS MODES
RECLAIM POLICY   STATUS   CLAIM              STORAGECLASS   REASON   AGE
pvc-8a814d62-bd58-4253-b0d1-82f2885db671   2Gi        RWO
Delete           Bound    default/san-pvc    ontap-san               12m
tridentctl get volumes -n trident
+-----------------------------------------+---------+--------------
+----------+-----------------------------------------+--------+---------+
|                   NAME                  |  SIZE   | STORAGE CLASS |
PROTOCOL |             BACKEND UUID              | STATE  | MANAGED |
+-----------------------------------------+---------+--------------
+----------+-----------------------------------------+--------+---------+
| pvc-8a814d62-bd58-4253-b0d1-82f2885db671 | 2.0 GiB | ontap-san     |
block    | a9b7bfff-0505-4e31-b6c5-59f492e02d33 | online | true    |
+-----------------------------------------+---------+--------------
+----------+-----------------------------------------+--------+---------+
```

Espandere un volume FC

È possibile espandere un FC Persistent Volume (PV) utilizzando il CSI provisioner.

NOTA L'espansione del volume FC è supportata dal driver `ontap-san` e richiede Kubernetes 1.16 e versioni successive.

Passaggio 1: configurare la StorageClass per supportare l'espansione del volume

Modifica la definizione di StorageClass per impostare il campo `allowVolumeExpansion` su `true`.

```
cat storageclass-ontapsan.yaml
```

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontap-san
provisioner: csi.trident.netapp.io
parameters:
  backendType: "ontap-san"
allowVolumeExpansion: True
```

Per un StorageClass già esistente, modificarlo per includere il `allowVolumeExpansion` parametro.

Passaggio 2: crea un PVC con la StorageClass che hai creato

Modifica la definizione del PVC e aggiorna il `spec.resources.requests.storage` per riflettere la nuova dimensione desiderata, che deve essere maggiore della dimensione originale.

```
cat pvc-ontapsan.yaml
```

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: san-pvc
spec:
  accessModes:
  - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: ontap-san
```

Trident crea un Persistent Volume (PV) e lo associa a questo Persistent Volume Claim (PVC).

```
kubectl get pvc
NAME      STATUS   VOLUME                                     CAPACITY
ACCESS MODES   STORAGECLASS   AGE
san-pvc   Bound    pvc-8a814d62-bd58-4253-b0d1-82f2885db671   1Gi
RWO            ontap-san      8s

kubectl get pv
NAME                                       CAPACITY   ACCESS MODES
```

```
RECLAIM POLICY   STATUS   CLAIM             STORAGECLASS   REASON   AGE
pvc-8a814d62-bd58-4253-b0d1-82f2885db671   1Gi        RWO
Delete           Bound    default/san-pvc   ontap-san               10s
```

Fase 3: definire un pod che collega il PVC

Collega il PV a un pod per ridimensionarlo. Esistono due scenari per il ridimensionamento di un PV FC:

- Se il PV è collegato a un pod, Trident espande il volume sul backend di archiviazione, esegue una nuova scansione del dispositivo e ridimensiona il filesystem.
- Quando si tenta di ridimensionare un PV non collegato, Trident espande il volume sul backend di storage. Dopo che il PVC è stato associato a un pod, Trident esegue una nuova scansione del dispositivo e ridimensiona il filesystem. Kubernetes aggiorna la dimensione del PVC dopo che l'operazione di espansione è stata completata con successo.

In questo esempio, viene creato un pod che utilizza il `san-pvc`.

```
 kubectl get pod
NAME         READY   STATUS    RESTARTS   AGE
ubuntu-pod   1/1     Running   0          65s

 kubectl describe pvc san-pvc
Name:          san-pvc
Namespace:     default
StorageClass:  ontap-san
Status:        Bound
Volume:        pvc-8a814d62-bd58-4253-b0d1-82f2885db671
Labels:        <none>
Annotations:   pv.kubernetes.io/bind-completed: yes
               pv.kubernetes.io/bound-by-controller: yes
               volume.beta.kubernetes.io/storage-provisioner:
csi.trident.netapp.io
Finalizers:    [kubernetes.io/pvc-protection]
Capacity:      1Gi
Access Modes:  RWO
VolumeMode:    Filesystem
Mounted By:    ubuntu-pod
```

Fase 4: Espandere il PV

Per ridimensionare il PV creato da 1Gi a 2Gi, modificare la definizione del PVC e aggiornare la `spec.resources.requests.storage` a 2Gi.

```
kubectl edit pvc san-pvc
```

```
# Please edit the object below. Lines beginning with a '#' will be
ignored,
# and an empty file will abort the edit. If an error occurs while saving
this file will be
# reopened with the relevant failures.
#
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  annotations:
    pv.kubernetes.io/bind-completed: "yes"
    pv.kubernetes.io/bound-by-controller: "yes"
    volume.beta.kubernetes.io/storage-provisioner: csi.trident.netapp.io
  creationTimestamp: "2019-10-10T17:32:29Z"
  finalizers:
  - kubernetes.io/pvc-protection
  name: san-pvc
  namespace: default
  resourceVersion: "16609"
  selfLink: /api/v1/namespaces/default/persistentvolumeclaims/san-pvc
  uid: 8a814d62-bd58-4253-b0d1-82f2885db671
spec:
  accessModes:
  - ReadWriteOnce
  resources:
    requests:
      storage: 2Gi
 # ...
```

Fase 5: convalida l'espansione

È possibile verificare che l'espansione abbia funzionato correttamente controllando le dimensioni del PVC, del PV e del Trident volume:

```
kubectl get pvc san-pvc
NAME      STATUS   VOLUME                                     CAPACITY
ACCESS MODES   STORAGECLASS   AGE
san-pvc   Bound    pvc-8a814d62-bd58-4253-b0d1-82f2885db671   2Gi
RWO            ontap-san      11m
kubectl get pv
NAME                                       CAPACITY   ACCESS MODES
RECLAIM POLICY   STATUS   CLAIM             STORAGECLASS   REASON   AGE
pvc-8a814d62-bd58-4253-b0d1-82f2885db671   2Gi        RWO
Delete           Bound    default/san-pvc   ontap-san               12m
tridentctl get volumes -n trident
```

```
+-----------------------------------------+---------+---------------
+----------+-----------------------------------------+--------+---------+
|                   NAME                  |  SIZE   | STORAGE CLASS |
PROTOCOL |             BACKEND UUID             | STATE  | MANAGED |
+-----------------------------------------+---------+---------------
+----------+-----------------------------------------+--------+---------+
| pvc-8a814d62-bd58-4253-b0d1-82f2885db671 | 2.0 GiB | ontap-san     |
block    | a9b7bfff-0505-4e31-b6c5-59f492e02d33 | online | true    |
+-----------------------------------------+---------+---------------
+----------+-----------------------------------------+--------+---------+
```

Espandere un volume NFS

Trident supporta l'espansione del volume per i PV NFS forniti su `ontap-nas`, `ontap-nas-economy`, `ontap-nas-flexgroup` e `azure-netapp-files` backend.

Passaggio 1: configurare la StorageClass per supportare l'espansione del volume

Per ridimensionare un NFS PV, l'amministratore deve prima configurare la storage class per consentire l'espansione del volume impostando il `allowVolumeExpansion` field su `true`:

```
cat storageclass-ontapnas.yaml
```

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontapnas
provisioner: csi.trident.netapp.io
parameters:
  backendType: ontap-nas
allowVolumeExpansion: true
```

Se hai già creato una classe di archiviazione senza questa opzione, puoi semplicemente modificare la classe di archiviazione esistente utilizzando `kubectl edit storageclass` per consentire l'espansione del volume.

Passaggio 2: crea un PVC con la StorageClass che hai creato

```
cat pvc-ontapnas.yaml
```

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
```

```
  name: ontapnas20mb
spec:
  accessModes:
  - ReadWriteOnce
  resources:
    requests:
      storage: 20Mi
  storageClassName: ontapnas
```

Trident dovrebbe creare un PV NFS da 20 MiB per questo PVC:

```
kubectl get pvc
NAME           STATUS   VOLUME
CAPACITY     ACCESS MODES   STORAGECLASS    AGE
ontapnas20mb   Bound    pvc-08f3d561-b199-11e9-8d9f-5254004dfdb7   20Mi
RWO            ontapnas        9s

kubectl get pv pvc-08f3d561-b199-11e9-8d9f-5254004dfdb7
NAME                                       CAPACITY   ACCESS MODES
RECLAIM POLICY   STATUS   CLAIM                   STORAGECLASS    REASON
AGE
pvc-08f3d561-b199-11e9-8d9f-5254004dfdb7   20Mi       RWO
Delete           Bound    default/ontapnas20mb   ontapnas
2m42s
```

Fase 3: Espandere il PV

Per ridimensionare il PV da 20 MiB appena creato a 1 GiB, modifica il PVC e imposta `spec.resources.requests.storage` su 1 GiB:

```
kubectl edit pvc ontapnas20mb
```

```
# Please edit the object below. Lines beginning with a '#' will be
ignored,
# and an empty file will abort the edit. If an error occurs while saving
this file will be
# reopened with the relevant failures.
#
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  annotations:
    pv.kubernetes.io/bind-completed: "yes"
    pv.kubernetes.io/bound-by-controller: "yes"
```

```
    volume.beta.kubernetes.io/storage-provisioner: csi.trident.netapp.io
  creationTimestamp: 2018-08-21T18:26:44Z
  finalizers:
  - kubernetes.io/pvc-protection
  name: ontapnas20mb
  namespace: default
  resourceVersion: "1958015"
  selfLink: /api/v1/namespaces/default/persistentvolumeclaims/ontapnas20mb
  uid: c1bd7fa5-a56f-11e8-b8d7-fa163e59eaab
spec:
  accessModes:
  - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
# ...
```

Fase 4: convalida l'espansione

È possibile verificare che il ridimensionamento abbia funzionato correttamente controllando le dimensioni del PVC, del PV e del volume Trident:

```
kubectl get pvc ontapnas20mb
NAME           STATUS   VOLUME
CAPACITY   ACCESS MODES   STORAGECLASS    AGE
ontapnas20mb   Bound    pvc-08f3d561-b199-11e9-8d9f-5254004dfdb7   1Gi
RWO            ontapnas        4m44s

kubectl get pv pvc-08f3d561-b199-11e9-8d9f-5254004dfdb7
NAME                                       CAPACITY   ACCESS MODES
RECLAIM POLICY   STATUS   CLAIM                  STORAGECLASS    REASON
AGE
pvc-08f3d561-b199-11e9-8d9f-5254004dfdb7   1Gi        RWO
Delete           Bound    default/ontapnas20mb   ontapnas
5m35s

tridentctl get volume pvc-08f3d561-b199-11e9-8d9f-5254004dfdb7 -n trident
+-----------------------------------------+---------+---------------
+----------+-----------------------------------------+--------+---------+
|                  NAME                   |  SIZE   | STORAGE CLASS |
PROTOCOL |             BACKEND UUID              | STATE  | MANAGED |
+-----------------------------------------+---------+---------------
+----------+-----------------------------------------+--------+---------+
| pvc-08f3d561-b199-11e9-8d9f-5254004dfdb7 | 1.0 GiB | ontapnas      |
file     | c5a6f6a4-b052-423b-80d4-8fb491a14a22 | online | true    |
+-----------------------------------------+---------+---------------
```

```
+----------+--------------------------------------+--------+---------+
```

Comprendere i limiti del sottosistema RWX NVMe

ReadWriteMany (RWX) volumi che utilizzano il protocollo NVMe hanno un limite di scalabilità di 64 nodi per volume. Di seguito sono illustrate le limitazioni, viene spiegata l'architettura del sottosistema NVMe coinvolto e vengono delineati i passaggi di risoluzione richiesti.

Comprendere il limite di 64 nodi

Se prevedi di usare i volumi ReadWriteMany (RWX) con il protocollo NVMe, un singolo volume RWX NVMe non può essere montato da più di 64 nodi in un cluster Kubernetes.

Non pianificare carichi di lavoro che montano lo stesso RWX NVMe PersistentVolumeClaim su più di 64 nodi.

Questa limitazione si applica solo ai volumi RWX che utilizzano il protocollo NVMe.

Comprendere i modelli di sottosistema NVMe

Modello di sottosistema per volume (versioni di Trident precedenti alla 26.02)

Nelle versioni di Trident precedenti alla 26.02, i volumi RWX NVMe vengono forniti utilizzando un modello di sottosistema per volume. Ogni volume RWX NVMe è mappato al proprio sottosistema NVMe dedicato su ONTAP.

Questo modello è semplice, ma presenta un limite di scalabilità inferiore. Nei cluster Kubernetes di grandi dimensioni, i limiti del controller del sottosistema vengono raggiunti rapidamente perché ogni volume RWX consuma un sottosistema dedicato.

Modello super-sottosistema (introdotto in Trident 26.02)

A partire da Trident 26.02, i volumi RWX NVMe utilizzano un modello di super-sottosistema condiviso. Più volumi RWX NVMe condividono lo stesso sottosistema NVMe.

Ogni super-sottosistema supporta fino a 1024 namespace (volumi). Questo modello migliora significativamente la scalabilità per i carichi di lavoro RWX e riduce la probabilità di raggiungere i limiti del sottosistema ONTAP.

Ogni volume RWX NVMe supporta fino a 64 nodi.

Identificare i sintomi di errore

Se crei o colleghi volumi RWX NVMe su larga scala, potresti osservare errori simili ai seguenti:

```
Maximum number of controllers reached. No more controllers can be created.
```

Questo errore indica che è stato raggiunto il limite del controller del sottosistema ONTAP NVMe.

Risolvere gli errori di limite del sottosistema

Per superare le limitazioni del sottosistema per volume e sfruttare il modello super-sottosistema, esegui l'aggiornamento a Trident 26.02 o versioni successive.

Aggiorna Trident per applicare il modello super-subsystem

Per applicare il modello super-sottosistema ai volumi RWX NVMe:

1. Aggiorna Trident alla versione 26.02 o successiva.
2. Riduci a zero le repliche di tutti i pod che utilizzano volumi RWX NVMe.
3. Verificare che nessun carico di lavoro stia utilizzando attivamente volumi RWX NVMe.
4. Aumenta nuovamente il numero di pod.

Questa sequenza di riavvio garantisce che i volumi RWX NVMe siano collegati utilizzando il modello super-subsystem.

- Questa limitazione si applica solo ai volumi RWX che utilizzano il protocollo NVMe.
- Il limite di 64 nodi si applica per ogni volume RWX NVMe.
- Altre modalità di accesso e altri protocolli non sono interessati.

Scalabilità del controller

Trident migliora la scalabilità del controller grazie a una maggiore concorrenza tra più driver di storage. Quando abiliti la scalabilità del controller, il controller Trident elabora le operazioni di storage in parallelo invece di serializzarle, il che aumenta il throughput negli ambienti Kubernetes con molte operazioni simultanee.

Prima di distribuire, determina quali driver Trident supportano la scalabilità del controller nella disponibilità generale e quali driver sono disponibili come anteprima tecnica in Trident 26.06. Questo ti aiuta a prendere decisioni di distribuzione informate e a gestire il rischio. La scalabilità del controller è disabilitata per impostazione predefinita.

Concetti chiave e definizioni

Scalabilità del controller

La scalabilità del controller si riferisce alla capacità del controller Trident di elaborare più operazioni di storage in parallelo, anziché serializzarle dietro un singolo blocco. Queste operazioni includono la creazione, l'eliminazione e il ridimensionamento dei volumi; la creazione e l'eliminazione di snapshot; la pubblicazione e l'annullamento della pubblicazione dei volumi; e la gestione del backend.

Quando abiliti la scalabilità del controller, le operazioni su volumi e backend diversi procedono contemporaneamente. Questo aumenta il throughput e riduce il tempo di esecuzione end-to-end in ambienti con un elevato numero di operazioni PersistentVolumeClaim e VolumeSnapshot simultanee.

Comportamento predefinito (modalità seriale)

Per impostazione predefinita, il controller Trident elabora le operazioni una alla volta. Ogni richiesta di creazione, eliminazione, ridimensionamento o snapshot viene completata prima che Trident avvii la successiva. La modalità seriale è quella predefinita supportata per tutte le installazioni e non richiede

configurazione.

La modalità seriale è sufficiente per la maggior parte dei carichi di lavoro. Abilita la scalabilità del controller solo quando il volume delle operazioni crea un accumulo durante l'elaborazione in modalità seriale.

Supporto per la scalabilità del controller

Trident supporta la scalabilità del controller a diversi livelli di maturità per diversi driver di storage.

Disponibilità generale

I seguenti driver supportano la scalabilità del controller alla disponibilità generale in Trident 26.06:

- `ontap-san`
- `ontap-nas`
- `ontap-nas-economy`
- `ontap-san-economy`
- `google-cloud-netapp-volumes`
- `azure-netapp-files`
- `solidfire-san`

NOTA I `google-cloud-netapp-volumes` e `google-cloud-netapp-volumes-san` driver sono diversi. Trident supporta solo `google-cloud-netapp-volumes`. Non usare `google-cloud-netapp-volumes-san` nelle configurazioni o negli esempi di backend.

Anteprima tecnica

Il seguente driver supporta la scalabilità del controller come funzionalità in anteprima tecnica in Trident 26.06:

- `asa-r2` (SAN e NVMe)

Questo driver presenta le seguenti limitazioni:

- La funzionalità di concorrenza del controller è disponibile solo per valutazione e test.
- Il comportamento può cambiare da una versione all'altra.
- NetApp non consiglia l'utilizzo in ambienti di produzione.

Come funziona enableConcurrency

Quando imposti `enableConcurrency` su `true`, Trident applica l'elaborazione simultanea a tutti i backend gestiti dal controller. L'impostazione si applica a tutti i backend contemporaneamente. Non puoi abilitarla per singoli backend o singoli driver.

Ogni backend configurato deve utilizzare un driver presente nell'elenco di disponibilità generale o di anteprima tecnica. Se un backend utilizza un driver non supportato, Trident non si avvia e Trident non aggiunge un backend che utilizza un driver non supportato.

Per ripristinare l'avvio, rimuovi o riconfigura qualsiasi backend che utilizza un driver non supportato oppure disabilita la scalabilità del controller.

Prima di abilitare

IMPORTANTE	Prima di abilitare la scalabilità del controller, assicurati che ogni backend configurato utilizzi un driver presente nell'elenco di disponibilità generale o di anteprima tecnica. Se un backend utilizza un driver non supportato, Trident non si avvia dopo che hai abilitato la funzionalità.

Utilizza la tabella seguente per decidere se abilitare la scalabilità del controller.

Se il tuo ambiente	Poi
Utilizzi solo driver supportati e riscontri un accumulo di operazioni del controller	Abilita la scalabilità del controller.
Utilizza un driver non supportato	Mantieni la modalità seriale predefinita. Non abilitare la scalabilità del controller.
Gestisce bassi volumi di operazioni senza arretrati	Mantieni la modalità seriale predefinita.

Abilita la scalabilità del controller

L' `enableConcurrency`opzione di configurazione controlla la scalabilità del controller. Devi abilitare esplicitamente questa opzione durante l'installazione di Trident o quando aggiorni una distribuzione esistente.

Distribuzione dell'operatore Trident

Per abilitare la scalabilità del controller con l'operatore Trident, imposta `enableConcurrency` su `true` nella `TridentOrchestrator` risorsa personalizzata (CR).

Nuova installazione

Crea o modifica il `TridentOrchestrator` CR e imposta `enableConcurrency` su `true`:

```
apiVersion: trident.netapp.io/v1
kind: TridentOrchestrator
metadata:
  name: trident
spec:
  namespace: trident
  enableConcurrency: true
```

Applica il CR:

```
kubectl apply -f tridentorchestrator_cr.yaml
```

Installazione esistente

Applicare una patch al `TridentOrchestrator` CR esistente per abilitare la scalabilità del controller:

```
kubectl patch torc trident --type=merge -p
'{"spec":{"enableConcurrency":true}}'
```

Verifica che Trident abbia applicato l'impostazione:

```
kubectl get torc trident -o
jsonpath='{.status.currentInstallationParams.enableConcurrency}'
```

Distribuzione Helm

Per abilitare la scalabilità del controller con Helm, impostare il `enableConcurrency` valore su `true`.

Nuova installazione

```
helm install trident netapp-trident/trident-operator --namespace trident
--create-namespace --set enableConcurrency=true
```

Installazione esistente

```
helm upgrade trident netapp-trident/trident-operator --namespace trident
--set enableConcurrency=true
```

In alternativa, imposta `enableConcurrency` su `true` in un file `values.yaml` personalizzato:

```
# values.yaml
enableConcurrency: true
```

Quindi installa o aggiorna utilizzando il file dei valori:

```
helm install trident netapp-trident/trident-operator --namespace trident
--create-namespace -f values.yaml
```

implementazione tridentctl

Per abilitare la scalabilità del controller con `tridentctl`, passare il `--enable-concurrency` flag durante l'installazione.

Nuova installazione

```
tridentctl install -n trident --enable-concurrency
```

Installazione esistente

Per abilitare la scalabilità del controller su una distribuzione `tridentctl`esistente, disinstalla Trident e reinstallalo con il flag:

```
tridentctl uninstall -n trident
tridentctl install -n trident --enable-concurrency
```

Verifica che la scalabilità del controller sia abilitata

Dopo aver abilitato la scalabilità del controller, verifica che il controller Trident sia in esecuzione con la concorrenza abilitata. Controlla i log del pod del controller:

```
kubectl logs -n trident deploy/trident-controller | grep -i concurrency
```

L'output include una voce di registro che conferma che la concorrenza è abilitata.

Comportamento della concorrenza

Quando la scalabilità del controller è abilitata, il controller Trident adotta il seguente comportamento:

- Trident sostituisce il blocco globale singolo con un blocco granulare per risorsa.
- Trident serializza le operazioni che modificano la stessa risorsa per mantenere la coerenza dei dati.
- Le operazioni che si limitano a leggere da una risorsa procedono contemporaneamente ad altre operazioni di lettura su quella risorsa.
- Trident limita le richieste API ONTAP simultanee a 20 per LIF di gestione per evitare il sovraccarico dei sistemi di storage di backend.
- Se più backend condividono lo stesso LIF di gestione, condividono questo limite di 20 richieste.

Avvertenze e limitazioni

Le seguenti considerazioni si applicano alla scalabilità del controller in Trident 26.06:

- La scalabilità del controller supporta solo i driver presenti negli elenchi di disponibilità generale e di anteprima tecnica. Per i dettagli, vedi Prima di abilitare.
- Il controller Trident gestisce la concorrenza internamente. Questa versione non prevede limiti di concorrenza configurabili dall'utente.
- Il throughput complessivo dipende dal driver di storage in uso, dalla reattività del backend e dalle prestazioni del server API di Kubernetes.
- Un'elevata concorrenza può aumentare il carico sui sistemi di storage back-end.
- Il comportamento di scalabilità del controller non è identico per tutti i driver.
- Il driver di anteprima tecnica può presentare prestazioni incoerenti sotto carico elevato e può cambiare comportamento tra una versione e l'altra.
- Il debug di operazioni concorrenti può risultare più complesso a causa dell'esecuzione parallela. Metriche e log possono mostrare l'output delle operazioni interlacciate.

Raccomandazioni

Prima di abilitare la scalabilità del controller, completa i seguenti passaggi:

1. Conferma che ogni backend configurato utilizza un driver presente nell'elenco di disponibilità generale o di anteprima tecnica.
2. Testa la modifica in un cluster non di produzione prima di applicarla in produzione.
3. Verifica che la scalabilità del controller sia abilitata dopo aver applicato la modifica.

Applica le seguenti raccomandazioni generali quando lavori con la scalabilità del controller:

- Usa i driver di disponibilità generale per gli ambienti di produzione che richiedono un'elevata scalabilità.
- Valuta il driver di anteprima tecnica solo in ambienti non di produzione.
- Monitora le prestazioni del backend e del controller quando operi su larga scala.
- Non dare per scontato l'ordine delle operazioni negli script di automazione.

Espansione automatica del volume

L'espansione automatica del volume consente ai Persistent Volumes forniti da Trident di aumentare automaticamente quando la capacità utilizzata raggiunge una soglia definita. Questa funzionalità riduce il sovraccarico operativo e aiuta a prevenire l'interruzione delle applicazioni causata dall'esaurimento della capacità. L'espansione automatica del volume viene implementata utilizzando Autogrow Policies. Una Autogrow Policy definisce:

- La soglia di utilizzo che attiva l'espansione
- La quantità di cui cresce il volume
- La dimensione massima che il volume può raggiungere

NOTA	I volumi aumentano automaticamente di dimensione quando viene superata la soglia di utilizzo definita. I volumi non vengono mai ridotti automaticamente.

Requisiti

Prima di configurare l'espansione automatica del volume, assicurarsi che siano soddisfatti i seguenti requisiti:

- Trident 26.02 o versioni successive
- Autorizzazioni di controllo degli accessi basate sui ruoli per creare `TridentAutogrowPolicy` risorse personalizzate
- StorageClasses configurati con `allowVolumeExpansion: true`
- Protocolli ONTAP supportati:
  - Network File System (NFS)
  - Internet Small Computer Systems Interface (iSCSI)
  - Non-Volatile Memory Express (NVMe)
  - Protocollo Fibre Channel (FCP)

Limitazioni

- I volumi raw block ONTAP Non-Volatile Memory Express precedenti a ONTAP 9.16.1 non supportano l'espansione automatica.
- Per i volumi di storage area network, se `growthAmount` è inferiore o uguale a 50 mebibyte, Trident aumenta automaticamente il valore a 51 mebibyte prima del ridimensionamento, a condizione che la dimensione risultante non superi `maxSize`.
- Negli ambienti brownfield, l'espansione automatica potrebbe non funzionare per alcuni volumi esistenti a causa del comportamento di migrazione della pubblicazione del volume.
- Quando un volume raggiunge `maxSize`, non si verifica alcuna ulteriore espansione.
- Protocolli supportati per l'espansione automatica del volume:
  - Network File System (NFS)
  - Internet Small Computer Systems Interface (iSCSI)
  - Non-Volatile Memory Express (NVMe)
  - Protocollo Fibre Channel (FCP)

Provision Volumes con criterio di Autogrow

La policy di Autogrow può essere configurata a due livelli:

- Livello di classe di archiviazione: imposta il valore predefinito per tutti i volumi (usando l'annotazione)
- Livello PVC: sostituisce la classe di archiviazione predefinita (utilizzando annotazione)

Crea una policy di Autogrow

Le Autogrow Policies consentono l'espansione automatica del volume quando i volumi raggiungono una soglia di capacità definita.

Assicurati di avere:

- Trident 26.02 o versioni successive installato
- Autorizzazioni di controllo degli accessi basate sui ruoli per creare `TridentAutogrowPolicy` risorse
- Comprensione dei requisiti di crescita del carico di lavoro

Una Autogrow Policy definisce come i volumi si espandono automaticamente quando raggiungono una soglia di capacità definita.

È possibile creare Autogrow Policies in qualsiasi momento del flusso di lavoro:

- Prima che StorageClasses e i volumi vengano creati
- Dopo che StorageClasses esistono
- Dopo che i volumi sono stati effettuati il provisioning

Questa flessibilità consente di introdurre l'espansione automatica senza ricreare le risorse esistenti.

Specifiche della Autogrow Policy

Le Autogrow Policies sono risorse personalizzate di Kubernetes definite come segue:

Campo	Descrizione	Formato	Richiesto	Esempio	Predefinito
nome	Identificatore univoco della policy	Stringa	Sì	production-db-policy	Nessuno
usedThreshold	Percentuale di capacità che innesca l'espansione	Stringa percentuale	Sì	"80%"	Nessuno
growthAmount	Quantità da aumentare quando viene raggiunta la soglia	Percentuale o dimensione	No	"10%" o "5Gi"	"10%"
maxSize	Limite massimo della dimensione del volume	Quantità Kubernetes	No	"500Gi"	Illimitato

Crea una policy di Autogrow

Passaggi

1. Crea un file YAML che definisce la tua Autogrow Policy:

```
apiVersion: trident.netapp.io/v1
kind: TridentAutogrowPolicy
metadata:
  name: standard-autogrow
spec:
  usedThreshold: "80%"
  growthAmount: "10%"
  maxSize: "500Gi"
```

2. Applica la policy al tuo cluster:

```
kubectl apply -f autogrow-policy.yaml
```

3. Verificare che la policy sia stata creata:

```
kubectl get tridentautogrowpolicy standard-autogrow
```

Output previsto

```
NAME                 USED THRESHOLD   GROWTH AMOUNT   STATE
standard-autogrow    80%              10%             Success
```

Stati delle policy

Dopo aver creato una policy, Trident convalida la specifica e assegna uno dei seguenti stati:

Stato	Descrizione	Azione richiesta
Successo	La policy è convalidata e pronta per l'uso.	Nessuno.
Fallito	Sono stati rilevati errori di convalida.	Rivedi e correggi la specifica.
Eliminazione	L'eliminazione è in corso.	Attendere il completamento.

Associa una policy a un StorageClass

È possibile associare una Autogrow Policy a una StorageClass utilizzando l' `trident.netapp.io/autogrowPolicy`annotazione. Tutti i volumi forniti da quella StorageClass ereditano la policy.

NOTA Il StorageClass deve avere `allowVolumeExpansion: true`.

Passaggi

1. Crea o modifica una StorageClass con l'annotazione Autogrow Policy:

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontap-gold
  annotations:
    trident.netapp.io/autogrowPolicy: "production-db-policy"
provisioner: csi.trident.netapp.io
parameters:
  backendType: "ontap-san"
  fsType: "ext4"
allowVolumeExpansion: true
```

2. Applica il StorageClass:

```
kubectl apply -f storageclass.yaml
```

3. Verifica l'annotazione:

```
kubectl get storageclass ontap-gold -o
jsonpath='{.metadata.annotations.trident\.netapp\.io/autogrowPolicy}'
```

Output previsto

```
production-db-policy
```

Precedenza delle policy

Quando le annotazioni Autogrow Policy sono impostate sia su una StorageClass che su un PVC, Trident applica le seguenti regole di precedenza:

1. **L'annotazione PVC ha la priorità.** Se un PVC imposta `trident.netapp.io/autogrowPolicy`, quel valore viene sempre utilizzato.
2. **StorageClass annotation si applica solo quando il PVC non ha annotazione.**
3. **Se nessuno dei due ha l'annotazione,** non viene applicata alcuna Autogrow Policy.

Annotazione StorageClass	Annotazione PVC	Comportamento efficace
`trident.netapp.io/autogrowPolicy: standard-agp`	Non impostato	Usi `standard-agp`.
`trident.netapp.io/autogrowPolicy: standard-agp`	`trident.netapp.io/autogrowPolicy: logs-policy`	Utilizza `logs-policy` (PVC sostituisce StorageClass).
`trident.netapp.io/autogrowPolicy: standard-agp`	`trident.netapp.io/autogrowPolicy: "none"`	No Autogrow Policy (PVC disabilita autogrow).
Non impostato	`trident.netapp.io/autogrowPolicy: dev-policy`	Usi `dev-policy`.
Non impostato	Non impostato	Nessuna Autogrow Policy.

Esempi di configurazione

Gli esempi seguenti mostrano configurazioni comuni di Autogrow Policy per diversi casi d'uso.

Politica conservativa per database di produzione

```
apiVersion: trident.netapp.io/v1
kind: TridentAutogrowPolicy
metadata:
  name: production-db-policy
spec:
  usedThreshold: "75%"
  growthAmount: "20%"
  maxSize: "5Ti"
```

Archiviazione dei log con incrementi di crescita fissi

```
apiVersion: trident.netapp.io/v1
kind: TridentAutogrowPolicy
metadata:
```

```
  name: log-storage-policy
spec:
  usedThreshold: "90%"
  growthAmount: "10Gi"
  maxSize: "100Gi"
```

Politica minima con valori predefiniti

Quando si omettono `growthAmount` e `maxSize`, Trident utilizza i valori predefiniti (`10%` crescita, dimensione illimitata):

```
apiVersion: trident.netapp.io/v1
kind: TridentAutogrowPolicy
metadata:
  name: logs-policy
spec:
  usedThreshold: "85%"
```

Policy con un maxSize personalizzato e growthAmount predefinito

```
apiVersion: trident.netapp.io/v1
kind: TridentAutogrowPolicy
metadata:
  name: default-ga-policy
spec:
  usedThreshold: "70%"
  maxSize: "100Gi"
```

Crescita aggressiva con maxSize illimitata

```
apiVersion: trident.netapp.io/v1
kind: TridentAutogrowPolicy
metadata:
  name: aggressive-growth-policy
spec:
  usedThreshold: "80%"
  growthAmount: "150%"
```

Policy con percentuali frazionarie

```
apiVersion: trident.netapp.io/v1
kind: TridentAutogrowPolicy
metadata:
```

```
  name: precise-policy
spec:
  usedThreshold: "80.28%"
  growthAmount: "10.65%"
  maxSize: "100Gi"
```

NOTA Sono supportate le percentuali frazionarie. Se si specificano più di tre cifre decimali, Trident arrotonda il valore a tre cifre decimali.

NAS StorageClass con crescita automatica

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontap-nas-autogrow
  annotations:
    trident.netapp.io/autogrowPolicy: "standard-autogrow"
provisioner: csi.trident.netapp.io
parameters:
  backendType: "ontap-nas"
  fsType: "ext4"
allowVolumeExpansion: true
```

SAN StorageClass con crescita automatica

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: database-storage
  annotations:
    trident.netapp.io/autogrowPolicy: "production-db-policy"
provisioner: csi.trident.netapp.io
parameters:
  backendType: "ontap-san"
  fsType: "ext4"
allowVolumeExpansion: true
```

Gestisci le policy di Autogrow

Dopo che crei le Autogrow Policies, puoi visualizzarle, aggiornarle ed eliminarle. Puoi anche monitorare quali volumi usano una determinata policy.

Visualizza le policy di crescita automatica

Elenca tutte le politiche

Usa `kubectl` per elencare tutte le Autogrow Policies nel tuo cluster:

```
kubectl get tridentautogrowpolicy
```

In alternativa, usa `tridentctl`:

```
tridentctl get autogrowpolicy
```

Visualizza dettagli della polizza

Per visualizzare le specifiche complete e lo stato di una policy:

```
kubectl describe tridentautogrowpolicy production-db-policy
```

Per visualizzare una policy con i volumi associati in formato YAML:

```
tridentctl get autogrowpolicy production-db-policy -o yaml
```

Aggiorna una policy di Autogrow

È possibile modificare una policy esistente per cambiarne la soglia, l'entità della crescita o la dimensione massima. Le modifiche hanno effetto immediato per tutti i volumi che utilizzano la policy.

IMPORTANTE	Le modifiche influiscono su tutti i volumi che utilizzano la policy. Quando possibile, testa prima le modifiche in un ambiente non di produzione.

Passaggi

1. Modifica la policy:

   ```
   kubectl edit tridentautogrowpolicy production-db-policy
   ```

2. Modifica i `spec` campi secondo necessità:

   ```
   spec:
     usedThreshold: "75%"     # Changed from 80%
     growthAmount: "20%"      # Changed from 10%
     maxSize: "1Ti"           # Changed from 500Gi
   ```

3. Salva ed esci. Le modifiche hanno effetto immediato.

Considerazioni sull'aggiornamento

- **Effetto immediato:** Tutti i volumi che utilizzano la policy adottano i nuovi parametri alla successiva valutazione della crescita.
- **Non è necessario riavviare il volume:** Le modifiche si applicano alla prossima operazione di crescita.
- **Test first:** convalidare le modifiche in un ambiente non di produzione quando possibile.
- **Comunica le modifiche:** Notifica ai team quando modifichi le policy condivise.

Elimina un criterio Autogrow

Le policy di crescita automatica utilizzano la protezione del finalizzatore per impedire la cancellazione accidentale mentre i volumi le utilizzano attivamente.

Passaggi

1. Elimina la policy:

```
kubectl delete tridentautogrowpolicy production-db-policy
```

2. Se i volumi utilizzano ancora la policy, l'eliminazione entra in uno stato `Deleting`. Verifica quali volumi sono interessati:

```
tridentctl get autogrowpolicy production-db-policy -o yaml
```

3. Rimuovi il criterio da ogni volume interessato. Scegli una delle seguenti opzioni:

   - **Opzione A: Disattivare esplicitamente l'autogrow** impostando l'annotazione su `"none"`:

```
kubectl annotate pvc <pvc-name> \
  trident.netapp.io/autogrowPolicy="none" \
  --overwrite
```

   - **Opzione B: Rimuovi completamente l'annotazione:**

```
kubectl annotate pvc <pvc-name> \
  trident.netapp.io/autogrowPolicy-
```

Comportamento di eliminazione

Scenario	Comportamento
Nessun volume utilizza la policy	La policy viene eliminata immediatamente.

Scenario	Comportamento
I volumi utilizzano la policy	La policy entra `Deleting` nello stato. Un finalizzatore blocca il completamento finché tutti i volumi non vengono rimossi.
Tutti i volumi vengono rimossi dalla policy	I finalizzatori vengono rimossi e la policy viene eliminata.

Monitorare l'utilizzo della policy Autogrow

Verificare i volumi utilizzando una policy

```
tridentctl get autogrowpolicy production-db-policy -o json | jq '.volumes'
```

Scopri quale policy utilizza un volume

```
kubectl get pvc database-pvc -o
jsonpath='{.metadata.annotations.trident\.netapp\.io/autogrowPolicy}'
```

Monitorare gli eventi delle policy

```
kubectl get events --field-selector
involvedObject.kind=TridentAutogrowPolicy
```

Protocolli supportati

Autogrow supporta i seguenti protocolli di archiviazione:

- NFS
- iSCSI
- FCP
- NVMe

NOTA	Per i volumi SAN, se la `growthAmount` configurata è pari o inferiore a 50 MiB, Trident aumenta automaticamente la quantità di crescita a 51 MB per l'operazione di ridimensionamento, a condizione che la dimensione risultante non superi `maxSize`.

Backend supportati

Autogrow supporta i backend ONTAP. Autogrow supporta anche Google Cloud NetApp Volumes (GCNV) sia per i volumi NAS che SAN.

Limitazioni note

- **ONTAP NVMe raw block volumes:** I volumi creati con versioni di ONTAP precedenti alla 9.16.1 non supportano l'autogrow.

- **Volumi esistenti (implementazioni brownfield):** L'espansione automatica potrebbe non funzionare per i volumi esistenti anche se è stata applicata una Autogrow Policy valida. Questa limitazione è dovuta a una migrazione in corso delle pubblicazioni dei volumi. Per verificare che la migrazione sia completa, controlla i log del controller Trident per messaggi `"Migration completed"`.

Domande frequenti

Quando Trident valuta la soglia?

Trident monitora costantemente l'utilizzo del volume. Quando la capacità utilizzata supera il `usedThreshold`, Trident crea una richiesta di ridimensionamento interna ed espande il volume della `growthAmount` configurata.

Ad esempio, questa policy attiva l'espansione all'80% della capacità e aumenta il volume del 10% ogni volta, fino a un massimo di 500 GiB:

```
apiVersion: trident.netapp.io/v1
kind: TridentAutogrowPolicy
metadata:
  name: standard-autogrow
spec:
  usedThreshold: "80%"
  growthAmount: "10%"
  maxSize: "500Gi"
```

Posso applicare una policy dopo che i volumi sono già stati forniti?

Sì. È possibile creare una Autogrow Policy in qualsiasi momento e applicarla ai PVC esistenti aggiungendo o aggiornando l' `trident.netapp.io/autogrowPolicy`annotazione. Non è necessario ricreare il PVC o il StorageClass.

Per applicare una policy a un PVC esistente:

```
kubectl annotate pvc <pvc-name> \
  trident.netapp.io/autogrowPolicy="production-db-policy" \
  --overwrite
```

Per applicare una policy a un'istanza esistente di StorageClass:

```
kubectl annotate storageclass ontap-gold \
  trident.netapp.io/autogrowPolicy="production-db-policy" \
  --overwrite
```

Cosa succede se imposto una politica di crescita automatica sia sulla StorageClass che sul PVC?

L'annotazione PVC ha sempre la precedenza. Se un PVC ha l'annotazione `trident.netapp.io/autogrowPolicy`, Trident utilizza quel valore indipendentemente da quanto

specificato in StorageClass. Per i dettagli, fai riferimento a "Precedenza delle policy".

Ad esempio, dato questo StorageClass:

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontap-gold
  annotations:
    trident.netapp.io/autogrowPolicy: "standard-agp"
provisioner: csi.trident.netapp.io
allowVolumeExpansion: true
```

E questo PVC che annulla la politica StorageClass:

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: database-pvc
  annotations:
    trident.netapp.io/autogrowPolicy: "logs-policy"
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 50Gi
  storageClassName: ontap-gold
```

Trident usa `logs-policy` per `database-pvc`, non `standard-agp`.

Come faccio a disattivare l'autogrow per un volume specifico?

Imposta l'annotazione PVC su `"none"`. Questa impostazione sovrascrive qualsiasi policy a livello di StorageClass per quel volume:

```
kubectl annotate pvc <pvc-name> \
  trident.netapp.io/autogrowPolicy="none" \
  --overwrite
```

Puoi verificare che autogrow sia disabilitato:

```
kubectl get pvc <pvc-name> -o jsonpath
```

```
='{.metadata.annotations.trident\.netapp\.io/autogrowPolicy}'
```

Output previsto

```
none
```

Cosa succede quando un volume raggiunge maxSize?

Trident interrompe l'espansione del volume. Non crea ulteriori richieste di ridimensionamento per quel volume, anche se l'utilizzo continua ad aumentare oltre il `usedThreshold`.

Ad esempio, con questa policy, Trident interrompe la crescita del volume dopo che raggiunge i 100 GiB:

```
apiVersion: trident.netapp.io/v1
kind: TridentAutogrowPolicy
metadata:
  name: capped-policy
spec:
  usedThreshold: "90%"
  growthAmount: "10Gi"
  maxSize: "100Gi"
```

Per consentire una crescita illimitata, ometti `maxSize` o impostalo su `0`:

```
apiVersion: trident.netapp.io/v1
kind: TridentAutogrowPolicy
metadata:
  name: unlimited-policy
spec:
  usedThreshold: "85%"
  growthAmount: "10%"
```

Posso modificare una policy senza riavviare i volumi?

Sì. Quando aggiorni una policy, tutti i volumi che utilizzano quella policy adottano i nuovi parametri alla successiva valutazione della crescita. Non è necessario riavviare i volumi.

Per aggiornare una policy in vigore:

```
kubectl edit tridentautogrowpolicy production-db-policy
```

Modifica i campi secondo necessità:

```
spec:
```

```
  usedThreshold: "75%"    # Changed from 80%
  growthAmount: "20%"     # Changed from 10%
  maxSize: "1Ti"          # Changed from 500Gi
```

Salva ed esci. Verifica la policy aggiornata:

```
kubectl get tridentautogrowpolicy production-db-policy
```

Output previsto

```
NAME                    USED THRESHOLD   GROWTH AMOUNT   STATE
production-db-policy    75%              20%             Success
```

Perché la mia policy è in stato Failed?

Uno `Failed` stato indica che la specifica della policy contiene errori di convalida. Esegui il seguente comando per visualizzare i dettagli dell'errore:

```
kubectl describe tridentautogrowpolicy <policy-name>
```

Le cause più comuni includono un `usedThreshold` non valido (deve essere compreso tra 1 e 99%), un `growthAmount` che supera `maxSize`, o un formato di quantità Kubernetes non valido. Correggere la specifica e riapplicarla:

```
kubectl apply -f autogrow-policy.yaml
```

Perché non posso eliminare una policy?

Le policy utilizzano la protezione del finalizzatore. Se i volumi utilizzano ancora la policy, l'eliminazione entra in uno stato `Deleting` e attende finché non rimuovi tutti i volumi dalla policy.

Identificare i volumi interessati:

```
tridentctl get autogrowpolicy production-db-policy -o yaml
```

Quindi rimuovere l'annotazione da ogni PVC:

```
# Option A: Explicitly disable autogrow
kubectl annotate pvc <pvc-name> \
  trident.netapp.io/autogrowPolicy="none" \
  --overwrite

# Option B: Remove the annotation entirely
```

```
kubectl annotate pvc <pvc-name> \
  trident.netapp.io/autogrowPolicy-
```

Dopo aver rimosso tutti i volumi, Trident rilascia il finalizzatore ed elimina la policy.

Quali backend sono supportati da autogrow?

Autogrow supporta i backend ONTAP. Supporta inoltre Google Cloud NetApp Volumes (GCNV) sia per i volumi NAS che SAN.

Autogrow supporta i protocolli NFS, iSCSI, FCP e NVMe. I volumi a blocchi raw NVMe richiedono ONTAP 9.16.1 o versioni successive.

I volumi esistenti nelle implementazioni brownfield potrebbero richiedere il completamento della migrazione della pubblicazione dei volumi prima che l'espansione automatica abbia effetto. Per verificare lo stato della migrazione, controlla i log del controller Trident:

```
kubectl logs -l app=trident-controller -n trident | grep "Migration
completed"
```

I seguenti esempi di StorageClass mostrano l'autogrow configurato per i backend NAS e SAN:

Backend NAS

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontap-nas-autogrow
  annotations:
    trident.netapp.io/autogrowPolicy: "standard-autogrow"
provisioner: csi.trident.netapp.io
parameters:
  backendType: "ontap-nas"
  fsType: "ext4"
allowVolumeExpansion: true
```

Backend SAN

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: database-storage
  annotations:
    trident.netapp.io/autogrowPolicy: "production-db-policy"
provisioner: csi.trident.netapp.io
parameters:
  backendType: "ontap-san"
```

```
  fsType: "ext4"
allowVolumeExpansion: true
```

Qual è la quantità minima di crescita per i volumi SAN?

Per i volumi SAN, la crescita minima effettiva è di 51 MB. Se si configura un `growthAmount` di 50 MiB o meno, Trident aumenta automaticamente la crescita a 51 MB per l'operazione di ridimensionamento.

Ad esempio, questa policy imposta una `growthAmount` di `"40Mi"`, ma Trident applica una crescita di 51 MB per qualsiasi volume SAN che la utilizza:

```
apiVersion: trident.netapp.io/v1
kind: TridentAutogrowPolicy
metadata:
  name: san-minimal-policy
spec:
  usedThreshold: "85%"
  growthAmount: "40Mi"
  maxSize: "100Gi"
```

Per evitare questa regolazione automatica, impostare `growthAmount` su un valore maggiore di 50 MiB:

```
apiVersion: trident.netapp.io/v1
kind: TridentAutogrowPolicy
metadata:
  name: san-policy
spec:
  usedThreshold: "85%"
  growthAmount: "100Mi"
  maxSize: "500Gi"
```

Importa i volumi Trident

È possibile importare volumi di archiviazione esistenti come PV Kubernetes utilizzando `tridentctl import` o creando un Persistent Volume Claim (PVC) con annotazioni di importazione Trident.

Panoramica e considerazioni

È possibile importare un volume in Trident per:

- Containerizzare un'applicazione e riutilizzare il set di dati esistente
- Utilizza un clone di un set di dati per un'applicazione effimera
- Ricostruire un cluster Kubernetes guasto

- Migrare i dati delle applicazioni durante il disaster recovery

Considerazioni

Prima di importare un volume, esaminare le seguenti considerazioni.

- Trident può importare solo volumi ONTAP di tipo RW (read-write). I volumi di tipo DP (data protection) sono volumi di destinazione di SnapMirror. È necessario interrompere la relazione di mirroring prima di importare il volume in Trident.
- Si consiglia di importare volumi senza connessioni attive. Per importare un volume utilizzato attivamente, clonare il volume e poi eseguire l'importazione.

ATTENZIONE	Questo è particolarmente importante per i volumi a blocchi, poiché Kubernetes non sarebbe a conoscenza della connessione precedente e potrebbe facilmente collegare un volume attivo a un pod. Questo può causare la corruzione dei dati.

- Sebbene `StorageClass` debba essere specificato su un PVC, Trident non utilizza questo parametro durante l'importazione. Le classi di archiviazione vengono utilizzate durante la creazione del volume per selezionare tra i pool disponibili in base alle caratteristiche di archiviazione. Poiché il volume esiste già, durante l'importazione non è richiesta la selezione del pool. Pertanto, l'importazione non fallirà anche se il volume esiste su un backend o un pool che non corrisponde alla classe di archiviazione specificata nel PVC.
- La dimensione del volume esistente viene determinata e impostata nel PVC. Dopo che il volume è stato importato dal driver di archiviazione, il PV viene creato con un ClaimRef al PVC.
  - La politica di recupero è inizialmente impostata su `retain` nel PV. Dopo che Kubernetes esegue correttamente il binding del PVC e del PV, la politica di recupero viene aggiornata per corrispondere alla politica di recupero della Storage Class.
  - Se la reclaim policy della Storage Class è `delete`, Trident elimina il volume di storage quando il PV viene eliminato.
- Per impostazione predefinita, Trident gestisce il PVC e rinomina il FlexVol volume e la LUN o qtree sul backend. Puoi passare il `--no-manage` flag per importare un volume non gestito e il `--no-rename` flag per mantenere il nome del volume.
  - `--no-manage`* - Se si utilizza il `--no-manage` flag, Trident non esegue alcuna operazione aggiuntiva sul PVC o sul PV per il ciclo di vita degli oggetti. Il volume di archiviazione non viene eliminato quando il PV viene eliminato e altre operazioni come il clone del volume e il ridimensionamento del volume vengono anch'esse ignorate.
  - `--no-rename`* - Se usi il `--no-rename`flag, Trident mantiene il nome del volume esistente durante l'importazione dei volumi e gestisce il ciclo di vita dei volumi. Questa opzione è supportata solo per i `ontap-nas`, `ontap-san` (inclusi i sistemi ASA r2), `ontap-san-economy` e `ontap-nas-economy` (importazione qtree) driver.

SUGGERIMENTO	Queste opzioni sono utili se si desidera utilizzare Kubernetes per i carichi di lavoro containerizzati, ma altrimenti si desidera gestire il ciclo di vita del volume di archiviazione al di fuori di Kubernetes.

- Al PVC e al PV viene aggiunta un'annotazione che ha il duplice scopo di indicare che il volume è stato importato e se il PVC e il PV sono gestiti. Questa annotazione non deve essere modificata o rimossa.

Importare un volume

È possibile importare un volume utilizzando `tridentctl import` oppure creando un PVC con annotazioni di importazione Trident.

NOTA Se si utilizzano annotazioni PVC, non è necessario scaricare o utilizzare `tridentctl` per importare il volume.

Utilizzo di tridentctl

Passaggi

1. Crea un file PVC (ad esempio, `pvc.yaml`) che verrà utilizzato per creare il PVC. Il file PVC dovrebbe includere `name`, `namespace`, `accessModes` e `storageClassName`. Facoltativamente, puoi specificare `unixPermissions` nella definizione del PVC.

   Di seguito è riportato un esempio di specifica minima:

   ```
   kind: PersistentVolumeClaim
   apiVersion: v1
   metadata:
     name: my_claim
     namespace: my_namespace
   spec:
     accessModes:
       - ReadWriteOnce
     storageClassName: my_storage_class
   ```

   NOTA Includere solo i parametri obbligatori. Parametri aggiuntivi come il nome del PV o la dimensione del volume possono causare il fallimento del comando di importazione.

2. Usa il comando `tridentctl import` per specificare il nome del backend Trident contenente il volume e il nome che identifica in modo univoco il volume sullo storage (ad esempio: ONTAP FlexVol, Element Volume). L'argomento `-f` è necessario per specificare il percorso del file PVC.

   ```
   tridentctl import volume <backendName> <volumeName> -f <path-to-pvc-file>
   ```

Utilizzo delle annotazioni PVC

Passaggi

1. Creare un file PVC YAML (ad esempio, `pvc.yaml`) con le annotazioni di importazione Trident richieste. Il file PVC deve includere:

   - `name` and `namespace` nei metadati
   - `accessModes`, `resources.requests.storage`, e `storageClassName` nelle specifiche
   - Annotazioni:

- `trident.netapp.io/importOriginalName`: Nome volume sul backend
- `trident.netapp.io/importBackendUUID`: UUID del backend in cui esiste il volume
- `trident.netapp.io/notManaged` (*Facoltativo*): Impostare su `"true"` per i volumi non gestiti. L'impostazione predefinita è `"false"`.

  Di seguito è riportato un esempio di specifica per l'importazione di un volume gestito:

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: <pvc-name>
  namespace: <namespace>
  annotations:
    trident.netapp.io/importOriginalName: "<volume-name>"
    trident.netapp.io/importBackendUUID: "<backend-uuid>"
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: <size>
  storageClassName: <storage-class-name>
```

2. Applica il file PVC YAML al tuo cluster Kubernetes:

```
kubectl apply -f <pvc-file>.yaml
```

Trident importerà automaticamente il volume e lo assocerà al PVC.

Esempi

Esaminare i seguenti esempi di importazione di volumi per i driver supportati.

ONTAP NAS e ONTAP NAS FlexGroup

Trident supporta l'importazione di volumi utilizzando i `ontap-nas` e `ontap-nas-flexgroup` driver.

NOTA

- Il `ontap-nas-economy` driver gestisce lo storage a livello di qtree, quindi non supporta l'importazione di un volume FlexVol autonomamente. Per importare qtree con il `ontap-nas-economy` driver, consulta Importa qtree (Tech Preview). Quando Trident importa un qtree, importa anche il FlexVol genitore come bucket che contiene il qtree.
- I `ontap-nas` e `ontap-nas-flexgroup` driver non consentono nomi di volumi duplicati.

Ogni volume creato con il `ontap-nas` driver è un volume FlexVol sul cluster ONTAP. L'importazione di volumi FlexVol con il `ontap-nas` driver funziona allo stesso modo. Un volume FlexVol già esistente su un cluster

ONTAP può essere importato come un `ontap-nas` PVC. Allo stesso modo, i volumi FlexGroup possono essere importati come `ontap-nas-flexgroup` PVC.

Esempi di ONTAP NAS utilizzando tridentctl

Gli esempi seguenti mostrano come importare volumi gestiti e non gestiti utilizzando `tridentctl`.

Volume gestito

L'esempio seguente importa un volume denominato `managed_volume` su un backend denominato `ontap_nas`:

```
tridentctl import volume ontap_nas managed_volume -f <path-to-pvc-file>

+----------------------------------------+---------+--------------
+----------+----------------------------------------+---------+----------+
|                   NAME                 |  SIZE   | STORAGE CLASS |
PROTOCOL |             BACKEND UUID             | STATE  | MANAGED |
+----------------------------------------+---------+--------------
+----------+----------------------------------------+---------+----------+
| pvc-bf5ad463-afbb-11e9-8d9f-5254004dfdb7 | 1.0 GiB | standard      |
file     | c5a6f6a4-b052-423b-80d4-8fb491a14a22 | online | true    |
+----------------------------------------+---------+--------------
+----------+----------------------------------------+---------+----------+
```

Volume non gestito

Quando si utilizza l' `--no-manage`argomento, Trident non rinomina il volume.

Il seguente esempio importa `unmanaged_volume` sul `ontap_nas` backend:

```
tridentctl import volume nas_blog unmanaged_volume -f <path-to-pvc-
file> --no-manage

+----------------------------------------+---------+--------------
+----------+----------------------------------------+---------+----------+
|                   NAME                 |  SIZE   | STORAGE CLASS |
PROTOCOL |             BACKEND UUID             | STATE  | MANAGED |
+----------------------------------------+---------+--------------
+----------+----------------------------------------+---------+----------+
| pvc-df07d542-afbc-11e9-8d9f-5254004dfdb7 | 1.0 GiB | standard      |
file     | c5a6f6a4-b052-423b-80d4-8fb491a14a22 | online | false   |
+----------------------------------------+---------+--------------
+----------+----------------------------------------+---------+----------+
```

Esempi ONTAP NAS che utilizzano annotazioni PVC

I seguenti esempi mostrano come importare volumi gestiti e non gestiti utilizzando annotazioni PVC.

Volume gestito

Il seguente esempio importa un volume da 1GiB `ontap-nas` denominato `ontap_volume1` dal backend `81abcb27-ea63-49bb-b606-0a5315ac5f21` con modalità di accesso RWO impostata tramite annotazioni PVC:

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: <managed-imported-volume>
  namespace: <namespace>
  annotations:
    trident.netapp.io/importOriginalName: "ontap_volume1"
    trident.netapp.io/importBackendUUID: "81abcb27-ea63-49bb-b606-
0a5315ac5f21"
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: <storage-class-name>
```

Volume non gestito

Il seguente esempio importa 1Gi `ontap-nas` volume denominato `ontap-volume2` dal backend `34abcb27-ea63-49bb-b606-0a5315ac5f34` con modalità di accesso RWO impostata tramite annotazioni PVC:

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: <umanaged-imported-volume>
  namespace: <namespace>
  annotations:
    trident.netapp.io/importOriginalName: "ontap-volume2"
    trident.netapp.io/importBackendUUID: "34abcb27-ea63-49bb-b606-
0a5315ac5f34"
    trident.netapp.io/notManaged: "true"
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
```

```
  storageClassName: <storage-class-name>
```

Importa qtree

NOTA L'importazione qtree è una funzionalità Tech Preview in Trident 26.06. Le funzionalità Tech Preview non sono supportate per l'utilizzo in produzione.

A partire da Trident 26.06, puoi importare i qtree ONTAP esistenti in Trident usando il `ontap-nas-economy` driver. Identifica il qtree tramite i nomi del volume FlexVol e del qtree nel formato `flexvol/qtree`, insieme all'UUID del backend che lo contiene. Le considerazioni generali sull'importazione in Panoramica e considerazioni si applicano anche all'importazione dei qtree.

Importa un qtree usando tridentctl

Specifica il qtree nel formato `flexvol/qtree` come nome del volume. Il seguente esempio importa il qtree `qtree1` dal volume FlexVol `flexvol1` sul backend `ontap_nas_economy`:

```
tridentctl import volume ontap_nas_economy flexvol1/qtree1 -f <path-to-pvc-file>
```

Per importare il qtree senza rinominarlo, aggiungi il `--no-rename` flag. Per importare il qtree senza che Trident ne gestisca il ciclo di vita, aggiungi il `--no-manage` flag.

Importa un qtree usando le annotazioni PVC

Crea un PVC con le seguenti annotazioni:

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: <pvc-name>
  namespace: <namespace>
  annotations:
    trident.netapp.io/importOriginalName: "flexvol1/qtree1"
    trident.netapp.io/importBackendUUID: "<backend-uuid>"
spec:
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: <size>
  storageClassName: <storage-class-name>
```

Per mantenere i nomi esistenti del volume FlexVol e del qtree, aggiungi l' `trident.netapp.io/importNoRename: "true"`annotazione. Per importare il qtree come non gestito, aggiungi l' `trident.netapp.io/notManaged: "true"`annotazione.

Denominazione e comportamento gestionale

Importazione gestita (impostazione predefinita)

Trident rinomina il qtree in modo che corrisponda al nome del PVC e rinomina il FlexVol volume secondo la `ontap-nas-economy` convenzione di naming. Il percorso di giunzione non cambia.

`--no-rename`

Trident mantiene i nomi esistenti dei FlexVol volume e dei qtree. I nomi devono già seguire le convenzioni di naming di Trident, altrimenti Trident non può gestire le operazioni di eliminazione o montaggio del qtree.

`--no-manage`

Trident fa riferimento al qtree ma non ne gestisce il ciclo di vita. I nomi devono seguire le convenzioni di naming di Trident e Trident non utilizza né gestisce policy di esportazione automatica per il qtree.

Comportamento della policy di esportazione

Se il backend utilizza le policy di esportazione automatica, il qtree importato utilizza il modello di policy di esportazione per qtree, in cui ogni qtree ha la propria policy di esportazione. Trident assegna al qtree una policy di esportazione vuota al momento dell'importazione. Al momento della pubblicazione, Trident aggiorna il volume FlexVol alla policy di esportazione per backend se l'esportazione automatica è abilitata e la policy di esportazione FlexVol esistente è `default` o vuota.

AVVERTENZA	L'aggiornamento della policy di esportazione del volume FlexVol può interrompere altri qtree nello stesso volume FlexVol che sono attualmente montati. Importa i qtree solo quando i qtree nel volume FlexVol sono smontati.
ATTENZIONE	Non indirizzare due cluster Kubernetes allo stesso SVM attivo e non importare un qtree che un altro cluster gestisce ancora. Se entrambi i cluster gestiscono lo stesso volume, operazioni in conflitto possono corrompere o distruggere i dati.

ONTAP SAN

Trident supporta l'importazione di volumi utilizzando i `ontap-san` (iSCSI, NVMe/TCP e FC) e `ontap-san-economy` driver.

Trident può importare volumi ONTAP SAN FlexVol che contengono una singola LUN. Questo è coerente con il driver `ontap-san`, che crea un volume FlexVol per ogni PVC e una LUN all'interno del volume FlexVol. Trident importa il volume FlexVol e lo associa alla definizione del PVC. Trident può importare volumi `ontap-san-economy` che contengono più LUN.

I seguenti esempi mostrano come importare volumi gestiti e non gestiti:

Volume gestito

Per i volumi gestiti, Trident rinomina il FlexVol volume nel `pvc-<uuid>` formato e il LUN all'interno del FlexVol volume in `lun0`.

Il seguente esempio importa il `ontap-san-managed` FlexVol volume presente sul `ontap_san_default` backend:

```
tridentctl import volume ontapsan_san_default ontap-san-managed -f pvc-basic-import.yaml -n trident -d
```

```
+-----------------------------------------+--------+--------------
+---------+-----------------------------------------+--------+---------+
|                   NAME                  |  SIZE  | STORAGE CLASS |
PROTOCOL |             BACKEND UUID              | STATE  | MANAGED |
+-----------------------------------------+--------+--------------
+---------+-----------------------------------------+--------+---------+
| pvc-d6ee4f54-4e40-4454-92fd-d00fc228d74a | 20 MiB | basic         |
block    | cd394786-ddd5-4470-adc3-10c5ce4ca757 | online | true    |
+-----------------------------------------+--------+--------------
+---------+-----------------------------------------+--------+---------+
```

Volume non gestito

Il seguente esempio importa `unmanaged_example_volume` sul `ontap_san` backend:

```
tridentctl import volume -n trident san_blog unmanaged_example_volume
-f pvc-import.yaml --no-manage
+-----------------------------------------+---------+--------------
+---------+-----------------------------------------+--------+---------+
|                   NAME                  |  SIZE   | STORAGE CLASS |
PROTOCOL |             BACKEND UUID              | STATE  | MANAGED |
+-----------------------------------------+---------+--------------
+---------+-----------------------------------------+--------+---------+
| pvc-1fc999c9-ce8c-459c-82e4-ed4380a4b228 | 1.0 GiB | san-blog      |
block    | e3275890-7d80-4af6-90cc-c7a0759f555a | online | false   |
+-----------------------------------------+---------+--------------
+---------+-----------------------------------------+--------+---------+
```

Se hai LUN mappate su igroups che condividono un IQN con un IQN di un nodo Kubernetes, come mostrato nell'esempio seguente, riceverai l'errore: `LUN already mapped to initiator(s) in this group`. Dovrai rimuovere l'initiator o annullare la mappatura della LUN per importare il volume.

```
Vserver   Igroup       Protocol OS Type  Initiators
--------- ------------ -------- -------- ------------------------------------
svm0      k8s-nodename.example.com-fe5d36f2-cded-4f38-9eb0-c7719fc2f9f3
                       iscsi    linux    iqn.1994-05.com.redhat:4c2e1cf35e0
svm0      unmanaged-example-igroup
                       mixed    linux    iqn.1994-05.com.redhat:4c2e1cf35e0
```

Esempi di ONTAP SAN-economy

I seguenti esempi mostrano come importare volumi gestiti e non gestiti per il backend. `ontap-san-economy`

Volume gestito

Quando si importa un volume gestito, Trident assume la proprietà del FlexVol e lo rinomina. È necessario tenere conto di questa ridenominazione quando si importano più LUN dallo stesso FlexVol.

Il seguente esempio importa `lun1` dal FlexVol `toimport` come volume gestito denominato `vol-managed-saneco`:

```
tridentctl import volume vol-managed-saneco toimport/lun1 -f
import1.yaml
```

Dopo l'importazione `lun1`, Trident rinomina il FlexVol (ad esempio, in `trident_lun_pool_xyz`). Per importare ulteriori LUN dallo stesso FlexVol, utilizzare il nuovo nome FlexVol:

```
tridentctl import volume vol-managed-saneco trident_lun_pool_xyz/lun2
-f import2.yaml
```

NOTA Il `ontap-san-economy` backend importa un LUN alla volta. Puoi automatizzare più importazioni utilizzando uno script.

Volume non gestito

Quando si importa un volume non gestito, Trident non assume la proprietà del FlexVol. Tuttavia, il FlexVol e la LUN devono seguire le convenzioni di denominazione di Trident.

Formato di denominazione FlexVol

```
trident_lun_pool_STORAGEPREFIX_RANDOMSTRING
```

- `STORAGEPREFIX` è il valore di `storagePrefix` nella configurazione del backend. Il valore predefinito è `trident`.
- `RANDOMSTRING` è una qualsiasi stringa tu scelga.

Requisito di denominazione LUN

Il LUN deve essere denominato `lun0`.

Esempio

Se il tuo `storagePrefix` è `xyz`, il full path al LUN è:

```
trident_lun_pool_xyz_randomstring/lun0
```

Elemento

Trident supporta il software NetApp Element e l'importazione di volumi NetApp HCI tramite il `solidfire-san` driver.

NOTA	Il driver Element supporta nomi di volume duplicati. Tuttavia, Trident restituisce un errore se ci sono nomi di volume duplicati. Come soluzione alternativa, clona il volume, fornisci un nome di volume univoco e importa il volume clonato.

Il seguente esempio importa un `element-managed` volume sul backend `element_default`.

```
tridentctl import volume element_default element-managed -f pvc-basic-
import.yaml -n trident -d

+----------------------------------------+--------+---------------
+----------+----------------------------------------+--------+---------+
|                   NAME                 |  SIZE  | STORAGE CLASS |
PROTOCOL |             BACKEND UUID             | STATE  | MANAGED |
+----------------------------------------+--------+---------------
+----------+----------------------------------------+--------+---------+
| pvc-970ce1ca-2096-4ecd-8545-ac7edc24a8fe | 10 GiB | basic-element |
block    | d3ba047a-ea0b-43f9-9c42-e38e58301c49 | online | true    |
+----------------------------------------+--------+---------------
+----------+----------------------------------------+--------+---------+
```

Azure NetApp Files

Trident supporta l'importazione di volumi utilizzando il `azure-netapp-files` driver.

NOTA	Per importare un volume di Azure NetApp Files, identifica il volume tramite il suo percorso volume. Il percorso volume è la parte del percorso di esportazione del volume dopo il `:/`. Ad esempio, se il percorso di montaggio è `10.0.0.2:/importvol1`, il percorso volume è `importvol1`.

Il seguente esempio importa un `azure-netapp-files` volume sul backend `azurenetappfiles_40517` con il percorso del volume `importvol1`.

```
tridentctl import volume azurenetappfiles_40517 importvol1 -f <path-to-
pvc-file> -n trident

+----------------------------------------+---------+---------------
+----------+----------------------------------------+--------+---------+
|                   NAME                 |  SIZE   | STORAGE CLASS |
PROTOCOL |             BACKEND UUID             | STATE  | MANAGED |
+----------------------------------------+---------+---------------
+----------+----------------------------------------+--------+---------+
| pvc-0ee95d60-fd5c-448d-b505-b72901b3a4ab | 100 GiB | anf-storage   |
```

```
file     | 1c01274f-d94b-44a3-98a3-04c953c9a51e | online | true    |
+----------+--------------------------------------+--------+---------+
```

Google Cloud NetApp Volumes

Trident supporta l'importazione di volumi utilizzando il `google-cloud-netapp-volumes` driver.

Il seguente esempio importa un volume su backend `backend-tbc-gcnv1` con il volume `testvoleasiaeast1`.

```
tridentctl import volume backend-tbc-gcnv1 "testvoleasiaeast1" -f < path-
to-pvc> -n trident

+------------------------------------------+---------
+----------------------+----------+--------------------------------------
+--------+---------+
|                   NAME                   |  SIZE   | STORAGE CLASS
| PROTOCOL |             BACKEND UUID             | STATE  | MANAGED |
+------------------------------------------+---------
+----------------------+----------+--------------------------------------
+--------+---------+
| pvc-a69cda19-218c-4ca9-a941-aea05dd13dc0 |  10 GiB | gcnv-nfs-sc-
identity | file     | 8c18cdf1-0770-4bc0-bcc5-c6295fe6d837 | online | true
|
+------------------------------------------+---------
+----------------------+----------+--------------------------------------
+--------+---------+
```

Il seguente esempio importa un `google-cloud-netapp-volumes` volume quando sono presenti due volumi nella stessa regione:

```
tridentctl import volume backend-tbc-gcnv1
"projects/123456789100/locations/asia-east1-a/volumes/testvoleasiaeast1"
-f <path-to-pvc> -n trident

+------------------------------------------+---------
+----------------------+----------+--------------------------------------
+--------+---------+
|                   NAME                   |  SIZE   | STORAGE CLASS
| PROTOCOL |             BACKEND UUID             | STATE  | MANAGED |
+------------------------------------------+---------
+----------------------+----------+--------------------------------------
+--------+---------+
| pvc-a69cda19-218c-4ca9-a941-aea05dd13dc0 |  10 GiB | gcnv-nfs-sc-
```

```
identity | file     | 8c18cdf1-0770-4bc0-bcc5-c6295fe6d837 | online | true
|
+-----------------------------------------+---------
+---------------------+----------+-------------------------------------
+--------+---------+
```

Personalizza i nomi e le etichette dei volumi di Trident

Con Trident, puoi assegnare nomi ed etichette significativi ai volumi che crei. Questo ti aiuta a identificare e mappare facilmente i volumi alle rispettive risorse Kubernetes (PVC). Puoi anche definire modelli a livello di backend per creare nomi di volumi personalizzati ed etichette personalizzate; tutti i volumi che crei, importi o cloni aderiranno ai modelli.

Prima di iniziare

Supporto per nomi ed etichette di volume personalizzabili:

- Operazioni di creazione, importazione e clonazione di volume.
- Nel caso del `ontap-nas-economy` driver, solo il nome del volume Qtree è conforme al modello di nome.
- Nel caso del `ontap-san-economy` driver, solo il nome LUN è conforme al modello di nome.

Limitazioni

- I nomi dei volumi personalizzati sono compatibili solo con i driver ONTAP on-premises.
- Le etichette personalizzate sono supportate solo per i `ontap-san`, `ontap-nas` e `ontap-nas-flexgroup` driver.
- I nomi dei volumi personalizzati non si applicano ai volumi esistenti.

Comportamenti chiave dei nomi di volume personalizzabili

- Se si verifica un errore a causa di una sintassi non valida in un modello di nome, la creazione del backend fallisce. Tuttavia, se l'applicazione del modello fallisce, il volume verrà nominato secondo la convenzione di naming esistente.
- Il prefisso di archiviazione non è applicabile quando un volume viene nominato utilizzando un name template dalla configurazione del backend. Qualsiasi valore di prefisso desiderato può essere aggiunto direttamente al template.

Esempi di configurazione del backend con name template ed etichette

I modelli di nome personalizzati possono essere definiti a livello di root e/o pool.

Esempio di livello radice

```
{
  "version": 1,
  "storageDriverName": "ontap-nas",
  "backendName": "ontap-nfs-backend",
```

```
  "managementLIF": "<ip address>",
  "svm": "svm0",
  "username": "<admin>",
  "password": "<password>",
  "defaults": {
    "nameTemplate":
"{{.volume.Name}}_{{.labels.cluster}}_{{.volume.Namespace}}_{{.volume.Requ
estName}}"
  },
  "labels": {
    "cluster": "ClusterA",
    "PVC": "{{.volume.Namespace}}_{{.volume.RequestName}}"
  }
}
```

Esempio a livello di pool

```
{
  "version": 1,
  "storageDriverName": "ontap-nas",
  "backendName": "ontap-nfs-backend",
  "managementLIF": "<ip address>",
  "svm": "svm0",
  "username": "<admin>",
  "password": "<password>",
  "useREST": true,
  "storage": [
    {
      "labels": {
        "labelname": "label1",
        "name": "{{ .volume.Name }}"
      },
      "defaults": {
        "nameTemplate": "pool01_{{ .volume.Name }}_{{ .labels.cluster
}}_{{ .volume.Namespace }}_{{ .volume.RequestName }}"
      }
    },
    {
      "labels": {
        "cluster": "label2",
        "name": "{{ .volume.Name }}"
      },
      "defaults": {
        "nameTemplate": "pool02_{{ .volume.Name }}_{{ .labels.cluster
}}_{{ .volume.Namespace }}_{{ .volume.RequestName }}"
      }
```

```
    }
  ]
}
```

Esempi di template di nome

Esempio 1:

```
"nameTemplate": "{{ .config.StoragePrefix }}_{{ .volume.Name }}_{{
.config.BackendName }}"
```

Esempio 2:

```
"nameTemplate": "pool_{{ .config.StoragePrefix }}_{{ .volume.Name }}_{{
slice .volume.RequestName 1 5 }}""
```

Punti da considerare

1. Nel caso di importazione di volumi, le etichette vengono aggiornate solo se il volume esistente ha etichette in un formato specifico. Ad esempio: `{"provisioning":{"Cluster":"ClusterA", "PVC": "pvcname"}}`.
2. Nel caso di importazioni di volumi gestiti, il nome del volume segue il modello di nome definito a livello di root nella definizione del backend.
3. Trident non supporta l'uso di un operatore slice con il prefisso dello storage.
4. Se i modelli non producono nomi di volume univoci, Trident aggiungerà alcuni caratteri casuali per creare nomi di volume univoci.
5. Se il nome personalizzato per un volume NAS economy supera i 64 caratteri, Trident nominerà i volumi in base alla convenzione di naming esistente. Per tutti gli altri driver ONTAP, se il nome del volume supera il limite di nome, il processo di creazione del volume non riesce.

Condividi un volume NFS Trident tra namespace

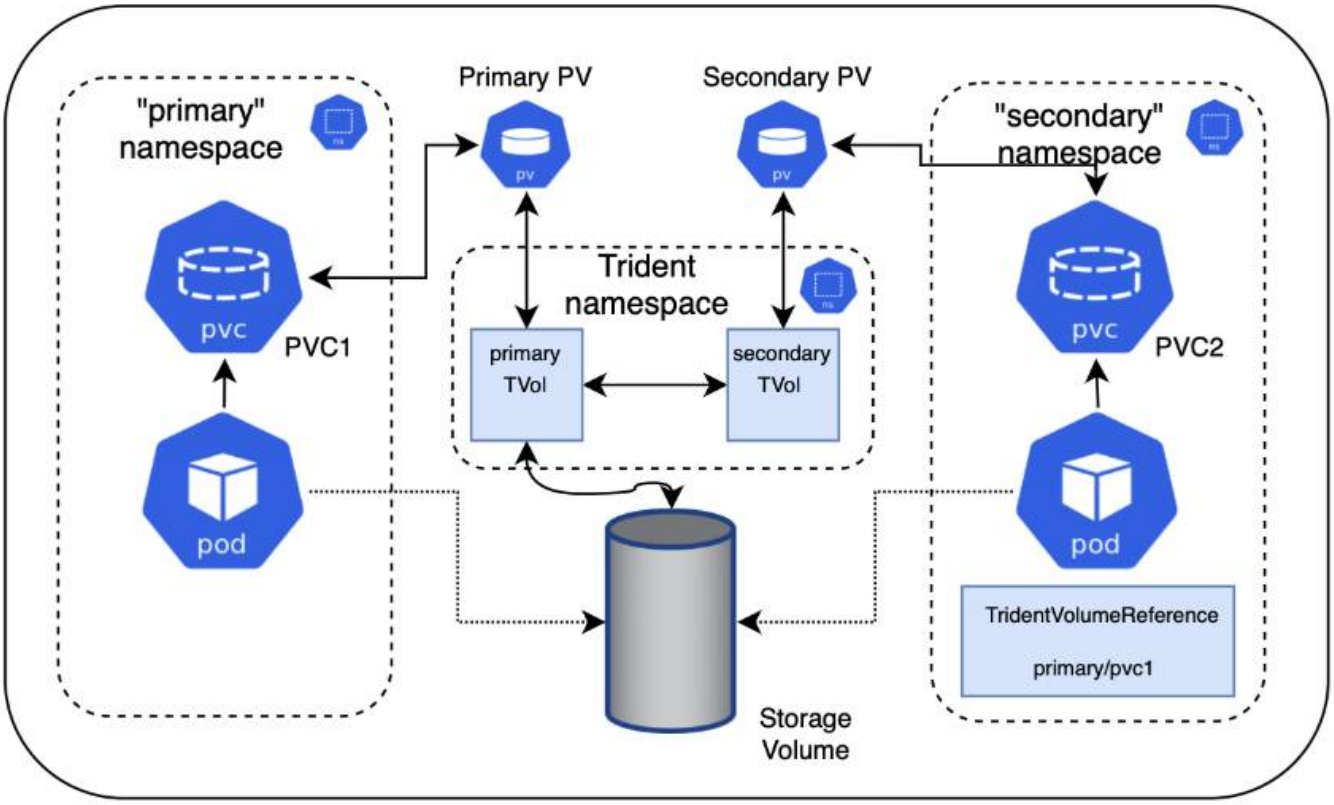
Utilizzando Trident, puoi creare un volume in uno spazio dei nomi primario e condividerlo in uno o più spazi dei nomi secondari.

Caratteristiche

La CR TridentVolumeReference consente di condividere in modo sicuro volumi NFS ReadWriteMany (RWX) su uno o più namespace Kubernetes. Questa soluzione nativa per Kubernetes offre i seguenti vantaggi:

- Più livelli di controllo degli accessi per garantire la sicurezza
- Funziona con tutti i driver di volume Trident NFS
- Nessuna dipendenza da tridentctl o da qualsiasi altra funzionalità non nativa di Kubernetes

Questo diagramma illustra la condivisione del volume NFS tra due namespace Kubernetes.



Avvio rapido

È possibile configurare la condivisione del volume NFS in pochi semplici passaggi.

1 **Configurare il PVC di origine per condividere il volume**

Il proprietario dello spazio dei nomi di origine concede il permesso di accedere ai dati nel source PVC.

2 **Concedere il permesso di creare un CR nello spazio dei nomi di destinazione**

L'amministratore del cluster concede il permesso al proprietario dello spazio dei nomi di destinazione di creare il CR TridentVolumeReference.

3 **Creare TridentVolumeReference nello spazio dei nomi di destinazione**

Il proprietario dello spazio dei nomi di destinazione crea il TridentVolumeReference CR per fare riferimento al PVC di origine.

4 **Crea il PVC subordinato nello spazio dei nomi di destinazione**

Il proprietario dello spazio dei nomi di destinazione crea il PVC subordinato per utilizzare l'origine dati dal PVC di origine.

Configurare gli spazi dei nomi di origine e di destinazione

Per garantire la sicurezza, la condivisione tra namespace richiede la collaborazione e l'azione del proprietario del namespace di origine, dell'amministratore del cluster e del proprietario del namespace di destinazione. Il ruolo dell'utente viene assegnato in ogni fase.

Passaggi

1. **Proprietario dello spazio dei nomi di origine:** crea il PVC (`pvc1`) nello spazio dei nomi di origine che concede l'autorizzazione alla condivisione con lo spazio dei nomi di destinazione (`namespace2`) utilizzando l'annotazione `shareToNamespace`.

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: pvc1
  namespace: namespace1
  annotations:
    trident.netapp.io/shareToNamespace: namespace2
spec:
  accessModes:
    - ReadWriteMany
  storageClassName: trident-csi
  resources:
    requests:
      storage: 100Gi
```

   Trident crea il PV e il suo volume di archiviazione NFS backend.

   NOTA

   - È possibile condividere il PVC con più namespace utilizzando un elenco delimitato da virgole. Ad esempio, `trident.netapp.io/shareToNamespace: namespace2,namespace3,namespace4`.
   - Puoi condividere con tutti gli spazi dei nomi utilizzando `*`. Ad esempio, `trident.netapp.io/shareToNamespace: *`
   - È possibile aggiornare il PVC per includere l `shareToNamespace`annotazione in qualsiasi momento.

2. **Amministratore del cluster:** assicurarsi che sia presente il corretto RBAC per concedere l'autorizzazione al proprietario dello spazio dei nomi di destinazione di creare il CR TridentVolumeReference nello spazio dei nomi di destinazione.
3. **Proprietario dello spazio dei nomi di destinazione:** Creare un TridentVolumeReference CR nello spazio dei nomi di destinazione che faccia riferimento allo spazio dei nomi di origine `pvc1`.

```
apiVersion: trident.netapp.io/v1
kind: TridentVolumeReference
metadata:
  name: my-first-tvr
```

```
  namespace: namespace2
spec:
  pvcName: pvc1
  pvcNamespace: namespace1
```

4. **Proprietario dello spazio dei nomi di destinazione:** Crea un PVC (`pvc2` nello spazio dei nomi di destinazione (`namespace2` utilizzando l'annotazione `shareFromPVC` per designare il PVC di origine.

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  annotations:
    trident.netapp.io/shareFromPVC: namespace1/pvc1
  name: pvc2
  namespace: namespace2
spec:
  accessModes:
    - ReadWriteMany
  storageClassName: trident-csi
  resources:
    requests:
      storage: 100Gi
```

NOTA La dimensione del PVC di destinazione deve essere inferiore o uguale a quella del PVC di origine.

Risultati

Trident legge l' `shareFromPVC`annotazione sul PVC di destinazione e crea il PV di destinazione come volume subordinato, senza risorse di storage proprie, che punta al PV di origine e condivide la risorsa di storage del PV di origine. Il PVC di destinazione e il PV di destinazione appaiono vincolati come di consueto.

Elimina un volume condiviso

È possibile eliminare un volume condiviso tra più namespace. Trident rimuoverà l'accesso al volume nel namespace di origine e manterrà l'accesso per gli altri namespace che condividono il volume. Quando tutti i namespace che fanno riferimento al volume vengono rimossi, Trident elimina il volume.

Utilizzare `tridentctl get` per interrogare i volumi subordinati

Utilizzando l'[`tridentctl`utility, puoi eseguire il `get`comando per ottenere i volumi subordinati. Per ulteriori informazioni, consultare `tridentctl` comandi e opzioni.

```
Usage:
  tridentctl get [option]
```

Flag:

- `-h, --help`: Aiuto per i volumi.
- `--parentOfSubordinate string`: Limita la query al volume sorgente subordinato.
- `--subordinateOf string`: Limita la query ai subordinati del volume.

Limitazioni

- Trident non può impedire ai namespace di destinazione di scrivere sul volume condiviso. È consigliabile utilizzare il blocco dei file o altri processi per impedire la sovrascrittura dei dati del volume condiviso.
- Non è possibile revocare l'accesso al PVC sorgente rimuovendo le `shareToNamespace` o `shareFromNamespace` annotazioni o eliminando il `TridentVolumeReference` CR. Per revocare l'accesso, è necessario eliminare il PVC subordinato.
- Snapshot, cloni e mirroring non sono possibili sui volumi subordinati.

Per ulteriori informazioni

Per saperne di più sull'accesso ai volumi tra namespace:

- Guarda la demo su "NetAppTV".

Clona i volumi Trident tra namespace

Utilizzando Trident, è possibile creare nuovi volumi utilizzando volumi esistenti o volumesnapshots da un namespace diverso all'interno dello stesso cluster Kubernetes.

Prerequisiti

Prima di clonare i volumi, assicurarsi che i backend di origine e di destinazione siano dello stesso tipo e abbiano la stessa classe di storage.

NOTA	La clonazione tra spazi dei nomi è supportata solo per i driver di archiviazione `ontap-san` e `ontap-nas`. Le clonazioni di sola lettura non sono supportate.

Avvio rapido

È possibile configurare la clonazione dei volumi in pochi passaggi.

1. **Configura il PVC sorgente per clonare il volume**

Il proprietario dello spazio dei nomi di origine concede il permesso di accedere ai dati nel source PVC.

2. **Concedere il permesso di creare un CR nello spazio dei nomi di destinazione**

L'amministratore del cluster concede il permesso al proprietario dello spazio dei nomi di destinazione di creare il CR TridentVolumeReference.

3 Creare TridentVolumeReference nello spazio dei nomi di destinazione

Il proprietario dello spazio dei nomi di destinazione crea il TridentVolumeReference CR per fare riferimento al PVC di origine.

4 Creare il PVC clone nello spazio dei nomi di destinazione

Il proprietario dello spazio dei nomi di destinazione crea un PVC per clonare il PVC dallo spazio dei nomi di origine.

Configurare gli spazi dei nomi di origine e di destinazione

Per garantire la sicurezza, la clonazione di volumi tra spazi dei nomi richiede la collaborazione e l'azione del proprietario dello spazio dei nomi di origine, dell'amministratore del cluster e del proprietario dello spazio dei nomi di destinazione. Il ruolo dell'utente viene assegnato in ogni fase.

Passaggi

1. **Proprietario dello spazio dei nomi di origine:** Crea il PVC (`pvc1`) nello spazio dei nomi di origine (`namespace1`) che concede il permesso di condividere con lo spazio dei nomi di destinazione (`namespace2`) usando l'annotazione `cloneToNamespace`.

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: pvc1
  namespace: namespace1
  annotations:
    trident.netapp.io/cloneToNamespace: namespace2
spec:
  accessModes:
    - ReadWriteMany
  storageClassName: trident-csi
  resources:
    requests:
      storage: 100Gi
```

Trident crea il PV e il suo volume di storage backend.

NOTA

- È possibile condividere il PVC con più namespace utilizzando un elenco delimitato da virgole. Ad esempio, `trident.netapp.io/cloneToNamespace: namespace2,namespace3,namespace4`.
- Puoi condividere con tutti gli spazi dei nomi utilizzando `*`. Ad esempio, `trident.netapp.io/cloneToNamespace: *`
- È possibile aggiornare il PVC per includere l `cloneToNamespace`annotazione in qualsiasi momento.

2. **Cluster admin:** Assicurarsi che sia presente un RBAC appropriato per concedere il permesso al proprietario dello spazio dei nomi destinazione di creare il TridentVolumeReference CR nello spazio dei nomi destinazione (`namespace2`).
3. **Proprietario dello spazio dei nomi di destinazione:** Crea un CR TridentVolumeReference nello spazio dei nomi di destinazione che faccia riferimento al PVC di origine `pvc1`.

```
apiVersion: trident.netapp.io/v1
kind: TridentVolumeReference
metadata:
  name: my-first-tvr
  namespace: namespace2
spec:
  pvcName: pvc1
  pvcNamespace: namespace1
```

4. **Proprietario dello spazio dei nomi di destinazione:** Creare un PVC (`pvc2`) nello spazio dei nomi di destinazione (`namespace2`) usando le `cloneFromPVC` o `cloneFromSnapshot`, e `cloneFromNamespace` annotazioni per designare il PVC di origine.

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  annotations:
    trident.netapp.io/cloneFromPVC: pvc1
    trident.netapp.io/cloneFromNamespace: namespace1
  name: pvc2
  namespace: namespace2
spec:
  accessModes:
    - ReadWriteMany
  storageClassName: trident-csi
  resources:
    requests:
      storage: 100Gi
```

Limitazioni

- Per i PVC provisionati utilizzando i driver ontap-nas-economy, i cloni di sola lettura non sono supportati.

Replicare i volumi utilizzando SnapMirror

Trident supporta le relazioni di mirroring tra un volume di origine su un cluster e il volume di destinazione sul cluster peered per replicare i dati per il disaster recovery. Puoi utilizzare una Custom Resource Definition (CRD) namespaced, chiamata Trident Mirror Relationship (TMR), per eseguire le seguenti operazioni:

- Crea relazioni di mirroring tra volumi (PVC)
- Rimuovere le relazioni di mirroring tra i volumi
- Interrompere le relazioni a specchio
- Promuovere il volume secondario durante le condizioni di disastro (failover)
- Esegui la transizione senza perdite delle applicazioni da un cluster a un altro cluster (durante i failover o le migrazioni pianificate)

Prerequisiti della replica

Assicurarsi che siano soddisfatti i seguenti prerequisiti prima di iniziare:

Cluster ONTAP

- **Trident**: Trident versione 22.10 o successiva deve essere presente sia sul cluster Kubernetes di origine che su quello di destinazione che utilizzano ONTAP come backend.
- **Licenze**: Le licenze asincrone ONTAP SnapMirror che utilizzano il bundle Data Protection devono essere abilitate sia sul cluster ONTAP di origine che su quello di destinazione. Consulta "Panoramica delle licenze SnapMirror in ONTAP" per maggiori informazioni.

  A partire da ONTAP 9.10.1, tutte le licenze vengono fornite come NetApp license file (NLF), ovvero un singolo file che abilita più funzionalità. Per ulteriori informazioni, consulta "Licenze incluse con ONTAP One".

NOTA	È supportata solo la protezione asincrona SnapMirror.

Peering

- **Cluster e SVM**: I backend di storage ONTAP devono essere sottoposti a peering. Consulta "Panoramica del peering di cluster e SVM" per maggiori informazioni.

IMPORTANTE	Assicurarsi che i nomi SVM utilizzati nella relazione di replica tra due cluster ONTAP siano univoci.

- **Trident e SVM**: le SVM remote peered devono essere disponibili per Trident sul cluster di destinazione.

Driver supportati

NetApp Trident supporta la replicazione del volume con NetApp SnapMirror technology utilizzando classi di archiviazione supportate dai seguenti driver: **`ontap-nas`: NFS** `ontap-san`: iSCSI **`ontap-san`: FC** `ontap-san`: NVMe/TCP (richiede la versione di ONTAP minima 9.15.1)

NOTA	La replicazione del volume tramite SnapMirror non è supportata per i sistemi ASA r2. Per informazioni sui sistemi ASA r2, vedere "Scopri i sistemi di storage ASA r2".

Crea un PVC specchiato

Seguire questi passaggi e utilizzare gli esempi CRD per creare una relazione mirror tra volumi primari e secondari.

Passaggi

1. Eseguire i seguenti passaggi sul cluster Kubernetes primario:
   a. Crea un oggetto StorageClass con il `trident.netapp.io/replication: true` parametro.

Esempio

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: csi-nas
provisioner: csi.trident.netapp.io
parameters:
  backendType: "ontap-nas"
  fsType: "nfs"
  trident.netapp.io/replication: "true"
```

b. Crea un PVC con la StorageClass creata in precedenza.

Esempio

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: csi-nas
spec:
  accessModes:
  - ReadWriteMany
  resources:
    requests:
      storage: 1Gi
  storageClassName: csi-nas
```

c. Crea un MirrorRelationship CR con informazioni locali.

Esempio

```
kind: TridentMirrorRelationship
apiVersion: trident.netapp.io/v1
metadata:
  name: csi-nas
spec:
  state: promoted
  volumeMappings:
  - localPVCName: csi-nas
```

Trident recupera le informazioni interne per il volume e lo stato corrente di protezione dei dati (DP) del volume, quindi popola il campo di stato del MirrorRelationship.

d. Ottieni il TridentMirrorRelationship CR per ottenere il nome interno e l'SVM del PVC.

```
kubectl get tmr csi-nas
```

```
kind: TridentMirrorRelationship
apiVersion: trident.netapp.io/v1
metadata:
  name: csi-nas
  generation: 1
spec:
  state: promoted
  volumeMappings:
  - localPVCName: csi-nas
status:
  conditions:
  - state: promoted
    localVolumeHandle:
"datavserver:trident_pvc_3bedd23c_46a8_4384_b12b_3c38b313c1e1"
    localPVCName: csi-nas
    observedGeneration: 1
```

2. Eseguire i seguenti passaggi sul cluster Kubernetes secondario:
   a. Crea un StorageClass con il parametro trident.netapp.io/replication: true.

      Esempio

      ```
      apiVersion: storage.k8s.io/v1
      kind: StorageClass
      metadata:
        name: csi-nas
      provisioner: csi.trident.netapp.io
      parameters:
        trident.netapp.io/replication: true
      ```

   b. Crea un MirrorRelationship CR con informazioni sulla destinazione e sulla sorgente.

      Esempio

      ```
      kind: TridentMirrorRelationship
      apiVersion: trident.netapp.io/v1
      metadata:
        name: csi-nas
      spec:
        state: established
        replicationPolicy: MirrorAllSnapshots
        replicationSchedule: 5min
      ```

```
  volumeMappings:
  - localPVCName: csi-nas
    remoteVolumeHandle:
"datavserver:trident_pvc_3bedd23c_46a8_4384_b12b_3c38b313c1e1"
```

Il TMR `spec` ha quattro campi di primo livello: `state`, che imposta lo stato di relazione desiderato (vedi Stati di replicazione del volume); `volumeMappings`, che associa i PVC locali ai relativi handle di volume remoti; e i campi facoltativi `replicationPolicy` e `replicationSchedule` descritti di seguito.

I campi `replicationPolicy` e `replicationSchedule` sono facoltativi. Imposta `replicationPolicy` sul nome di un criterio asincrono di SnapMirror su ONTAP e `replicationSchedule` sul nome di una pianificazione ONTAP. Se ometti uno dei due campi, Trident utilizza il criterio di replica o la pianificazione corrispondente configurata sul backend. Imposta questi campi quando crei la relazione; non puoi modificare il criterio o la pianificazione dopo che la relazione è stata stabilita. Per utilizzare impostazioni diverse, elimina e ricrea il TMR.

Trident crea e inizializza una relazione di SnapMirror utilizzando la policy di replica e la pianificazione configurate.

1. Creare un PVC con la StorageClass precedentemente creata per fungere da secondario (SnapMirror destinazione).

   Esempio

   ```
   kind: PersistentVolumeClaim
   apiVersion: v1
   metadata:
     name: csi-nas
     annotations:
       trident.netapp.io/mirrorRelationship: csi-nas
   spec:
     accessModes:
     - ReadWriteMany
   resources:
     requests:
       storage: 1Gi
   storageClassName: csi-nas
   ```

   Trident verificherà la presenza del TridentMirrorRelationship CRD e non riuscirà a creare il volume se la relazione non esiste. Se la relazione esiste, Trident assicurerà che il nuovo FlexVol volume venga posizionato su una SVM in peering con la SVM remota definita nel MirrorRelationship.

Stati di replicazione del volume

Una Trident Mirror Relationship (TMR) è un CRD che rappresenta un'estremità di una relazione di replicazione tra PVC. La TMR di destinazione ha uno stato, che indica a Trident qual è lo stato desiderato. La TMR di destinazione ha i seguenti stati:

- **Stabilito**: il PVC locale è il volume di destinazione di una relazione mirror e questa è una nuova relazione.
- **Promosso**: il PVC locale è ReadWrite e montabile, con nessuna relazione speculare attualmente in vigore.
- **Ristabilito**: il PVC locale è il volume di destinazione di una relazione di SnapMirror ed era anche precedentemente in quella relazione di SnapMirror.
  - Lo stato ristabilito deve essere utilizzato se il volume di destinazione è mai stato in una relazione con il volume di origine, perché sovrascrive il contenuto del volume di destinazione.
  - Lo stato ripristinato non riuscirà se il volume non era precedentemente in una relazione con la sorgente.

Promuovere il PVC secondario durante un failover non pianificato

Eseguire il seguente passaggio sul cluster Kubernetes secondario:

- Aggiorna il campo *spec.state* di TridentMirrorRelationship a `promoted`.

Promuovere il PVC secondario durante un failover pianificato

Durante un failover pianificato (migrazione), eseguire i seguenti passaggi per promuovere il PVC secondario:

Passaggi

1. Sul cluster Kubernetes primario, crea uno snapshot del PVC e attendi fino a quando lo snapshot viene creato.
2. Sul cluster Kubernetes primario, crea il TridentTridentSnapshotInfoCR per ottenere i dettagli interni.

   Esempio

   ```
   kind: TridentSnapshotInfo
   apiVersion: trident.netapp.io/v1
   metadata:
     name: csi-nas
   spec:
     snapshotName csi-nas-snapshot
   ```

   a. Sul cluster Kubernetes secondario, aggiorna il campo *spec.state* del CR *TridentMirrorRelationship* a *promoted* e *spec.volumeMappings[].promotedSnapshotHandle* al internalName dello snapshot.
3. Nel cluster Kubernetes secondario, confermare lo stato (campo status.state) di TridentMirrorRelationship su promoted.

Ripristina una relazione mirror dopo un failover

Prima di ripristinare una relazione speculare, scegli il lato che vuoi rendere come nuovo primario.

Passaggi

1. Nel cluster Kubernetes secondario, assicurati che i valori per il campo *spec.remoteVolumeHandle* su TridentMirrorRelationship siano aggiornati.
2. Sul cluster Kubernetes secondario, aggiorna il campo *spec.state* di TridentMirrorRelationship in `reestablished`.

Operazioni aggiuntive

Trident supporta le seguenti operazioni sui volumi primario e secondario:

Replica il PVC primario in un nuovo PVC secondario

Assicurati di avere già un PVC primario e un PVC secondario.

Passaggi

1. Eliminare i CRD PersistentVolumeClaim e TridentMirrorRelationship dal cluster secondario (di destinazione) stabilito.
2. Eliminare il TridentMirrorRelationship CRD dal cluster primario (sorgente).
3. Crea un nuovo TridentMirrorRelationship CRD sul cluster primario (sorgente) per il nuovo PVC secondario (destinazione) che si desidera stabilire.

Ridimensiona un PVC mirrorato, primario o secondario

Il PVC può essere ridimensionato normalmente, ONTAP espanderà automaticamente tutti i flevxols di destinazione se la quantità di dati supera la dimensione corrente.

Rimuovi la replicazione da un PVC

Per rimuovere la replica, eseguire una delle seguenti operazioni sul volume secondario corrente:

- Eliminare il MirrorRelationship sul PVC secondario. Questo interrompe la relazione di replicazione.
- Oppure, aggiorna il campo spec.state su *promoted*.

Elimina un PVC (che era stato precedentemente mirrorato)

Trident verifica la presenza di PVC replicati e rilascia la relazione di replicazione prima di tentare di eliminare il volume.

Elimina un TMR

L'eliminazione di un TMR su un lato di una relazione mirror fa sì che il TMR rimanente passi allo stato *promoted* prima che Trident completi l'eliminazione. Se il TMR selezionato per l'eliminazione è già nello stato *promoted*, non esiste alcuna relazione mirror e il TMR verrà rimosso e Trident promuoverà il PVC locale a *ReadWrite*. Questa eliminazione rilascia i metadati SnapMirror per il volume locale in ONTAP. Se questo volume verrà utilizzato in una relazione mirror in futuro, dovrà utilizzare un nuovo TMR con uno stato di replica del volume *established* durante la creazione della nuova relazione mirror.

Aggiorna le relazioni mirror quando ONTAP è online

Le relazioni mirror possono essere aggiornate in qualsiasi momento dopo essere state stabilite. È possibile utilizzare i campi `state: promoted` o `state: reestablished` per aggiornare le relazioni. Quando si promuove un volume di destinazione a un volume ReadWrite normale, è possibile utilizzare *promotedSnapshotHandle* per specificare uno snapshot specifico a cui ripristinare il volume corrente.

Aggiorna le relazioni mirror quando ONTAP è offline

È possibile utilizzare un CRD per eseguire un aggiornamento SnapMirror senza che Trident abbia connettività diretta al cluster ONTAP. Fare riferimento al seguente formato di esempio di TridentActionMirrorUpdate:

Esempio

```
apiVersion: trident.netapp.io/v1
kind: TridentActionMirrorUpdate
metadata:
  name: update-mirror-b
spec:
  snapshotHandle: "pvc-1234/snapshot-1234"
  tridentMirrorRelationshipName: mirror-b
```

`status.state` riflette lo stato del TridentActionMirrorUpdate CRD. Può assumere un valore tra *Succeeded*, *In Progress* o *Failed*.

Usa la topologia CSI

Trident può creare e collegare selettivamente i volumi ai nodi presenti in un cluster Kubernetes facendo uso del "Funzione Topology CSI".

Panoramica

Utilizzando la funzionalità CSI Topology, l'accesso ai volumi può essere limitato a un sottoinsieme di nodi, in base alle regioni e alle zone di disponibilità. I provider di cloud oggi consentono agli amministratori di Kubernetes di creare nodi basati sulle zone. I nodi possono essere situati in diverse zone di disponibilità all'interno di una regione o in varie regioni. Per facilitare il provisioning dei volumi per i carichi di lavoro in un'architettura multizona, Trident utilizza CSI Topology.

SUGGERIMENTO Scopri di più sulla funzione CSI Topology "qui".

Kubernetes offre due modalità uniche di binding dei volumi:

- Con `VolumeBindingMode` impostato su `Immediate`, Trident crea il volume senza alcuna consapevolezza della topologia. Il binding del volume e il provisioning dinamico vengono gestiti quando viene creato il PVC. Questo è il valore predefinito `VolumeBindingMode` ed è adatto per i cluster che non applicano vincoli topologici. I volumi persistenti vengono creati senza alcuna dipendenza dai requisiti di pianificazione del pod richiedente.
- Con `VolumeBindingMode` impostato su `WaitForFirstConsumer`, la creazione e il binding di un Persistent Volume per un PVC vengono ritardati fino a quando un pod che utilizza il PVC viene pianificato e creato. In questo modo, i volumi vengono creati per soddisfare i vincoli di pianificazione imposti dai requisiti topologici.

NOTA La `WaitForFirstConsumer` modalità di binding non richiede etichette di topologia. Questo può essere utilizzato indipendentemente dalla funzione Topologia CSI.

Cosa ti servirà

Per utilizzare la Topologia CSI, è necessario quanto segue:

- Un cluster Kubernetes che esegue un "versione supportata di Kubernetes"

```
kubectl version
```

```
Client Version: version.Info{Major:"1", Minor:"19",
GitVersion:"v1.19.3",
GitCommit:"1e11e4a2108024935ecfcb2912226cedeafd99df",
GitTreeState:"clean", BuildDate:"2020-10-14T12:50:19Z",
GoVersion:"go1.15.2", Compiler:"gc", Platform:"linux/amd64"}
Server Version: version.Info{Major:"1", Minor:"19",
GitVersion:"v1.19.3",
GitCommit:"1e11e4a2108024935ecfcb2912226cedeafd99df",
GitTreeState:"clean", BuildDate:"2020-10-14T12:41:49Z",
GoVersion:"go1.15.2", Compiler:"gc", Platform:"linux/amd64"}
```

- I nodi del cluster dovrebbero avere etichette che introducono la consapevolezza della topologia (`topology.kubernetes.io/region` e `topology.kubernetes.io/zone`). Queste etichette **dovrebbero essere presenti sui nodi del cluster** prima che Trident sia installato affinché Trident sia consapevole della topologia.

```
kubectl get nodes -o=jsonpath='{range .items[*]}[{.metadata.name},
{.metadata.labels}]{"\n"}{end}' | grep --color "topology.kubernetes.io"
[node1,
{"beta.kubernetes.io/arch":"amd64","beta.kubernetes.io/os":"linux","kube
rnetes.io/arch":"amd64","kubernetes.io/hostname":"node1","kubernetes.io/
os":"linux","node-
role.kubernetes.io/master":"","topology.kubernetes.io/region":"us-
east1","topology.kubernetes.io/zone":"us-east1-a"}]
[node2,
{"beta.kubernetes.io/arch":"amd64","beta.kubernetes.io/os":"linux","kube
rnetes.io/arch":"amd64","kubernetes.io/hostname":"node2","kubernetes.io/
os":"linux","node-
role.kubernetes.io/worker":"","topology.kubernetes.io/region":"us-
east1","topology.kubernetes.io/zone":"us-east1-b"}]
[node3,
{"beta.kubernetes.io/arch":"amd64","beta.kubernetes.io/os":"linux","kube
rnetes.io/arch":"amd64","kubernetes.io/hostname":"node3","kubernetes.io/
os":"linux","node-
role.kubernetes.io/worker":"","topology.kubernetes.io/region":"us-
east1","topology.kubernetes.io/zone":"us-east1-c"}]
```

Passo 1: Creare un backend consapevole della topologia

I backend di storage Trident possono essere progettati per fornire selettivamente volumi in base alle zone di disponibilità. Ogni backend può contenere un blocco opzionale `supportedTopologies` che rappresenta un elenco di zone e regioni supportate. Per le StorageClasses che fanno uso di un backend di questo tipo, un volume viene creato solo se richiesto da un'applicazione pianificata in una regione/zona supportata.

Ecco un esempio di definizione backend:

YAML

```yaml
---
version: 1
storageDriverName: ontap-san
backendName: san-backend-us-east1
managementLIF: 192.168.27.5
svm: iscsi_svm
username: admin
password: password
supportedTopologies:
  - topology.kubernetes.io/region: us-east1
    topology.kubernetes.io/zone: us-east1-a
  - topology.kubernetes.io/region: us-east1
    topology.kubernetes.io/zone: us-east1-b
```

JSON

```json
{
  "version": 1,
  "storageDriverName": "ontap-san",
  "backendName": "san-backend-us-east1",
  "managementLIF": "192.168.27.5",
  "svm": "iscsi_svm",
  "username": "admin",
  "password": "password",
  "supportedTopologies": [
    {
      "topology.kubernetes.io/region": "us-east1",
      "topology.kubernetes.io/zone": "us-east1-a"
    },
    {
      "topology.kubernetes.io/region": "us-east1",
      "topology.kubernetes.io/zone": "us-east1-b"
    }
  ]
}
```

NOTA

`supportedTopologies` è usato per fornire un elenco di regioni e zone per backend. Queste regioni e zone rappresentano l'elenco dei valori consentiti che possono essere forniti in un StorageClass. Per StorageClasses che contengono un sottoinsieme delle regioni e zone fornite in un backend, Trident crea un volume sul backend.

È possibile definire `supportedTopologies` per pool di storage anche. Vedere il seguente esempio:

```
---
version: 1
storageDriverName: ontap-nas
backendName: nas-backend-us-central1
managementLIF: 172.16.238.5
svm: nfs_svm
username: admin
password: password
supportedTopologies:
  - topology.kubernetes.io/region: us-central1
    topology.kubernetes.io/zone: us-central1-a
  - topology.kubernetes.io/region: us-central1
    topology.kubernetes.io/zone: us-central1-b
storage:
  - labels:
      workload: production
    supportedTopologies:
      - topology.kubernetes.io/region: us-central1
        topology.kubernetes.io/zone: us-central1-a
  - labels:
      workload: dev
    supportedTopologies:
      - topology.kubernetes.io/region: us-central1
        topology.kubernetes.io/zone: us-central1-b
```

In questo esempio, le etichette `region` e `zone` indicano la posizione del pool di storage. `topology.kubernetes.io/region` e `topology.kubernetes.io/zone` stabiliscono da dove possono essere consumati i pool di storage.

Passo 2: Definire StorageClasses che sono consapevoli della topologia

In base alle etichette topologiche fornite ai nodi nel cluster, StorageClasses può essere definito per contenere informazioni sulla topologia. Questo determinerà i pool di storage che servono come candidati per le richieste di PVC effettuate e il sottoinsieme di nodi che possono utilizzare i volumi forniti da Trident.

Vedere il seguente esempio:

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: netapp-san-us-east1
provisioner: csi.trident.netapp.io
volumeBindingMode: WaitForFirstConsumer
allowedTopologies:
  - matchLabelExpressions:
    - key: topology.kubernetes.io/zone
```

```
      values:
        - us-east1-a
        - us-east1-b
    - key: topology.kubernetes.io/region
      values:
        - us-east1
parameters:
  fsType: ext4
```

Nella definizione di StorageClass fornita sopra, `volumeBindingMode` è impostato su `WaitForFirstConsumer`. I PVC richiesti con questo StorageClass non saranno utilizzati finché non saranno referenziati in un pod. E, `allowedTopologies` fornisce le zone e la regione da utilizzare. Il `netapp-san-us-east1` StorageClass crea i PVC sul `san-backend-us-east1` backend definito sopra.

Fase 3: Creare e utilizzare un PVC

Con la StorageClass creata e mappata su un backend, ora puoi creare i PVC.

Vedi l'esempio `spec` qui sotto:

```
---
kind: PersistentVolumeClaim
apiVersion: v1
metadata: null
name: pvc-san
spec: null
accessModes:
  - ReadWriteOnce
resources:
  requests:
    storage: 300Mi
storageClassName: netapp-san-us-east1
```

La creazione di un PVC utilizzando questo manifest avrebbe il seguente risultato:

```
kubectl create -f pvc.yaml
persistentvolumeclaim/pvc-san created
kubectl get pvc
NAME      STATUS    VOLUME   CAPACITY   ACCESS MODES   STORAGECLASS
AGE
pvc-san   Pending                                      netapp-san-us-east1
2s
kubectl describe pvc
Name:          pvc-san
Namespace:     default
```

```
StorageClass:  netapp-san-us-east1
Status:        Pending
Volume:
Labels:        <none>
Annotations:   <none>
Finalizers:    [kubernetes.io/pvc-protection]
Capacity:
Access Modes:
VolumeMode:    Filesystem
Mounted By:    <none>
Events:
  Type    Reason                Age   From                         Message
  ----    ------                ----  ----                         -------
  Normal  WaitForFirstConsumer  6s    persistentvolume-controller  waiting
for first consumer to be created before binding
```

Per consentire a Trident di creare un volume e associarlo al PVC, utilizzare il PVC in un pod. Vedere il seguente esempio:

```
apiVersion: v1
kind: Pod
metadata:
  name: app-pod-1
spec:
  affinity:
    nodeAffinity:
      requiredDuringSchedulingIgnoredDuringExecution:
        nodeSelectorTerms:
        - matchExpressions:
          - key: topology.kubernetes.io/region
            operator: In
            values:
            - us-east1
      preferredDuringSchedulingIgnoredDuringExecution:
      - weight: 1
        preference:
          matchExpressions:
          - key: topology.kubernetes.io/zone
            operator: In
            values:
            - us-east1-a
            - us-east1-b
  securityContext:
    runAsUser: 1000
    runAsGroup: 3000
```

```
    fsGroup: 2000
  volumes:
  - name: vol1
    persistentVolumeClaim:
      claimName: pvc-san
  containers:
  - name: sec-ctx-demo
    image: busybox
    command: [ "sh", "-c", "sleep 1h" ]
    volumeMounts:
    - name: vol1
      mountPath: /data/demo
    securityContext:
      allowPrivilegeEscalation: false
```

Questo podSpec indica a Kubernetes di programmare il pod sui nodi presenti nella `us-east1` regione, e di scegliere tra qualsiasi nodo presente nella `us-east1-a` o `us-east1-b` zone.

Vedi il seguente output:

```
kubectl get pods -o wide
NAME        READY   STATUS    RESTARTS   AGE   IP               NODE
NOMINATED NODE   READINESS GATES
app-pod-1   1/1     Running   0          19s   192.168.25.131   node2
<none>           <none>
kubectl get pvc -o wide
NAME      STATUS   VOLUME                                     CAPACITY
ACCESS MODES   STORAGECLASS          AGE   VOLUMEMODE
pvc-san   Bound    pvc-ecb1e1a0-840c-463b-8b65-b3d033e2e62b   300Mi
RWO            netapp-san-us-east1   48s   Filesystem
```

Aggiorna i backend per includere `supportedTopologies`

I backend preesistenti possono essere aggiornati per includere un elenco di `supportedTopologies` utilizzando `tridentctl backend update`. Questo non influirà sui volumi che sono già stati forniti e sarà utilizzato solo per i PVC successivi.

Trova ulteriori informazioni

- "Gestisci le risorse per i contenitori"
- "nodeSelector"
- "Affinità e anti-affinità"
- "Taints e Tolleranze"

Lavora con gli snapshot di Trident

Gli snapshot dei volumi persistenti (PV) di Kubernetes consentono copie point-in-time dei volumi. È possibile creare uno snapshot di un volume creato utilizzando Trident, importare uno snapshot creato al di fuori di Trident, creare un nuovo volume da uno snapshot esistente e recuperare i dati del volume dagli snapshot.

Panoramica

Lo snapshot del volume è supportato dai `ontap-nas`, `ontap-nas-flexgroup`, `ontap-san`, `ontap-san-economy`, `solidfire-san`, `azure-netapp-files` e `google-cloud-netapp-volumes` driver.

Prima di iniziare

Per lavorare con gli snapshot, è necessario disporre di un controller snapshot esterno e di Custom Resource Definitions (CRD). Questa è responsabilità dell'orchestratore Kubernetes (ad esempio: Kubeadm, GKE, OpenShift).

Se la tua distribuzione Kubernetes non include il controller snapshot e i CRD, consulta Distribuire un controller snapshot del volume.

NOTA	Non creare un controller snapshot se si creano snapshot di volumi on-demand in un ambiente GKE. GKE utilizza un controller snapshot integrato e nascosto.

Crea uno Snapshot del volume

Passaggi

1. Crea un `VolumeSnapshotClass`. Per ulteriori informazioni, fare riferimento a "VolumeSnapshotClass".
   - Il `driver` punta al driver Trident CSI.
   - `deletionPolicy` può essere `Delete` o `Retain`. Quando impostato su `Retain`, lo snapshot fisico sottostante sullo storage cluster viene mantenuto anche quando l'oggetto `VolumeSnapshot` viene eliminato.

     Esempio

     ```
     cat snap-sc.yaml
     ```

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

2. Crea una Snapshot di un PVC esistente.

   Esempi

   - Questo esempio crea uno snapshot di un PVC esistente.

```
cat snap.yaml
```

```
apiVersion: snapshot.storage.k8s.io/v1
kind: VolumeSnapshot
metadata:
  name: pvc1-snap
spec:
  volumeSnapshotClassName: csi-snapclass
  source:
    persistentVolumeClaimName: pvc1
```

- In questo esempio viene creato un oggetto snapshot del volume per un PVC denominato `pvc1` e il nome dello snapshot è impostato su `pvc1-snap`. Un VolumeSnapshot è analogo a un PVC ed è associato a un oggetto `VolumeSnapshotContent` che rappresenta lo snapshot effettivo.

```
kubectl create -f snap.yaml
volumesnapshot.snapshot.storage.k8s.io/pvc1-snap created

kubectl get volumesnapshots
NAME                    AGE
pvc1-snap               50s
```

- È possibile identificare l `VolumeSnapshotContent'oggetto per il `pvc1-snap` VolumeSnapshot descrivendolo. L `Snapshot Content Name'oggetto identifica l'oggetto VolumeSnapshotContent che serve questa snapshot. Il parametro `Ready To Use` indica che la snapshot può essere utilizzata per creare un nuovo PVC.

```
kubectl describe volumesnapshots pvc1-snap
Name:         pvc1-snap
Namespace:    default
...
Spec:
  Snapshot Class Name:    pvc1-snap
  Snapshot Content Name:  snapcontent-e8d8a0ca-9826-11e9-9807-
525400f3f660
  Source:
    API Group:
    Kind:       PersistentVolumeClaim
    Name:       pvc1
Status:
  Creation Time:  2019-06-26T15:27:29Z
  Ready To Use:   true
  Restore Size:   3Gi
```

```
...
```

Crea un PVC da una Snapshot del volume

Puoi usare `dataSource` per creare un PVC usando un VolumeSnapshot denominato `<pvc-name>` come origine dei dati. Dopo che il PVC è stato creato, può essere collegato a un pod e usato come qualsiasi altro PVC.

ATTENZIONE	Il PVC verrà creato nello stesso backend del volume sorgente. Fai riferimento a "KB: Creazione di un PVC da un Trident PVC Snapshot non può essere creata in un backend alternativo".

L'esempio seguente crea il PVC utilizzando `pvc1-snap` come origine dati.

```
cat pvc-from-snap.yaml
```

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: pvc-from-snap
spec:
  accessModes:
    - ReadWriteOnce
  storageClassName: golden
  resources:
    requests:
      storage: 3Gi
  dataSource:
    name: pvc1-snap
    kind: VolumeSnapshot
    apiGroup: snapshot.storage.k8s.io
```

Importa una Snapshot del volume

Trident supporta la "Processo di snapshot pre-provisionato Kubernetes" per consentire all'amministratore del cluster di creare un oggetto `VolumeSnapshotContent` e importare Snapshot creati al di fuori di Trident.

Prima di iniziare

Trident deve aver creato o importato il volume d'origine dello Snapshot.

Passaggi

1. **Cluster admin:** Creare un `VolumeSnapshotContent` oggetto che fa riferimento allo snapshot del backend. Questo avvia il flusso di lavoro dello snapshot in Trident.
   - Specificare il nome dello snapshot del backend in `annotations` come `trident.netapp.io/internalSnapshotName: <"backend-snapshot-name">`.

- Specificare `<name-of-parent-volume-in-trident>/<volume-snapshot-content-name>` in `snapshotHandle`. Questa è l'unica informazione fornita a Trident dallo snapshotter esterno nella chiamata `ListSnapshots`.

> **NOTA** Il `<volumeSnapshotContentName>` non può sempre corrispondere al nome dello snapshot del backend a causa dei vincoli di denominazione del CR.

Esempio

Il seguente esempio crea un oggetto `VolumeSnapshotContent` che fa riferimento allo snapshot del backend `snap-01`.

```
apiVersion: snapshot.storage.k8s.io/v1
kind: VolumeSnapshotContent
metadata:
  name: import-snap-content
  annotations:
    trident.netapp.io/internalSnapshotName: "snap-01"  # This is the
name of the snapshot on the backend
spec:
  deletionPolicy: Retain
  driver: csi.trident.netapp.io
  source:
    snapshotHandle: pvc-f71223b5-23b9-4235-bbfe-e269ac7b84b0/import-
snap-content # <import PV name or source PV name>/<volume-snapshot-
content-name>
  volumeSnapshotRef:
    name: import-snap
    namespace: default
```

2. **Cluster admin:** Crea il `VolumeSnapshot` CR che fa riferimento all' `VolumeSnapshotContent` oggetto. Questa richiesta consente l'accesso all'utilizzo di `VolumeSnapshot` in un determinato namespace.

Esempio

Il seguente esempio crea un `VolumeSnapshot` CR chiamato `import-snap` che fa riferimento al `VolumeSnapshotContent` chiamato `import-snap-content`.

```
apiVersion: snapshot.storage.k8s.io/v1
kind: VolumeSnapshot
metadata:
  name: import-snap
spec:
  # volumeSnapshotClassName: csi-snapclass (not required for pre-
provisioned or imported snapshots)
  source:
    volumeSnapshotContentName: import-snap-content
```

3. **Elaborazione interna (nessuna azione richiesta):** Lo snapshotter esterno riconosce il nuovo `VolumeSnapshotContent` e esegue la chiamata `ListSnapshots`. Trident crea il `TridentSnapshot`.
   - L'external snapshotter imposta `VolumeSnapshotContent` su `readyToUse` e `VolumeSnapshot` su `true`.
   - Trident restituisce `readyToUse=true`.
4. **Qualsiasi utente:** Crea un `PersistentVolumeClaim` per fare riferimento al nuovo `VolumeSnapshot`, dove il `spec.dataSource` (o `spec.dataSourceRef`) nome è il `VolumeSnapshot` nome.

   Esempio

   Il seguente esempio crea un PVC che fa riferimento al `VolumeSnapshot` denominato `import-snap`.

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: pvc-from-snap
spec:
  accessModes:
    - ReadWriteOnce
  storageClassName: simple-sc
  resources:
    requests:
      storage: 1Gi
  dataSource:
    name: import-snap
    kind: VolumeSnapshot
    apiGroup: snapshot.storage.k8s.io
```

Recupera i dati del volume utilizzando le Snapshot

La directory delle snapshot è nascosta per impostazione predefinita per facilitare la massima compatibilità dei volumi forniti utilizzando i driver `ontap-nas` e `ontap-nas-economy`. Abilita la directory `.snapshot` per recuperare i dati direttamente dalle snapshot.

Utilizzare il comando ONTAP CLI di ripristino dell'istantanea del volume per ripristinare un volume a uno stato registrato in una precedente Snapshot.

```
cluster1::*> volume snapshot restore -vserver vs0 -volume vol3 -snapshot
vol3_snap_archive
```

NOTA	Quando si ripristina una copia Snapshot, la configurazione del volume esistente viene sovrascritta. Le modifiche apportate ai dati del volume dopo la creazione della copia Snapshot vengono perse.

Ripristino del volume in loco da una Snapshot

Trident offre un rapido ripristino in-place del volume da una snapshot utilizzando il `TridentActionSnapshotRestore` (TASR) CR. Questo CR funziona come un'azione Kubernetes imperativa e non persiste dopo il completamento dell'operazione.

Trident supporta il ripristino delle snapshot sui driver `ontap-san`, `ontap-san-economy`, `ontap-nas`, `ontap-nas-flexgroup`, `azure-netapp-files`, `google-cloud-netapp-volumes` e `solidfire-san`.

Prima di iniziare

È necessario disporre di un PVC vincolato e di una Snapshot del volume disponibile.

- Verificare che lo stato del PVC sia bound.

```
kubectl get pvc
```

- Verificare che la snapshot del volume sia pronta per l'uso.

```
kubectl get vs
```

Passaggi

1. Crea il CR TASR. Questo esempio crea un CR per PVC `pvc1` e volume snapshot `pvc1-snapshot`.

   NOTA Il TASR CR deve trovarsi in uno spazio dei nomi in cui esistono il PVC e il VS.

```
cat tasr-pvc1-snapshot.yaml
```

```
apiVersion: trident.netapp.io/v1
kind: TridentActionSnapshotRestore
metadata:
  name: trident-snap
  namespace: trident
spec:
  pvcName: pvc1
  volumeSnapshotName: pvc1-snapshot
```

2. Applica il CR per ripristinare dallo snapshot. Questo esempio ripristina dallo snapshot `pvc1`.

```
kubectl create -f tasr-pvc1-snapshot.yaml
```

```
tridentactionsnapshotrestore.trident.netapp.io/trident-snap created
```

Risultati

Trident ripristina i dati dallo snapshot. Puoi verificare lo stato di ripristino dello snapshot:

```
kubectl get tasr -o yaml
```

```
apiVersion: trident.netapp.io/v1
items:
- apiVersion: trident.netapp.io/v1
  kind: TridentActionSnapshotRestore
  metadata:
    creationTimestamp: "2023-04-14T00:20:33Z"
    generation: 3
    name: trident-snap
    namespace: trident
    resourceVersion: "3453847"
    uid: <uid>
  spec:
    pvcName: pvc1
    volumeSnapshotName: pvc1-snapshot
  status:
    startTime: "2023-04-14T00:20:34Z"
    completionTime: "2023-04-14T00:20:37Z"
    state: Succeeded
kind: List
metadata:
  resourceVersion: ""
```

NOTA

- Nella maggior parte dei casi, Trident non riproverà automaticamente l'operazione in caso di fallimento. Dovrai eseguire nuovamente l'operazione.
- Gli utenti Kubernetes senza accesso admin potrebbero dover ottenere il permesso dall'admin per creare un TASR CR nel loro namespace dell'applicazione.

Eliminare un PV con le relative Snapshot

Quando si elimina un Persistent Volume con le relative Snapshot, il volume Trident corrispondente viene aggiornato a uno "Deleting state". Rimuovere le Snapshot del volume per eliminare il volume Trident.

Distribuire un controller snapshot del volume

Se la tua distribuzione di Kubernetes non include il controller di snapshot e i CRD, puoi distribuirli come segue.

Passaggi

1. Crea CRD di Snapshot del volume.

```
cat snapshot-setup.sh
```

```
#!/bin/bash
# Create volume snapshot CRDs
kubectl apply -f https://raw.githubusercontent.com/kubernetes-csi/external-snapshotter/release-6.1/client/config/crd/snapshot.storage.k8s.io_volumesnapshotclasses.yaml
kubectl apply -f https://raw.githubusercontent.com/kubernetes-csi/external-snapshotter/release-6.1/client/config/crd/snapshot.storage.k8s.io_volumesnapshotcontents.yaml
kubectl apply -f https://raw.githubusercontent.com/kubernetes-csi/external-snapshotter/release-6.1/client/config/crd/snapshot.storage.k8s.io_volumesnapshots.yaml
```

2. Crea il controller di snapshot.

```
kubectl apply -f https://raw.githubusercontent.com/kubernetes-csi/external-snapshotter/release-6.1/deploy/kubernetes/snapshot-controller/rbac-snapshot-controller.yaml
```

```
kubectl apply -f https://raw.githubusercontent.com/kubernetes-csi/external-snapshotter/release-6.1/deploy/kubernetes/snapshot-controller/setup-snapshot-controller.yaml
```

NOTA Se necessario, apri `deploy/kubernetes/snapshot-controller/rbac-snapshot-controller.yaml` e aggiorna `namespace` al tuo namespace.

Link correlati

- "Istantanee del volume"
- "VolumeSnapshotClass"

Lavora con gli snapshot dei gruppi di volumi di Trident

Snapshot del gruppo di volumi Kubernetes dei volumi persistenti (PV) NetApp Trident offre la possibilità di creare snapshot di più volumi (un gruppo di snapshot di volumi). Questo snapshot del gruppo di volumi rappresenta copie di più volumi acquisite nello stesso point-in-time.

NOTA VolumeGroupSnapshot is a beta feature in Kubernetes con API beta. Kubernetes 1.32 è la

versione minima richiesta per VolumeGroupSnapshot.

Crea snapshot del gruppo di volumi

Lo snapshot del gruppo di volumi è supportato con i seguenti driver di archiviazione:

- `ontap-san` driver - solo per i protocolli iSCSI e FC, non per il protocollo NVMe/TCP.
- `ontap-san-economy` - solo per il protocollo iSCSI.
- `ontap-nas`

NOTA Lo snapshot del gruppo di volumi non è supportato per i sistemi di archiviazione NetApp ASA r2 o AFX.

Prima di iniziare

- Assicurati che la versione di Kubernetes sia K8s 1.32 o superiore.
- Per lavorare con gli snapshot, è necessario disporre di un controller snapshot esterno e di Custom Resource Definitions (CRD). Questa è responsabilità dell'orchestratore Kubernetes (ad esempio: Kubeadm, GKE, OpenShift).

  Se la tua distribuzione Kubernetes non include il controller snapshot esterno e i CRD, consulta Distribuire un controller snapshot del volume.

  NOTA Non creare un controller snapshot se si creano snapshot di gruppi di volumi on-demand in un ambiente GKE. GKE utilizza un controller snapshot integrato e nascosto.

- Nel file YAML del controller snapshot, impostare il `CSIVolumeGroupSnapshot` feature gate su 'true' per garantire che la funzionalità di snapshot del gruppo di volumi sia abilitata.
- Crea le classi di snapshot del gruppo di volumi richieste prima di creare uno snapshot del gruppo di volumi.
- Assicurati che tutti i PVC/volumi siano sullo stesso SVM per poter creare VolumeGroupSnapshot.

Passaggi

- Crea un VolumeGroupSnapshotClass prima di creare un VolumeGroupSnapshot. Per ulteriori informazioni, consulta "VolumeGroupSnapshotClass".

```
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
```

- Crea PVC con le etichette richieste utilizzando le classi di storage esistenti oppure aggiungi queste etichette ai PVC esistenti.

  L'esempio seguente crea il PVC utilizzando `pvc1-group-snap` come origine dati ed etichetta

`consistentGroupSnapshot: groupA`. Definisci la chiave e il valore dell'etichetta in base alle tue esigenze.

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: pvc1-group-snap
  labels:
    consistentGroupSnapshot: groupA
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 100Mi
  storageClassName: sc1-1
```

- Crea un VolumeGroupSnapshot con la stessa etichetta (`consistentGroupSnapshot: groupA` specificata nel PVC.

  Questo esempio crea una Snapshot di un gruppo di volumi:

```
apiVersion: groupsnapshot.storage.k8s.io/v1beta1
kind: VolumeGroupSnapshot
metadata:
  name: "vgs1"
  namespace: trident
spec:
  volumeGroupSnapshotClassName: csi-group-snap-class
  source:
    selector:
      matchLabels:
        consistentGroupSnapshot: groupA
```

Recupera i dati del volume utilizzando una Snapshot di gruppo

È possibile ripristinare i singoli Persistent Volumes utilizzando le singole Snapshot che sono state create come parte del Volume Group Snapshot. Non è possibile recuperare il Volume Group Snapshot come unità.

Utilizzare il comando ONTAP CLI di ripristino dell'istantanea del volume per ripristinare un volume a uno stato registrato in una precedente Snapshot.

```
cluster1::*> volume snapshot restore -vserver vs0 -volume vol3 -snapshot
vol3_snap_archive
```

NOTA	Quando si ripristina una copia Snapshot, la configurazione del volume esistente viene sovrascritta. Le modifiche apportate ai dati del volume dopo la creazione della copia Snapshot vengono perse.

Ripristino del volume in loco da una Snapshot

Trident offre un rapido ripristino in-place del volume da una snapshot utilizzando il `TridentActionSnapshotRestore` (TASR) CR. Questo CR funziona come un'azione Kubernetes imperativa e non persiste dopo il completamento dell'operazione.

Per ulteriori informazioni, consulta "Ripristino del volume in loco da una Snapshot".

Eliminare un PV con le Snapshot di gruppo associate

Quando si elimina una Snapshot di un volume di gruppo:

- È possibile eliminare VolumeGroupSnapshots come un tutto, non le singole snapshot nel gruppo.
- Se i PersistentVolumes vengono eliminati mentre esiste una snapshot per quel PersistentVolume, Trident sposterà quel volume in uno stato di "eliminazione" perché la snapshot deve essere rimossa prima che il volume possa essere rimosso in modo sicuro.
- Se è stato creato un clone utilizzando una snapshot raggruppata e poi il gruppo deve essere eliminato, inizierà un'operazione di split-on-clone e il gruppo non potrà essere eliminato fino al completamento della divisione.

Distribuire un controller snapshot del volume

Se la tua distribuzione di Kubernetes non include il controller di snapshot e i CRD, puoi distribuirli come segue.

Passaggi

1. Crea CRD di Snapshot del volume.

```
cat snapshot-setup.sh
```

```
#!/bin/bash
# Create volume snapshot CRDs
kubectl apply -f https://raw.githubusercontent.com/kubernetes-
csi/external-snapshotter/release-
8.2/client/config/crd/groupsnapshot.storage.k8s.io_volumegroupsnapshotcl
asses.yaml
kubectl apply -f https://raw.githubusercontent.com/kubernetes-
csi/external-snapshotter/release-
8.2/client/config/crd/groupsnapshot.storage.k8s.io_volumegroupsnapshotco
ntents.yaml
kubectl apply -f https://raw.githubusercontent.com/kubernetes-
csi/external-snapshotter/release-
8.2/client/config/crd/groupsnapshot.storage.k8s.io_volumegroupsnapshots.
yaml
```

2. Crea il controller di snapshot.

```
kubectl apply -f https://raw.githubusercontent.com/kubernetes-csi/external-snapshotter/release-8.2/deploy/kubernetes/snapshot-controller/rbac-snapshot-controller.yaml
```

```
kubectl apply -f https://raw.githubusercontent.com/kubernetes-csi/external-snapshotter/release-8.2/deploy/kubernetes/snapshot-controller/setup-snapshot-controller.yaml
```

NOTA Se necessario, apri `deploy/kubernetes/snapshot-controller/rbac-snapshot-controller.yaml` e aggiorna `namespace` al tuo namespace.

Link correlati

- "VolumeGroupSnapshotClass"
- "Istantanee del volume"

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.