



Usa Trident

Trident

NetApp
January 15, 2026

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

Sommario

Usa Trident	1
Preparare il nodo worker	1
Selezionare gli strumenti giusti	1
Rilevamento del servizio nodo	1
Volumi NFS	2
volumi iSCSI	2
Volumi NVMe/TCP	6
SCSI su volumi FC	7
Configurare e gestire i backend	10
Configurare i backend	10
Azure NetApp Files	10
Google Cloud NetApp Volumes	29
Configurare un Cloud Volumes Service per il backend di Google Cloud	46
Configurare un backend NetApp HCI o SolidFire	58
Driver ONTAP SAN	63
Driver NAS ONTAP	92
Amazon FSx for NetApp ONTAP	128
Crea backend con kubectl	163
Gestire i backend	171
Creare e gestire classi di archiviazione	181
Creare una classe di archiviazione	181
Gestire le classi di archiviazione	184
Fornire e gestire i volumi	186
Fornire un volume	186
Espandi i volumi	190
Volumi di importazione	201
Personalizza i nomi e le etichette dei volumi	209
Condividere un volume NFS tra gli spazi dei nomi	212
Clonazione di volumi tra spazi dei nomi	216
Replicare i volumi utilizzando SnapMirror	219
Utilizzare la topologia CSI	225
Lavorare con gli snapshot	233
Lavorare con gli snapshot del gruppo di volumi	241

Usa Trident

Preparare il nodo worker

Tutti i nodi worker nel cluster Kubernetes devono essere in grado di montare i volumi forniti per i 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. Nelle versioni recenti di Red Hat Enterprise Linux CoreOS (RHCOS) gli strumenti sono installati per impostazione predefinita.

Strumenti NFS

"[Installa gli strumenti NFS](#)" se stai utilizzando: `ontap-nas`, `ontap-nas-economy`, `ontap-nas-flexgroup`, `azure-netapp-files`, `gcp-cvs`.

strumenti iSCSI

"[Installare gli strumenti iSCSI](#)" se stai utilizzando: `ontap-san`, `ontap-san-economy`, `solidfire-san`.

Strumenti NVMe

"[Installa gli strumenti NVMe](#)" se stai usando `ontap-san` per il protocollo NVMe (Nonvolatile Memory Express) su TCP (NVMe/TCP).



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

Strumenti SCSI su FC

Fare riferimento a "[Modalità di configurazione degli host SAN FC e FC-NVMe](#)" per ulteriori informazioni sulla configurazione degli host SAN FC e FC-NVMe.

"[Installa gli strumenti FC](#)" se stai usando `ontap-san` con `sanType fc` (SCSI su FC).

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

Rilevamento del servizio nodo

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



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

Rivedi gli eventi

Trident crea eventi affinché il nodo identifichi i servizi rilevati. Per rivedere 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 CR del nodo Trident . Per visualizzare i servizi rilevati, eseguire:

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

Volumi NFS

Installare gli strumenti NFS utilizzando i comandi per il sistema operativo in uso. Assicurarsi che il servizio NFS venga avviato durante l'avvio.

RHEL 8+

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

Ubuntu

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



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

volumi iSCSI

Trident può stabilire automaticamente una sessione iSCSI, analizzare le LUN e 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 anticiparle.
3. **Eseguire le riparazioni** necessarie per riportare lo stato della sessione iSCSI corrente allo stato desiderato.



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

Le funzionalità di auto-riparazione iSCSI 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 inattiva, Trident attende sette minuti prima di disconnettersi per ristabilire la connessione con un portale.



Ad esempio, se i segreti CHAP venissero ruotati sul controller di archiviazione e la rete perdesse la connettività, i vecchi segreti CHAP (*obsoleti*) potrebbero persistere. La funzione di auto-riparazione è in grado di riconoscere questo problema e di 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 individuare 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

Installare gli strumenti iSCSI utilizzando i comandi del sistema operativo in uso.

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 `solidfire-san` driver e Element OS 12.5 o precedente, assicurarsi che l'algoritmo di autenticazione CHAP sia impostato su MD5 in `/etc/iscsi/iscsid.conf`. Con Element 12.7 sono disponibili gli algoritmi CHAP sicuri conformi a FIPS SHA1, SHA-256 e SHA3-256.

```
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 PV iSCSI, specificare `discard mountOption` in `StorageClass` per eseguire il recupero dello spazio in linea. Fare riferimento a ["Documentazione Red Hat"](#).
- Assicurati di aver aggiornato all'ultima versione 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 multipathing:

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



Garantire /etc/multipath.conf contiene find_multipaths no Sotto defaults.

5. Assicurare che iscsid E multipathd stanno correndo:

```
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 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 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
```



Garantire /etc/multipath.conf contiene find_multipaths no Sotto defaults.

5. Assicurare che open-iscsi E multipath-tools sono abilitati e in esecuzione:

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



Per Ubuntu 18.04, è necessario scoprire le porte di destinazione con `iscsiadm` prima di iniziare `open-iscsi` per l'avvio del demone iSCSI. In alternativa, puoi modificare il `iscsi` servizio da avviare `iscsid` automaticamente.

Configurare o disabilitare l'auto-riparazione iSCSI

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

- **Intervallo di auto-riparazione iSCSI:** determina la frequenza con cui viene richiamata l'auto-riparazione iSCSI (predefinito: 5 minuti). È possibile configurarlo in modo che venga eseguito più frequentemente impostando un numero più piccolo o meno frequentemente impostando un numero più grande.



Impostando l'intervallo di auto-riparazione iSCSI su 0, l'auto-riparazione iSCSI viene interrotta completamente. Non consigliamo di disabilitare la funzione di auto-riparazione iSCSI; dovrebbe essere disabilitata solo in determinati scenari, quando la funzione di auto-riparazione iSCSI non funziona come previsto o per scopi di debug.

- **Tempo di attesa per l'auto-riparazione iSCSI:** determina la durata di attesa dell'auto-riparazione iSCSI prima di disconnettersi da una sessione non integra e tentare di accedere nuovamente (impostazione predefinita: 7 minuti). È possibile configurarlo su un numero maggiore in modo che le sessioni identificate come non integre debbano attendere più a lungo prima di essere disconnesse e poi venga effettuato un tentativo di accesso successivo, oppure su un numero inferiore per disconnettersi e accedere prima.

Timone

Per configurare o modificare le impostazioni di auto-riparazione iSCSI, passare il `iscsiSelfHealingInterval` E `iscsiSelfHealingWaitTime` parametri durante l'installazione o l'aggiornamento di Helm.

Nell'esempio seguente l'intervallo di auto-riparazione iSCSI viene impostato su 3 minuti e il tempo di attesa per l'auto-riparazione su 6 minuti:

```
helm install trident trident-operator-100.2506.0.tgz --set
iscsiSelfHealingInterval=3m0s --set iscsiSelfHealingWaitTime=6m0s -n
trident
```

tridentctl

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

Nell'esempio seguente l'intervallo di auto-riparazione iSCSI viene impostato su 3 minuti e il tempo di attesa per l'auto-riparazione su 6 minuti:

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

Volumi NVMe/TCP

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



- NVMe richiede RHEL 9 o versione successiva.
- 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 l'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 in caso di problemi. Non modificare questa impostazione.

SCSI su volumi FC

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

Prerequisiti

Configurare le impostazioni di rete e nodo richieste per FC.

Impostazioni di rete

1. Ottieni il WWPN delle interfacce di destinazione. Fare riferimento a ["mostra interfaccia di rete"](#) per maggiori informazioni.
2. Ottieni il WWPN per le interfacce sull'iniziatore (host).

Fare riferimento alle utilità del sistema operativo host corrispondenti.

3. Configurare la suddivisione in zone sullo switch FC utilizzando i WWPN dell'host e della destinazione.

Per informazioni, fare riferimento alla documentazione del rispettivo fornitore dello switch.

Per maggiori dettagli, fare riferimento alla seguente documentazione ONTAP :

- ["Panoramica sulla zonizzazione Fibre Channel e FCoE"](#)

- ["Modalità di configurazione degli host SAN FC e FC-NVMe"](#)

Installa gli strumenti FC

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

- Quando si utilizzano nodi worker che eseguono RHEL/Red Hat Enterprise Linux CoreOS (RHCOS) con FC PV, specificare `discard mountOption` in StorageClass per eseguire il recupero dello spazio in linea. Fare riferimento a ["Documentazione Red Hat"](#) .

RHEL 8+

1. Installare i seguenti pacchetti di sistema:

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

2. Abilita multipathing:

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



Garantire /etc/multipath.conf contiene find_multipaths no Sotto defaults.

3. Assicurare che multipathd è 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 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
```



Garantire /etc/multipath.conf contiene find_multipaths no Sotto defaults.

3. Assicurare che multipath-tools è abilitato e in esecuzione:

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

Configurare e gestire i backend

Configurare i backend

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

Trident offre automaticamente pool di archiviazione da backend che corrispondono ai requisiti definiti da una classe di archiviazione. Scopri come configurare il backend per il tuo sistema di archiviazione.

- ["Configurare un backend di Azure NetApp Files"](#)
- ["Configurare un backend Google Cloud NetApp Volumes"](#)
- ["Configurare un Cloud Volumes Service per il backend di Google Cloud Platform"](#)
- ["Configurare un backend NetApp HCI o SolidFire"](#)
- ["Configurare un backend con driver ONTAP o Cloud Volumes ONTAP NAS"](#)
- ["Configurare un backend con driver ONTAP o Cloud Volumes ONTAP SAN"](#)
- ["Utilizzare Trident con Amazon FSx for NetApp ONTAP"](#)

Azure NetApp Files

Configurare un backend di Azure NetApp Files

È possibile configurare Azure NetApp Files come backend per Trident. È possibile collegare volumi NFS e SMB utilizzando un backend di Azure NetApp Files. Trident supporta anche la gestione delle credenziali mediante identità gestite per i cluster di Azure Kubernetes Services (AKS).

Dettagli del driver di Azure NetApp Files

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

Autista	Protocollo	Modalità volume	Modalità di accesso supportate	Sistemi di file supportati
azure-netapp-files	NFS SMB	File system	RWO, ROX, RWX, RWOP	nfs, smb

Considerazioni

- Il servizio Azure NetApp Files non supporta volumi inferiori a 50 GiB. Trident crea automaticamente volumi da 50 GiB se viene richiesto un volume più piccolo.
- Trident supporta volumi SMB montati su pod in esecuzione solo su nodi Windows.

Identità gestite per AKS

Supporti Trident **"identità gestite"** per i cluster di Azure Kubernetes Services. Per sfruttare la gestione semplificata delle credenziali offerta dalle identità gestite, è necessario disporre di:

- Un cluster Kubernetes distribuito tramite AKS
- Identità gestite configurate sul cluster AKS Kubernetes
- Trident installato che include il `cloudProvider` specificare "Azure" .

Operatore Trident

Per installare Trident utilizzando l'operatore Trident , modificare `tridentorchestrator_cr.yaml` impostare `cloudProvider` a "Azure" . Per esempio:

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

Timone

L'esempio seguente installa i set Trident `cloudProvider` ad Azure utilizzando la variabile di ambiente `$CP` :

```
helm install trident trident-operator-100.2506.0.tgz --create
--namespace --namespace <trident-namespace> --set cloudProvider=$CP
```

`<code>tridentctl</code>`

L'esempio seguente installa Trident e imposta il `cloudProvider` bandiera a Azure :

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

Identità cloud per AKS

L'identità cloud consente ai pod Kubernetes di accedere alle risorse di Azure autenticandosi come identità del carico di lavoro anziché fornire credenziali Azure esplicite.

Per sfruttare i vantaggi dell'identità cloud in Azure, è necessario disporre di:

- Un cluster Kubernetes distribuito tramite AKS

- Identità del carico di lavoro e oidc-issuer configurati sul cluster AKS Kubernetes
- Trident installato che include il `cloudProvider` specificare "Azure" E `cloudIdentity` specificando l'identità del carico di lavoro

Operatore Trident

Per installare Trident utilizzando l'operatore Trident , modificare `tridentorchestrator_cr.yaml` impostare `cloudProvider` A "Azure" e impostare `cloudIdentity` A `azure.workload.identity/client-id: xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxx` .

Per esempio:

```
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-xxxxxxxxxxxxx' # Edit
```

Timone

Impostare 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-
xxxxxxxxxxxxx'"
```

L'esempio seguente installa Trident e imposta `cloudProvider` ad Azure utilizzando la variabile di ambiente `$CP` e imposta il `cloudIdentity` utilizzando la variabile d'ambiente `$CI` :

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

<code>tridentctl</code>

Impostare 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-
xxxxxxxxxxxxx"
```

L'esempio seguente installa Trident e imposta il `cloud-provider` bandiera a `$CP` , E `cloud-identity` A `$CI` :

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

Prepararsi a configurare un backend di Azure NetApp Files

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

Prerequisiti per i 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. Fare riferimento a ["Azure: configura Azure NetApp Files e crea un volume NFS"](#) .

Per configurare e utilizzare un ["Azure NetApp Files"](#) backend, hai bisogno di quanto segue:



- `subscriptionID`, `tenantID`, `clientID`, `location`, E `clientSecret` sono facoltativi quando si utilizzano identità gestite su un cluster AKS.
- `tenantID`, `clientID`, E `clientSecret` sono facoltativi quando si utilizza un'identità cloud su un cluster AKS.

- Un bacino di capacità. Fare riferimento a ["Microsoft: creare un pool di capacità per Azure NetApp Files"](#) .
- Una subnet delegata ad Azure NetApp Files. Fare riferimento a ["Microsoft: delegare una subnet ad Azure NetApp Files"](#) .
- `subscriptionID` da una sottoscrizione Azure con Azure NetApp Files abilitato.
- `tenantID`, `clientID`, E `clientSecret` da un ["Registrazione dell'app"](#) in Azure Active Directory con autorizzazioni sufficienti per il servizio Azure NetApp Files . La registrazione dell'app deve utilizzare:
 - Il ruolo di Proprietario o Collaboratore ["predefinito da Azure"](#) .
 - UN ["ruolo di collaboratore personalizzato"](#) a livello di abbonamento(`assignableScopes`) con le seguenti autorizzazioni limitate solo a quanto richiesto Trident . Dopo aver creato il ruolo personalizzato, ["assegnare il ruolo utilizzando il portale di Azure"](#) .

```
{
  "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/read",

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

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

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

          "Microsoft.Features/featureProviders/subscriptionFeatureRegistrations/read",

          "Microsoft.Features/featureProviders/subscriptionFeatureRegistrations/write",

          "Microsoft.Features/featureProviders/subscriptionFeatureRegistrations/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'azzurro `location` che contiene almeno uno ["sottorete delegata"](#) . A partire dal Trident 22.01, il `location` Il parametro è un campo obbligatorio al livello superiore del file di configurazione del backend. I valori di posizione specificati nei pool virtuali vengono ignorati.
- Per usare Cloud Identity, ottenere il client ID da un ["identità gestita assegnata dall'utente"](#) e specificare quell'ID in `azure.workload.identity/client-id: xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx` .

Requisiti aggiuntivi per i volumi SMB

Per creare un volume SMB, è necessario disporre di:

- Active Directory configurato e connesso ad Azure NetApp Files. Fare riferimento a ["Microsoft: creare e gestire 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 su pod in esecuzione solo su nodi Windows.
- Almeno un segreto Trident contenente le credenziali di Active Directory, in modo che Azure NetApp Files possa autenticarsi in Active Directory. Per generare segreto `smbcreds` :

```

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

```

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

Opzioni ed esempi di configurazione del backend Azure NetApp Files

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

gli esempi di configurazione.

Opzioni di configurazione del backend

Trident utilizza la configurazione 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.



* A partire dalla versione NetApp Trident 25.06, i pool di capacità QoS manuali sono supportati come anteprima tecnologica.*

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

Parametro	Descrizione	Predefinito
version		Sempre 1
storageDriverName	Nome del driver di archiviazione	"file-azure-netapp"
backendName	Nome personalizzato o backend di archiviazione	Nome del conducente + "_" + caratteri casuali
subscriptionID	ID sottoscrizione della sottoscrizione di Azure. Facoltativo quando le identità gestite sono abilitate su un cluster AKS.	
tenantID	L'ID tenant da una registrazione app facoltativa quando le identità gestite o l'identità cloud vengono utilizzate su un cluster AKS.	
clientID	ID client da una registrazione app Facoltativo quando si utilizzano identità gestite o identità cloud su un cluster AKS.	
clientSecret	Il segreto client di una registrazione app facoltativa quando vengono utilizzate identità gestite o identità cloud 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.	
resourceGroups	Elenco dei gruppi di risorse per filtrare le risorse scoperte	[] (nessun filtro)
netappAccounts	Elenco degli account NetApp per filtrare le risorse rilevate	[] (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 . Le funzionalità di rete non sono disponibili in tutte le regioni e potrebbero dover essere abilitate tramite un abbonamento. Specificando networkFeatures quando la funzionalità non è abilitata, il provisioning del volume non riesce.	""
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 delimitate da virgole per scegliere NFS v4.1. Le opzioni di montaggio impostate in una definizione di classe di archiviazione sostituiscono le opzioni di montaggio impostate nella configurazione del backend.	"nfsvers=3"
limitVolumeSize	Il provisioning non riesce se la dimensione del volume richiesto è superiore a questo valore	"" (non applicato di default)
debugTraceFlags	Flag di debug da utilizzare durante la risoluzione dei problemi. Esempio, \{"api": false, "method": true, "discovery": true\} . Non utilizzare questa funzione a meno che non si desideri risolvere un problema e richiedere un dump dettagliato del registro.	null
nasType	Configurare la creazione di volumi NFS o SMB. Le opzioni sono nfs , smb o nullo. Impostando il valore su null, i volumi NFS vengono impostati di default.	nfs

Parametro	Descrizione	Predefinito
supportedTopologies	Rappresenta un elenco di regioni e zone supportate da questo backend. Per maggiori informazioni, fare riferimento a "Utilizzare la topologia CSI" .	
qosType	Rappresenta il tipo di QoS: automatico o manuale. Anteprima tecnica per Trident 25.06	Auto
maxThroughput	Imposta la velocità massima consentita in MiB/sec. Supportato solo per pool di capacità QoS manuali. Anteprima tecnica per Trident 25.06	4 MiB/sec



Per ulteriori informazioni sulle funzionalità di rete, fare riferimento a ["Configurare le funzionalità di rete per un volume di Azure NetApp Files"](#) .

Autorizzazioni e risorse richieste

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 richieste (subnet, rete virtuale, pool di capacità) associate. Se il debug è abilitato, Trident registrerà 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 utilizzando nomi brevi o completamente qualificati. Nella maggior parte delle situazioni si consiglia di utilizzare nomi completi, poiché i nomi brevi possono corrispondere a più risorse con lo stesso nome.

IL `resourceGroups` , `netappAccounts` , E `capacityPools` I valori sono filtri che limitano l'insieme delle risorse rilevate a quelle disponibili per questo backend di archiviazione e possono essere specificati in qualsiasi combinazione. I nomi completi seguono questo formato:

Tipo	Formato
Gruppo di risorse	<gruppo di risorse>
Conto NetApp	<gruppo di risorse>/<account NetApp>
Capacità di pool	<gruppo di risorse>/<account NetApp>/<pool di capacità>
Rete virtuale	<gruppo di risorse>/<rete virtuale>
Sottorete	<gruppo di risorse>/<rete virtuale>/<sottorete>

Provisioning del volume

È possibile controllare il provisioning predefinito del volume specificando le seguenti opzioni in una sezione speciale del file di configurazione. Fare riferimento a [Configurazioni di esempio](#) per i dettagli.

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 subnet IPv4 in notazione CIDR. Ignorato per i volumi SMB.	"0.0.0.0/0"
snapshotDir	Controlla la visibilità della directory .snapshot	"true" per NFSv4 "false" per NFSv3
size	La dimensione predefinita dei nuovi volumi	"100G"
unixPermissions	I permessi Unix dei nuovi volumi (4 cifre ottali). Ignorato per i volumi SMB.	"" (funzione di anteprima, richiede l'inserimento nella whitelist nell'abbonamento)

Configurazioni di esempio

Gli esempi seguenti mostrano configurazioni di base che lasciano la maggior parte dei parametri 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 gli account NetApp, i pool di capacità e le subnet delegate ad Azure NetApp Files nella posizione configurata e posiziona casualmente i nuovi volumi su uno di questi pool e subnet. Perché `nasType` viene omissso, il `nfs` si applica l'impostazione predefinita e il backend provvederà al provisioning dei volumi NFS.

Questa configurazione è ideale quando si inizia a utilizzare Azure NetApp Files e si provano le cose, ma in pratica si vorrà fornire un ambito aggiuntivo per i volumi di cui si esegue il provisioning.

```
---
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 facoltativi quando si utilizzano 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:
    - ultra-pool
  resourceGroups:
    - aks-ami-eastus-rg
  netappAccounts:
    - smb-na
  virtualNetwork: eastus-prod-vnet
  subnet: eastus-anf-subnet
```

Identità cloud per AKS

Questa configurazione del backend omette `tenantID`, `clientID`, E `clientSecret`, che sono facoltativi 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 pool di capacità

Questa configurazione del backend posiziona i volumi in Azure eastus posizione in un Ultra capacità del pool. Trident rileva automaticamente tutte le subnet delegate ad Azure NetApp Files in quella posizione e posiziona casualmente un nuovo volume su una di esse.

```
---
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 del backend posiziona i volumi in Azure `eastus` posizione con pool di capacità QoS manuali. **Anteprima tecnologica in NetApp Trident 25.06.**

```
---
version: 1
storageDriverName: azure-netapp-files
backendName: anfl
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 l'ambito del posizionamento del volume a una singola subnet e modifica anche alcune impostazioni predefinite di provisioning del volume.

```
---
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: my-virtual-network
subnet: 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 backend definisce più pool di archiviazione in un singolo file. Questa funzionalità è utile quando si hanno più pool di capacità che supportano diversi livelli di servizio e si desidera creare classi di archiviazione in Kubernetes che li rappresentino. 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:
        - ultra-1
        - ultra-2
      networkFeatures: Standard
  - labels:
      performance: silver
      serviceLevel: Premium
      capacityPools:
        - premium-1
  - labels:
      performance: bronze
      serviceLevel: Standard
      capacityPools:
        - standard-1
        - standard-2
```

Configurazione delle topologie supportate

Trident semplifica il provisioning dei volumi per carichi di lavoro in base alle regioni e alle zone di disponibilità. IL `supportedTopologies` Il blocco in questa configurazione 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 archiviazione. Per le classi di archiviazione che contengono un sottoinsieme delle regioni e delle zone fornite in un backend, Trident crea volumi nella regione e nella zona menzionate. Per maggiori informazioni, fare riferimento a "[Utilizzare 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 archiviazione

Il seguente `StorageClass` le definizioni si riferiscono ai pool di archiviazione sopra indicati.

Definizioni di esempio utilizzando `parameter.selector` campo

Utilizzando `parameter.selector` puoi specificare per ciascuno `StorageClass` il pool virtuale 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

```

Definizioni di esempio 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 sullo spazio dei nomi 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 pool che supportano volumi SMB. `nasType: nfs O
nasType: null filtri per 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 la creazione del backend fallisce, c'è qualcosa che non va nella configurazione del backend. È possibile visualizzare i registri per determinare la causa eseguendo il seguente comando:

```
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 un backend Google Cloud NetApp Volumes

Ora puoi configurare Google Cloud NetApp Volumes come backend per Trident. È possibile collegare volumi NFS e SMB utilizzando un backend Google Cloud NetApp Volumes .

Dettagli del driver Google Cloud NetApp Volumes

Trident fornisce il `google-cloud-netapp-volumes` driver per comunicare con il cluster. Le modalità di accesso supportate sono: *ReadWriteOnce* (RWO), *ReadOnlyMany* (ROX), *ReadWriteMany* (RWX), *ReadWriteOncePod* (RWOP).

Autista	Protocollo	Modalità volume	Modalità di accesso supportate	Sistemi di file supportati
google-cloud-netapp-volumes	NFS SMB	File system	RWO, ROX, RWX, RWOP	nfs, smb

Identità cloud per GKE

Cloud Identity consente ai pod Kubernetes di accedere alle risorse di Google Cloud autenticandosi come identità del carico di lavoro anziché fornire credenziali Google Cloud esplicite.

Per sfruttare i vantaggi dell'identità cloud in Google Cloud, è necessario disporre di:

- Un cluster Kubernetes distribuito tramite GKE.
- Identità del carico di lavoro configurata sul cluster GKE e GKE MetaData Server configurato sui pool di nodi.
- Un account di servizio GCP con il ruolo Google Cloud NetApp Volumes Admin (`roles/netapp.admin`) o un

ruolo personalizzato.

- Trident installato che include cloudProvider per specificare "GCP" e cloudIdentity che specifica il nuovo account di servizio GCP. Di seguito è riportato un esempio.

Operatore Trident

Per installare Trident utilizzando l'operatore Trident , modificare `tridentorchestrator_cr.yaml` impostare `cloudProvider` A "GCP" e impostare `cloudIdentity` A `iam.gke.io/gcp-service-account: cloudvolumes-admin-sa@mygcpproject.iam.gserviceaccount.com` .

Per esempio:

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

Timone

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

```
export CP="GCP"
export ANNOTATION="'iam.gke.io/gcp-service-account: cloudvolumes-admin-
sa@mygcpproject.iam.gserviceaccount.com'"
```

L'esempio seguente installa Trident e imposta `cloudProvider` a GCP utilizzando la variabile di ambiente `$CP` e imposta il `cloudIdentity` utilizzando la variabile d'ambiente `$ANNOTATION` :

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

<code>tridentctl</code>

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

```
export CP="GCP"
export ANNOTATION="'iam.gke.io/gcp-service-account: cloudvolumes-admin-
sa@mygcpproject.iam.gserviceaccount.com'"
```

L'esempio seguente installa Trident e imposta il `cloud-provider` bandiera a `$CP` , E `cloud-identity` A `$ANNOTATION` :

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

Preparati a configurare un backend Google Cloud NetApp Volumes

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

Prerequisiti per i volumi NFS

Se si utilizza 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. Fare riferimento a ["Prima di iniziare"](#) .

Prima di configurare il backend di Google Cloud NetApp Volumes, assicurati di disporre di quanto segue:

- Un account Google Cloud configurato con il servizio Google Cloud NetApp Volumes . Fare riferimento a ["Google Cloud NetApp Volumes"](#) .
- Numero di progetto del tuo account Google Cloud. Fare riferimento a ["Identificazione dei progetti"](#) .
- Un account di servizio Google Cloud con NetApp Volumes Admin(roles/netapp.admin) ruolo. Fare riferimento a ["Ruoli e autorizzazioni di gestione dell'identità e dell'accesso"](#) .
- File chiave API per il tuo account GCNV. Fare riferimento a ["Crea una chiave dell'account di servizio"](#)
- Un pool di stoccaggio. Fare riferimento a ["Panoramica dei pool di archiviazione"](#) .

Per ulteriori informazioni su come configurare l'accesso a Google Cloud NetApp Volumes, fare riferimento a ["Configurare l'accesso a Google Cloud NetApp Volumes"](#) .

Opzioni ed esempi di configurazione del backend Google Cloud NetApp Volumes

Scopri le opzioni di configurazione backend per Google Cloud NetApp Volumes e rivedi gli esempi di configurazione.

Opzioni di configurazione del backend

Ogni backend esegue il provisioning dei volumi in una singola regione di Google Cloud. Per creare volumi in altre regioni, è possibile 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 del backend di archiviazione	Nome del driver + "_" + parte della chiave API

Parametro	Descrizione	Predefinito
storagePools	Parametro facoltativo utilizzato per specificare i pool di archiviazione per la creazione del volume.	
projectNumber	Numero di progetto dell'account Google Cloud. Il valore si trova nella home page del portale Google Cloud.	
location	Posizione di Google Cloud in cui Trident crea i volumi GCNV. Quando si creano cluster Kubernetes interregionali, i volumi creati in un location può essere utilizzato nei carichi di lavoro pianificati sui nodi in più regioni di Google Cloud. Il traffico interregionale comporta un costo aggiuntivo.	
apiKey	Chiave API per l'account del servizio Google Cloud con netapp.admin ruolo. Include il contenuto in formato JSON del file della chiave privata di un account di servizio Google Cloud (copiato letteralmente 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 non riesce se la dimensione del volume richiesto è superiore a questo valore.	"" (non applicato di default)
serviceLevel	Il livello di servizio di un pool di archiviazione e dei suoi volumi. I valori sono flex , standard , premium , O extreme .	
labels	Insieme di etichette arbitrarie formattate in JSON da applicare ai volumi	""
network	Rete Google Cloud utilizzata per i volumi GCNV.	
debugTraceFlags	Flag di debug da utilizzare durante la risoluzione dei problemi. Esempio, {"api":false, "method":true} . Non utilizzare questa funzione a meno che non si desideri risolvere un problema e richiedere un dump dettagliato del registro.	null
nasType	Configurare la creazione di volumi NFS o SMB. Le opzioni sono nfs , smb o nullo. Impostando il valore su null, i volumi NFS vengono impostati di default.	nfs

Parametro	Descrizione	Predefinito
supportedTopologies	Rappresenta un elenco di regioni e zone supportate da questo backend. Per maggiori informazioni, fare riferimento a "Utilizzare la topologia CSI" . Per 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 del volume predefinito in `defaults` sezione 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"
snapshotDir	Accesso al <code>.snapshot</code> elenco	"true" per NFSv4 "false" per NFSv3
snapshotReserve	Percentuale di volume riservata agli snapshot	"" (accetta il valore predefinito 0)
unixPermissions	I permessi Unix dei nuovi volumi (4 cifre ottali).	""

Configurazioni di esempio

Gli esempi seguenti mostrano configurazioni di base che lasciano la maggior parte dei parametri 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 pool di archiviazione delegati a Google Cloud NetApp Volumes nella posizione configurata e posiziona casualmente i nuovi volumi su uno di questi pool. Perché `nasType` viene omissso, il `nfs` si applica l'impostazione predefinita e il backend provvederà al provisioning dei volumi NFS.

Questa configurazione è ideale quando si inizia a utilizzare Google Cloud NetApp Volumes e si provano le funzionalità, ma in pratica sarà molto probabilmente necessario fornire un ambito aggiuntivo per i volumi di cui si esegue il provisioning.

```

---
apiVersion: v1
kind: Secret
metadata:
  name: backend-tbc-gcnv-secret
type: Opaque
stringData:
  private_key_id: f2cb6ed6d7cc10c453f7d3406fc700c5df0ab9ec
  private_key: |
    -----BEGIN PRIVATE KEY-----\n
    znHczZsrtrHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m\n
    znHczZsrtrHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m\n
    znHczZsrtrHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m\n
    XsYg6gyxy4zq7OlwWgLwGa==\n
    -----END PRIVATE KEY-----\n

---
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 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
```



```

---
apiVersion: v1
kind: Secret
metadata:
  name: backend-tbc-gcnv-secret
type: Opaque
stringData:
  private_key_id: f2cb6ed6d7cc10c453f7d3406fc700c5df0ab9ec
  private_key: |
    -----BEGIN PRIVATE KEY-----
    znHczZsrtrHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrtrHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrtrHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m
    XsYg6gyxy4zq7OlwWgLwGa==
    -----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 backend definisce più pool virtuali in un singolo file. I pool virtuali sono definiti nel `storage` sezione. Sono utili quando si hanno più pool di archiviazione che supportano diversi livelli di servizio e si desidera creare classi di archiviazione in Kubernetes che li rappresentino. Per differenziare i pool vengono utilizzate etichette virtuali. Ad esempio, nell'esempio seguente `performance` etichetta e `serviceLevel` Il tipo viene utilizzato per differenziare i pool virtuali.

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

Per maggiori informazioni, fare riferimento a ["Pool virtuali"](#) .

```
---
apiVersion: v1
kind: Secret
metadata:
  name: backend-tbc-gcnv-secret
type: Opaque
stringData:
  private_key_id: f2cb6ed6d7cc10c453f7d3406fc700c5df0ab9ec
  private_key: |
    -----BEGIN PRIVATE KEY-----
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3b1/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3b1/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3b1/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3b1/qp8B4Kws8zX5ojY9m
    XsYg6gyxy4zq7OlwWgLwGa==
    -----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

```

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

```

Configurazione delle topologie supportate

Trident semplifica il provisioning dei volumi per carichi di lavoro in base alle regioni e alle zone di disponibilità. Il `supportedTopologies` Il blocco in questa configurazione 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 archiviazione. Per le classi di archiviazione che contengono un sottoinsieme delle regioni e delle zone fornite in un backend, Trident crea volumi nella regione e nella zona menzionate. Per maggiori informazioni, fare riferimento a ["Utilizzare 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
```

Cosa succederà 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
backend-tbc-gcnv	backend-tbc-gcnv	b2fd1ff9-b234-477e-88fd-713913294f65
Bound	Success	

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

```
tridentctl logs
```

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

Definizioni delle classi di archiviazione

Quello che segue è un esempio di base `StorageClass` definizione che si riferisce al backend di cui sopra.

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

Definizioni di esempio utilizzando il `parameter.selector` campo:

Utilizzando `parameter.selector` puoi specificare per ciascuno `StorageClass` IL "piscina 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 maggiori dettagli sulle classi di archiviazione, fare riferimento a ["Creare una classe di archiviazione"](#) .

Definizioni di esempio 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 della fase del nodo è possibile utilizzare qualsiasi utente/password di Active Directory con qualsiasi/nessuna autorizzazione.

Configurazione di base sullo spazio dei nomi 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 pool che supportano volumi SMB. `nasType: nfs O
nasType: null filtri per pool NFS.

Esempio di definizione di 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	

Configurare un Cloud Volumes Service per il backend di Google Cloud

Scopri come configurare NetApp Cloud Volumes Service per Google Cloud come backend per la tua installazione Trident utilizzando le configurazioni di esempio fornite.

Dettagli del driver Google Cloud

Trident fornisce il `gcp-cvs` driver per comunicare con il cluster. Le modalità di accesso supportate sono: *ReadWriteOnce* (RWO), *ReadOnlyMany* (ROX), *ReadWriteMany* (RWX), *ReadWriteOncePod* (RWOP).

Autista	Protocollo	Modalità volume	Modalità di accesso supportate	Sistemi di file supportati
gcp-cvs	NFS	File system	RWO, ROX, RWX, RWOP	nfs

Scopri di più sul supporto Trident per Cloud Volumes Service per Google Cloud

Trident può creare volumi Cloud Volumes Service in uno dei due [tipi di servizio](#) :

- **CVS-Performance:** il tipo di servizio Trident predefinito. Questo tipo di servizio ottimizzato per le prestazioni è particolarmente adatto ai carichi di lavoro di produzione che danno valore alle prestazioni. Il tipo di servizio CVS-Performance è un'opzione hardware che supporta volumi con una dimensione minima di 100 GiB. Puoi scegliere uno dei ["tre livelli di servizio"](#) :

- standard
- premium
- extreme

- **CVS:** Il tipo di servizio CVS fornisce un'elevata disponibilità zonale con livelli di prestazioni da limitati a moderati. Il tipo di servizio CVS è un'opzione software che utilizza pool di archiviazione per supportare volumi piccoli fino a 1 GiB. Il pool di archiviazione può contenere fino a 50 volumi, tutti i quali condividono la capacità e le prestazioni del pool. Puoi scegliere uno dei ["due livelli di servizio"](#) :

- standardsw
- zoneredundantstandardsw

Cosa ti servirà

Per configurare e utilizzare il ["Cloud Volumes Service per Google Cloud"](#) backend, hai bisogno di quanto segue:

- Un account Google Cloud configurato con il servizio NetApp Cloud Volumes Service
- Numero di progetto del tuo account Google Cloud
- Account di servizio Google Cloud con `netappcloudvolumes.admin` ruolo
- File chiave API per il tuo account Cloud Volumes Service

Opzioni di configurazione del backend

Ogni backend esegue il provisioning dei volumi in una singola regione di Google Cloud. Per creare volumi in altre regioni, è possibile definire backend aggiuntivi.

Parametro	Descrizione	Predefinito
<code>version</code>		Sempre 1
<code>storageDriverName</code>	Nome del driver di archiviazione	"gcp-cvs"
<code>backendName</code>	Nome personalizzato o backend di archiviazione	Nome del driver + "_" + parte della chiave API
<code>storageClass</code>	Parametro facoltativo utilizzato per specificare il tipo di servizio CVS. Utilizzo <code>software</code> per selezionare il tipo di servizio CVS. In caso contrario, Trident presuppone il tipo di servizio CVS-Performance(<code>hardware</code>).	
<code>storagePools</code>	Solo tipo di servizio CVS. Parametro facoltativo utilizzato per specificare i pool di archiviazione per la creazione del volume.	
<code>projectNumber</code>	Numero di progetto dell'account Google Cloud. Il valore si trova nella home page del portale Google Cloud.	

Parametro	Descrizione	Predefinito
hostProjectNumber	Obbligatorio se si utilizza una rete VPC condivisa. In questo scenario, <code>projectNumber</code> è il progetto di servizio, e <code>hostProjectNumber</code> è il progetto ospitante.	
apiRegion	La regione di Google Cloud in cui Trident crea i volumi Cloud Volumes Service . Quando si creano cluster Kubernetes interregionali, i volumi creati in un <code>apiRegion</code> può essere utilizzato nei carichi di lavoro pianificati sui nodi in più regioni di Google Cloud. Il traffico interregionale comporta un costo aggiuntivo.	
apiKey	Chiave API per l'account del servizio Google Cloud con <code>netappcloudvolumes.admin</code> ruolo. Include il contenuto in formato JSON del file della chiave privata di un account di servizio Google Cloud (copiato letteralmente nel file di configurazione del backend).	
proxyURL	URL proxy se è necessario un server proxy per connettersi all'account CVS. Il server proxy può essere un proxy HTTP o un proxy HTTPS. Per un proxy HTTPS, la convalida del certificato viene ignorata per consentire l'utilizzo di certificati autofirmati nel server proxy. I server proxy con autenticazione abilitata non sono supportati.	
nfsMountOptions	Controllo dettagliato delle opzioni di montaggio NFS.	"nfsvers=3"
limitVolumeSize	Il provisioning non riesce se la dimensione del volume richiesto è superiore a questo valore.	"" (non applicato di default)
serviceLevel	Il livello di servizio CVS-Performance o CVS per i nuovi volumi. I valori CVS-Performance sono <code>standard</code> , <code>premium</code> , O <code>extreme</code> . I valori CVS sono <code>standardsw</code> O <code>zoneredundantstandardsw</code> .	L'impostazione predefinita di CVS-Performance è "standard". L'impostazione predefinita di CVS è "standardsw".
network	Rete Google Cloud utilizzata per i volumi Cloud Volumes Service .	"predefinito"
debugTraceFlags	Flag di debug da utilizzare durante la risoluzione dei problemi. Esempio, <code>\{"api":false,"method":true\}</code> . Non utilizzare questa funzione a meno che non si desideri risolvere un problema e richiedere un dump dettagliato del registro.	null
allowedTopologies	Per abilitare l'accesso tra regioni, la definizione <code>StorageClass</code> per <code>allowedTopologies</code> deve includere tutte le regioni. Per esempio: <ul style="list-style-type: none"> - <code>key: topology.kubernetes.io/region</code> <code>values:</code> - <code>us-east1</code> - <code>europa-west1</code> 	

Opzioni di provisioning del volume

È possibile controllare il provisioning del volume predefinito in `defaults` sezione del file di configurazione.

Parametro	Descrizione	Predefinito
<code>exportRule</code>	Le regole di esportazione per i nuovi volumi. Deve essere un elenco separato da virgole di qualsiasi combinazione di indirizzi IPv4 o subnet IPv4 in notazione CIDR.	"0.0.0.0/0"
<code>snapshotDir</code>	Accesso al <code>.snapshot</code> elenco	"falso"
<code>snapshotReserve</code>	Percentuale di volume riservata agli snapshot	"" (accetta il valore predefinito CVS pari a 0)
<code>size</code>	Le dimensioni dei nuovi volumi. Il requisito minimo per le prestazioni CVS è 100 GiB. Il minimo CVS è 1 GiB.	Il tipo di servizio CVS-Performance è impostato per impostazione predefinita su "100GiB". Il tipo di servizio CVS non imposta un valore predefinito ma richiede almeno 1 GiB.

Esempi di tipi di servizio CVS-Performance

Gli esempi seguenti forniscono configurazioni di esempio per il tipo di servizio CVS-Performance.

Esempio 1: Configurazione minima

Questa è la configurazione minima del backend che utilizza il tipo di servizio CVS-Performance predefinito con il livello di servizio "standard" predefinito.

```
---
version: 1
storageDriverName: gcp-cvs
projectNumber: "012345678901"
apiRegion: us-west2
apiKey:
  type: service_account
  project_id: my-gcp-project
  private_key_id: <id_value>
  private_key: |
    -----BEGIN PRIVATE KEY-----
    <key_value>
    -----END PRIVATE KEY-----
  client_email: cloudvolumes-admin-sa@my-gcp-
project.iam.gserviceaccount.com
  client_id: "123456789012345678901"
  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/cloudvolumes-admin-
sa%40my-gcp-project.iam.gserviceaccount.com
```

Esempio 2: Configurazione del livello di servizio

Questo esempio illustra le opzioni di configurazione del backend, tra cui il livello di servizio e i valori predefiniti del volume.

```
---
version: 1
storageDriverName: gcp-cvs
projectNumber: '012345678901'
apiRegion: us-west2
apiKey:
  type: service_account
  project_id: my-gcp-project
  private_key_id: "<id_value>"
  private_key: |
    -----BEGIN PRIVATE KEY-----
    <key_value>
    -----END PRIVATE KEY-----
  client_email: cloudvolumes-admin-sa@my-gcp-
project.iam.gserviceaccount.com
  client_id: '123456789012345678901'
  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/cloudvolumes-admin-
sa%40my-gcp-project.iam.gserviceaccount.com
proxyURL: http://proxy-server-hostname/
nfsMountOptions: vers=3,proto=tcp,timeo=600
limitVolumeSize: 10Ti
serviceLevel: premium
defaults:
  snapshotDir: 'true'
  snapshotReserve: '5'
  exportRule: 10.0.0.0/24,10.0.1.0/24,10.0.2.100
  size: 5Ti
```

Esempio 3: Configurazione del pool virtuale

Questo campione utilizza `storage` per configurare pool virtuali e `StorageClasses` che rimandano ad essi. Fare riferimento a [Definizioni delle classi di archiviazione](#) per vedere come sono state definite le classi di archiviazione.

Qui vengono impostati valori predefiniti specifici per tutti i pool virtuali, che impostano il `snapshotReserve` al 5% e il `exportRule` a 0.0.0.0/0. I pool virtuali sono definiti nel `storage` sezione. Ogni singolo pool virtuale definisce il proprio `serviceLevel` e alcuni pool sovrascrivono i valori predefiniti. Le etichette dei pool virtuali sono state utilizzate per differenziare i pool in base a `performance` e `protection`.

```
---
version: 1
storageDriverName: gcp-cvs
projectNumber: '012345678901'
apiRegion: us-west2
apiKey:
  type: service_account
  project_id: my-gcp-project
  private_key_id: "<id_value>"
  private_key: |
    -----BEGIN PRIVATE KEY-----
    <key_value>
    -----END PRIVATE KEY-----
  client_email: cloudvolumes-admin-sa@my-gcp-
project.iam.gserviceaccount.com
  client_id: '123456789012345678901'
  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/cloudvolumes-admin-
sa%40my-gcp-project.iam.gserviceaccount.com
nfsMountOptions: vers=3,proto=tcp,timeo=600
defaults:
  snapshotReserve: '5'
  exportRule: 0.0.0.0/0
labels:
  cloud: gcp
region: us-west2
storage:
- labels:
  performance: extreme
  protection: extra
  serviceLevel: extreme
```

```

defaults:
  snapshotDir: 'true'
  snapshotReserve: '10'
  exportRule: 10.0.0.0/24
- labels:
  performance: extreme
  protection: standard
  serviceLevel: extreme
- labels:
  performance: premium
  protection: extra
  serviceLevel: premium
defaults:
  snapshotDir: 'true'
  snapshotReserve: '10'
- labels:
  performance: premium
  protection: standard
  serviceLevel: premium
- labels:
  performance: standard
  serviceLevel: standard

```

Definizioni delle classi di archiviazione

Le seguenti definizioni StorageClass si applicano all'esempio di configurazione del pool virtuale. Utilizzando `parameters.selector`, è possibile specificare per ogni StorageClass il pool virtuale utilizzato per ospitare un volume. Il volume avrà gli aspetti definiti nel pool scelto.

Esempio di classe di archiviazione

```
---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: cvs-extreme-extra-protection
provisioner: csi.trident.netapp.io
parameters:
  selector: performance=extreme; protection=extra
allowVolumeExpansion: true
---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: cvs-extreme-standard-protection
provisioner: csi.trident.netapp.io
parameters:
  selector: performance=premium; protection=standard
allowVolumeExpansion: true
---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: cvs-premium-extra-protection
provisioner: csi.trident.netapp.io
parameters:
  selector: performance=premium; protection=extra
allowVolumeExpansion: true
---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: cvs-premium
provisioner: csi.trident.netapp.io
parameters:
  selector: performance=premium; protection=standard
allowVolumeExpansion: true
---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: cvs-standard
provisioner: csi.trident.netapp.io
parameters:
  selector: performance=standard
```

```

allowVolumeExpansion: true
---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: cvs-extra-protection
provisioner: csi.trident.netapp.io
parameters:
  selector: protection=extra
allowVolumeExpansion: true

```

- Il primo StorageClass(`cvs-extreme-extra-protection`) corrisponde al primo pool virtuale. Questa è l'unica piscina che offre prestazioni estreme con una riserva istantanea del 10%.
- L'ultima StorageClass(`cvs-extra-protection`) richiama qualsiasi pool di archiviazione che fornisca una riserva di snapshot del 10%. Trident decide quale pool virtuale selezionare e garantisce che venga soddisfatto il requisito di riserva degli snapshot.

Esempi di tipi di servizio CVS

Gli esempi seguenti forniscono configurazioni di esempio per il tipo di servizio CVS.

Esempio 1: Configurazione minima

Questa è la configurazione minima del backend utilizzando `storageClass` per specificare il tipo di servizio CVS e quello predefinito `standardsw` livello di servizio.

```
---
version: 1
storageDriverName: gcp-cvs
projectNumber: '012345678901'
storageClass: software
apiRegion: us-east4
apiKey:
  type: service_account
  project_id: my-gcp-project
  private_key_id: "<id_value>"
  private_key: |
    -----BEGIN PRIVATE KEY-----
    <key_value>
    -----END PRIVATE KEY-----
  client_email: cloudvolumes-admin-sa@my-gcp-
project.iam.gserviceaccount.com
  client_id: '123456789012345678901'
  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/cloudvolumes-admin-
sa%40my-gcp-project.iam.gserviceaccount.com
serviceLevel: standardsw
```

Esempio 2: Configurazione del pool di archiviazione

Questa configurazione di backend di esempio utilizza `storagePools` per configurare un pool di archiviazione.

```
---
version: 1
storageDriverName: gcp-cvs
backendName: gcp-std-so-with-pool
projectNumber: '531265380079'
apiRegion: europe-west1
apiKey:
  type: service_account
  project_id: cloud-native-data
  private_key_id: "<id_value>"
  private_key: |-
    -----BEGIN PRIVATE KEY-----
    <key_value>
    -----END PRIVATE KEY-----
  client_email: cloudvolumes-admin-sa@cloud-native-
data.iam.gserviceaccount.com
  client_id: '107071413297115343396'
  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/cloudvolumes-admin-
sa%40cloud-native-data.iam.gserviceaccount.com
storageClass: software
zone: europe-west1-b
network: default
storagePools:
- 1bc7f380-3314-6005-45e9-c7dc8c2d7509
serviceLevel: Standardsw
```

Cosa succederà ora?

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

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

Se la creazione del backend fallisce, c'è qualcosa che non va nella configurazione del backend. È possibile visualizzare i registri per determinare la causa eseguendo il seguente comando:

```
tridentctl logs
```

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

Configurare un backend NetApp HCI o SolidFire

Scopri come creare e utilizzare un backend Element con la tua installazione Trident .

Dettagli del driver dell'elemento

Trident fornisce il `solidfire-san` driver di archiviazione per comunicare con il cluster. Le modalità di accesso supportate sono: *ReadWriteOnce* (RWO), *ReadOnlyMany* (ROX), *ReadWriteMany* (RWX), *ReadWriteOncePod* (RWOP).

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

Autista	Protocollo	Modalità Volume	Modalità di accesso supportate	Sistemi di file supportati
solidfire-san	iSCSI	Bloccare	RWO, ROX, RWX, RWOP	Nessun file system. Dispositivo a blocchi grezzi.
solidfire-san	iSCSI	File system	RWO, RWOP	xfs, ext3 , ext4

Prima di iniziare

Prima di creare un backend Element, avrai bisogno di quanto segue.

- Un sistema di archiviazione supportato che esegue il software Element.
- Credenziali per un amministratore del cluster NetApp HCI/ SolidFire o un utente tenant in grado di gestire i volumi.
- Tutti i nodi worker di Kubernetes devono avere installati gli strumenti iSCSI appropriati. Fare riferimento a ["informazioni sulla preparazione del nodo worker"](#) .

Opzioni di configurazione del backend

Per le opzioni di configurazione del backend, consultare la seguente tabella:

Parametro	Descrizione	Predefinito
version		Sempre 1
storageDriverName	Nome del driver di archiviazione	Sempre "solidfire-san"
backendName	Nome personalizzato o backend di archiviazione	"solidfire_" + indirizzo IP di archiviazione (iSCSI)

Parametro	Descrizione	Predefinito
Endpoint	MVIP per il cluster SolidFire con credenziali tenant	
SVIP	Indirizzo IP e porta di archiviazione (iSCSI)	
labels	Insieme 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 un'interfaccia host specifica	"predefinito"
UseCHAP	Utilizzare CHAP per autenticare iSCSI. Trident utilizza CHAP.	VERO
AccessGroups	Elenco degli ID dei gruppi di accesso da utilizzare	Trova l'ID di un gruppo di accesso denominato "trident"
Types	Specifiche QoS	
limitVolumeSize	Il provisioning non riesce se la dimensione del volume richiesto è superiore a questo valore	"" (non applicato di default)
debugTraceFlags	Flag di debug da utilizzare durante la risoluzione dei problemi. Esempio, {"api":false, "method":true}	null



Non usare `debugTraceFlags` a meno che non si stia risolvendo un problema e si necessiti di un dump di registro dettagliato.

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

Questo esempio mostra un file backend che utilizza l'autenticazione CHAP e modella tre tipi di volume con garanzie QoS specifiche. Molto probabilmente definiresti quindi classi di archiviazione per consumare ciascuna di queste utilizzando `IOPS` parametro della classe di archiviazione.

```

---
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 della classe di backend e di archiviazione per solidfire-san autista con pool virtuali

Questo esempio mostra il file di definizione del backend configurato con pool virtuali insieme alle StorageClass che vi fanno riferimento.

Trident copia le etichette presenti su un pool di archiviazione nella LUN di archiviazione back-end durante il provisioning. Per comodità, gli amministratori di storage possono definire etichette per pool virtuale e raggruppare i volumi in base all'etichetta.

Nel file di definizione backend di esempio mostrato di seguito, vengono impostati valori predefiniti specifici per tutti i pool di archiviazione, che impostano `type` presso Silver. I pool virtuali sono definiti nel `storage` sezione. In questo esempio, alcuni pool di archiviazione impostano il proprio tipo e altri 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 StorageClass fanno riferimento ai pool virtuali sopra indicati. Utilizzando il `parameters.selector` campo, ogni StorageClass richiama quale pool virtuale può essere utilizzato per

ospitare un volume. Il volume avrà gli aspetti definiti nel pool virtuale scelto.

Il primo StorageClass(solidfire-gold-four) verrà mappato sul primo pool virtuale. Questa è l'unica piscina che offre prestazioni d'oro con un Volume Type QoS d'oro. L'ultima StorageClass(solidfire-silver) richiama qualsiasi pool di archiviazione che offra prestazioni Silver. Trident deciderà quale pool virtuale selezionare e garantirà che i requisiti di archiviazione siano soddisfatti.

```
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 maggiori informazioni

- ["Gruppi di accesso al volume"](#)

Driver ONTAP SAN

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 archiviazione SAN per comunicare con il cluster ONTAP . Le modalità di accesso supportate sono: *ReadWriteOnce* (RWO), *ReadOnlyMany* (ROX), *ReadWriteMany* (RWX), *ReadWriteOncePod* (RWOP).

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

Autista	Protocollo	Modalità volume	Modalità di accesso supportate	Sistemi di file supportati
ontap-san	NVMe/TCP Fare riferimento a Considerazioni aggiuntive per NVMe/TCP .	File system	RWO, RWOP ROX e RWX non sono disponibili nella modalità volume Filesystem.	xfs, ext3 , ext4
ontap-san-economy	iSCSI	Bloccare	RWO, ROX, RWX, RWOP	Nessun file system; dispositivo a blocchi raw
ontap-san-economy	iSCSI	File system	RWO, RWOP ROX e RWX non sono disponibili nella modalità volume Filesystem.	xfs, ext3 , ext4



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

Permessi utente

Trident prevede di essere eseguito come amministratore ONTAP o SVM, in genere utilizzando `admin` utente del cluster o un `vsadmin` Utente SVM o 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 il cluster `fsxadmin` utente o un `vsadmin` Utente SVM o un utente con un nome diverso che ha lo stesso ruolo. IL `fsxadmin` L'utente è un sostituto limitato dell'utente amministratore del cluster.



Se usi il `limitAggregateUsage` parametro, sono richieste le autorizzazioni di amministratore del cluster. Quando si utilizza Amazon FSx for NetApp ONTAP con Trident, il `limitAggregateUsage` il parametro non funzionerà con il `vsadmin` E `fsxadmin` account utente. Se si specifica questo parametro, l'operazione di configurazione non riuscirà.

Sebbene sia possibile creare un ruolo più restrittivo all'interno di ONTAP utilizzabile da un conducente Trident , non lo consigliamo. La maggior parte delle nuove versioni di Trident richiameranno API aggiuntive di cui

bisognerebbe tenere conto, rendendo gli aggiornamenti difficili e soggetti a errori.

Considerazioni aggiuntive per NVMe/TCP

Trident supporta il protocollo NVMe (Non-Volatile Memory Express) utilizzando `ontap-san` autista 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 supportato nativamente da NVMe
- Multipathing del mappatore di dispositivi (DM)
- Crittografia LUKS



NVMe è supportato solo con le API REST ONTAP e non 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.



"Sistemi ASA r2" differiscono dagli altri sistemi ONTAP (ASA, AFF e FAS) nell'implementazione del loro livello di archiviazione. Nei sistemi ASA r2, al posto degli aggregati vengono utilizzate zone di disponibilità dello storage. Fare riferimento a ["Questo"](#) Articolo della Knowledge Base su come assegnare 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, potresti configurare un `san-dev` classe che utilizza il `ontap-san` autista e un `san-default` classe che utilizza il `ontap-san-economy` uno.

Tutti i nodi worker di Kubernetes devono avere installati gli strumenti iSCSI appropriati. Fare riferimento a ["Preparare il nodo worker"](#) per i dettagli.

Autenticare il backend ONTAP

Trident offre due modalità di autenticazione di un backend ONTAP .

- Basato su credenziali: nome utente e password di un utente ONTAP con le autorizzazioni richieste. Si consiglia di utilizzare un ruolo di accesso di sicurezza predefinito, ad esempio `admin` o `vsadmin` per garantire la massima compatibilità con le versioni ONTAP .

- Basato su certificato: Trident può anche comunicare con un cluster ONTAP utilizzando un certificato installato sul backend. Qui, la definizione del backend deve contenere 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 da metodi basati su credenziali a metodi basati su certificati. Tuttavia, è supportato solo un metodo di autenticazione alla volta. Per passare a un metodo di autenticazione diverso, è necessario rimuovere il metodo esistente dalla configurazione del backend.



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

Abilita l'autenticazione basata sulle credenziali

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

Un esempio di definizione del backend sarà simile a questo:

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"
}
```

Tieni presente che la definizione del backend è l'unico posto in cui le credenziali vengono archiviate in testo normale. Dopo aver creato il backend, i nomi utente e le password vengono codificati con Base64 e

memorizzati come segreti Kubernetes. La creazione o l'aggiornamento di un backend è l'unico passaggio che richiede la conoscenza delle credenziali. Si tratta pertanto di un'operazione riservata esclusivamente all'amministratore, che deve essere eseguita dall'amministratore di Kubernetes/archiviazione.

Abilita l'autenticazione basata sul certificato

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

- `clientCertificate`: valore codificato in Base64 del certificato client.
- `clientPrivateKey`: valore codificato in Base64 della chiave privata associata.
- `trustedCACertificate`: valore codificato in Base64 del certificato CA attendibile. Se si utilizza una CA attendibile, è necessario fornire questo parametro. Questo può essere ignorato se non viene utilizzata alcuna CA attendibile.

Un flusso di lavoro tipico prevede i seguenti passaggi.

Passi

1. Genera un certificato client e una chiave. Durante la generazione, impostare il nome comune (CN) sull'utente ONTAP con cui effettuare l'autenticazione.

```
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 dell'archiviazione. Ignora se non viene utilizzata alcuna 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 client e la chiave (dal passaggio 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. Conferma che il ruolo di accesso alla sicurezza ONTAP supporta cert 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. Sostituire < ONTAP Management LIF> e <nome vserver> con l'IP Management LIF 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 attendibile con Base64.

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

7. Creare il backend utilizzando i valori ottenuti dal passaggio precedente.

```
cat cert-backend.json  
{  
  "version": 1,  
  "storageDriverName": "ontap-san",  
  "backendName": "SanBackend",  
  "managementLIF": "1.2.3.4",  
  "svm": "vserver_test",  
  "clientCertificate": "Faaaakkkkeeee...Vaaalllluuuuueeee",  
  "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. 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 |
+-----+-----+-----+-----+
+-----+-----+
```



Quando si ruotano le password, l'amministratore dell'archiviazione deve prima aggiornare la password per l'utente su ONTAP. Segue un aggiornamento del backend. Durante la rotazione dei certificati, è possibile aggiungere più certificati all'utente. Il backend viene quindi aggiornato per utilizzare il nuovo certificato, dopodiché 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 ai volumi effettuate in seguito. Un aggiornamento del backend riuscito indica che Trident può comunicare con il backend ONTAP e gestire le future operazioni sui volumi.

Crea un ruolo ONTAP personalizzato per Trident

È possibile creare un ruolo 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 Trident, Trident utilizza il ruolo del 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

Eeguire i seguenti passaggi in ONTAP System Manager:

1. **Crea un ruolo personalizzato:**

- a. Per creare un ruolo personalizzato a livello di cluster, selezionare **Cluster > Impostazioni**.

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

- b. Selezionare l'icona della freccia (→) accanto a **Utenti e ruoli**.
- c. Selezionare **+Aggiungi in Ruoli**.
- d. Definisci le regole per il ruolo e clicca su **Salva**.

2. **Assegnare il ruolo all'utente Trident *: + Eseguire i seguenti passaggi nella pagina *Utenti e ruoli:**

- a. Selezionare Aggiungi icona **+** in **Utenti**.
- b. Selezionare il nome utente richiesto e selezionare un ruolo nel menu a discesa per **Ruolo**.
- c. Fare clic su **Salva**.

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 `ontap-san` e `ontap-san-economy` conducenti. Ciò richiede l'abilitazione del `useCHAP` opzione nella definizione del backend. Quando impostato su `true` Trident configura la sicurezza dell'iniziatore predefinito dell'SVM su CHAP bidirezionale e imposta il nome utente e i segreti dal file 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
```



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

Inoltre `useCHAP=true`, IL `chapInitiatorSecret`, `chapTargetInitiatorSecret`, `chapTargetUsername`, E `chapUsername` i campi 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 dell'archiviazione indica a Trident di configurare CHAP sul backend di archiviazione. Ciò include quanto segue:

- Impostazione di CHAP sull'SVM:
 - Se il tipo di sicurezza dell'iniziatore predefinito dell'SVM è nessuno (impostato per impostazione predefinita) e non ci sono LUN preesistenti già presenti nel volume, Trident imposterà il tipo di sicurezza predefinito su CHAP e procedere alla configurazione dell'iniziatore CHAP e del nome utente e dei segreti di destinazione.
 - Se l'SVM contiene LUN, Trident non abiliterà CHAP sull'SVM. Ciò garantisce che l'accesso ai LUN già presenti sulla SVM non sia limitato.
- Configurazione dell'iniziatore CHAP e del nome utente e dei segreti di destinazione; queste opzioni devono essere specificate nella configurazione del backend (come mostrato sopra).

Dopo aver creato il 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 in `backend.json` file. Ciò richiederà l'aggiornamento dei segreti CHAP e l'utilizzo di `tridentctl update` comando per riflettere questi cambiamenti.



Quando si aggiornano i segreti CHAP per un backend, è necessario utilizzare `tridentctl` per aggiornare il backend. Non aggiornare le credenziali sul cluster di archiviazione 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-4378-8a33-1a472ffbeeb5c |
online |        7 |
+-----+-----+-----+-----+
+-----+-----+
```

Le connessioni esistenti non saranno interessate e continueranno a rimanere attive se le credenziali vengono aggiornate da Trident sull'SVM. Le nuove connessioni utilizzano le credenziali aggiornate, mentre le connessioni esistenti continuano a rimanere attive. Scollegando e ricollegando i vecchi PV, questi utilizzeranno le credenziali aggiornate.

Opzioni ed esempi di configurazione SAN ONTAP

Scopri come creare e utilizzare i driver ONTAP SAN con la tua installazione Trident .

Questa sezione fornisce esempi di configurazione del backend e dettagli per il mapping dei backend alle StorageClass.

"Sistemi ASA r2" differiscono dagli altri sistemi ONTAP (ASA, AFF e FAS) nell'implementazione del loro livello di archiviazione. Tali variazioni incidono sull'utilizzo di determinati parametri come indicato. ["Scopri di più sulle differenze tra i sistemi ASA r2 e gli altri sistemi ONTAP"](#).




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


Nella configurazione del backend Trident non è necessario specificare che il sistema è ASA r2. Quando selezioni `ontap-san` come il `storageDriverName`, Trident rileva automaticamente il ASA r2 o il tradizionale sistema ONTAP. Alcuni parametri di configurazione del backend non sono applicabili ai sistemi ASA r2, come indicato nella tabella seguente.


Opzioni di configurazione del backend

Per le opzioni di configurazione del backend, consultare la seguente tabella:

Parametro	Descrizione	Predefinito
<code>version</code>		Sempre 1
<code>storageDriverName</code>	Nome del driver di archiviazione	<code>ontap-san`O`ontap-san-economy</code>
<code>backendName</code>	Nome personalizzato o backend di archiviazione	Nome del driver + "_" + <code>dataLIF</code>
<code>managementLIF</code>	<p>Indirizzo IP di un cluster o di un LIF di gestione SVM.</p> <p>È possibile specificare un nome di dominio completo (FQDN).</p> <p>Può essere impostato per utilizzare indirizzi IPv6 se Trident è stato installato utilizzando il flag IPv6. Gli indirizzi IPv6 devono essere definiti tra parentesi quadre, ad esempio [28e8:d9fb:a825:b7bf:69a8:d02f:9e7b:3555].</p> <p>Per un passaggio senza interruzioni a MetroCluster, vedere Esempio MetroCluster.</p> <div><p>Se si utilizzano le credenziali "vsadmin", <code>managementLIF</code> deve essere quello dell'SVM; se si utilizzano le credenziali "admin", <code>managementLIF</code> deve essere quello del cluster.</p></div>	"10.0.0.1", "[2001:1234:abcd::fefe]"

Parametro	Descrizione	Predefinito
dataLIF	Indirizzo IP del protocollo LIF. Può essere impostato per utilizzare indirizzi IPv6 se Trident è stato installato utilizzando 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. Usi Trident " Mappa LUN selettiva ONTAP " per scoprire gli iSCSI LIF necessari per stabilire una sessione multi-percorso. Viene generato un avviso se dataLIF è definito esplicitamente. Omettere per Metrocluster. Vedi il Esempio MetroCluster .	Derivato dall'SVM
svm	Macchina virtuale di archiviazione da utilizzare Ometti per Metrocluster. Vedi il Esempio MetroCluster .	Derivato se un SVM managementLIF è specificato
useCHAP	Utilizzare CHAP per autenticare iSCSI per i driver ONTAP SAN [Booleano]. Impostato su true affinché Trident configuri e utilizzi CHAP bidirezionale come autenticazione predefinita per l'SVM fornito nel backend. Fare riferimento a " Prepararsi a configurare il backend con i driver ONTAP SAN " per i dettagli. Non supportato per FCP o NVMe/TCP.	false
chapInitiatorSecret	Segreto dell'iniziatore CHAP. Obbligatorio se useCHAP=true	""
labels	Insieme di etichette arbitrarie formattate in JSON da applicare ai volumi	""
chapTargetInitiatorSecret	Segreto dell'iniziatore del target 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 . Utilizzato per l'autenticazione basata sulle credenziali. Per l'autenticazione di Active Directory, vedere " Autenticare Trident su un SVM backend utilizzando le credenziali di Active Directory ".	""

Parametro	Descrizione	Predefinito
password	Password necessaria per comunicare con il cluster ONTAP . Utilizzato per l'autenticazione basata sulle credenziali. Per l'autenticazione di Active Directory, vedere "Autenticare Trident su un SVM backend utilizzando le credenziali di Active Directory" .	""
svm	Macchina virtuale di archiviazione da utilizzare	Derivato se un SVM managementLIF è specificato
storagePrefix	Prefisso utilizzato durante il provisioning di nuovi volumi nell'SVM. Non può essere modificato in seguito. Per aggiornare questo parametro, sarà necessario creare un nuovo backend.	trident
aggregate	<p>Aggregato per il provisioning (facoltativo; se impostato, deve essere assegnato all'SVM). Per il <code>ontap-nas-flexgroup</code> driver, questa opzione viene ignorata. Se non assegnato, è possibile utilizzare uno qualsiasi degli aggregati disponibili per eseguire il provisioning di un volume FlexGroup .</p> <div>  <p>Quando l'aggregato viene aggiornato in SVM, viene aggiornato automaticamente in Trident interrogando SVM senza dover riavviare Trident Controller. Dopo aver configurato un aggregato specifico in Trident per il provisioning dei volumi, se l'aggregato viene rinominato o spostato fuori dall'SVM, il backend passerà allo stato di errore in Trident durante l'interrogazione dell'aggregato SVM. Per riportare online il backend, è necessario modificare l'aggregato con uno presente sull'SVM oppure rimuoverlo del tutto.</p> </div> <p>Non specificare per i sistemi ASA r2.</p>	""
limitAggregateUsage	Il provisioning non riesce se l'utilizzo è superiore a questa percentuale. Se si utilizza un backend Amazon FSx for NetApp ONTAP , non specificare <code>limitAggregateUsage</code> . Il fornito <code>fsxadmin</code> E <code>vsadmin</code> non contengono le autorizzazioni richieste per recuperare l'utilizzo aggregato e limitarlo tramite Trident. Non specificare per i sistemi ASA r2.	"" (non applicato di default)
limitVolumeSize	Il provisioning non riesce se la dimensione del volume richiesto è superiore a questo valore. Limita inoltre la dimensione massima dei volumi gestiti per le LUN.	"" (non applicato di default)
lunsPerFlexvol	Numero massimo di LUN per Flexvol, deve essere compreso nell'intervallo [50, 200]	100

Parametro	Descrizione	Predefinito
debugTraceFlags	Flag di debug da utilizzare durante la risoluzione dei problemi. Esempio: {"api":false, "method":true} Non utilizzare a meno che non si stia risolvendo un problema e si necessiti di un dump di registro dettagliato.	null
useREST	<p>Parametro booleano per utilizzare le API REST ONTAP .</p> <div> <p>`useREST` Quando impostato su `true` , Trident utilizza le API REST ONTAP per comunicare con il backend; quando impostato su `false` Trident utilizza chiamate ONTAPI (ZAPI) per comunicare con il backend. Questa funzionalità richiede ONTAP 9.11.1 e versioni successive. Inoltre, il ruolo di accesso ONTAP utilizzato deve avere accesso a `ontapi` applicazione. Ciò è soddisfatto dal predefinito `vsadmin` E `cluster-admin` ruoli. A partire dalla versione Trident 24.06 e ONTAP 9.15.1 o successiva, `useREST` è impostato su `true` per impostazione predefinita; modifica `useREST` A `false` per utilizzare le chiamate ONTAPI (ZAPI).</p> </div> <p>`useREST` è completamente qualificato per NVMe/TCP.</p> <div>  <p>NVMe è supportato solo con le API REST ONTAP e non con ONTAPI (ZAPI).</p> </div> <p>Se specificato, impostare sempre su <code>true</code> per sistemi ASA r2.</p>	true`per ONTAP 9.15.1 o successivo, altrimenti `false`.
sanType	Utilizzare per selezionare <code>iscsi</code> per iSCSI, <code>nvme</code> per NVMe/TCP o <code>fc</code> per SCSI su Fibre Channel (FC).	`iscsi` se vuoto

Parametro	Descrizione	Predefinito
formatOptions	Utilizzo formatOptions per specificare gli argomenti della riga di comando per mkfs comando, che verrà applicato ogni volta che un volume viene formattato. Ciò consente di formattare il volume in base alle proprie preferenze. Assicurarsi di specificare formatOptions in modo simile a quello delle 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 i sistemi ASA r2 quando si utilizzano i protocolli iSCSI e NVMe/TCP.	
limitVolumePoolSize	Dimensione massima FlexVol richiedibile quando si utilizzano LUN nel backend ontap-san-economy.	"" (non applicato di default)
denyNewVolumePools	Limita ontap-san-economy backend dalla creazione di nuovi volumi FlexVol per contenere i loro LUN. Per il provisioning di nuovi PV vengono utilizzati solo i Flexvol preesistenti.	

Consigli per l'utilizzo di formatOptions

Trident consiglia la seguente opzione per velocizzare il processo di formattazione:

-E nodiscard:

- Mantieni, non tentare di scartare i blocchi in fase di mkfs (scartare i blocchi inizialmente è utile su dispositivi a stato solido e storage sparse/thin-provisioned). Sostituisce l'opzione obsoleta "-K" ed è applicabile a tutti i file system (xfs, ext3 ed ext4).

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

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

passi

1. Configurare le impostazioni del Domain Name System (DNS) per un SVM 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 l'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 username E password parametri rispettivamente per il nome utente o gruppo AD e la password.

Opzioni di configurazione del backend per il provisioning dei volumi

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

Parametro	Descrizione	Predefinito
spaceAllocation	Assegnazione dello spazio per LUN	"true" Se specificato, impostare su true per sistemi ASA r2.
spaceReserve	Modalità di prenotazione dello spazio: "nessuno" (sottile) o "volume" (spesso). Impostato su none per sistemi ASA r2.	"nessuno"
snapshotPolicy	Criterio di snapshot da utilizzare. Impostato su none per sistemi ASA r2.	"nessuno"
qosPolicy	Gruppo di criteri QoS da assegnare ai volumi creati. Scegliere tra qosPolicy o adaptiveQosPolicy per ogni pool di archiviazione/backend. Per utilizzare i gruppi di policy QoS con Trident è necessario ONTAP 9.8 o versione successiva. È necessario utilizzare un gruppo di policy QoS non condiviso e assicurarsi che il gruppo di policy venga applicato individualmente a ciascun componente. Un gruppo di policy QoS condiviso impone il limite massimo per la produttività totale di tutti i carichi di lavoro.	""
adaptiveQosPolicy	Gruppo di policy QoS adattivo da assegnare ai volumi creati. Scegli uno tra qosPolicy o adaptiveQosPolicy per pool di archiviazione/backend	""
snapshotReserve	Percentuale di volume riservata agli snapshot. Non specificare per i sistemi ASA r2.	"0" se snapshotPolicy è "nessuno", altrimenti ""
splitOnClone	Dividere un clone dal suo genitore al momento della creazione	"falso"
encryption	Abilita NetApp Volume Encryption (NVE) sul nuovo volume; l'impostazione predefinita è false . Per utilizzare questa opzione, NVE deve essere concesso in licenza e abilitato sul cluster. Se NAE è abilitato sul backend, qualsiasi volume fornito in Trident sarà abilitato per NAE. Per maggiori informazioni, fare riferimento a: "Come funziona Trident con NVE e NAE" .	"false" Se specificato, impostare su true per sistemi ASA r2.

Parametro	Descrizione	Predefinito
luksEncryption	Abilita la crittografia LUKS. Fare riferimento a "Utilizzare Linux Unified Key Setup (LUKS)" .	"" Impostato su false per sistemi ASA r2.
tieringPolicy	Criterio di suddivisione in livelli per utilizzare "nessuno" 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 definiti:

```
---
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'
```



Per tutti i volumi creati utilizzando ontap-san driver, Trident aggiunge un ulteriore 10 percento di capacità al FlexVol per ospitare i metadati LUN. La LUN verrà fornita con le dimensioni esatte richieste dall'utente nel PVC. Trident aggiunge il 10 percento al FlexVol (mostrato come dimensione disponibile in ONTAP). Gli utenti riceveranno ora la quantità di capacità utilizzabile richiesta. Questa modifica impedisce inoltre che i LUN diventino di sola lettura, a meno che lo spazio disponibile non sia completamente utilizzato. Ciò 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 per cento in più Trident aggiunge al FlexVol per ospitare i metadati LUN. Per `snapshotReserve = 5%` e richiesta PVC = 5 GiB, la dimensione totale del volume è 5,79 GiB e la dimensione disponibile è 5,5 GiB. IL `volume show` il comando dovrebbe mostrare risultati simili a questo esempio:

Vserver	Volume	Aggregate	State	Type	Size	Available	Used%
		_pvc_89f1c156_3801_4de4_9f9d_034d54c395f4	online	RW	10GB	5.00GB	0%
		_pvc_e42ec6fe_3baa_4af6_996d_134adbbb8e6d	online	RW	5.79GB	5.50GB	0%
		_pvc_e8372153_9ad9_474a_951a_08ae15e1c0ba	online	RW	1GB	511.8MB	0%

3 entries were displayed.

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 predefiniti. Questo è il modo più semplice per definire un backend.



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

Esempio ONTAP SAN

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

```
---
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 MetroCluster

È possibile configurare il backend per evitare di dover aggiornare manualmente la definizione del backend dopo il passaggio e il ritorno durante ["Replica e ripristino SVM"](#) .

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

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

Esempio di economia ONTAP SAN

```
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) vengono popolati in `backend.json` e prendere 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 CHAP bidirezionali

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 economico 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 NVMe/TCP

È necessario disporre di una SVM configurata con NVMe sul backend ONTAP . Questa è una configurazione backend di base 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 SCSI su FC (FCP)

È necessario disporre di un SVM configurato con FC sul backend ONTAP . Questa è una configurazione backend di base 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}}"
```

Esempio di formatOptions 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 archiviazione, ad esempio `spaceReserve` a nessuno, `spaceAllocation` a falso, e `encryption` a falso. 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 nel volume di archiviazione al momento del provisioning. Per comodità, gli amministratori di storage possono definire etichette per pool virtuale e raggruppare i volumi in base all'etichetta.

In questi esempi, alcuni dei pool di archiviazione impostano i propri `spaceReserve`, `spaceAllocation`, `Encryption` valori e alcuni pool sovrascrivono i valori predefiniti.



```

---
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"

```

```
---
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 su StorageClasses

Le seguenti definizioni StorageClass fanno riferimento a [Esempi di backend con pool virtuali](#) . Utilizzando il `parameters.selector` campo, ogni StorageClass richiama quali pool virtuali possono essere utilizzati per ospitare un volume. Il volume avrà gli aspetti definiti nel pool virtuale scelto.

- IL `protection-gold` StorageClass verrà mappato sul primo pool virtuale nel `ontap-san` backend. Questa è l'unica piscina che offre una protezione di livello Gold.

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

- IL protection-not-gold StorageClass verrà mappato sul secondo e terzo pool virtuale in ontap-san backend. Questi sono gli unici pool che offrono un livello di protezione diverso dall'oro.

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

- IL app-mysqldb StorageClass verrà mappato sul terzo pool virtuale in ontap-san-economy backend. Questo è l'unico pool che offre la configurazione del pool di archiviazione per l'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"
```

- IL protection-silver-creditpoints-20k StorageClass verrà mappato sul secondo pool virtuale in ontap-san backend. Questo è l'unico pool che offre protezione di livello Silver e 20.000 punti di 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"
```

- IL creditpoints-5k StorageClass verrà mappato sul terzo pool virtuale in ontap-san backend e il quarto pool virtuale nel ontap-san-economy backend. Queste sono le uniche offerte di pool con 5000 punti di credito.

```

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

```

- IL my-test-app-sc StorageClass verrà mappato su testAPP piscina virtuale nel ontap-san autista con sanType: nvme . Questa è l'unica piscina 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 deciderà quale pool virtuale selezionare e garantirà che i requisiti di archiviazione siano soddisfatti.

Driver NAS ONTAP

Panoramica del driver NAS ONTAP

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 archiviazione NAS per comunicare con il cluster ONTAP . Le modalità di accesso supportate sono: *ReadWriteOnce* (RWO), *ReadOnlyMany* (ROX), *ReadWriteMany* (RWX), *ReadWriteOncePod* (RWOP).

Autista	Protocollo	Modalità volume	Modalità di accesso supportate	Sistemi di file supportati
ontap-nas	NFS SMB	File system	RWO, ROX, RWX, RWOP	"" , nfs , smb
ontap-nas-economy	NFS SMB	File system	RWO, ROX, RWX, RWOP	"" , nfs , smb
ontap-nas-flexgroup	NFS SMB	File system	RWO, ROX, RWX, RWOP	"" , nfs , smb



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

Permessi utente

Trident prevede di essere eseguito come amministratore ONTAP o SVM, in genere utilizzando `admin` utente del cluster o un `vsadmin` Utente SVM o 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 il cluster `fsxadmin` utente o un `vsadmin` Utente SVM o un utente con un nome diverso che ha lo stesso ruolo. IL `fsxadmin` L'utente è un sostituto limitato dell'utente amministratore del cluster.



Se usi il `limitAggregateUsage` parametro, sono richieste le autorizzazioni di amministratore del cluster. Quando si utilizza Amazon FSx for NetApp ONTAP con Trident, il `limitAggregateUsage` il parametro non funzionerà con il `vsadmin` E `fsxadmin` account utente. Se si specifica questo parametro, l'operazione di configurazione non riuscirà.

Sebbene sia possibile creare un ruolo più restrittivo all'interno di ONTAP utilizzabile da un conducente Trident , non lo consigliamo. La maggior parte delle nuove versioni di Trident richiameranno 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.

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, è possibile configurare una classe Gold che utilizza `ontap-nas` driver e una classe Bronze che utilizza il `ontap-nas-economy` uno.
- Tutti i nodi worker di Kubernetes devono avere installati gli strumenti NFS appropriati. Fare riferimento a ["Qui"](#) per maggiori dettagli.
- Trident supporta volumi SMB montati su pod in esecuzione solo su nodi Windows. Fare riferimento a [Prepararsi al provisioning dei volumi SMB](#) per i dettagli.

Autenticare il backend ONTAP

Trident offre due modalità di autenticazione di un backend ONTAP .

- Basato su credenziali: questa modalità richiede autorizzazioni sufficienti per il backend ONTAP . Si consiglia di utilizzare un account associato a un ruolo di accesso di sicurezza predefinito, ad esempio `admin` o `vsadmin` per garantire la massima compatibilità con le versioni ONTAP .
- Basato su certificato: questa modalità richiede un certificato installato sul backend affinché Trident possa comunicare con un cluster ONTAP . Qui, la definizione del backend deve contenere 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 da metodi basati su credenziali a metodi basati su certificati. Tuttavia, è supportato solo un metodo di autenticazione alla volta. Per passare a un metodo di autenticazione diverso, è necessario rimuovere il metodo esistente dalla configurazione del backend.



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

Abilita l'autenticazione basata sulle credenziali

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

Un esempio di definizione del backend sarà simile a questo:

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

```
{
  "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"
  }
}
```

Tieni presente che la definizione del backend è l'unico posto in cui le credenziali vengono archiviate in testo normale. Dopo aver creato il backend, i nomi utente e le password vengono codificati con Base64 e memorizzati come segreti Kubernetes. La creazione/aggiornamento di un backend è l'unico passaggio che richiede la conoscenza delle credenziali. Si tratta pertanto di un'operazione riservata esclusivamente all'amministratore, che deve essere eseguita dall'amministratore di Kubernetes/archiviazione.

Abilita l'autenticazione basata su certificato

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

- `clientCertificate`: valore codificato in Base64 del certificato client.
- `clientPrivateKey`: valore codificato in Base64 della chiave privata associata.
- `trustedCACertificate`: valore codificato in Base64 del certificato CA attendibile. Se si utilizza una CA attendibile, è necessario fornire questo parametro. Questo può essere ignorato se non viene utilizzata alcuna CA attendibile.

Un flusso di lavoro tipico prevede i seguenti passaggi.

Passi

1. Genera un certificato client e una chiave. Durante la generazione, impostare il nome comune (CN) sull'utente ONTAP con cui effettuare l'autenticazione.

```
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 dell'archiviazione. Ignora se non viene utilizzata alcuna 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 client e la chiave (dal passaggio 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. Conferma che il ruolo di accesso alla sicurezza ONTAP supporta cert 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. Sostituire < ONTAP Management LIF> e <nome vservice> con l'IP Management LIF e il nome SVM. È necessario assicurarsi che la politica di servizio del LIF sia 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 attendibile con Base64.

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

7. Creare il backend utilizzando i valori ottenuti dal 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...Vaaalllluuuuueeee",
  "clientPrivateKey": "LS0tFaKE...0VaLuES0tLS0K",
  "storagePrefix": "myPrefix_"
}

#Update backend with tridentctl
tridentctl update backend NasBackend -f cert-backend-updated.json -n
trident
```

NAME	STORAGE DRIVER	UUID
NasBackend	ontap-nas	98e19b74-aec7-4a3d-8dcf-128e5033b214

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. 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
NasBackend	ontap-nas	98e19b74-aec7-4a3d-8dcf-128e5033b214



Quando si ruotano le password, l'amministratore dell'archiviazione deve prima aggiornare la password per l'utente su ONTAP. Segue un aggiornamento del backend. Durante la rotazione dei certificati, è possibile aggiungere più certificati all'utente. Il backend viene quindi aggiornato per utilizzare il nuovo certificato, dopodiché 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 ai volumi effettuate in seguito. Un aggiornamento del backend riuscito indica che Trident può comunicare con il backend ONTAP e gestire le future operazioni sui volumi.

Crea un ruolo ONTAP personalizzato per Trident

È possibile creare un ruolo 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 Trident, Trident utilizza il ruolo del 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

Eeguire i seguenti passaggi in ONTAP System Manager:

1. **Crea un ruolo personalizzato:**

- a. Per creare un ruolo personalizzato a livello di cluster, selezionare **Cluster > Impostazioni**.

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

- b. Selezionare l'icona della freccia (→) accanto a **Utenti e ruoli**.

- c. Selezionare **+Aggiungi in Ruoli**.

- d. Definisci le regole per il ruolo e clicca su **Salva**.

2. **Assegnare il ruolo all'utente Trident *: + Eseguire i seguenti passaggi nella pagina *Utenti e ruoli:**

- a. Selezionare Aggiungi icona **+** in **Utenti**.

- b. Selezionare il nome utente richiesto e selezionare un ruolo nel menu a discesa per **Ruolo**.

- c. Fare clic su **Salva**.

Per maggiori informazioni consultare le seguenti pagine:

- ["Ruoli personalizzati per l'amministrazione di ONTAP"](#) o ["Definisci ruoli personalizzati"](#)
- ["Lavorare con ruoli e utenti"](#)

Gestire le policy di esportazione NFS

Trident utilizza criteri di esportazione NFS per controllare l'accesso ai volumi di cui si occupa.

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

- Trident può gestire dinamicamente la politica di esportazione autonomamente; in questa modalità di

funzionamento, l'amministratore dell'archiviazione specifica un elenco di blocchi CIDR che rappresentano indirizzi IP ammissibili. Trident aggiunge automaticamente alla policy di esportazione gli IP dei nodi applicabili che rientrano in questi intervalli al momento della pubblicazione. In alternativa, se non vengono specificati CIDR, tutti gli IP unicast con ambito globale trovati sul nodo su cui viene pubblicato il volume verranno aggiunti alla policy di esportazione.

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

Gestire dinamicamente le politiche di esportazione

Trident offre la possibilità di gestire dinamicamente le policy di esportazione per i backend ONTAP . Ciò consente all'amministratore dell'archiviazione di specificare uno spazio di indirizzamento consentito per gli IP dei nodi worker, anziché definire manualmente regole esplicite. Semplifica notevolmente la gestione delle policy di esportazione: le modifiche alle policy di esportazione non richiedono più un intervento manuale sul cluster di storage. Inoltre, ciò consente di limitare l'accesso al cluster di archiviazione solo ai nodi worker che montano volumi e hanno IP compresi nell'intervallo specificato, supportando una gestione automatizzata e dettagliata.



Non utilizzare Network Address Translation (NAT) quando si utilizzano criteri di esportazione dinamici. Con NAT, il controller di archiviazione vede l'indirizzo NAT frontend e non l'indirizzo host IP effettivo, quindi l'accesso verrà negato se 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 del 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
```



Quando si utilizza questa funzionalità, è necessario assicurarsi che la giunzione radice nella SVM disponga di una policy di esportazione creata in precedenza con una regola di esportazione che consenta il blocco CIDR del nodo (ad esempio la policy di esportazione predefinita). Seguire sempre le best practice consigliate NetApp per dedicare una SVM a Trident.

Ecco una spiegazione del funzionamento di questa funzionalità utilizzando l'esempio sopra riportato:

- `autoExportPolicy` è impostato su `true`. Ciò indica che Trident crea una policy di esportazione per ogni volume fornito con questo backend per il `svm1` SVM e gestire l'aggiunta e l'eliminazione delle regole utilizzando `autoexportCIDRs` blocchi di indirizzi. Finché un volume non viene collegato a un nodo, il volume utilizza una policy di esportazione vuota, senza regole, per impedire l'accesso indesiderato a tale 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 aggiunti anche alla policy di esportazione utilizzata dal FlexVol volume padre
 - Per esempio:
 - UUID backend `403b5326-8482-40db-96d0-d83fb3f4daec`
 - `autoExportPolicy` impostato su `true`
 - prefisso di archiviazione `trident`
 - Codice UUID PVC `a79bcf5f-7b6d-4a40-9876-e2551f159c1c`
 - qtree denominato `trident_pvc_a79bcf5f_7b6d_4a40_9876_e2551f159c1c` crea una politica di esportazione per FlexVol denominata `trident-403b5326-8482-40db96d0-d83fb3f4daec`, una politica di esportazione per il qtree denominato `trident_pvc_a79bcf5f_7b6d_4a40_9876_e2551f159c1c` e una politica di esportazione vuota denominata `trident_empty` sull'SVM. Le regole per la policy di esportazione FlexVol saranno un superset di tutte le regole contenute nelle policy di esportazione qtree. La policy di esportazione vuota verrà riutilizzata da tutti i volumi non collegati.
- `autoExportCIDRs` contiene un elenco di blocchi di indirizzi. Questo campo è facoltativo e il suo valore predefinito è `["0.0.0.0/0", "::/0"]`. Se non definito, Trident aggiunge tutti gli indirizzi unicast con ambito globale trovati sui nodi worker con pubblicazioni.

In questo esempio, il `192.168.0.0/24` è fornito lo spazio di indirizzamento. Ciò indica che gli IP dei nodi Kubernetes che rientrano in questo intervallo di indirizzi con pubblicazioni verranno aggiunti alla policy di esportazione creata Trident. Quando Trident registra un nodo su cui è in esecuzione, recupera gli indirizzi IP del nodo e li confronta con i blocchi di indirizzi forniti in `autoExportCIDRs`. Al momento della pubblicazione, dopo aver filtrato gli IP, Trident crea le regole della policy di esportazione per gli IP client del nodo su cui sta pubblicando.

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

Dopo che Trident crea o aggiorna un backend, puoi controllare 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>
      fileType: 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 mount non autorizzati, a meno che questo IP non venga riutilizzato da un nuovo nodo nel cluster.

Per i backend già esistenti, aggiornare il backend con `tridentctl update backend` garantisce che Trident gestisca automaticamente le politiche di esportazione. In questo modo vengono create due nuove policy di esportazione denominate in base all'UUID del backend e al nome `qtree` quando necessario. I volumi presenti nel backend utilizzeranno i criteri di esportazione appena creati dopo essere stati smontati e montati nuovamente.



L'eliminazione di un backend con criteri di esportazione gestiti automaticamente eliminerà il criterio di esportazione creato 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 attivo viene aggiornato, è necessario riavviare il pod Trident sul nodo. Trident aggiornerà quindi la politica di esportazione per i backend che gestisce per riflettere questa modifica dell'IP.

Prepararsi al provisioning dei volumi SMB

Con un po' di preparazione aggiuntiva, è possibile eseguire il provisioning dei volumi SMB utilizzando `ontap-nas` conducenti.



È necessario configurare entrambi i protocolli NFS e SMB/CIFS sull'SVM per creare un `ontap-nas-economy` Volume SMB per cluster ONTAP on-premise. 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

Prima di poter effettuare 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 su pod in esecuzione solo su nodi Windows.
- Almeno un segreto Trident contenente le credenziali di Active Directory. Per generare segreto `smbcreds`:

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

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

Passi

1. Per ONTAP on-premise, 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 amministrative SMB in uno dei due modi seguenti: utilizzando ["Console di gestione Microsoft"](#) Snap-in Cartelle condivise o tramite ONTAP CLI. Per creare le condivisioni SMB utilizzando ONTAP CLI:

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

Il `vserver cifs share create` Il 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 all'SVM specificato:

```
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
```



Fare riferimento a ["Crea una condivisione SMB"](#) per maggiori dettagli.

2. Durante la creazione del backend, è necessario configurare quanto segue per specificare i volumi SMB. Per tutte le opzioni di configurazione del backend FSx for ONTAP, fare riferimento a ["Opzioni di configurazione ed esempi di FSx per ONTAP"](#).

Parametro	Descrizione	Esempio
smbShare	È possibile specificare uno dei seguenti elementi: 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 comune alla condivisione ai volumi. Questo parametro è facoltativo per ONTAP on-premise. Questo parametro è obbligatorio per i backend Amazon FSx per ONTAP e non può essere vuoto.	smb-share
nasType	Deve essere impostato su smb . Se nullo, il valore predefinito è <code>nfs</code> .	smb
securityStyle	Stile di sicurezza per i nuovi volumi. Deve essere impostato su ntfs O mixed per volumi SMB.	ntfs`O `mixed per volumi SMB
unixPermissions	Modalità per nuovi volumi. Deve essere lasciato vuoto per i volumi SMB.	""

Abilita SMB sicuro

A partire dalla versione 25.06, NetApp Trident supporta il provisioning sicuro dei volumi SMB creati utilizzando `ontap-nas` E `ontap-nas-economy` backend. Quando è abilitato SMB sicuro, è possibile fornire un accesso controllato alle condivisioni SMB per gli utenti e i gruppi di utenti di Active Directory (AD) utilizzando gli elenchi di controllo di accesso (ACL).

Punti da ricordare

- Importazione `ontap-nas-economy` volumi non è supportato.
- Sono supportati solo i cloni di sola lettura per `ontap-nas-economy` volumi.
- Se Secure SMB è abilitato, Trident ignorerà la condivisione SMB menzionata nel backend.
- L'aggiornamento dell'annotazione PVC, dell'annotazione della classe di archiviazione e del campo backend non aggiorna l'ACL della condivisione SMB.
- L'ACL di condivisione SMB specificato nell'annotazione del PVC clone avrà la precedenza su quelli presenti nel PVC di origine.
- Assicuratevi di fornire utenti AD validi quando abiliti SMB sicuro. Gli utenti non validi non verranno aggiunti all'ACL.
- Se si forniscono autorizzazioni diverse allo stesso utente AD nel backend, nella classe di archiviazione e nel PVC, la priorità delle autorizzazioni sarà: PVC, classe di archiviazione e quindi backend.
- Secure SMB è supportato per `ontap-nas` importazioni di volumi gestiti e non applicabile alle importazioni di volumi non gestiti.

Passi

1. Specificare `adAdminUser` in `TridentBackendConfig` come mostrato nell'esempio seguente:

```

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. Aggiungere l'annotazione nella classe di archiviazione.

Aggiungi il `trident.netapp.io/smbShareAdUser` annotazione alla classe di archiviazione per abilitare SMB sicuro senza errori. Il valore utente specificato per l'annotazione `trident.netapp.io/smbShareAdUser` dovrebbe essere lo stesso del nome utente specificato nel `smbcreds` segreto. Puoi 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. Creare un PVC.

L'esempio seguente 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 del NAS ONTAP



Scopri come creare e utilizzare i driver ONTAP NAS con l'installazione Trident . Questa sezione fornisce esempi di configurazione del backend e dettagli per il mapping dei backend alle StorageClass.


Opzioni di configurazione del backend

Per le opzioni di configurazione del backend, consultare la seguente tabella:

Parametro	Descrizione	Predefinito
version		Sempre 1
storageDriverName	Nome del driver di archiviazione	ontap-nas, ontap-nas-economy , O ontap-nas-flexgroup
backendName	Nome personalizzato o backend di archiviazione	Nome del driver + "_" + dataLIF
managementLIF	Indirizzo IP di un cluster o di un LIF di gestione SVM. È possibile specificare un nome di dominio completo (FQDN). Può essere impostato per utilizzare indirizzi IPv6 se Trident è stato installato utilizzando 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 a MetroCluster , vedere Esempio MetroCluster .	"10.0.0.1", "[2001:1234:abcd::fefe]"

Parametro	Descrizione	Predefinito
dataLIF	Indirizzo IP del protocollo LIF. NetApp consiglia di specificare dataLIF . Se non specificato, Trident recupera i dataLIF dall'SVM. È possibile specificare un nome di dominio completo (FQDN) da utilizzare per le operazioni di montaggio NFS, consentendo di creare un DNS round-robin per bilanciare il carico su più dataLIF. Può essere modificato dopo l'impostazione iniziale. Fare riferimento a . Può essere impostato per utilizzare indirizzi IPv6 se Trident è stato installato utilizzando 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 MetroCluster .	Indirizzo specificato o derivato da SVM, se non specificato (non consigliato)
svm	Macchina virtuale di archiviazione da utilizzare Ometti per Metrocluster. Vedi il Esempio MetroCluster .	Derivato se un SVM managementLIF è specificato
autoExportPolicy	Abilita la creazione e l'aggiornamento automatici dei criteri di esportazione [Booleano]. Utilizzando il autoExportPolicy E autoExportCIDRs opzioni, Trident può gestire automaticamente le policy di esportazione.	falso
autoExportCIDRs	Elenco di CIDR per filtrare gli IP dei nodi di Kubernetes quando autoExportPolicy è abilitato. Utilizzando il autoExportPolicy E autoExportCIDRs opzioni, Trident può gestire automaticamente le policy di esportazione.	["0.0.0.0/0", ":::0"]
labels	Insieme di etichette arbitrarie formattate in 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	""
username	Nome utente per connettersi al cluster/SVM. Utilizzato per l'autenticazione basata sulle credenziali. Per l'autenticazione di Active Directory, vedere "Autenticare Trident su un SVM backend utilizzando le credenziali di Active Directory" .	
password	Password per connettersi al cluster/SVM. Utilizzato per l'autenticazione basata sulle credenziali. Per l'autenticazione di Active Directory, vedere "Autenticare Trident su un SVM backend utilizzando le credenziali di Active Directory" .	

Parametro	Descrizione	Predefinito
storagePrefix	<p>Prefisso utilizzato durante il provisioning di nuovi volumi nell'SVM. Non può essere aggiornato dopo averlo impostato</p> <div>  <p>Quando si utilizza ontap-nas-economy e un prefisso storage di 24 o più caratteri, i qtrees non avranno il prefisso storage incorporato, sebbene sarà presente nel nome del volume.</p> </div>	"tridente"
aggregate	<p>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 eseguire il provisioning di un volume FlexGroup .</p> <div>  <p>Quando l'aggregato viene aggiornato in SVM, viene aggiornato automaticamente in Trident interrogando SVM senza dover riavviare Trident Controller. Dopo aver configurato un aggregato specifico in Trident per il provisioning dei volumi, se l'aggregato viene rinominato o spostato fuori dall'SVM, il backend passerà allo stato di errore in Trident durante l'interrogazione dell'aggregato SVM. Per riportare online il backend, è necessario modificare l'aggregato con uno presente sull'SVM oppure rimuoverlo del tutto.</p> </div>	""
limitAggregateUsage	<p>Il provisioning non riesce se l'utilizzo è superiore a questa percentuale. Non si applica ad Amazon FSx per ONTAP.</p>	"" (non applicato di default)

Parametro	Descrizione	Predefinito
flexgroupAggregateList	<p>Elenco degli aggregati per il provisioning (facoltativo; se impostato, deve essere assegnato all'SVM). Tutti gli aggregati assegnati all'SVM vengono utilizzati per effettuare il provisioning di un volume FlexGroup . Supportato per il driver di archiviazione ontap-nas-flexgroup.</p> <p> Quando l'elenco aggregato viene aggiornato in SVM, l'elenco viene aggiornato automaticamente in Trident interrogando SVM senza dover riavviare Trident Controller. Dopo aver configurato un elenco di aggregati specifico in Trident per il provisioning dei volumi, se l'elenco di aggregati viene rinominato o spostato fuori da SVM, il backend passerà allo stato di errore in Trident durante l'interrogazione dell'aggregato SVM. Per riportare online il backend, è necessario modificare l'elenco aggregato con uno presente sull'SVM oppure rimuoverlo del tutto.</p>	""
limitVolumeSize	Il provisioning non riesce se la dimensione del volume richiesto è superiore a questo valore. Limita inoltre la dimensione massima dei volumi che gestisce per i <code>qtree</code> e <code>qtreesPerFlexvol</code> l'opzione consente di personalizzare il numero massimo di <code>qtree</code> per FlexVol volume	"" (non applicato di default)
debugTraceFlags	Flag di debug da utilizzare durante la risoluzione dei problemi. Esempio, {"api":false, "method":true} Non utilizzare <code>debugTraceFlags</code> a meno che non si stia risolvendo un problema e si necessiti di un dump di registro dettagliato.	null
nasType	Configurare la creazione di volumi NFS o SMB. Le opzioni sono <code>nfs</code> , <code>smb</code> o nullo. Impostando il valore su null, i volumi NFS vengono impostati di default.	<code>nfs</code>
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 archiviazione, ma se non vengono specificate opzioni di montaggio in una classe di archiviazione, Trident utilizzerà le opzioni di montaggio specificate nel file di configurazione del backend di archiviazione. Se non vengono specificate opzioni di montaggio nella classe di archiviazione o nel file di configurazione, Trident non imposterà alcuna opzione di montaggio su un volume persistente associato.	""

Parametro	Descrizione	Predefinito
qtreesPerFlexvol	Il numero massimo di Qtree per FlexVol deve essere compreso nell'intervallo [50, 300]	"200"
smbShare	È possibile specificare uno dei seguenti elementi: 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 comune alla condivisione ai volumi. Questo parametro è facoltativo per ONTAP on-premise. Questo parametro è obbligatorio per i backend Amazon FSx per ONTAP e non può essere vuoto.	smb-share
useREST	Parametro booleano per utilizzare le API REST ONTAP. <code>useREST</code> Quando impostato su <code>true</code> , Trident utilizza le API REST ONTAP per comunicare con il backend; quando impostato su <code>false</code> Trident utilizza chiamate ONTAPI (ZAPI) per comunicare con il backend. Questa funzionalità richiede ONTAP 9.11.1 e versioni successive. Inoltre, il ruolo di accesso ONTAP utilizzato deve avere accesso a <code>ontapi</code> applicazione. Ciò è soddisfatto dal predefinito <code>vsadmin</code> E <code>cluster-admin</code> ruoli. A partire dalla versione Trident 24.06 e ONTAP 9.15.1 o successiva, <code>useREST</code> è impostato su <code>true</code> per impostazione predefinita; modifica <code>useREST</code> A <code>false</code> per utilizzare le chiamate ONTAPI (ZAPI).	<code>true</code> per ONTAP 9.15.1 o successivo, altrimenti <code>false</code> .
limitVolumePoolSize	Dimensione massima FlexVol richiedibile quando si utilizzano Qtrees nel backend <code>ontap-nas-economy</code> .	"" (non applicato di default)
denyNewVolumePools	Limita <code>ontap-nas-economy</code> backend dalla creazione di nuovi volumi FlexVol per contenere i loro Qtree. Per il provisioning di nuovi PV vengono utilizzati solo i Flexvol preesistenti.	
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 in `defaults` sezione della configurazione. Per un esempio, vedere gli esempi di configurazione riportati di seguito.

Parametro	Descrizione	Predefinito
spaceAllocation	Assegnazione dello spazio per Qtrees	"VERO"

Parametro	Descrizione	Predefinito
spaceReserve	Modalità di prenotazione dello spazio; "nessuno" (sottile) o "volume" (spesso)	"nessuno"
snapshotPolicy	Criterio di snapshot da utilizzare	"nessuno"
qosPolicy	Gruppo di criteri QoS da assegnare ai volumi creati. Scegli uno tra qosPolicy o adaptiveQosPolicy per pool di archiviazione/backend	""
adaptiveQosPolicy	Gruppo di policy QoS adattivo da assegnare ai volumi creati. Scegliere tra qosPolicy o adaptiveQosPolicy per ogni pool di archiviazione/backend. Non supportato da ontap-nas-economy.	""
snapshotReserve	Percentuale di volume riservata agli snapshot	"0" se snapshotPolicy è "nessuno", altrimenti ""
splitOnClone	Dividere un clone dal suo genitore al momento della creazione	"falso"
encryption	Abilita NetApp Volume Encryption (NVE) sul nuovo volume; l'impostazione predefinita è false . Per utilizzare questa opzione, NVE deve essere concesso in licenza e abilitato sul cluster. Se NAE è abilitato sul backend, qualsiasi volume fornito in Trident sarà abilitato per NAE. Per maggiori informazioni, fare riferimento a: "Come funziona Trident con NVE e NAE" .	"falso"
tieringPolicy	Criterio di tiering per utilizzare "nessuno"	
unixPermissions	Modalità per nuovi volumi	"777" per i volumi NFS; vuoto (non applicabile) per i volumi SMB
snapshotDir	Controlla l'accesso al .snapshot elenco	"true" per NFSv4 "false" per NFSv3
exportPolicy	Politica di esportazione da utilizzare	"predefinito"
securityStyle	Stile di sicurezza per i nuovi volumi. Supporti NFS mixed E unix stili di sicurezza. Supporti SMB 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.	""



Per utilizzare i gruppi di policy QoS con Trident è necessario ONTAP 9.8 o versione successiva. Dovresti utilizzare un gruppo di policy QoS non condiviso e assicurarti che il gruppo di policy venga applicato individualmente a ciascun componente. Un gruppo di policy QoS condiviso impone il limite massimo per la produttività totale di tutti i carichi di lavoro.

Esempi di provisioning del volume

Ecco un esempio con i valori predefiniti definiti:

```

---
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 FlexVol sia dimensionato correttamente con la percentuale `snapshotReserve` e 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 uno spazio inferiore a quello richiesto. Prima della versione 21.07, quando l'utente richiedeva un PVC (ad esempio, 5 GiB), con `snapshotReserve` al 50%, otteneva solo 2,5 GiB di spazio scrivibile. Questo perché ciò che l'utente ha richiesto è l'intero volume e `snapshotReserve` è una percentuale di quello. Con Trident 21.07, ciò che l'utente richiede è lo spazio scrivibile e Trident definisce lo `snapshotReserve` numero come percentuale del volume totale. Questo non si applica a `ontap-nas-economy`. Per vedere come funziona, vedere l'esempio seguente

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 $5/.5 = 10$ GiB e la dimensione disponibile è 5 GiB, che è ciò che l'utente ha richiesto nella richiesta PVC IL `volume show` il comando dovrebbe mostrare risultati simili a questo esempio:

Vserver	Volume	Aggregate	State	Type	Size	Available	Used%
		_pvc_89f1c156_3801_4de4_9f9d_034d54c395f4	online	RW	10GB	5.00GB	0%
		_pvc_e8372153_9ad9_474a_951a_08ae15e1c0ba	online	RW	1GB	511.8MB	0%

2 entries were displayed.

I backend esistenti delle installazioni precedenti eseguiranno il provisioning dei volumi come spiegato sopra durante l'aggiornamento Trident. Per i volumi creati prima dell'aggiornamento, è necessario ridimensionarli affinché la modifica venga visualizzata. Ad esempio, un PVC da 2 GiB con `snapshotReserve=50` in precedenza produceva un volume che forniva 1 GiB di spazio scrivibile. Ridimensionando il volume a 3 GiB, ad esempio, l'applicazione ottiene 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 predefiniti. Questo è il modo più semplice per definire un backend.



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 MetroCluster

È possibile configurare il backend per evitare di dover aggiornare manualmente la definizione del backend dopo il passaggio e il ritorno durante ["Replica e ripristino SVM"](#) .

Per un passaggio e un ritorno senza interruzioni, specificare l'SVM utilizzando `managementLIF` e omettere il `dataLIF` E `svm` parametri. Per 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 minimo di configurazione backend. `clientCertificate`, `clientPrivateKey`, E `trustedCACertificate` (facoltativo, se si utilizza una CA attendibile) vengono popolati in `backend.json` e prendere 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 è possibile istruire Trident a utilizzare criteri di esportazione dinamici per creare e gestire automaticamente i criteri di esportazione. Funziona allo stesso modo per il `ontap-nas-economy` E `ontap-nas-flexgroup` conducenti.

```
---
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 indirizzi IPv6

Questo esempio mostra managementLIF utilizzando 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 Amazon FSx per ONTAP che utilizza volumi SMB

IL smbShare il parametro è obbligatorio per FSx per 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 backend di esempio mostrati di seguito, vengono impostati valori predefiniti specifici per tutti i pool di archiviazione, ad esempio `spaceReserve` a nessuno, `spaceAllocation` a falso, e `encryption` a falso. 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 nel volume di archiviazione al momento del provisioning. Per comodità, gli amministratori di storage possono definire etichette per pool virtuale e raggruppare i volumi in base all'etichetta.

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

Esempio ONTAP NAS

```
---
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: mysqlldb
  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"
```

```
---
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 su StorageClasses

Le seguenti definizioni StorageClass fanno riferimento a [Esempi di backend con pool virtuali](#) . Utilizzando il `parameters.selector` campo, ogni StorageClass richiama quali pool virtuali possono essere utilizzati per ospitare un volume. Il volume avrà gli aspetti definiti nel pool virtuale scelto.

- IL `protection-gold` StorageClass verrà mappato sul primo e sul secondo pool virtuale in `ontap-nas-flexgroup` backend. Queste sono le uniche piscine che offrono una protezione di livello Gold.

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

- IL `protection-not-gold` StorageClass verrà mappato sul terzo e quarto pool virtuale in `ontap-nas-flexgroup` backend. Questi sono gli unici pool che offrono un livello di protezione diverso dall'oro.

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

- IL `app-mysqldb` StorageClass verrà mappato sul quarto pool virtuale nel `ontap-nas` backend. Questo è l'unico pool che offre la configurazione del pool di archiviazione per l'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"

```

- Il protection-silver-creditpoints-20k StorageClass verrà mappato sul terzo pool virtuale in ontap-nas-flexgroup backend. Questo è l'unico pool che offre protezione di livello Silver e 20.000 punti di 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"

```

- IL creditpoints-5k StorageClass verrà mappato sul terzo pool virtuale in ontap-nas backend e il secondo pool virtuale nel ontap-nas-economy backend. Queste sono le uniche offerte di pool con 5000 punti di credito.

```

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

```

Trident deciderà quale pool virtuale selezionare e garantirà che i requisiti di archiviazione siano soddisfatti.

Aggiornamento 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>
```



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 sicuri

Configurazione 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 archiviazione

```
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 classe di archiviazione 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
```



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

Esempio di classe di archiviazione 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 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 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

```

Amazon FSx for NetApp ONTAP

Utilizzare Trident con Amazon FSx for NetApp ONTAP

"Amazon FSx for NetApp ONTAP" è un servizio AWS completamente gestito che consente ai clienti di avviare ed eseguire file system basati sul sistema operativo di storage NetApp ONTAP. FSx for ONTAP ti consente di sfruttare le funzionalità, le prestazioni e le capacità amministrative NetApp che conosci, sfruttando al contempo la semplicità, l'agilità, la sicurezza e la scalabilità dell'archiviazione dei dati su AWS. FSx per ONTAP supporta le funzionalità del file system ONTAP e le API di amministrazione.

Puoi integrare il tuo file system Amazon FSx for NetApp ONTAP con Trident per garantire che i cluster Kubernetes in esecuzione in Amazon Elastic Kubernetes Service (EKS) possano fornire volumi persistenti a blocchi e file supportati da ONTAP.

Un file system è la risorsa principale in Amazon FSx, analogamente a un cluster ONTAP in locale. All'interno di ogni SVM è possibile creare uno o più volumi, ovvero contenitori di dati in cui vengono archiviati i file e le cartelle nel file system. Con Amazon FSx for NetApp ONTAP verrà fornito come file system gestito nel cloud. Il

nuovo tipo di file system si chiama * NetApp ONTAP*.

Utilizzando Trident con Amazon FSx for NetApp ONTAP, puoi garantire che i cluster Kubernetes in esecuzione in Amazon Elastic Kubernetes Service (EKS) possano fornire volumi persistenti di file e blocchi supportati da ONTAP.

Requisiti

Inoltre ["Requisiti Trident"](#) Per integrare FSx per 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) raggiungibile dai nodi worker del cluster.
- Nodi worker preparati per ["NFS o iSCSI"](#) .



Assicurati di seguire i passaggi di preparazione del nodo richiesti per Amazon Linux e Ubuntu ["Immagini della macchina Amazon"](#) (AMI) a seconda del tipo di AMI EKS.

Considerazioni

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



Se si specifica un ruolo IAM per Trident, è possibile omettere di specificare il `apiRegion`, `apiKey`, E `secretKey` campi a Trident in modo esplicito. Per maggiori informazioni, fare riferimento a ["Opzioni di configurazione ed esempi di FSx per ONTAP"](#) .

Utilizzo simultaneo del driver Trident SAN/iSCSI ed EBS-CSI

Se si prevede di utilizzare i 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 usare il `fsxadmin` utente per il tuo file system o il `vsadmin` utente configurato per il tuo SVM.



Trident si aspetta di essere gestito come un `vsadmin` Utente SVM o come utente con un nome diverso che ha lo stesso ruolo. Amazon FSx for NetApp ONTAP ha un `fsxadmin` utente che è una sostituzione limitata ONTAP `admin` utente del cluster. Consigliamo vivamente di utilizzare `vsadmin` con Trident.

- Basato su certificato: Trident comunicherà con l'SVM sul file system FSx utilizzando un certificato installato sull'SVM.

Per i dettagli sull'abilitazione dell'autenticazione, fare riferimento all'autenticazione per il tipo di driver:

- ["Autenticazione NAS ONTAP"](#)
- ["Autenticazione ONTAP SAN"](#)

Immagini macchina Amazon (AMI) testate

Il cluster EKS supporta vari sistemi operativi, ma AWS ha ottimizzato alcune Amazon Machine Image (AMI) per container ed EKS. Le seguenti AMI sono state testate con NetApp Trident 25.02.

AMI	NAS	NAS-economia	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.



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 gli AMI di cui è noto il funzionamento.

Test eseguiti con:

- Versione EKS: 1.32
- Metodo di installazione: Helm 25.06 e come componente aggiuntivo AWS 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: Classe di archiviazione, pvc, pod
- Elimina: pod, pvc (normale, qtree/lun – economy, NAS con backup AWS)

Trova maggiori informazioni

- ["Documentazione Amazon FSx for NetApp ONTAP"](#)
- ["Post del blog su Amazon FSx for NetApp ONTAP"](#)

Crea un ruolo IAM e un segreto AWS

È possibile configurare i pod Kubernetes per accedere alle risorse AWS autenticandosi come ruolo AWS IAM anziché fornire credenziali AWS esplicite.



Per eseguire l'autenticazione tramite un ruolo AWS IAM, è necessario disporre di un cluster Kubernetes distribuito tramite EKS.

Crea il segreto di AWS Secrets Manager

Poiché Trident emetterà API su un server virtuale FSx per gestire l'archiviazione per te, avrà bisogno delle 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

Per funzionare correttamente, Trident necessita anche delle autorizzazioni AWS. Pertanto, è necessario creare una policy che fornisca a Trident le autorizzazioni di cui ha bisogno.

Gli esempi seguenti 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 JSON della policy:

```
{  
  "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 identità Pod o ruolo IAM per l'associazione dell'account di servizio (IRSA)

È possibile configurare un account di servizio Kubernetes affinché assuma un ruolo AWS Identity and Access Management (IAM) con EKS Pod Identity o un ruolo IAM per l'associazione dell'account di servizio (IRSA). Tutti i Pod configurati per utilizzare l'account di servizio possono quindi accedere a qualsiasi servizio AWS per

il quale il ruolo dispone delle autorizzazioni di accesso.

Identità del pod

Le associazioni Amazon EKS Pod Identity consentono di gestire le credenziali per le applicazioni, in modo simile a come i profili delle istanze Amazon EC2 forniscono le 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 maggiori informazioni fare riferimento a ["Configurare l'agente di identità del pod Amazon EKS"](#).

Crea trust-relationship.json:

Creare trust-relationship.json per consentire al Service Principal EKS di assumere questo ruolo per Pod Identity. Quindi crea un ruolo con questa policy di attendibilità:

```
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 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 l'account del servizio 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"  
        }  
      }  
    }  
  ]  
}
```

Aggiornare i seguenti valori nel `trust-relationship.json` file:

- **<account_id>** - ID del tuo account AWS
- **<oidc_provider>** - L'OIDC del cluster EKS. È possibile ottenere `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 (creato nel passaggio precedente) al ruolo utilizzando questo comando:

```
aws iam attach-role-policy --role-name my-role --policy-arn <IAM policy
ARN>
```

Verifica che il fornitore OICD sia associato:

Verifica che il tuo provider OIDC sia associato al tuo cluster. Puoi verificarlo usando 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 l'esempio seguente per creare un ruolo IAM per l'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
```

Installa 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:

- Timone
- Componente aggiuntivo EKS

Se si desidera utilizzare la funzionalità snapshot, installare il componente aggiuntivo CSI Snapshot Controller. Fare riferimento a ["Abilita la funzionalità snapshot per i volumi CSI"](#) per maggiori informazioni.

Installa Trident tramite helm

Identità del pod

1. Aggiungere 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 usare il `helm list` comando per rivedere i dettagli dell'installazione quali nome, spazio dei nomi, grafico, 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. Aggiungere 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 usare il `helm list` comando per rivedere i dettagli dell'installazione quali nome, spazio dei nomi, grafico, 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-
100.2506.0	25.06.0		

Se intendi utilizzare iSCSI, assicurati che iSCSI sia abilitato sul tuo 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 la sicurezza e la stabilità dei cluster Amazon EKS e di ridurre la quantità di lavoro necessaria per installare, configurare e aggiornare i componenti aggiuntivi.

Prerequisiti

Prima di configurare il componente aggiuntivo Trident per AWS EKS, assicurati di disporre di quanto segue:

- Un account cluster Amazon EKS con abbonamento aggiuntivo
- Autorizzazioni AWS per il marketplace AWS:
"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 for NetApp ONTAP esistente

Abilita il componente aggiuntivo Trident per AWS

Console di gestione

1. Aprire la console Amazon EKS all'indirizzo <https://console.aws.amazon.com/eks/home#/clusters>.
2. Nel riquadro di navigazione a sinistra, seleziona **Cluster**.
3. Selezionare il nome del cluster per il quale si desidera configurare il componente aggiuntivo NetApp Trident CSI.
4. Seleziona **Componenti aggiuntivi** e poi **Ottieni altri componenti aggiuntivi**.
5. Per selezionare il componente aggiuntivo, seguire questi passaggi:
 - a. Scorri verso il basso fino alla sezione **Componenti aggiuntivi di AWS Marketplace** e digita **"Trident"** nella casella di ricerca.
 - b. Selezionare la casella di controllo nell'angolo in alto a destra della casella Trident by NetApp.
 - c. Selezionare **Avanti**.
6. Nella pagina delle impostazioni **Configura componenti aggiuntivi selezionati**, procedi come segue:



Salta questi passaggi se utilizzi l'associazione Pod Identity.

- a. Seleziona la **Versione** che desideri utilizzare.
- b. Se si utilizza l'autenticazione IRSA, assicurarsi di impostare i valori di configurazione disponibili nelle impostazioni di configurazione facoltative:
 - Seleziona la **Versione** che desideri utilizzare.
 - Segui lo **schema di configurazione del componente aggiuntivo** e imposta il parametro **configurationValues** nella sezione **Valori di configurazione** sul role-arn creato nel passaggio precedente (il valore deve essere nel seguente formato):

```
{  
  
  "cloudIdentity": "'eks.amazonaws.com/role-arn: <role ARN>'",  
  "cloudProvider": "AWS"  
  
}
```

+

Se si seleziona Sostituisci per il metodo di risoluzione dei conflitti, una o più impostazioni del componente aggiuntivo esistente possono essere sovrascritte con le impostazioni del componente aggiuntivo Amazon EKS. Se non si abilita questa opzione e si verifica un conflitto con le impostazioni esistenti, l'operazione fallisce. È possibile utilizzare il messaggio di errore risultante per risolvere il conflitto. Prima di selezionare questa opzione, assicurati che il componente aggiuntivo Amazon EKS non gestisca impostazioni che devi gestire autonomamente.

7. Selezionare **Avanti**.
8. Nella pagina **Revisiona e aggiungi**, seleziona **Crea**.

Una volta completata l'installazione del componente aggiuntivo, verrà visualizzato il componente aggiuntivo installato.

Interfaccia a riga di comando AWS

1. Crea il add-on.json file:

Per l'identità del pod, utilizzare il seguente formato:

```
{
  "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"
  }
}
```



Sostituire <role ARN> con l'ARN del ruolo creato nel passaggio precedente.

2. Installa il componente aggiuntivo Trident EKS.

```
aws eks create-addon --cli-input-json file://add-on.json
```

eksctl

Il seguente comando di esempio installa il componente aggiuntivo Trident EKS:

```
eksctl create addon --name netapp_trident-operator --cluster
<cluster_name> --force
```

Aggiorna il componente aggiuntivo 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 **Cluster**.
3. Selezionare il nome del cluster per il quale si desidera aggiornare il componente aggiuntivo NetApp Trident CSI.
4. Selezionare la scheda **Componenti aggiuntivi**.
5. Selezionare * Trident by NetApp* e quindi **Modifica**.
6. Nella pagina **Configura Trident di NetApp**, procedere come segue:
 - a. Seleziona la **Versione** che desideri utilizzare.
 - b. Espandi le **Impostazioni di configurazione facoltative** e modificalo secondo necessità.
 - c. Seleziona **Salva modifiche**.

Interfaccia a riga di comando AWS

L'esempio seguente aggiorna il componente aggiuntivo 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 componente aggiuntivo FSxN Trident CSI. Sostituire 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 componente aggiuntivo alla versione restituita in AGGIORNAMENTO DISPONIBILE 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 il `--force` opzione e una qualsiasi delle impostazioni del componente aggiuntivo Amazon EKS è in conflitto con le impostazioni esistenti, l'aggiornamento del componente aggiuntivo Amazon EKS non riesce e viene visualizzato un messaggio di errore che consente di risolvere il conflitto. Prima di specificare questa opzione, assicurati che il componente aggiuntivo Amazon EKS non gestisca le impostazioni che devi gestire, perché tali impostazioni vengono sovrascritte con questa opzione. Per ulteriori informazioni sulle altre opzioni per questa impostazione, vedere ["Componenti aggiuntivi"](#) . Per ulteriori informazioni sulla gestione sul campo di Amazon EKS Kubernetes, vedere ["Gestione del campo Kubernetes"](#) .

Disinstallare/rimuovere il componente aggiuntivo Trident EKS

Per rimuovere un componente aggiuntivo Amazon EKS sono disponibili due opzioni:

- **Conserva il software aggiuntivo sul tuo cluster:** questa opzione rimuove la gestione di qualsiasi impostazione da parte di Amazon EKS. Rimuove inoltre la possibilità per Amazon EKS di notificare gli aggiornamenti e di aggiornare automaticamente il componente aggiuntivo Amazon EKS dopo aver avviato un aggiornamento. Tuttavia, mantiene il software aggiuntivo sul cluster. Questa opzione rende il componente aggiuntivo un'installazione autogestita, anziché un componente aggiuntivo Amazon EKS. Con questa opzione non ci saranno tempi di inattività per il componente aggiuntivo. Conservare il `--preserve` opzione nel comando per preservare il componente aggiuntivo.
- **Rimuovere completamente il software aggiuntivo dal cluster:** NetApp consiglia di rimuovere il componente aggiuntivo Amazon EKS dal cluster solo se nel cluster non sono presenti risorse che dipendono da esso. Rimuovere il `--preserve` opzione dal `delete` comando per rimuovere il componente aggiuntivo.



Se al componente aggiuntivo è associato un account IAM, l'account IAM non viene rimosso.

Console di gestione

1. Aprire la console Amazon EKS all'indirizzo <https://console.aws.amazon.com/eks/home#/clusters>.
2. Nel riquadro di navigazione a sinistra, seleziona **Cluster**.
3. Selezionare il nome del cluster per il quale si desidera rimuovere il componente aggiuntivo NetApp Trident CSI.
4. Selezionare la scheda **Componenti aggiuntivi** e quindi selezionare * Trident by NetApp*.
5. Seleziona **Rimuovi**.
6. Nella finestra di dialogo **Rimuovi conferma netapp_trident-operator**, procedere come segue:
 - a. Se desideri che Amazon EKS interrompa la gestione delle impostazioni per il componente aggiuntivo, seleziona **Conserva nel cluster**. Eseguire questa operazione se si desidera mantenere il software aggiuntivo sul cluster, in modo da poter gestire autonomamente tutte le impostazioni dell'add-on.
 - b. Inserisci **netapp_trident-operator**.
 - c. Seleziona **Rimuovi**.

Interfaccia a riga di comando AWS

Sostituire `my-cluster` con il nome del tuo cluster, quindi esegui 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 il backend di archiviazione

Integrazione dei driver ONTAP SAN e NAS

Per creare un backend di archiviazione, 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 e l'SVM da cui ottenerlo e come autenticarsi. L'esempio seguente mostra come definire l'archiviazione basata su NAS e utilizzare un segreto AWS per archiviare le credenziali nell'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:xxxxxxx: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:xxxxxxx:secret:secret-
name",
      "type": "awsarn"
    }
  }
}
```

Eseguire i seguenti comandi per creare e convalidare la configurazione del backend Trident (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
```

- Convalida 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

Dettagli del driver FSx per ONTAP

È possibile integrare Trident con Amazon FSx for NetApp ONTAP utilizzando i seguenti driver:

- **ontap-san:** Ogni PV fornito è una LUN all'interno del proprio volume Amazon FSx for NetApp ONTAP . Consigliato per l'archiviazione 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 fornito è una LUN con un numero configurabile di LUN per volume Amazon FSx for NetApp ONTAP .
- **ontap-nas-economy:** Ogni PV fornito è un qtree, con un numero configurabile di qtree per volume Amazon FSx for NetApp ONTAP .
- **ontap-nas-flexgroup:** Ogni PV fornito è un volume Amazon FSx for NetApp ONTAP FlexGroup completo.

Per i dettagli del driver, fare riferimento a ["Driver NAS"](#) E ["Driver SAN"](#) .

Una volta creato il file di configurazione, esegui questo comando per crearlo all'interno del tuo EKS:

```
kubectl create -f configuration_file
```

Per verificare lo stato, eseguire questo comando:

```
kubectl get tbc -n trident
```

NAME	BACKEND NAME	BACKEND UUID
PHASE STATUS		
backend-fsx-ontap-nas	backend-fsx-ontap-nas	7a551921-997c-4c37-a1d1-f2f4c87fa629
Bound	Success	

Configurazione avanzata del backend ed esempi

Per le opzioni di configurazione del backend, consultare la seguente tabella:

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 backend di archiviazione	Nome del driver + "_" + dataLIF
managementLIF	Indirizzo IP di un cluster o di un LIF di gestione SVM. È possibile specificare un nome di dominio completo (FQDN). Può essere impostato per utilizzare indirizzi IPv6 se Trident è stato installato utilizzando il flag IPv6. Gli indirizzi IPv6 devono essere definiti tra parentesi quadre, ad esempio [28e8:d9fb:a825:b7bf:69a8:d02f:9e7b:3555]. Se fornisci il fsxFilesystemID sotto il aws campo, non è necessario fornire il managementLIF perché Trident recupera l'SVM managementLIF informazioni da AWS. Quindi, è necessario fornire le credenziali per un utente sotto l'SVM (ad esempio: vsadmin) e l'utente deve avere il vsadmin ruolo.	"10.0.0.1", "[2001:1234:abcd::fefe]"

Parametro	Descrizione	Esempio
dataLIF	Indirizzo IP del protocollo LIF. * Driver ONTAP NAS*: NetApp consiglia di specificare dataLIF. Se non specificato, Trident recupera i dataLIF dall'SVM. È possibile specificare un nome di dominio completo (FQDN) da utilizzare per le operazioni di montaggio NFS, consentendo di creare un DNS round-robin per bilanciare il carico su più dataLIF. Può essere modificato dopo l'impostazione iniziale. Fare riferimento a . * Driver ONTAP SAN*: Non specificare per iSCSI. Trident utilizza ONTAP Selective LUN Map per scoprire gli iSCSI LIF necessari per stabilire una sessione multi-percorso. Se dataLIF è definito in modo esplicito, viene generato un avviso. Può essere impostato per utilizzare indirizzi IPv6 se Trident è stato installato utilizzando 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 dei criteri di esportazione [Booleano]. Utilizzando il autoExportPolicy E autoExportCIDRs opzioni, Trident può gestire automaticamente le policy di esportazione.	false
autoExportCIDRs	Elenco di CIDR per filtrare gli IP dei nodi di Kubernetes quando autoExportPolicy è abilitato. Utilizzando il autoExportPolicy E autoExportCIDRs opzioni, Trident può gestire automaticamente le policy di esportazione.	"["0.0.0.0/0", "::/0"]"
labels	Insieme di etichette arbitrarie formattate in 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. Utilizzato per l'autenticazione basata sulle credenziali. Ad esempio, vsadmin.	
password	Password per connettersi al cluster o alla SVM. Utilizzato per l'autenticazione basata sulle credenziali.	
svm	Macchina virtuale di archiviazione da utilizzare	Derivato se è specificato un managementLIF SVM.
storagePrefix	Prefisso utilizzato durante il provisioning di nuovi volumi nell'SVM. Non può essere modificato dopo la creazione. Per aggiornare questo parametro, sarà necessario creare un nuovo backend.	trident
limitAggregateUsage	Non specificare per Amazon FSx for NetApp ONTAP. Il fornito fsxadmin E vsadmin non contengono le autorizzazioni richieste per recuperare l'utilizzo aggregato e limitarlo tramite Trident.	Non utilizzare.
limitVolumeSize	Il provisioning non riesce se la dimensione del volume richiesto è superiore a questo valore. Limita inoltre la dimensione massima dei volumi che gestisce per qtree e LUN e qtreesPerFlexvol l'opzione consente di personalizzare il numero massimo di qtree per FlexVol volume	"" (non applicato di default)
lunsPerFlexvol	Il numero massimo di LUN per volume Flexvol deve essere compreso nell'intervallo [50, 200]. Solo SAN.	"100"

Parametro	Descrizione	Esempio
debugTraceFlags	Flag di debug da utilizzare durante la risoluzione dei problemi. Esempio, {"api":false, "method":true} Non utilizzare debugTraceFlags a meno che non si stia risolvendo un problema e si necessiti di un dump di registro dettagliato.	null
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 archiviazione, ma se non vengono specificate opzioni di montaggio in una classe di archiviazione, Trident utilizzerà le opzioni di montaggio specificate nel file di configurazione del backend di archiviazione. Se non vengono specificate opzioni di montaggio nella classe di archiviazione o nel file di configurazione, Trident non imposterà alcuna opzione di montaggio su un volume persistente associato.	""
nasType	Configurare la creazione di volumi NFS o SMB. Le opzioni sono <code>nfs</code> , <code>smb</code> , o nullo. Deve essere impostato su <code>smb</code> per volumi SMB. Impostando il valore su <code>null</code> , i volumi NFS vengono impostati di default.	<code>nfs</code>
qtreesPerFlexvol	Numero massimo di Qtree per FlexVol volume, deve essere compreso nell'intervallo [50, 300]	"200"
smbShare	È possibile specificare uno dei seguenti elementi: 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 i backend Amazon FSx for ONTAP .	<code>smb-share</code>

Parametro	Descrizione	Esempio
useREST	Parametro booleano per utilizzare le API REST ONTAP . Quando impostato su <code>true</code> Trident utilizzerà le API REST ONTAP per comunicare con il backend. Questa funzionalità richiede ONTAP 9.11.1 e versioni successive. Inoltre, il ruolo di accesso ONTAP utilizzato deve avere accesso a <code>ontap</code> applicazione. Ciò è soddisfatto dal predefinito <code>vsadmin</code> E <code>cluster-admin</code> ruoli.	<code>false</code>
aws	È possibile specificare quanto segue nel file di configurazione per AWS FSx per ONTAP: - <code>fsxFilesystemID</code> : Specificare l'ID del file system AWS FSx. - <code>apiRegion</code> : Nome della regione API AWS. - <code>apikey</code> : Chiave API AWS. - <code>secretKey</code> : Chiave segreta AWS.	<code>""</code> <code>""</code> <code>""</code>
credentials	Specificare le credenziali FSx SVM da archiviare in AWS Secrets Manager. - <code>name</code> : Amazon Resource Name (ARN) del segreto, che contiene le credenziali di SVM. - <code>type</code> : Impostato su <code>awsarn</code> . Fare riferimento a "Crea un segreto AWS Secrets Manager" per maggiori informazioni.	

Opzioni di configurazione del backend per il provisioning dei volumi

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

Parametro	Descrizione	Predefinito
spaceAllocation	Assegnazione dello spazio per LUN	<code>true</code>
spaceReserve	Modalità di prenotazione dello spazio; "nessuno" (sottile) o "volume" (spesso)	<code>none</code>
snapshotPolicy	Criterio di snapshot da utilizzare	<code>none</code>

Parametro	Descrizione	Predefinito
qosPolicy	Gruppo di criteri QoS da assegnare ai volumi creati. Scegliere tra qosPolicy o adaptiveQosPolicy per ogni pool di archiviazione o backend. Per utilizzare i gruppi di policy QoS con Trident è necessario ONTAP 9.8 o versione successiva. È necessario utilizzare un gruppo di policy QoS non condiviso e assicurarsi che il gruppo di policy venga applicato individualmente a ciascun componente. Un gruppo di policy QoS condiviso impone il limite massimo per la produttività totale di tutti i carichi di lavoro.	""
adaptiveQosPolicy	Gruppo di policy QoS adattivo da assegnare ai volumi creati. Scegliere tra qosPolicy o adaptiveQosPolicy per ogni pool di archiviazione o backend. Non supportato da ontap-nas-economy.	""
snapshotReserve	Percentuale di volume riservata agli 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; l'impostazione predefinita è false . Per utilizzare questa opzione, NVE deve essere concesso in licenza e abilitato sul cluster. Se NAE è abilitato sul backend, qualsiasi volume fornito in Trident sarà abilitato per NAE. Per maggiori informazioni, fare riferimento a: "Come funziona Trident con NVE e NAE" .	false
luksEncryption	Abilita la crittografia LUKS. Fare riferimento a "Utilizzare Linux Unified Key Setup (LUKS)" . Solo SAN.	""
tieringPolicy	Criterio di tiering da utilizzare none	
unixPermissions	Modalità per nuovi volumi. Lasciare vuoto per i volumi SMB.	""

Parametro	Descrizione	Predefinito
securityStyle	Stile di sicurezza per i nuovi volumi. Supporti NFS <code>mixed</code> E <code>unix</code> stili di sicurezza. Supporti SMB <code>mixed</code> E <code>ntfs</code> stili di sicurezza.	L'impostazione predefinita di NFS è <code>unix</code> . L'impostazione predefinita di SMB è <code>ntfs</code> .

Prepararsi al provisioning dei volumi SMB

È possibile eseguire il provisioning dei volumi SMB utilizzando `ontap-nas` autista. Prima di completare [Integrazione dei driver ONTAP SAN e NAS](#) completare i seguenti passaggi.

Prima di iniziare

Prima di poter effettuare il provisioning dei volumi SMB utilizzando `ontap-nas` conducente, devi avere quanto segue.

- Un cluster Kubernetes con un nodo controller Linux e almeno un nodo worker Windows che esegue Windows Server 2019. Trident supporta volumi SMB montati su pod in esecuzione solo su nodi Windows.
- Almeno un segreto Trident contenente le credenziali di Active Directory. Per generare segreto `smbcreds` :

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

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

Passi

1. Crea condivisioni SMB. È possibile creare le condivisioni amministrative SMB in uno dei due modi seguenti: utilizzando ["Console di gestione Microsoft"](#) Snap-in Cartelle condivise o tramite ONTAP CLI. Per creare le condivisioni SMB utilizzando ONTAP CLI:

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

Il `vserver cifs share create` Il 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 all'SVM specificato:

```
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
```



Fare riferimento a "[Crea una condivisione SMB](#)" per maggiori dettagli.

2. Durante la creazione del backend, è necessario configurare quanto segue per specificare i volumi SMB. Per tutte le opzioni di configurazione del backend FSx for ONTAP , fare riferimento a "[Opzioni di configurazione ed esempi di FSx per ONTAP](#)" .

Parametro	Descrizione	Esempio
smbShare	È possibile specificare uno dei seguenti elementi: 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 i backend Amazon FSx for ONTAP .	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 volumi SMB.	ntfs`O `mixed per volumi SMB
unixPermissions	Modalità per nuovi volumi. Deve essere lasciato vuoto per i volumi SMB.	""

Configurare una classe di archiviazione e PVC

Configurare un oggetto StorageClass di Kubernetes e creare la classe di archiviazione per indicare a Trident come effettuare il provisioning dei volumi. Creare un PersistentVolumeClaim (PVC) che utilizzi la StorageClass Kubernetes configurata per richiedere l'accesso al PV. È quindi possibile montare il fotovoltaico su un pod.

Creare una classe di archiviazione

Configurare un oggetto StorageClass di Kubernetes

IL "[Oggetto StorageClass di Kubernetes](#)" L'oggetto identifica Trident come il provisioner utilizzato per quella classe e indica a Trident come effettuare il provisioning di un volume. Utilizzare questo esempio per configurare Storageclass per i 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"
```

Utilizzare questo esempio per configurare Storageclass per i volumi che utilizzano 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"
```

Per effettuare il provisioning dei volumi NFSv3 su AWS Bottlerocket, aggiungere i requisiti mountOptions alla classe di archiviazione:

```
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
```

Fare riferimento a ["Oggetti Kubernetes e Trident"](#) per i dettagli su come le classi di archiviazione interagiscono con PersistentVolumeClaim e parametri per controllare il modo in cui Trident approvvigiona i volumi.

Creare una classe di archiviazione

Passi

1. Questo è un oggetto Kubernetes, quindi usa `kubectl` per crearlo in Kubernetes.

```
kubectl create -f storage-class-ontapnas.yaml
```

2. Ora dovresti vedere una classe di archiviazione **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

Creare il PVC

UN "[*PersistentVolumeClaim*](#)" (PVC) è una richiesta di accesso al PersistentVolume sul cluster.

Il PVC può essere configurato per richiedere una determinata dimensione di archiviazione o modalità di accesso. Utilizzando la StorageClass associata, l'amministratore del cluster può controllare molto più della dimensione e della modalità di accesso di PersistentVolume, ad esempio le prestazioni o il livello di servizio.

Dopo aver creato il PVC, è possibile montare il volume in un pod.

Esempi di manifesti

Manifesti di esempio PersistentVolumeClaim

Questi 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 una StorageClass denominata `basic-csi`.

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: pvc-storage
spec:
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 1Gi
  storageClassName: ontap-gold
```

PVC utilizzando l'esempio iSCSI

Questo esempio mostra un PVC di base per iSCSI con accesso RWO associato a una StorageClass denominata `protection-gold`.

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: pvc-san
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: protection-gold
```

Crea PVC

Passi

1. Creare 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

Fare riferimento a "[Oggetti Kubernetes e Trident](#)" per i dettagli su come le classi di archiviazione interagiscono con `PersistentVolumeClaim` e parametri per controllare il modo in cui Trident approvvigiona i volumi.

Attributi Trident

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

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

Attributo	Tipo	Valori	Offerta	Richiesta	Supportato da
crittografia	bool	vero, falso	Il pool supporta volumi crittografati	Volume con crittografia abilitata	ontap-nas, ontap-nas-economy, ontap-nas-flexgroups, ontap-san
IOPS	interno	intero positivo	Pool è in grado di garantire IOPS in questo intervallo	Volume garantito per questi IOPS	solidfire-san

¹: Non supportato dai sistemi ONTAP Select

Distribuisci l'applicazione di esempio

Una volta creata la classe di accumulo e il PVC, è possibile montare il fotovoltaico su un pod. Questa sezione elenca il comando di esempio e la configurazione per collegare il PV a un pod.

Passi

1. Montare il volume in un pod.

```
kubectl create -f pv-pod.yaml
```

Questi esempi mostrano le configurazioni di base per fissare 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
```



Puoi monitorare i progressi utilizzando `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
```

Configurare 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 la sicurezza e la stabilità dei cluster Amazon EKS e di ridurre la quantità di lavoro necessaria per installare, configurare e aggiornare i componenti aggiuntivi.

Prerequisiti

Prima di configurare il componente aggiuntivo Trident per AWS EKS, assicurati di disporre di quanto segue:

- Un account cluster Amazon EKS con autorizzazioni per lavorare con i componenti aggiuntivi. Fare riferimento a ["Componenti aggiuntivi Amazon EKS"](#).
- Autorizzazioni AWS per il marketplace AWS:
"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 for NetApp ONTAP esistente

Passi

1. Assicurati di creare il ruolo IAM e il segreto AWS per consentire ai pod EKS di accedere alle risorse AWS. Per le istruzioni, vedere ["Crea un ruolo IAM e un segreto AWS"](#).

2. Nel cluster EKS Kubernetes, vai alla scheda **Componenti aggiuntivi**.

The screenshot shows the AWS EKS console interface for a cluster named 'tri-env-eks'. At the top, there are buttons for 'Delete cluster', 'Upgrade version', and 'View dashboard'. A notification banner at the top states: 'End of standard support for Kubernetes version 1.30 is July 28, 2025. On that date, your cluster will enter the extended support period with additional fees. For more information, see the [pricing page](#).' Below this, the 'Cluster info' section displays: Status: Active, Kubernetes version: 1.30, Support period: Standard support until July 28, 2025, and Provider: EKS. It also shows 'Cluster health issues' and 'Upgrade insights', both with a green checkmark and a '0' indicating no issues or insights. A navigation bar includes tabs for Overview, Resources, Compute, Networking, Add-ons (selected with a count of 1), Access, Observability, Update history, and Tags. A message states: 'New versions are available for 1 add-on.' The 'Add-ons (3)' section includes a search bar, filters for 'Any category...', 'Any status', and shows '3 matches'. Buttons for 'View details', 'Edit', 'Remove', and 'Get more add-ons' are present.

3. Vai su **Componenti aggiuntivi di AWS Marketplace** e scegli la categoria *archiviazione*.

The screenshot shows the 'AWS Marketplace add-ons (1)' page. It includes a search bar with the placeholder 'Find add-on'. Under 'Filtering options', there are buttons for 'Any category', 'NetApp, Inc.', 'Any pricing model', and a 'Clear filters' button. A tag for 'NetApp, Inc.' is shown with a close button. The main content area features the 'NetApp Trident' add-on. It includes the NetApp logo, the title 'NetApp Trident', and a description: 'NetApp Trident streamlines Amazon FSx for NetApp ONTAP storage management in Kubernetes to let your developers and administrators focus on application deployment. FSx for ONTAP flexibility, scalability, and integration capabilities make it the ideal choice for organizations seeking efficient containerized storage workflows. [Product details](#)'. Below the description is a 'Standard Contract' button. A table-like layout shows: Category: storage, Listed by: [NetApp, Inc.](#), Supported versions: 1.31, 1.30, 1.29, 1.28, 1.27, 1.26, 1.25, 1.24, 1.23, and Pricing starting at: [View pricing details](#). At the bottom right, there are 'Cancel' and 'Next' buttons.

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 del componente aggiuntivo.

Configure selected add-ons settings

Configure the add-ons for your cluster by selecting settings.


NetApp Trident

Remove add-on


Listed by


Category

Status




storage

 Ready to install

 You're subscribed to this software

You can view the terms and pricing details for this product or choose another offer if one is available.


View subscription



Version

Select the version for this add-on.

v25.6.0-eksbuild.1



Optional configuration settings

Cancel

Previous

Next

6. Configurare le impostazioni aggiuntive richieste.

Review and add

Step 1: Select add-ons

[Edit](#)

Selected add-ons (1)

 Find add-on



 1 

Add-on name	Type	Status
netapp_trident-operator	storage	 Ready to install

Step 2: Configure selected add-ons settings



[Edit](#)


Selected add-ons version (1)

 1 

Add-on name	Version	IAM role for service account (IRSA)
netapp_trident-operator	v24.10.0-eksbuild.1	Not set

EKS Pod Identity (0)

 1 

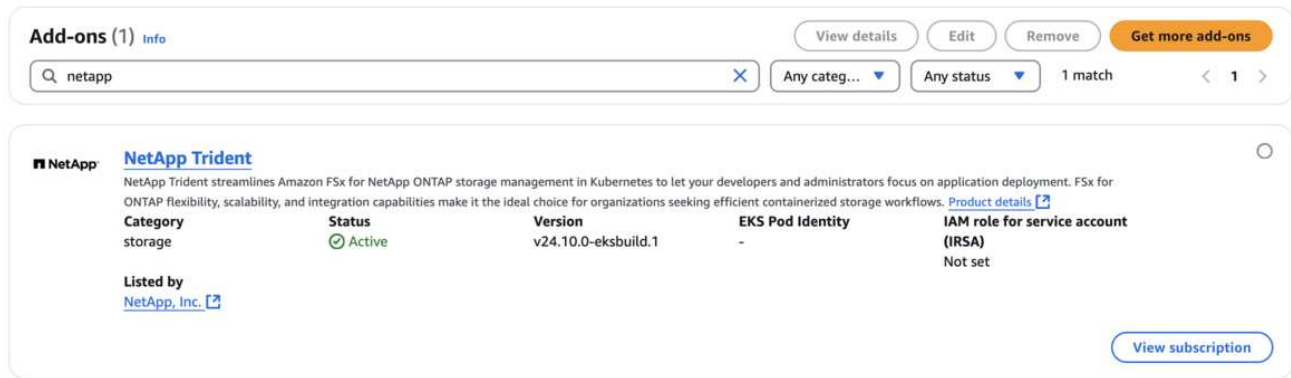
Add-on name	IAM role 	Service account
No Pod Identity associations None of the selected add-on(s) have Pod Identity associations.		

[Cancel](#)[Previous](#)[Create](#)

7. Se si utilizza IRSA (ruoli IAM per l'account di servizio), fare riferimento ai passaggi di configurazione aggiuntivi ["Qui"](#).

8. Seleziona **Crea**.

9. Verificare che lo stato del componente aggiuntivo sia *Attivo*.



10. Eseguire il seguente comando per verificare che Trident sia installato correttamente sul cluster:

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

11. Continuare l'installazione e configurare il backend di archiviazione. Per informazioni, vedere ["Configurare il backend di archiviazione"](#).

Installa/disinstalla il componente aggiuntivo Trident EKS tramite CLI

Installare il componente aggiuntivo NetApp Trident EKS tramite 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)
```

Disinstallare il componente aggiuntivo NetApp Trident EKS tramite 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 di archiviazione. Indica a Trident come comunicare con quel sistema di archiviazione e come Trident deve effettuare il provisioning dei volumi da esso. Dopo aver installato Trident, il passo successivo è creare un backend. IL `TridentBackendConfig` La definizione di risorse personalizzate (CRD) consente di creare e gestire i backend Trident direttamente tramite l'interfaccia Kubernetes. Puoi farlo usando `kubectl` o lo strumento CLI equivalente per la tua distribuzione Kubernetes.

`TridentBackendConfig`

`TridentBackendConfig` (`tbc`, `tbconfig`, `tbackendconfig`) è un frontend, un CRD con namespace che consente di gestire i backend Trident utilizzando `kubectl`. Gli amministratori di Kubernetes e storage possono ora creare e gestire i backend direttamente tramite la CLI di Kubernetes senza richiedere un'utilità

della riga di comando dedicata(`tridentctl`).

Dopo la creazione di un `TridentBackendConfig` oggetto, accade quanto segue:

- Trident crea automaticamente un backend in base alla configurazione fornita. Questo è rappresentato internamente come un `TridentBackend (tbe , tridentbackend)` CR.
- IL `TridentBackendConfig` è unicamente legato a un `TridentBackend` che è stato creato da Trident.

Ogni `TridentBackendConfig` mantiene una mappatura uno a uno con un `TridentBackend` La prima è l'interfaccia fornita all'utente per progettare e configurare i backend; la seconda è il modo in cui Trident rappresenta l'oggetto backend effettivo.



`TridentBackend` I CR vengono creati automaticamente da Trident. **Non dovresti** modificarli. Se vuoi apportare aggiornamenti ai backend, fallo modificando il `TridentBackendConfig` oggetto.

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
```

Puoi anche dare un'occhiata agli esempi nel "[trident-installer](#)" directory per configurazioni di esempio per la piattaforma/servizio di archiviazione desiderato.

IL spec accetta parametri di configurazione specifici del backend. In questo esempio, il backend utilizza il `ontap-san` driver di archiviazione e utilizza i parametri di configurazione qui tabulati. Per l'elenco delle opzioni di configurazione per il driver di archiviazione desiderato, fare riferimento a "[informazioni sulla configurazione del backend per il driver di archiviazione](#)".

IL spec la sezione include anche `credentials` E `deletionPolicy` campi, che sono stati recentemente introdotti nel `TridentBackendConfig` CR:

- `credentials`: Questo parametro è un campo obbligatorio e contiene le credenziali utilizzate per l'autenticazione con il sistema/servizio di archiviazione. Viene impostato su un segreto Kubernetes creato dall'utente. Le credenziali non possono essere trasmesse in testo normale e ciò causerebbe un errore.
- `deletionPolicy`: Questo campo definisce cosa dovrebbe accadere quando il `TridentBackendConfig` viene eliminato. Può assumere uno dei due valori possibili:

- **delete**: Ciò comporta l'eliminazione di entrambi `TridentBackendConfig` CR e il 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`. Impostazione della politica di eliminazione su `retain` consente agli utenti di eseguire il downgrade a una versione precedente (precedente alla 21.04) e di mantenere i backend creati. Il valore di questo campo può essere aggiornato dopo un `TridentBackendConfig` è creato.



Il nome di un backend viene impostato utilizzando `spec.backendName`. Se non specificato, il nome del backend viene impostato sul nome del `TridentBackendConfig` oggetto (metadati.nome). Si consiglia di impostare esplicitamente i nomi del backend utilizzando `spec.backendName`.



Backend creati con `tridentctl` non hanno un associato `TridentBackendConfig` oggetto. Puoi scegliere di gestire tali backend con `kubectl` creando un `TridentBackendConfig` CR. Bisogna fare attenzione a specificare parametri di configurazione identici (come `spec.backendName`, `spec.storagePrefix`, `spec.storageDriverName`, e così via). Trident collegherà automaticamente il nuovo creato `TridentBackendConfig` con il backend preesistente.

Panoramica dei passaggi

Per creare un nuovo backend utilizzando `kubectl`, dovresti fare quanto segue:

1. Crea un "[Segreto di Kubernetes](#)" Il segreto contiene le credenziali di cui Trident ha bisogno per comunicare con il cluster/servizio di archiviazione.
2. Crea un `TridentBackendConfig` oggetto. Contiene informazioni specifiche sul cluster/servizio di archiviazione 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: creare un segreto Kubernetes

Crea un segreto che contenga le credenziali di accesso per il backend. Questa caratteristica è unica per ogni servizio/piattaforma di archiviazione. 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 Segreto per ogni piattaforma di archiviazione:

Descrizione dei campi segreti della piattaforma di archiviazione	Segreto	Descrizione dei campi
Azure NetApp Files	ID cliente	L'ID client da una registrazione dell'app
Cloud Volumes Service per GCP	id_chiave_privata	ID della chiave privata. Parte della chiave API per l'account di servizio GCP con ruolo di amministratore CVS
Cloud Volumes Service per GCP	chiave privata	Chiave privata. Parte della chiave API per l'account di servizio GCP con ruolo di amministratore CVS
Elemento (NetApp HCI/ SolidFire)	Punto finale	MVIP per il cluster SolidFire con credenziali tenant
ONTAP	nome utente	Nome utente per connettersi al cluster/SVM. Utilizzato per l'autenticazione basata sulle credenziali
ONTAP	password	Password per connettersi al cluster/SVM. Utilizzato per l'autenticazione basata sulle credenziali
ONTAP	chiave privata del cliente	Valore codificato in Base64 della chiave privata del client. Utilizzato per l'autenticazione basata su certificato

Descrizione dei campi segreti della piattaforma di archiviazione	Segreto	Descrizione dei campi
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 del target CHAP. Obbligatorio se useCHAP=true. Per ontap-san E ontap-san-economy

Il segreto creato in questo passaggio verrà referenziato nel `spec.credentials` campo del `TridentBackendConfig` oggetto che viene creato nel passaggio successivo.

Passaggio 2: creare il `TridentBackendConfig` CR

Ora sei pronto per creare il tuo `TridentBackendConfig` CR. In questo esempio, un backend che utilizza il `ontap-san` il driver viene creato utilizzando il `TridentBackendConfig` oggetto 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 il TridentBackendConfig CR, puoi verificare lo stato. Vedere il seguente esempio:

```

kubectl -n trident get tbc backend-tbc-ontap-san

```

NAME	BACKEND NAME	BACKEND UUID
backend-tbc-ontap-san	ontap-san-backend	8d24fce7-6f60-4d4a-8ef6-bab2699e6ab8
Bound	Success	

Un backend è stato creato con successo e associato a TridentBackendConfig CR.

La fase può assumere uno dei seguenti valori:

- **Bound:** IL TridentBackendConfig CR è associato a un backend e quel backend contiene configRef impostato su TridentBackendConfig Uid di CR.
- **Unbound:** Rappresentato utilizzando "". IL TridentBackendConfig l'oggetto non è vincolato a un backend. Tutti i nuovi creati TridentBackendConfig I CR si trovano in questa fase per impostazione predefinita. Dopo il cambio di fase, non è più possibile tornare allo stato Unbound.
- **Deleting:** IL TridentBackendConfig CR deletionPolicy era impostato per essere eliminato. Quando il TridentBackendConfig Una volta eliminato il CR, passa allo stato Eliminazione.
 - Se non esistono richieste di volume persistenti (PVC) sul backend, l'eliminazione TridentBackendConfig comporterà l'eliminazione del backend Trident e anche del TridentBackendConfig CR.
 - Se sul backend sono presenti uno o più PVC, questo passa allo stato di eliminazione. IL TridentBackendConfig Successivamente anche CR entra nella 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 ha ancora un riferimento al backend eliminato. IL TridentBackendConfig CR può ancora 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 Manca il CRD. Potrebbe essere necessario un intervento.

A questo punto, il backend è stato creato correttamente! Ci sono diverse operazioni che possono essere gestite ulteriormente, come ad esempio ["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 anche possibile ottenere 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 attivato dall'utente (ad esempio, l'utente ha modificato qualcosa in spec) o attivato da Trident (ad esempio, durante i riavvii Trident). Può essere un successo o un fallimento. phase rappresenta lo stato della relazione tra TridentBackendConfig CR e backend. Nell'esempio sopra, phase ha il valore Bound, il che significa che il TridentBackendConfig CR è associato al backend.

Puoi eseguire il `kubectl -n trident describe tbc <tbc-cr-name>` comando per ottenere i dettagli dei registri eventi.



Non è possibile aggiornare o eliminare un backend che contiene un associato TridentBackendConfig oggetto utilizzando `tridentctl`. Per comprendere i passaggi coinvolti nel passaggio tra `tridentctl` E TridentBackendConfig, "[vedi qui](#)".

Gestire i backend

Eseguire la gestione del backend con kubectl

Scopri come eseguire operazioni di gestione del backend utilizzando `kubectl`.

Elimina un backend

Eliminando un `TridentBackendConfig`, istruisci Trident a eliminare/conservare i backend (in base a `deletionPolicy`). Per eliminare un backend, assicurati che `deletionPolicy` è impostato per l'eliminazione. Per eliminare solo il `TridentBackendConfig`, assicurarsi che `deletionPolicy` è impostato per mantenere. Ciò garantisce che il backend sia ancora presente e possa essere gestito utilizzando `tridentctl`.

Esegui il seguente comando:

```
kubectl delete tbc <tbc-name> -n trident
```

Trident non elimina i segreti di Kubernetes che erano in uso da `TridentBackendConfig`. L'utente Kubernetes è responsabile della pulizia dei segreti. Bisogna fare attenzione quando si eliminano i segreti. Dovresti eliminare i segreti solo se non sono utilizzati dai backend.

Visualizza i backend esistenti

Esegui il seguente comando:

```
kubectl get tbc -n trident
```

Puoi anche correre `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 di archiviazione sono cambiate. Per aggiornare le credenziali, il segreto Kubernetes utilizzato in `TridentBackendConfig` l'oggetto deve essere aggiornato. Trident aggiornerà automaticamente il backend con le credenziali più recenti fornite. Eseguire il seguente comando per aggiornare il segreto di Kubernetes:

```
kubectl apply -f <updated-secret-file.yaml> -n trident
```

- È necessario aggiornare i parametri (ad esempio il nome ONTAP SVM utilizzato).
 - Puoi aggiornare `TridentBackendConfig` oggetti direttamente tramite Kubernetes utilizzando il seguente comando:

```
kubectl apply -f <updated-backend-file.yaml>
```

- In alternativa, è possibile apportare modifiche a quelle esistenti `TridentBackendConfig` CR utilizzando il seguente comando:

```
kubectl edit tbc <tbc-name> -n trident
```



- Se un aggiornamento del backend fallisce, il backend continua a mantenere la sua ultima configurazione nota. È possibile visualizzare i registri per determinare 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, è possibile eseguire nuovamente il comando di aggiornamento.

Eseguire la gestione del backend con `tridentctl`

Scopri come eseguire operazioni di gestione del backend utilizzando `tridentctl`.

Crea un backend

Dopo aver creato un ["file di configurazione del backend"](#), eseguire il seguente comando:

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

Se la creazione del backend fallisce, significa che c'è stato un errore nella configurazione del backend. È possibile visualizzare i registri per determinare 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 comando `create` comando di nuovo.

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
```



Se Trident ha eseguito il provisioning di volumi e snapshot da questo backend che esistono ancora, l'eliminazione del backend impedisce che vengano eseguiti il provisioning di nuovi volumi. Il backend continuerà a esistere nello stato "Eliminazione".

Visualizza i backend esistenti

Per visualizzare i backend noti a Trident , 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, c'è stato un problema con la configurazione del backend oppure hai tentato un aggiornamento non valido. È possibile visualizzare i registri per determinare 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 comando update comando di nuovo.

Identificare le classi di archiviazione che utilizzano un backend

Questo è un esempio del tipo di domande a cui puoi rispondere con il JSON che `tridentctl` output per oggetti backend. Questo utilizza il `jq` utilità che devi installare.

```
tridentctl get backend -o json | jq '[.items[] | {backend: .name, storageClasses: [.storage[].storageClasses]|unique}]'
```

Questo vale anche per i backend creati utilizzando `TridentBackendConfig`.

Spostarsi tra le opzioni di gestione del backend

Scopri i diversi modi di gestire i backend in Trident.

Opzioni per la gestione dei backend

Con l'introduzione di `TridentBackendConfig`, gli amministratori ora hanno due modi unici per gestire i backend. Ciò solleva le seguenti domande:

- I backend possono essere creati utilizzando `tridentctl` essere gestito con `TridentBackendConfig`?
- I backend possono essere creati utilizzando `TridentBackendConfig` essere gestito utilizzando `tridentctl`?

Maneggio `tridentctl` backend utilizzando `TridentBackendConfig`

Questa sezione illustra i passaggi necessari per gestire i backend creati utilizzando `tridentctl` direttamente tramite l'interfaccia Kubernetes creando `TridentBackendConfig` oggetti.

Ciò si applicherà ai seguenti scenari:

- Backend preesistenti, che non hanno un `TridentBackendConfig` perché sono stati creati con `tridentctl`.
- Nuovi backend creati con `tridentctl`, mentre altri `TridentBackendConfig` gli oggetti esistono.

In entrambi gli scenari, i backend continueranno a essere presenti, con Trident che pianifica i volumi e opera su di essi. In questo caso gli amministratori hanno due possibilità:

- Continua a usare `tridentctl` per gestire i backend creati utilizzandolo.
- Associa i backend creati utilizzando `tridentctl` a un nuovo `TridentBackendConfig` oggetto. Ciò significherebbe che i backend saranno gestiti utilizzando `kubectl` e non `tridentctl`.

Per gestire un backend preesistente utilizzando `kubectl`, dovrai creare un `TridentBackendConfig` che si collega al backend esistente. Ecco una panoramica di come funziona:

1. Crea un segreto Kubernetes. Il segreto contiene le credenziali di cui Trident ha bisogno per comunicare con il cluster/servizio di archiviazione.
2. Crea un `TridentBackendConfig` oggetto. Contiene informazioni specifiche sul cluster/servizio di archiviazione e fa riferimento al segreto creato nel passaggio precedente. Bisogna fare attenzione a specificare parametri di configurazione identici (come `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 collega a un backend esistente, sarà necessario ottenere la configurazione del backend. In questo esempio, supponiamo che sia stato creato un backend 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: creare un segreto Kubernetes

Crea un segreto che contenga 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: creare un TridentBackendConfig CR

Il passo successivo è creare un `TridentBackendConfig` CR che si collegherà automaticamente al 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 del backend originale.
- Le credenziali vengono fornite tramite un segreto Kubernetes e non in testo normale.

In questo caso, il `TridentBackendConfig` sarà simile a questo:

```
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: mysqlldb
      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 il TridentBackendConfig è stato creato, 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
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
ontap-nas-backend	ontap-nas	52f2eb10-e4c6-4160-99fc-96b3be5ab5d7
online	25	

Il backend sarà ora completamente gestito tramite tbc-ontap-nas-backend TridentBackendConfig oggetto.

Maneggio TridentBackendConfig **backend utilizzando** tridentctl

`tridentctl` può essere utilizzato per elencare i backend creati utilizzando `TridentBackendConfig`. Inoltre, gli amministratori possono anche scegliere di gestire completamente tali backend tramite `tridentctl` eliminando `TridentBackendConfig` e assicurandosi `spec.deletionPolicy` è 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
backend-tbc-ontap-san	ontap-san-backend	81abcb27-ea63-49bb-b606-0a5315ac5f82

```
tridentctl get backend ontap-san-backend -n trident
```

NAME	STORAGE DRIVER	UUID
ontap-san-backend	ontap-san	81abcb27-ea63-49bb-b606-0a5315ac5f82

Dall'output si vede che `TridentBackendConfig` è stato creato correttamente ed è associato a un backend [osservare l'UUID del backend].

Passaggio 1: conferma `deletionPolicy` è impostato su `retain`

Diamo un'occhiata al valore di `deletionPolicy`. Questo deve essere impostato su `retain`. Ciò 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
backend-tbc-ontap-san	ontap-san-backend	81abcb27-ea63-49bb-b606-0a5315ac5f82

```
# 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
backend-tbc-ontap-san	ontap-san-backend	81abcb27-ea63-49bb-b606-0a5315ac5f82



Non procedere al passaggio successivo a meno che `deletionPolicy` è impostato su `retain`

Passaggio 2: Eliminare il `TridentBackendConfig` CR

Il passaggio finale è quello di eliminare il `TridentBackendConfig` CR. Dopo aver confermato il `deletionPolicy` è impostato su `retain`, puoi 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 la cancellazione del `TridentBackendConfig` oggetto, Trident lo rimuove semplicemente senza eliminare effettivamente il backend stesso.

Creare e gestire classi di archiviazione

Creare una classe di archiviazione

Configurare un oggetto `StorageClass` di Kubernetes e creare la classe di archiviazione per indicare a Trident come effettuare il provisioning dei volumi.

Configurare un oggetto `StorageClass` di Kubernetes

IL "[Oggetto `StorageClass` di Kubernetes](#)" identifica Trident come il provisioner utilizzato per quella classe e indica a Trident come effettuare il provisioning di un volume. Per 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

```

Fare riferimento a ["Oggetti Kubernetes e Trident"](#) per i dettagli su come le classi di archiviazione interagiscono con PersistentVolumeClaim e parametri per controllare il modo in cui Trident approvvigiona i volumi.

Creare una classe di archiviazione

Dopo aver creato l'oggetto StorageClass, è possibile creare la classe di archiviazione. [Campioni di classe di archiviazione](#) fornisce alcuni esempi di base che puoi utilizzare o modificare.

Passi

1. Questo è 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 archiviazione **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"
        ]
      }
    }
  ]
}

```

Campioni di classe di archiviazione

Trident fornisce ["definizioni di classi di archiviazione semplici per backend specifici"](#) .

In alternativa, puoi modificare `sample-input/storage-class-csi.yaml.templ` file fornito con il programma di installazione e sostituisci `BACKEND_TYPE` con il nome del driver di archiviazione.

```
./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
```

Gestire le classi di archiviazione

È possibile visualizzare le classi di archiviazione esistenti, impostare una classe di archiviazione predefinita, identificare il backend della classe di archiviazione ed eliminare le classi di archiviazione.

Visualizza le classi di archiviazione esistenti

- Per visualizzare le classi di archiviazione Kubernetes esistenti, eseguire il seguente comando:

```
kubectl get storageclass
```

- Per visualizzare i dettagli della classe di archiviazione Kubernetes, eseguire il seguente comando:

```
kubectl get storageclass <storage-class> -o json
```

- Per visualizzare le classi di archiviazione 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 classe di archiviazione predefinita

Kubernetes 1.6 ha aggiunto la possibilità di impostare una classe di archiviazione predefinita. Questa è la classe di archiviazione che verrà utilizzata per fornire un volume persistente se un utente non ne specifica uno in una richiesta di volume persistente (PVC).

- Definisci una classe di archiviazione predefinita impostando l'annotazione `storageclass.kubernetes.io/is-default-class` su `true` nella definizione della classe di archiviazione. Secondo la specifica, qualsiasi altro valore o l'assenza dell'annotazione viene interpretato come falso.
- È possibile configurare una classe di archiviazione esistente come classe di archiviazione predefinita utilizzando il seguente comando:

```
kubectl patch storageclass <storage-class-name> -p '{"metadata":  
{"annotations":{"storageclass.kubernetes.io/is-default-class":"true"}}}'
```

- Allo stesso modo, è possibile rimuovere l'annotazione della classe di archiviazione 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 pacchetto di installazione Trident che includono questa annotazione.



Nel cluster dovrebbe essere presente una sola classe di archiviazione predefinita alla volta. Tecnicamente Kubernetes non impedisce di averne più di una, ma si comporterà come se non ci fosse alcuna classe di archiviazione predefinita.

Identificare il backend per una classe di archiviazione

Questo è un esempio del tipo di domande a cui puoi rispondere con il JSON che `tridentctl` output per gli oggetti backend Trident. Questo utilizza il `jq` utilità, che potrebbe essere necessario installare prima.

```
tridentctl get storageclass -o json | jq '[.items[] | {storageClass:  
.Config.name, backends: [.storage]|unique}]'
```

Elimina una classe di archiviazione

Per eliminare una classe di archiviazione da Kubernetes, esegui il seguente comando:

```
kubectl delete storageclass <storage-class>
```

`<storage-class>` dovrebbe essere sostituito con la tua classe di archiviazione.

Tutti i volumi persistenti creati tramite questa classe di archiviazione rimarranno intatti e Trident continuerà a gestirli.



Trident impone un vuoto `fsType` per i volumi che crea. Per i backend iSCSI, si consiglia di applicare `parameters.fsType` nella `StorageClass`. Dovresti eliminare le `StorageClass` esistenti e ricrearle con `parameters.fsType` specificato.

Fornire e gestire i volumi

Fornire un volume

Creare un `PersistentVolumeClaim` (PVC) che utilizzi la `StorageClass` Kubernetes configurata per richiedere l'accesso al PV. È quindi possibile montare il fotovoltaico su un pod.

Panoramica

UN "*PersistentVolumeClaim*" (PVC) è una richiesta di accesso al `PersistentVolume` sul cluster.

Il PVC può essere configurato per richiedere una determinata dimensione di archiviazione o modalità di accesso. Utilizzando la `StorageClass` associata, l'amministratore del cluster può controllare molto più della dimensione e della modalità di accesso di `PersistentVolume`, ad esempio le prestazioni o il livello di servizio.

Dopo aver creato il PVC, è possibile montare il volume in un pod.

Creare il PVC

Passi

1. Creare 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. Montare il volume in un pod.

```
kubectl create -f pv-pod.yaml
```



Puoi monitorare i progressi utilizzando `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 manifesti

Manifesti di esempio PersistentVolumeClaim

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 una StorageClass denominata `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 una StorageClass denominata `protection-gold`.

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: pvc-san-nvme
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 300Mi
  storageClassName: protection-gold
```

Campioni di manifesti del pod

Questi esempi mostrano le configurazioni di base per fissare 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 NVMe/TCP di base

```
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
```

Fare riferimento a ["Oggetti Kubernetes e Trident"](#) per i dettagli su come le classi di archiviazione interagiscono con PersistentVolumeClaim e parametri per controllare il modo in cui Trident approvvigiona i volumi.

Espandi i volumi

Trident offre agli utenti di Kubernetes la possibilità di espandere i propri volumi dopo averli 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.



L'espansione del volume iSCSI è supportata da `ontap-san`, `ontap-san-economy`, `solidfire-san` driver e richiede Kubernetes 1.16 e versioni successive.

Passaggio 1: configurare StorageClass per supportare l'espansione del volume

Modificare la definizione StorageClass per impostare `allowVolumeExpansion` campo a `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 una StorageClass già esistente, modificarla per includere `allowVolumeExpansion` parametro.

Passaggio 2: creare un PVC con la StorageClass creata

Modifica la definizione PVC e aggiornala `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 file system.
- Quando si tenta di ridimensionare un PV non collegato, Trident espande il volume sul backend di archiviazione. Dopo che il PVC è stato associato a un pod, Trident esegue una nuova scansione del dispositivo e ridimensiona il file system. Kubernetes aggiorna quindi la dimensione del PVC una volta completata correttamente l'operazione di espansione.

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: Espandi il PV

Per ridimensionare il PV creato da 1Gi a 2Gi, modificare la definizione PVC e aggiornare `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: convalidare l'espansione

È possibile verificare che l'espansione abbia funzionato correttamente controllando le dimensioni del PVC, del PV e del volume Trident :

```
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     |
+-----+-----+-----+-----+
+-----+-----+-----+-----+
```

Espandi un volume FC

È possibile espandere un FC Persistent Volume (PV) utilizzando il provisioner CSI.



L'espansione del volume FC è supportata da ontap-san driver e richiede Kubernetes 1.16 e versioni successive.

Passaggio 1: configurare StorageClass per supportare l'espansione del volume

Modificare la definizione StorageClass per impostare allowVolumeExpansion campo a 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 una StorageClass già esistente, modificarla per includere `allowVolumeExpansion` parametro.

Passaggio 2: creare un PVC con la StorageClass creata

Modifica la definizione PVC e aggiornala `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 FC PV:

- 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 file system.
- Quando si tenta di ridimensionare un PV non collegato, Trident espande il volume sul backend di archiviazione. Dopo che il PVC è stato associato a un pod, Trident esegue una nuova scansione del dispositivo e ridimensiona il file system. Kubernetes aggiorna quindi la dimensione del PVC una volta completata correttamente l'operazione di espansione.

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: Espandi il PV

Per ridimensionare il PV creato da 1Gi a 2Gi, modificare la definizione PVC e aggiornare `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: convalidare l'espansione

È possibile verificare che l'espansione abbia funzionato correttamente controllando le dimensioni del PVC, del PV e del volume Trident :

```
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`, `gcp-cvs`, `E azure-netapp-files` backend.

Passaggio 1: configurare StorageClass per supportare l'espansione del volume

Per ridimensionare un PV NFS, l'amministratore deve prima configurare la classe di archiviazione per consentire l'espansione del volume impostando `allowVolumeExpansion` campo a `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: creare un PVC con la StorageClass creata

```
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: Espandi il PV

Per ridimensionare il PV da 20 MiB appena creato a 1 GiB, modificare il PVC e impostare `spec.resources.requests.storage` a 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: convalidare 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 |
+-----+-----+-----+-----+
|          BACKEND UUID  | STATE | MANAGED  |
+-----+-----+-----+-----+
| pvc-08f3d561-b199-11e9-8d9f-5254004dfdb7 | 1.0 GiB | ontapnas      |
| file          | c5a6f6a4-b052-423b-80d4-8fb491a14a22 | online | true      |
+-----+-----+-----+-----+
+-----+-----+-----+-----+
```

Volumi di importazione

È possibile importare volumi di archiviazione esistenti come PV Kubernetes utilizzando `tridentctl import`.

Panoramica e considerazioni

È possibile importare un volume in Trident per:

- Containerizzare un'applicazione e riutilizzare il suo set di dati esistente
- Utilizzare un clone di un set di dati per un'applicazione effimera
- Ricostruisci un cluster Kubernetes non riuscito
- Migrare i dati dell'applicazione durante il ripristino di emergenza

Considerazioni

Prima di importare un volume, esaminare le seguenti considerazioni.

- Trident può importare solo volumi ONTAP di tipo RW (lettura-scrittura). I volumi di tipo DP (protezione dati) sono volumi di destinazione SnapMirror. Prima di importare il volume in Trident, è necessario interrompere la relazione mirror.

- Sugeriamo di importare volumi senza connessioni attive. Per importare un volume utilizzato attivamente, clonare il volume e quindi eseguire l'importazione.



Ciò 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. Ciò può causare il danneggiamento dei dati.

- Anche se `StorageClass` deve 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, non è richiesta alcuna selezione del pool durante l'importazione. Pertanto, l'importazione non fallirà anche se il volume esiste su un backend o 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 ha associato correttamente PVC e PV, la policy di recupero viene aggiornata per corrispondere alla policy di recupero della classe di archiviazione.
 - Se la politica di recupero della classe di archiviazione è `delete`, il volume di archiviazione verrà eliminato quando il PV verrà eliminato.
- Per impostazione predefinita, Trident gestisce il PVC e rinomina il FlexVol volume e il LUN sul backend. Puoi passare il `--no-manage` flag per importare un volume non gestito. Se usi `--no-manage`, Trident non esegue alcuna operazione aggiuntiva sul PVC o sul PV per l'intero ciclo di vita degli oggetti. Il volume di archiviazione non viene eliminato quando si elimina il PV e anche altre operazioni come la clonazione del volume e il ridimensionamento del volume vengono ignorate.



Questa opzione è utile se si desidera utilizzare Kubernetes per carichi di lavoro containerizzati ma si desidera gestire il ciclo di vita del volume di archiviazione al di fuori di Kubernetes.

- Viene aggiunta un'annotazione al PVC e al PV 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.

Importa un volume

Puoi usare `tridentctl import` per importare un volume.

Passi

1. Creare il file Persistent Volume Claim (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 tua definizione di 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
```



Non includere parametri aggiuntivi come il nome PV o la dimensione del volume. Ciò può causare il fallimento del comando di importazione.

2. Utilizzare il `tridentctl import volume` comando per specificare il nome del backend Trident contenente il volume e il nome che identifica in modo univoco il volume nello storage (ad esempio: ONTAP FlexVol, Element Volume, percorso Cloud Volumes Service). IL `-f` L'argomento è necessario per specificare il percorso del file PVC.

```
tridentctl import volume <backendName> <volumeName> -f <path-to-pvc-
file>
```

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 `ontap-nas` E `ontap-nas-flexgroup` conducenti.



- Trident non supporta l'importazione di volumi utilizzando `ontap-nas-economy` autista.
- IL `ontap-nas` E `ontap-nas-flexgroup` i driver non consentono nomi di volume duplicati.

Ogni volume creato con il `ontap-nas` il driver è un FlexVol volume sul cluster ONTAP . Importazione di volumi FlexVol con `ontap-nas` il driver funziona allo stesso modo. Un volume FlexVol già esistente su un cluster ONTAP può essere importato come `ontap-nas` PVC. Allo stesso modo, i volumi FlexGroup possono essere importati come `ontap-nas-flexgroup` PVC.

Esempi di ONTAP NAS

Di seguito è riportato un esempio di importazione di un volume gestito e di un volume non gestito.

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
pvc-bf5ad463-afbb-11e9-8d9f-5254004dfdb7	1.0 GiB	standard
file	online	true

Volume non gestito

Quando si utilizza il `--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
pvc-df07d542-afbc-11e9-8d9f-5254004dfdb7	1.0 GiB	standard
file	online	false

ONTAP SAN

Trident supporta l'importazione di volumi utilizzando `ontap-san` (iSCSI, NVMe/TCP e FC) e `ontap-san-economy` conducenti.

Trident può importare volumi ONTAP SAN FlexVol che contengono un singolo LUN. Ciò è coerente con il `ontap-san` driver, che crea un FlexVol volume per ogni PVC e un LUN all'interno del FlexVol volume. Trident importa il FlexVol volume e lo associa alla definizione PVC. Trident può importare `ontap-san-economy`

volumi che contengono più LUN.

Esempi ONTAP SAN

Di seguito è riportato un esempio di importazione di un volume gestito e di un volume non gestito.

Volume gestito

Per i volumi gestiti, Trident rinomina il FlexVol volume in `pvc-<uuid>` formato e LUN all'interno del FlexVol volume per `lun0`.

L'esempio seguente 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 LUNS mappati su igroup che condividono un IQN con un IQN del nodo Kubernetes, come mostrato nell'esempio seguente, riceverai l'errore: `LUN already mapped to initiator(s) in this group`.

Per importare il volume sarà necessario rimuovere l'iniziatore o annullare la mappatura del LUN.

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

Elemento

Trident supporta il software NetApp Element e l'importazione di volumi NetApp HCI utilizzando `solidfire-san` autista.



Il driver Element supporta nomi di volume duplicati. Tuttavia, Trident restituisce un errore se sono presenti nomi di volume duplicati. Come soluzione alternativa, clonare il volume, fornire un nome di volume univoco e importare il volume clonato.

Esempio di elemento

L'esempio seguente 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 |
+-----+-----+-----+-----+
|          BACKEND UUID  |      | STATE  | MANAGED |
+-----+-----+-----+-----+
+-----+-----+-----+-----+
| pvc-970ce1ca-2096-4ecd-8545-ac7edc24a8fe | 10 GiB | basic-element |
block    | d3ba047a-ea0b-43f9-9c42-e38e58301c49 | online | true    |
+-----+-----+-----+-----+
+-----+-----+-----+-----+
```

Piattaforma Google Cloud

Trident supporta l'importazione di volumi utilizzando `gcp-cvs` autista.



Per importare un volume supportato dal servizio NetApp Cloud Volumes Service in Google Cloud Platform, identifica il volume tramite il suo percorso. Il percorso del volume è la parte del percorso di esportazione del volume dopo il `:/`. Ad esempio, se il percorso di esportazione è `10.0.0.1:adroit-jolly-swift`, il percorso del volume è `adroit-jolly-swift`.

Esempio di Google Cloud Platform

L'esempio seguente importa un `gcp-cvs` volume sul backend `gcpcvs_YEppr` con il percorso del volume di `adroit-jolly-swift`.

```
tridentctl import volume gcpcvs_YEppr adroit-jolly-swift -f <path-to-pvc-
file> -n trident
```

	NAME	SIZE	STORAGE CLASS	
PROTOCOL	BACKEND UUID	STATE	MANAGED	
	pvc-a46ccab7-44aa-4433-94b1-e47fc8c0fa55	93 GiB	gcp-storage	file
	e1a6e65b-299e-4568-ad05-4f0a105c888f	online	true	

Azure NetApp Files

Trident supporta l'importazione di volumi utilizzando `azure-netapp-files` autista.



Per importare un volume Azure NetApp Files , identifica il volume tramite il suo percorso. Il percorso del 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 del volume è `importvol1` .

Esempio Azure NetApp Files

L'esempio seguente importa un `azure-netapp-files` volume sul backend `azurenetaappfiles_40517` con il percorso del volume `importvol1` .

```
tridentctl import volume azurenetaappfiles_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 `google-cloud-netapp-volumes` autista.

Esempio Google Cloud NetApp Volumes

L'esempio seguente importa un google-cloud-netapp-volumes volume sul 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

L'esempio seguente importa un `google-cloud-netapp-volumes` volume quando due volumi sono presenti 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

Con Trident puoi assegnare nomi ed etichette significativi ai volumi che crei. Ciò consente di identificare e mappare facilmente i volumi alle rispettive risorse Kubernetes (PVC). È anche possibile definire modelli a livello di backend per creare nomi di volume ed etichette personalizzate; tutti i volumi creati, importati o clonati aderiranno ai modelli.

Prima di iniziare

Supporto per nomi e etichette di volume personalizzabili:

1. Operazioni di creazione, importazione e clonazione del volume.
2. Nel caso del driver ontap-nas-economy, solo il nome del volume Qtree è conforme al modello di nome.
3. Nel caso del driver ontap-san-economy, solo il nome LUN è conforme al modello di nome.

Limitazioni

1. I nomi dei volumi personalizzabili sono compatibili solo con i driver ONTAP on-premises.
2. I nomi dei volumi personalizzabili non si applicano ai volumi esistenti.

Comportamenti chiave dei nomi dei volumi personalizzabili

1. 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 denominato in base alla convenzione di denominazione esistente.
2. Il prefisso di archiviazione non è applicabile quando un volume viene denominato utilizzando un modello di nome dalla configurazione backend. È possibile aggiungere direttamente al modello qualsiasi valore di

prefisso desiderato.

Esempi di configurazione del backend con modello di nome ed etichette

È possibile definire modelli di nomi personalizzati a livello di radice e/o di 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.RequestName}}"
  },
  "labels": {
    "cluster": "ClusterA",
    "PVC": "{{.volume.Namespace}}_{{.volume.RequestName}}"
  }
}
```

Esempio di livello della piscina

```
{
  "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 modelli di nomi

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 importazioni di volumi, le etichette vengono aggiornate solo se il volume esistente ha etichette in un formato specifico. Per 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 radice nella definizione del backend.
3. Trident non supporta l'uso di un operatore slice con il prefisso storage.
4. Se i modelli non generano nomi di volume univoci, Trident aggiungerà alcuni caratteri casuali per creare nomi di volume univoci.
5. Se il nome personalizzato per un volume economico NAS supera i 64 caratteri, Trident nominerà i volumi in base alla convenzione di denominazione esistente. Per tutti gli altri driver ONTAP , se il nome del volume supera il limite, il processo di creazione del volume fallisce.

Condividere un volume NFS tra gli spazi dei nomi

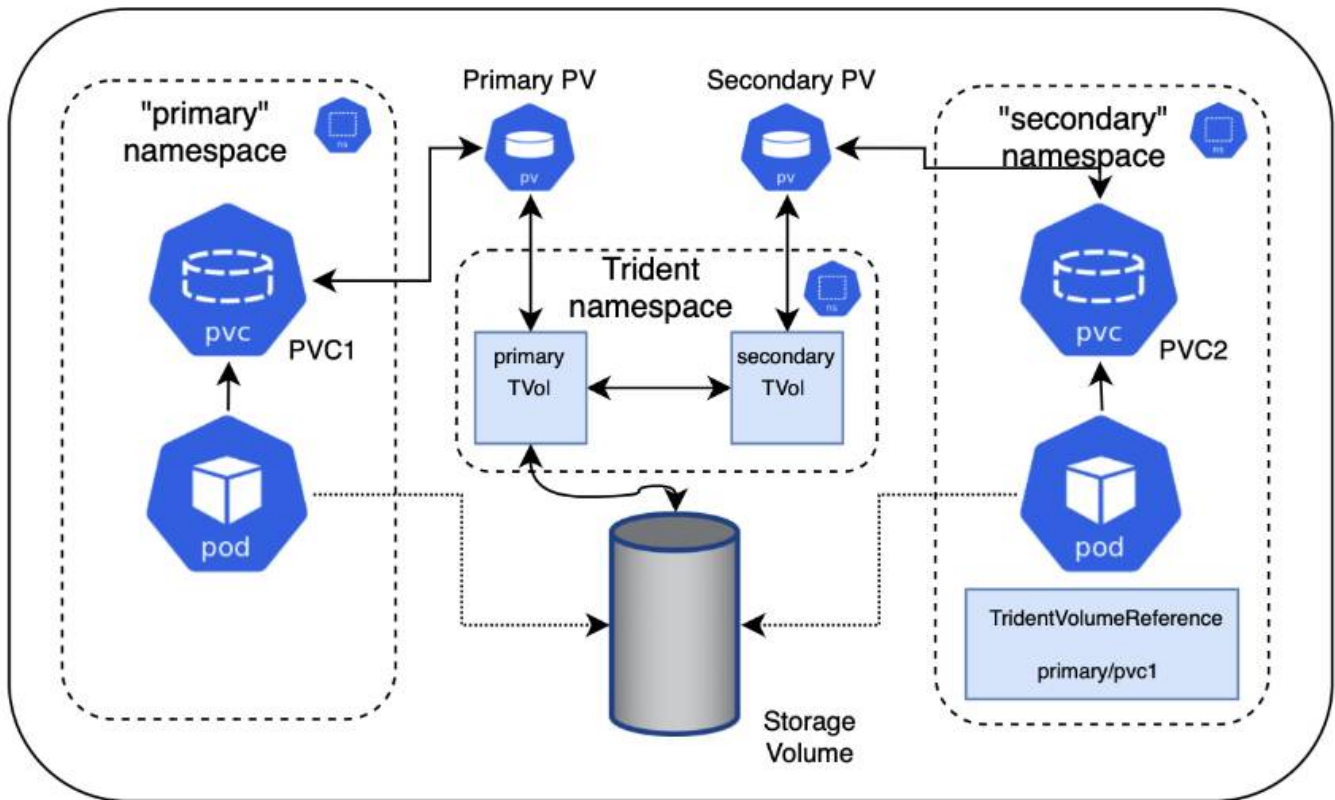
Utilizzando Trident, è possibile creare un volume in uno spazio dei nomi primario e dividerlo in uno o più spazi dei nomi secondari.

Caratteristiche

TridentVolumeReference CR consente di condividere in modo sicuro volumi NFS ReadWriteMany (RWX) su uno o più namespace Kubernetes. Questa soluzione nativa di 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 l'autorizzazione ad accedere ai dati nel PVC di origine.

2

Concedi l'autorizzazione per creare un CR nello spazio dei nomi di destinazione

L'amministratore del cluster concede l'autorizzazione al proprietario dello spazio dei nomi di destinazione per creare il CR `TridentVolumeReference`.

3

Crea `TridentVolumeReference` nello spazio dei nomi di destinazione

Il proprietario dello spazio dei nomi di destinazione crea il CR `TridentVolumeReference` 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 dello spazio dei nomi di origine, dell'amministratore del cluster e del proprietario dello spazio dei nomi di destinazione. Il ruolo dell'utente viene designato in ogni fase.

Passi

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 il `shareToNamespace` annotazione.

```
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.



- È possibile condividere il PVC con più namespace utilizzando un elenco delimitato da virgole. Per esempio, `trident.netapp.io/shareToNamespace: namespace2, namespace3, namespace4`.
- Puoi condividere con tutti gli spazi dei nomi utilizzando `*`. Per esempio, `trident.netapp.io/shareToNamespace: *`
- È possibile aggiornare il PVC per includere il `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 per creare il CR `TridentVolumeReference` nello spazio dei nomi di destinazione.
3. **Proprietario dello spazio dei nomi di destinazione:** Crea un CR `TridentVolumeReference` nello spazio dei nomi di destinazione che fa 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 il `shareFromPVC` annotazione per designare il PVC sorgente.

```

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

```



La dimensione del PVC di destinazione deve essere inferiore o uguale a quella del PVC di origine.

Risultati

Il Trident legge il `shareFromPVC` annotazione sul PVC di destinazione e crea il PV di destinazione come volume subordinato senza risorse di archiviazione proprie che punta al PV di origine e condivide la risorsa di archiviazione del PV di origine. La destinazione PVC e PV appaiono vincolate normalmente.

Elimina un volume condiviso

È possibile eliminare un volume condiviso tra più namespace. Trident rimuoverà l'accesso al volume nello spazio dei nomi di origine e manterrà l'accesso per gli altri spazi dei nomi che condividono il volume. Quando tutti gli spazi dei nomi che fanno riferimento al volume vengono rimossi, Trident elimina il volume.

Utilizzo `tridentctl` get per interrogare volumi subordinati

Utilizzando il `tridentctl` utilità, è possibile eseguire il `get` comando per ottenere volumi subordinati. Per maggiori informazioni, fare riferimento al `tridentctl` [comandi e opzioni](#).

```
Usage:
  tridentctl get [option]
```

Bandiere:

- `-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 agli spazi dei nomi di destinazione di scrivere sul volume condiviso. Per evitare la sovrascrittura dei dati del volume condiviso, è opportuno utilizzare il blocco dei file o altri processi.
- Non è possibile revocare l'accesso al PVC di origine rimuovendo il `shareToNamespace` O `shareFromNamespace` annotazioni o eliminazione del `TridentVolumeReference` CR. Per revocare l'accesso, è necessario eliminare il PVC subordinato.
- Non è possibile eseguire snapshot, cloni e mirroring sui volumi subordinati.

Per maggiori informazioni

Per saperne di più sull'accesso ai volumi tra più namespace:

- Visita ["Condivisione di volumi tra namespace: dai il benvenuto all'accesso ai volumi tra namespace"](#) .
- Guarda la demo su ["NetAppTV"](#) .

Clonazione di volumi tra spazi dei nomi

Utilizzando Trident, puoi creare nuovi volumi utilizzando volumi esistenti o snapshot di volume da uno spazio dei nomi 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 archiviazione.



La clonazione tra spazi dei nomi è supportata solo per `ontap-san` E `ontap-nas` driver di archiviazione. I cloni di sola lettura non sono supportati.

Avvio rapido

È possibile impostare la clonazione del volume in pochi semplici passaggi.



Configurare il PVC di origine per clonare il volume

Il proprietario dello spazio dei nomi di origine concede l'autorizzazione ad accedere ai dati nel PVC di origine.

2**Concedi l'autorizzazione per creare un CR nello spazio dei nomi di destinazione**

L'amministratore del cluster concede l'autorizzazione al proprietario dello spazio dei nomi di destinazione per creare il CR `TridentVolumeReference`.

3**Crea `TridentVolumeReference` nello spazio dei nomi di destinazione**

Il proprietario dello spazio dei nomi di destinazione crea il CR `TridentVolumeReference` per fare riferimento al PVC di origine.

4**Crea il PVC clone nello spazio dei nomi di destinazione**

Il proprietario dello spazio dei nomi di destinazione crea 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 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 designato in ogni fase.

Passi

1. **Proprietario dello spazio dei nomi di origine:** Crea il PVC(`pvc1`) nello spazio dei nomi di origine(`namespace1`) che concede l'autorizzazione alla condivisione con lo spazio dei nomi di destinazione(`namespace2`) utilizzando il `cloneToNamespace` annotazione.

```
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 archiviazione backend.



- È possibile condividere il PVC con più namespace utilizzando un elenco delimitato da virgole. Per esempio, `trident.netapp.io/cloneToNamespace: namespace2,namespace3,namespace4`.
- Puoi condividere con tutti gli spazi dei nomi utilizzando `*`. Per esempio, `trident.netapp.io/cloneToNamespace: *`
- È possibile aggiornare il PVC per includere il `cloneToNamespace` 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 per creare il CR `TridentVolumeReference` nello spazio dei nomi di destinazione(`namespace2`).
3. **Proprietario dello spazio dei nomi di destinazione:** Crea un CR `TridentVolumeReference` nello spazio dei nomi di destinazione che fa 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 il `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 forniti tramite driver `ontap-nas-economy`, i cloni di sola lettura non sono supportati.

Replicare i volumi utilizzando SnapMirror

Trident supporta relazioni mirror tra un volume di origine su un cluster e il volume di destinazione sul cluster peer per replicare i dati in caso di ripristino di emergenza. È possibile utilizzare una definizione di risorsa personalizzata (CRD) con namespace, denominata Trident Mirror Relationship (TMR), per eseguire le seguenti operazioni:

- Creare relazioni speculari tra volumi (PVC)
- Rimuovere le relazioni speculari tra i volumi
- Rompere le relazioni speculari
- Promuovere il volume secondario durante le condizioni di disastro (failover)
- Eseguire la transizione senza perdite delle applicazioni da un cluster all'altro (durante i failover o le migrazioni pianificate)

Prerequisiti di replicazione

Prima di iniziare, assicurati che siano soddisfatti i seguenti prerequisiti:

Cluster ONTAP

- *** Trident***: Trident versione 22.10 o successiva deve essere presente sia sui cluster Kubernetes di origine che su quelli 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. Fare riferimento a ["Panoramica delle licenze SnapMirror in ONTAP"](#) per maggiori informazioni.

A partire da ONTAP 9.10.1, tutte le licenze vengono fornite come file di licenza NetApp (NLF), ovvero un singolo file che abilita più funzionalità. Fare riferimento a ["Licenze incluse con ONTAP One"](#) per maggiori informazioni.



È supportata solo la protezione asincrona SnapMirror .

Sbirciando

- **Cluster e SVM**: i backend di archiviazione ONTAP devono essere peered. Fare riferimento a ["Panoramica del peering di cluster e SVM"](#) per maggiori informazioni.



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 dei volumi con la tecnologia NetApp SnapMirror utilizzando classi di archiviazione supportate dai seguenti driver: **ontap-nas : NFS** **ontap-san : iSCSI** **ontap-san : FC** **ontap-san : NVMe/TCP** (richiede almeno la versione ONTAP 9.15.1)



La replicazione del volume tramite SnapMirror non è supportata per i sistemi ASA r2. Per informazioni sui sistemi ASA r2, vedere ["Scopri di più sui sistemi di archiviazione ASA r2"](#).

Crea un PVC specchiato

Seguire questi passaggi e utilizzare gli esempi CRD per creare una relazione speculare tra volumi primari e secondari.

Passi

1. Eseguire i seguenti passaggi sul cluster Kubernetes primario:
 - a. Creare un oggetto StorageClass con `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. Creare un PVC con StorageClass creato in precedenza.

Esempio

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: csi-nas
spec:
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 1Gi
  storageClassName: csi-nas
```

- c. Creare una CR MirrorRelationship 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 di MirrorRelationship.

- d. Ottenere il CR TridentMirrorRelationship 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:
- Creare una 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. Creare una CR MirrorRelationship con informazioni sulla destinazione e sulla sorgente.

Esempio

```
kind: TridentMirrorRelationship
apiVersion: trident.netapp.io/v1
metadata:
  name: csi-nas
spec:
  state: established
  volumeMappings:
    - localPVCName: csi-nas
      remoteVolumeHandle:
        "datavserver:trident_pvc_3bedd23c_46a8_4384_b12b_3c38b313c1e1"
```

Trident creerà una relazione SnapMirror con il nome del criterio di relazione configurato (o predefinito per ONTAP) e la inizierà.

- c. Creare un PVC con StorageClass creato in precedenza che funga da destinazione secondaria (SnapMirror).

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 il CRD `TridentMirrorRelationship` 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 collegata alla SVM remota definita in `MirrorRelationship`.

Stati di replica del volume

Una relazione Trident Mirror (TMR) è una CRD che rappresenta un'estremità di una relazione di replicazione tra PVC. Il TMR di destinazione ha uno stato che indica a Trident qual è lo stato desiderato. La TMR di destinazione presenta i seguenti stati:

- **Stabilito:** il PVC locale è il volume di destinazione di una relazione speculare e questa è una nuova relazione.
- **Promosso:** il PVC locale è ReadWrite e montabile, senza alcuna relazione mirror attualmente in vigore.
- **Ristabilito:** il PVC locale è il volume di destinazione di una relazione speculare ed era anche precedentemente in quella relazione speculare.
 - 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 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 PVC secondario durante un failover pianificato

Durante un failover pianificato (migrazione), eseguire i seguenti passaggi per promuovere il PVC secondario:

Passi

1. Sul cluster Kubernetes primario, creare uno snapshot del PVC e attendere che venga creato.
2. Sul cluster Kubernetes primario, creare `SnapshotInfo` CR per ottenere dettagli interni.

Esempio

```
kind: SnapshotInfo
apiVersion: trident.netapp.io/v1
metadata:
  name: csi-nas
spec:
  snapshot-name: csi-nas-snapshot
```

3. Nel cluster Kubernetes secondario, aggiornare il campo `spec.state` del CR `TridentMirrorRelationship` in `promoted` e `spec.promotedSnapshotHandle` in modo che sia `internalName` dello snapshot.
4. Nel cluster Kubernetes secondario, confermare lo stato (campo `status.state`) di `TridentMirrorRelationship` su promosso.

Ripristinare una relazione mirror dopo un failover

Prima di ripristinare una relazione speculare, scegli il lato che vuoi rendere primario.

Passi

1. Nel cluster Kubernetes secondario, assicurarsi che i valori per il campo *spec.remoteVolumeHandle* su *TridentMirrorRelationship* siano aggiornati.
2. Sul cluster Kubernetes secondario, aggiorna il campo *spec.mirror* di *TridentMirrorRelationship* a *reestablished*.

Operazioni aggiuntive

Trident supporta le seguenti operazioni sui volumi primario e secondario:

Replicare il PVC primario in un nuovo PVC secondario

Assicurati di avere già un PVC primario e un PVC secondario.

Passi

1. Eliminare i CRD *PersistentVolumeClaim* e *TridentMirrorRelationship* dal cluster secondario (di destinazione) stabilito.
2. Eliminare il CRD *TridentMirrorRelationship* dal cluster primario (sorgente).
3. Creare un nuovo CRD *TridentMirrorRelationship* sul cluster primario (sorgente) per il nuovo PVC secondario (destinazione) che si desidera stabilire.

Ridimensiona un PVC specchiato, primario o secondario

Il PVC può essere ridimensionato normalmente, ONTAP espanderà automaticamente tutti i flevxol di destinazione se la quantità di dati supera la dimensione corrente.

Rimuovere la replicazione da un PVC

Per rimuovere la replica, eseguire una delle seguenti operazioni sul volume secondario corrente:

- Eliminare la *MirrorRelationship* sul PVC secondario. Ciò interrompe la relazione di replicazione.
- Oppure, aggiorna il campo *spec.state* in *promoted*.

Elimina un PVC (che era stato precedentemente specchiato)

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 speculare fa sì che il TMR rimanente passi allo stato *promosso* prima che Trident completi l'eliminazione. Se il TMR selezionato per l'eliminazione è già nello stato *promosso*, 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 in futuro questo volume verrà utilizzato in una relazione mirror, dovrà utilizzare un nuovo TMR con uno stato di replica del volume *stabilito* durante la creazione della nuova relazione mirror.

Aggiorna le relazioni mirror quando ONTAP è online

Le relazioni speculari possono essere aggiornate in qualsiasi momento dopo essere state stabilite. Puoi usare il `state: promoted` o `state: reestablished` campi per aggiornare le relazioni. Quando si promuove un volume di destinazione a un normale volume ReadWrite, è possibile utilizzare `promotedSnapshotHandle` per specificare uno snapshot specifico in 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
```

Il campo `status.state` riflette lo stato del CRD `TridentActionMirrorUpdate`. Può assumere un valore tra *Riuscito*, *In corso* o *Non riuscito*.

Utilizzare la topologia CSI

Trident può creare e collegare selettivamente volumi ai nodi presenti in un cluster Kubernetes utilizzando ["Funzionalità di topologia 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. Oggi i provider cloud consentono agli amministratori di Kubernetes di generare nodi basati su zone. I nodi possono essere ubicati in diverse zone di disponibilità all'interno di una regione o tra regioni diverse. Per facilitare il provisioning dei volumi per i carichi di lavoro in un'architettura multizona, Trident utilizza la topologia CSI.



Scopri di più sulla funzionalità CSI Topology ["Qui"](#).

Kubernetes offre due modalità di associazione dei volumi uniche:

- Con `VolumeBindingMode` impostato su `Immediate` Trident crea il volume senza alcuna consapevolezza della topologia. Il binding del volume e il provisioning dinamico vengono gestiti durante la creazione del PVC. Questo è l'impostazione predefinita `VolumeBindingMode` ed è adatto per cluster che non impongono vincoli di topologia. I volumi persistenti vengono creati senza alcuna dipendenza dai requisiti di pianificazione del pod richiedente.
- Con `VolumeBindingMode` impostato su `WaitForFirstConsumer`, la creazione e l'associazione di un volume persistente per un PVC vengono ritardate finché non viene pianificato e creato un pod che utilizza il PVC. In questo modo, i volumi vengono creati per soddisfare i vincoli di pianificazione imposti dai requisiti di topologia.



IL `WaitForFirstConsumer` la modalità di associazione non richiede etichette topologiche. Può essere utilizzato indipendentemente dalla funzionalità CSI Topology.

Cosa ti servirà

Per utilizzare la topologia CSI, è necessario quanto segue:

- Un cluster Kubernetes che esegue un **"versione Kubernetes supportata"**

```
kubectl version
Client Version: version.Info{Major:"1", Minor:"19",
GitVersion:"v1.19.3",
GitCommit:"1e11e4a2108024935ecfcb2912226cedeaafd99df",
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:"1e11e4a2108024935ecfcb2912226cedeaafd99df",
GitTreeState:"clean", BuildDate:"2020-10-14T12:41:49Z",
GoVersion:"go1.15.2", Compiler:"gc", Platform:"linux/amd64"}
```

- I nodi nel 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 dell'installazione di Trident affinché Trident sia a conoscenza 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","kubernetes.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","kubernetes.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","kubernetes.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"}]
```

Passaggio 1: creare un backend consapevole della topologia

I backend di archiviazione Trident possono essere progettati per effettuare il provisioning selettivo dei volumi in base alle zone di disponibilità. Ogni backend può trasportare un optional `supportedTopologies` blocco che rappresenta un elenco di zone e regioni supportate. Per le `StorageClass` che utilizzano tale backend, un volume verrà creato solo se richiesto da un'applicazione pianificata in una regione/zona supportata.

Ecco un esempio di definizione del backend:

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

```
{
  "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"
    }
  ]
}
```



`supportedTopologies` viene utilizzato 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 una StorageClass. Per le StorageClass che contengono un sottoinsieme delle regioni e delle zone fornite in un backend, Trident crea un volume sul backend.

Puoi definire `supportedTopologies` anche per pool di archiviazione. Vedere il seguente esempio:

```

---
version: 1
storageDriverName: ontap-nas
backendName: nas-backend-us-centrall
managementLIF: 172.16.238.5
svm: nfs_svm
username: admin
password: password
supportedTopologies:
  - topology.kubernetes.io/region: us-centrall
    topology.kubernetes.io/zone: us-centrall-a
  - topology.kubernetes.io/region: us-centrall
    topology.kubernetes.io/zone: us-centrall-b
storage:
  - labels:
      workload: production
    supportedTopologies:
      - topology.kubernetes.io/region: us-centrall
        topology.kubernetes.io/zone: us-centrall-a
  - labels:
      workload: dev
    supportedTopologies:
      - topology.kubernetes.io/region: us-centrall
        topology.kubernetes.io/zone: us-centrall-b

```

In questo esempio, il region E zone le etichette indicano la posizione del pool di archiviazione. topology.kubernetes.io/region E topology.kubernetes.io/zone determinano da dove possono essere consumati i pool di stoccaggio.

Passaggio 2: definire StorageClass che riconoscono la topologia

In base alle etichette topologiche fornite ai nodi del cluster, è possibile definire StorageClass in modo che contengano informazioni sulla topologia. Ciò determinerà i pool di archiviazione che fungono da candidati per le richieste 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: null
name: netapp-san-us-east1
provisioner: csi.trident.netapp.io
volumeBindingMode: WaitForFirstConsumer
allowedTopologies:
  - matchLabelExpressions: null
  - 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 StorageClass fornita sopra, volumeBindingMode è impostato su WaitForFirstConsumer. I PVC richiesti con questa StorageClass non verranno elaborati finché non verranno referenziati in un pod. E, allowedTopologies fornisce le zone e la regione da utilizzare. Il netapp-san-us-east1 StorageClass crea PVC su san-backend-us-east1 backend definito sopra.

Fase 3: creare e utilizzare un PVC

Dopo aver creato StorageClass e mappato un backend, è ora possibile creare PVC.

Vedi l'esempio spec 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 manifesto produrrebbe quanto segue:

```

kubect1 create -f pvc.yaml
persistentvolumeclaim/pvc-san created
kubect1 get pvc
NAME          STATUS      VOLUME      CAPACITY    ACCESS MODES    STORAGECLASS
AGE
pvc-san      Pending                                netapp-san-us-east1
2s
kubect1 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
  ----      -
  Normal    WaitForFirstConsumer  6s     persistentvolume-controller
waiting
for first consumer to be created before binding

```

Per far sì Trident crei un volume e lo legghi al PVC, utilizzare il PVC in un contenitore. 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: voll
      persistentVolumeClaim:
        claimName: pvc-san
  containers:
    - name: sec-ctx-demo
      image: busybox
      command: [ "sh", "-c", "sleep 1h" ]
      volumeMounts:
        - name: voll
          mountPath: /data/demo
      securityContext:
        allowPrivilegeEscalation: false

```

Questo podSpec istruisce Kubernetes a pianificare il pod sui nodi presenti nel us-east1 regione e sceglie tra qualsiasi nodo presente nella us-east1-a O us-east1-b zone.

Vedere 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` usando `tridentctl backend update`. Ciò non influirà sui volumi già forniti e verrà utilizzato solo per i PVC successivi.

Trova maggiori informazioni

- ["Gestire le risorse per i contenitori"](#)
- ["Selettore di nodi"](#)
- ["Affinità e anti-affinità"](#)
- ["Contaminazioni e tolleranze"](#)

Lavorare con gli snapshot

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 da `ontap-nas`, `ontap-nas-flexgroup`, `ontap-san`, `ontap-san-economy`, `solidfire-san`, `gcp-cvs`, `azure-netapp-files`, E `google-cloud-netapp-volumes` conducenti.

Prima di iniziare

Per lavorare con gli snapshot è necessario disporre di un controller snapshot esterno e di definizioni di risorse personalizzate (CRD). Questa è responsabilità dell'orchestratore di Kubernetes (ad esempio: Kubeadm, GKE, OpenShift).

Se la distribuzione Kubernetes non include il controller snapshot e i CRD, fare riferimento a [Distribuisci un controller di snapshot del volume](#).



Non creare un controller snapshot se si creano snapshot di volumi on-demand in un ambiente GKE. GKE utilizza un controller snapshot nascosto e integrato.

Crea uno snapshot del volume

Passi

1. Crea un `VolumeSnapshotClass` Per ulteriori informazioni, fare riferimento a ["Classe VolumeSnapshot"](#) .
 - IL driver indica il pilota Trident CSI.
 - `deletionPolicy` può essere `Delete` O `Retain` . Quando impostato su `Retain` , lo snapshot fisico sottostante sul cluster di archiviazione viene mantenuto anche quando `VolumeSnapshot` l'oggetto 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 un'istanza di un PVC esistente.

Esempi

- Questo esempio crea un'istanza 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
```

- Questo esempio crea 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 `VolumeSnapshotContent` oggetto che rappresenta l'istanza effettiva.

```
kubectl create -f snap.yaml
volumesnapshot.snapshot.storage.k8s.io/pvc1-snap created

kubectl get volumesnapshots
NAME                                AGE
pvc1-snap                          50s
```

- Puoi identificare il `VolumeSnapshotContent` oggetto per il `pvc1-snap` `VolumeSnapshot` descrivendolo. Il `Snapshot Content Name` identifica l'oggetto `VolumeSnapshotContent` che gestisce questo snapshot. Il `Ready To Use` parametro indica che lo snapshot può essere utilizzato 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
...
```

Creare un PVC da uno snapshot del volume

Puoi usare `dataSource` per creare un PVC utilizzando un `VolumeSnapshot` denominato `<pvc-name>` come fonte dei dati. Una volta creato, il PVC può essere attaccato a un contenitore e utilizzato come qualsiasi altro PVC.



Il PVC verrà creato nello stesso backend del volume sorgente. Fare riferimento a "[KB: La creazione di un PVC da uno snapshot PVC Trident non può essere creata in un backend alternativo](#)".

L'esempio seguente crea il PVC utilizzando `pvc1-snap` come fonte dei 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: pvcl-snap
    kind: VolumeSnapshot
    apiGroup: snapshot.storage.k8s.io

```

Importa uno snapshot del volume

Trident supporta il ["Processo di snapshot pre-provisioning di Kubernetes"](#) per consentire all'amministratore del cluster di creare un `VolumeSnapshotContent` oggetti e importare snapshot creati al di fuori di Trident.

Prima di iniziare

Trident deve aver creato o importato il volume padre dello snapshot.

Passi

1. **Amministratore del cluster:** Crea un `VolumeSnapshotContent` oggetto che fa riferimento allo snapshot del backend. In questo modo si avvia il flusso di lavoro degli snapshot in Trident.
 - Specificare il nome dello snapshot del backend in annotations `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 nel `ListSnapshots` chiamata.



IL `<volumeSnapshotContentName>` non può sempre corrispondere al nome dello snapshot backend a causa dei vincoli di denominazione CR.

Esempio

L'esempio seguente crea un `VolumeSnapshotContent` oggetto 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. **Amministratore del cluster:** Crea il VolumeSnapshot CR che fa riferimento al VolumeSnapshotContent oggetto. Questa richiesta richiede l'accesso per utilizzare il VolumeSnapshot in un dato spazio dei nomi.

Esempio

L'esempio seguente crea un VolumeSnapshot CR nominato import-snap che fa riferimento al VolumeSnapshotContent nominato 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 creato VolumeSnapshotContent e gestisce il ListSnapshots chiamata. Trident crea il TridentSnapshot .
 - Lo snapshotter esterno imposta il VolumeSnapshotContent A readyToUse e il VolumeSnapshot A true .
 - Il ritorno Trident readyToUse=true .
4. **Qualsiasi utente:** Crea un PersistentVolumeClaim per fare riferimento al nuovo VolumeSnapshot , dove il spec.dataSource (O spec.dataSourceRef) il nome è il VolumeSnapshot nome.

Esempio

L'esempio seguente crea un PVC che fa riferimento a VolumeSnapshot nominato `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 gli snapshot

La directory snapshot è nascosta per impostazione predefinita per facilitare la massima compatibilità dei volumi forniti tramite `ontap-nas` E `ontap-nas-economy` conducenti. Abilita il `.snapshot` directory per recuperare direttamente i dati dagli snapshot.

Utilizzare il ripristino dello snapshot del volume ONTAP CLI per ripristinare un volume a uno stato registrato in uno snapshot precedente.

```
cluster1::*> volume snapshot restore -vserver vs0 -volume vol3 -snapshot
vol3_snap_archive
```



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 andranno perse.

Ripristino del volume in loco da uno snapshot

Trident fornisce un rapido ripristino del volume in loco da uno snapshot utilizzando `TridentActionSnapshotRestore` (TASR) CR. Questa CR funziona come un'azione Kubernetes imperativa e non persiste dopo il completamento dell'operazione.

Trident supporta il ripristino degli snapshot su `ontap-san`, `ontap-san-economy`, `ontap-nas`, `ontap-nas-flexgroup`, `azure-netapp-files`, `gcp-cvs`, `google-cloud-netapp-volumes`, E `solidfire-san` conducenti.

Prima di iniziare

È necessario disporre di un PVC associato e di uno snapshot del volume disponibile.

- Verificare che lo stato del PVC sia vincolato.

```
kubectl get pvc
```

- Verificare che lo snapshot del volume sia pronto per l'uso.

```
kubectl get vs
```

Passi

1. Creare il CR TASR. Questo esempio crea un CR per PVC `pvc1` e snapshot del volume `pvc1-snapshot`.



Il CR TASR deve trovarsi in uno spazio dei nomi in cui sono presenti PVC e 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. Applicare il CR per ripristinare dallo snapshot. Questo esempio ripristina da uno snapshot `pvc1`.

```
kubectl create -f tasr-pvc1-snapshot.yaml
```

```
tridentactionsnapshotrestore.trident.netapp.io/trident-snap created
```

Risultati

Trident ripristina i dati dallo snapshot. È possibile 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: ""

```



- Nella maggior parte dei casi, Trident non riproverà automaticamente l'operazione in caso di errore. Sarà necessario eseguire nuovamente l'operazione.
- Gli utenti Kubernetes senza accesso amministrativo potrebbero dover ottenere l'autorizzazione dall'amministratore per creare un CR TASR nel namespace della propria applicazione.

Elimina un PV con gli snapshot associati

Quando si elimina un volume persistente con snapshot associati, il volume Trident corrispondente viene aggiornato allo "stato di eliminazione". Rimuovere gli snapshot del volume per eliminare il volume Trident .

Distribuisci un controller di snapshot del volume

Se la distribuzione Kubernetes non include il controller snapshot e i CRD, è possibile distribuirli come segue.

Passi

1. Creare 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
1
kubectl apply -f https://raw.githubusercontent.com/kubernetes-
csi/external-snapshotter/release-
6.1/client/config/crd/snapshot.storage.k8s.io_volumesnapshots.yaml
```

2. Creare il controller 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
```



Se necessario, aprire `deploy/kubernetes/snapshot-controller/rbac-snapshot-controller.yaml` e aggiornare namespace al tuo namespace.

Link correlati

- ["Istantanee del volume"](#)
- ["Classe VolumeSnapshot"](#)

Lavorare con gli snapshot del gruppo di volumi

Snapshot del gruppo di volumi Kubernetes di 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 eseguite nello stesso momento.



VolumeGroupSnapshot è una funzionalità beta di Kubernetes con API beta. Kubernetes 1.32 è la versione minima richiesta per VolumeGroupSnapshot.

Creare snapshot del gruppo di volumi

L'istantanea del gruppo di volumi è supportata con `ontap-san` driver, solo per protocollo iSCSI, non ancora supportato con Fibre Channel (FCP) né NVMe/TCP. Prima di iniziare

- Assicurati che la versione di Kubernetes sia K8s 1.32 o successiva.
- Per lavorare con gli snapshot è necessario disporre di un controller snapshot esterno e di definizioni di risorse personalizzate (CRD). Questa è responsabilità dell'orchestratore di Kubernetes (ad esempio: Kubeadm, GKE, OpenShift).

Se la distribuzione Kubernetes non include il controller snapshot esterno e i CRD, fare riferimento a [Distribuisci un controller di snapshot del volume](#).



Non creare un controller snapshot se si creano snapshot di gruppi di volumi su richiesta in un ambiente GKE. GKE utilizza un controller snapshot nascosto e integrato.

- Nel controller snapshot YAML, impostare `CSIVolumeGroupSnapshot` feature gate su 'true' per garantire che lo snapshot del gruppo di volumi sia abilitato.
- Creare le classi di snapshot del gruppo di volumi richieste prima di creare uno snapshot del gruppo di volumi.
- Assicurarsi che tutti i PVC/volumi siano sullo stesso SVM per poter creare `VolumeGroupSnapshot`.

Passi

- Creare una `VolumeGroupSnapshotClass` prima di creare una `VolumeGroupSnapshot`. Per maggiori informazioni, fare riferimento a ["ClasseSnapshotVolumeGroup"](#).

```
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
```

- Creare PVC con le etichette richieste utilizzando le classi di archiviazione esistenti oppure aggiungere queste etichette ai PVC esistenti.

L'esempio seguente crea il PVC utilizzando `pvc1-group-snap` come fonte di dati ed etichetta `consistentGroupSnapshot: groupA`. Definisci la chiave e il valore dell'etichetta in base alle tue esigenze.

```

kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: pvcl-group-snap
  labels:
    consistentGroupSnapshot: groupA
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 100Mi
  storageClassName: sc1-1

```

- Crea un VolumeGroupSnapshot con la stessa etichetta(`consistentGroupSnapshot: groupA`) specificato nel PVC.

Questo esempio crea uno snapshot del 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 uno snapshot di gruppo

È possibile ripristinare singoli volumi persistenti utilizzando gli snapshot individuali creati come parte dello snapshot del gruppo di volumi. Non è possibile ripristinare lo snapshot del gruppo di volumi come unità.

Utilizzare il ripristino dello snapshot del volume ONTAP CLI per ripristinare un volume a uno stato registrato in uno snapshot precedente.

```

cluster1::*> volume snapshot restore -vserver vs0 -volume vol3 -snapshot
vol3_snap_archive

```



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 andranno perse.

Ripristino del volume in loco da uno snapshot

Trident fornisce un rapido ripristino del volume in loco da uno snapshot utilizzando `TridentActionSnapshotRestore` (TASR) CR. Questa CR funziona come un'azione Kubernetes imperativa e non persiste dopo il completamento dell'operazione.

Per maggiori informazioni, vedere ["Ripristino del volume in loco da uno snapshot"](#).

Elimina un PV con snapshot di gruppo associati

Quando si elimina uno snapshot del volume di gruppo:

- È possibile eliminare i `VolumeGroupSnapshot` nel loro complesso, non i singoli snapshot del gruppo.
- Se i `PersistentVolume` vengono eliminati mentre esiste uno snapshot per quel `PersistentVolume`, Trident sposterà quel volume in uno stato di "eliminazione" perché lo snapshot deve essere rimosso prima che il volume possa essere rimosso in modo sicuro.
- Se è stato creato un clone utilizzando uno snapshot raggruppato e in seguito il gruppo deve essere eliminato, verrà avviata un'operazione di suddivisione su clone e il gruppo non potrà essere eliminato finché la suddivisione non sarà completata.

Distribuisci un controller di snapshot del volume

Se la distribuzione Kubernetes non include il controller snapshot e i CRD, è possibile distribuirli come segue.

Passi

1. Creare 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_volumegroupsnapshotclasses.yaml
kubectl apply -f https://raw.githubusercontent.com/kubernetes-csi/external-snapshotter/release-8.2/client/config/crd/groupsnapshot.storage.k8s.io_volumegroupsnapshotcontents.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. Creare il controller 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
```



Se necessario, aprire `deploy/kubernetes/snapshot-controller/rbac-snapshot-controller.yaml` e aggiornare namespace al tuo namespace.

Link correlati

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

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

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

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

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

Informazioni sul marchio commerciale

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