



# Configura e gestisci i backend

Trident

NetApp  
April 08, 2026

# Sommario

Configura e gestisci i backend	1
Configura i backend	1
Azure NetApp Files	1
Configura un backend Azure NetApp Files	1
Prepararsi a configurare un backend Azure NetApp Files	5
Opzioni di configurazione del backend di Azure NetApp Files ed esempi	7
Google Cloud NetApp Volumes	20
Configura un backend Google Cloud NetApp Volumes	20
Prepararsi a configurare un backend Google Cloud NetApp Volumes	23
Opzioni di configurazione del backend di Google Cloud NetApp Volumes ed esempi	23
Configura un backend NetApp HCI o SolidFire	37
Dettagli del driver Element	37
Prima di iniziare	38
Opzioni di configurazione del backend	38
Esempio 1: configurazione del backend per <code>solidfire-san</code> driver con tre tipi di volume	39
Esempio 2: Configurazione del backend e della storage class per <code>solidfire-san</code> driver con pool virtuali	40
Trova ulteriori informazioni	42
Driver SAN ONTAP	42
Panoramica del driver ONTAP SAN	42
Prepararsi a configurare il backend con i driver ONTAP SAN	45
Opzioni di configurazione SAN ONTAP ed esempi	52
Driver NAS ONTAP	72
Panoramica del driver ONTAP NAS	72
Prepararsi a configurare un backend con i driver ONTAP NAS	73
Opzioni ed esempi di configurazione NAS ONTAP	86
Amazon FSx for NetApp ONTAP	109
Usa Trident con Amazon FSx for NetApp ONTAP	109
Crea un ruolo IAM e un AWS Secret	112
Installare Trident	117
Configura il backend di storage	125
Configura una storage class e un PVC	134
Distribuisci l'applicazione di esempio	138
Configura il componente aggiuntivo Trident EKS su un cluster EKS	139
Crea backend con kubectl	142
TridentBackendConfig	143
Panoramica dei passaggi	144
Passaggio 1: crea un Kubernetes Secret	144
Passaggio 2: crea la TridentBackendConfig CR	146
Fase 3: verificare lo stato del TridentBackendConfig CR	146
(Facoltativo) Passaggio 4: Ottieni maggiori dettagli	147
Gestisci i backend	149

Esegui la gestione del backend con kubectf . . . . .	149
Esegui la gestione del backend con tridentctl . . . . .	150
Spostarsi tra le opzioni di gestione del backend . . . . .	152

# Configura e gestisci i backend

## Configura i backend

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

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

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

## Azure NetApp Files

### Configura un backend Azure NetApp Files

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

#### Dettagli del driver Azure NetApp Files

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

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

#### Considerazioni

- Il servizio Azure NetApp Files non supporta volumi più piccoli di 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

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

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

### Operatore Trident

Per installare Trident utilizzando l'operatore Trident, modifica `tridentorchestrator_cr.yaml` per impostare `cloudProvider` su "Azure". Ad esempio:

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

### Helm

Il seguente esempio installa Trident sets `cloudProvider` su Azure utilizzando la variabile di ambiente `$CP`:

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

### `tridentctl`

Il seguente esempio installa Trident e imposta il flag `cloudProvider` su Azure:

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

## Identità cloud per AKS

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

Per sfruttare l'identità del cloud in Azure, è necessario disporre di:

- Un cluster Kubernetes distribuito utilizzando AKS

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

## Operatore Trident

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

Ad 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-xxxxxxxxxxxx' # Edit
```

## Helm

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

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

L'esempio seguente installa Trident e imposta `cloudProvider` su Azure usando la variabile d'ambiente `$CP` e imposta `cloudIdentity` usando la variabile d'ambiente `$CI`:

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

## `tridentctl`

Imposta i valori dei flag **cloud provider** e **cloud identity** utilizzando le seguenti variabili d'ambiente:

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

L'esempio seguente installa Trident e imposta il `cloud-provider` flag su `$CP`, e `cloud-identity` su `$CI`:

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

## Prepararsi a configurare un backend Azure NetApp Files

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

### Prerequisiti per volumi NFS e SMB

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

Per configurare e utilizzare un ["Azure NetApp Files"](#) backend, è necessario quanto segue:



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

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

## Ruolo di collaboratore personalizzato

```
{
  "id": "/subscriptions/<subscription-
id>/providers/Microsoft.Authorization/roleDefinitions/<role-
definition-id>",
  "properties": {
    "roleName": "custom-role-with-limited-perms",
    "description": "custom role providing limited permissions",
    "assignableScopes": [
      "/subscriptions/<subscription-id>"
    ],
    "permissions": [
      {
        "actions": [
          "Microsoft.NetApp/netAppAccounts/capacityPools/read",
          "Microsoft.NetApp/netAppAccounts/capacityPools/write",
          "Microsoft.NetApp/netAppAccounts/capacityPools/volumes/read",
          "Microsoft.NetApp/netAppAccounts/capacityPools/volumes/write",
          "Microsoft.NetApp/netAppAccounts/capacityPools/volumes/delete",
          "Microsoft.NetApp/netAppAccounts/capacityPools/volumes/snapshots/
read",
          "Microsoft.NetApp/netAppAccounts/capacityPools/volumes/snapshots/
write",
          "Microsoft.NetApp/netAppAccounts/capacityPools/volumes/snapshots/
delete",
          "Microsoft.NetApp/netAppAccounts/capacityPools/volumes/MountTarge
ts/read",
          "Microsoft.Network/virtualNetworks/read",
          "Microsoft.Network/virtualNetworks/subnets/read",
          "Microsoft.Features/featureProviders/subscriptionFeatureRegistrat
ions/read",
          "Microsoft.Features/featureProviders/subscriptionFeatureRegistrat
ions/write",
          "Microsoft.Features/featureProviders/subscriptionFeatureRegistrat
ions/delete",
```

```

        "Microsoft.Features/features/read",
        "Microsoft.Features/operations/read",
        "Microsoft.Features/providers/features/read",

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

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

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

```

- L'location`Azure che contiene almeno un ["sottorete delegata"](#). A partire da Trident 22.01, il parametro `location` è un campo obbligatorio al livello superiore del file di configurazione del backend. I valori di posizione specificati nei pool virtuali vengono ignorati.
- Per utilizzare Cloud Identity, ottenere il client ID da un ["identità gestita assegnata all'utente"](#) e specificare quell'ID in `azure.workload.identity/client-id: xxxxxxxx-xxxx-xxxx-xxxx-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: Crea e gestisci le connessioni Active Directory per Azure NetApp Files"](#).
- Un cluster Kubernetes con un nodo controller Linux e almeno un nodo worker Windows che esegue Windows Server 2022. Trident supporta volumi SMB montati solo su pod in esecuzione su nodi Windows.
- Almeno un segreto Trident contenente le credenziali di Active Directory, così che Azure NetApp Files possa autenticarsi ad Active Directory. Per generare il segreto `smbcreds`:

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

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

### Opzioni di configurazione del backend di Azure NetApp Files ed esempi

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

rivedi esempi di configurazione.

## Opzioni di configurazione del backend

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

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

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

Parametro	Descrizione	Predefinito
virtualNetwork	Nome di una rete virtuale con una subnet delegata	""
subnet	Nome di una subnet delegata a Microsoft.Netapp/volumes	""
networkFeatures	Insieme di funzionalità VNet per un volume, può essere Basic o Standard. Network Features non è disponibile in tutte le regioni e potrebbe dover essere abilitato in un abbonamento. Specificare networkFeatures quando la funzionalità non è abilitata causa il fallimento del provisioning del volume.	""
nfsMountOptions	Controllo dettagliato delle opzioni di montaggio NFS. Ignorato per i volumi SMB. Per montare volumi utilizzando NFS versione 4.1, includere nfsvers=4 nell'elenco delle opzioni di montaggio separate da virgole per scegliere NFS v4.1. Le opzioni di montaggio impostate in una definizione di classe di archiviazione sostituiscono le opzioni di montaggio impostate nella configurazione del backend.	"nfsvers=3"
limitVolumeSize	Non eseguire il provisioning se la dimensione del volume richiesto è superiore a questo valore	"" (non applicato di default)
debugTraceFlags	Flag di debug da utilizzare durante la risoluzione dei problemi. Esempio, <code>\{"api": false, "method": true, "discovery": true\}</code> . Non utilizzare questa opzione a meno che non si stia risolvendo un problema e si richieda un dump dettagliato del log.	null
nasType	Configura la creazione di volumi NFS o SMB. Le opzioni sono nfs, smb o null. Impostando su null, vengono creati di default volumi NFS.	nfs
supportedTopologies	Rappresenta un elenco di regioni e zone supportate da questo backend. Per ulteriori informazioni, fare riferimento a <a href="#">"Usa la topologia CSI"</a> .	

Parametro	Descrizione	Predefinito
qosType	Rappresenta il tipo di QoS: Auto o Manual.	Auto
maxThroughput	Imposta il throughput massimo consentito in MiB/sec. Supportato solo per i pool di capacità QoS manuali.	4 MiB/sec



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

### Autorizzazioni e risorse necessarie

Se si riceve l'errore "No capacity pools found" durante la creazione di un PVC, è probabile che la registrazione dell'app non abbia le autorizzazioni e le risorse richieste (subnet, virtual network, capacity pool) associate. Se il debug è abilitato, Trident registrerà le risorse Azure rilevate quando il backend viene creato. Verificare che venga utilizzato un ruolo appropriato.

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



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

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

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

### Provisioning dei volumi

È possibile controllare il provisioning predefinito dei volumi specificando le seguenti opzioni in una sezione speciale del file di configurazione. Consultare [Esempi di configurazione](#) 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 sottoreti IPv4 in notazione CIDR. Ignorato per 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 in anteprima, richiede whitelisting nell'abbonamento)

## Esempi di configurazione

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

### Configurazione minima

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

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

```

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

```

## Identità gestite per AKS

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

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

## Identità cloud per AKS

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

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-anf-1
  namespace: trident
spec:
  version: 1
  storageDriverName: azure-netapp-files
  capacityPools:
    - ultra-pool
  resourceGroups:
    - aks-ami-eastus-rg
  netappAccounts:
    - smb-na
  virtualNetwork: eastus-prod-vnet
  subnet: eastus-anf-subnet
  location: eastus
  subscriptionID: 9f87c765-4774-fake-ae98-a721add45451
```

## Configurazione specifica del livello di servizio con filtri del capacity pool

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

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

## Esempio di backend con pool di capacità QoS manuali

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

```
---
version: 1
storageDriverName: azure-netapp-files
backendName: 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 la portata del posizionamento dei volumi a una singola subnet e modifica anche alcune impostazioni predefinite del provisioning dei volumi.

```
---
version: 1
storageDriverName: azure-netapp-files
subscriptionID: 9f87c765-4774-fake-ae98-a721add45451
tenantID: 68e4f836-edc1-fake-bff9-b2d865ee56cf
clientID: dd043f63-bf8e-fake-8076-8de91e5713aa
clientSecret: SECRET
location: eastus
serviceLevel: Ultra
capacityPools:
  - application-group-1/account-1/ultra-1
  - application-group-1/account-1/ultra-2
virtualNetwork: application-group-1/eastus-prod-vnet
subnet: application-group-1/eastus-prod-vnet/my-subnet
networkFeatures: Standard
nfsMountOptions: vers=3,proto=tcp,timeo=600
limitVolumeSize: 500Gi
defaults:
  exportRule: 10.0.0.0/24,10.0.1.0/24,10.0.2.100
  snapshotDir: "true"
  size: 200Gi
  unixPermissions: "0777"
```

## Configurazione del pool virtuale

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

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

## Configurazione delle topologie supportate

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

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

## Definizioni delle classi di storage

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

### Esempi di definizioni che utilizzano `parameter.selector` campo

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

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

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

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

### Esempi di definizioni per volumi SMB

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

## Configurazione di base sul namespace predefinito

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

## Utilizzo di segreti diversi per ogni namespace

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

## Utilizzo di segreti diversi per volume

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



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

## Crea il backend

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

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

Se la creazione del backend fallisce, c'è qualcosa di sbagliato nella configurazione del backend. Puoi visualizzare i log per determinare la causa eseguendo il seguente comando:

```
tridentctl logs
```

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

# Google Cloud NetApp Volumes

## Configura un backend Google Cloud NetApp Volumes

Ora puoi configurare Google Cloud NetApp Volumes come backend per Trident. Puoi 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).

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

### Identità cloud per GKE

Cloud identity consente ai pod Kubernetes di accedere alle risorse di Google Cloud autenticandosi come workload identity invece di fornire credenziali Google Cloud esplicite.

Per sfruttare l'identità cloud in Google Cloud, devi avere:

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

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

## Operatore Trident

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

Ad 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'
```

## Helm

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

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

Il seguente esempio installa Trident e imposta `cloudProvider` su GCP utilizzando la variabile di ambiente `$CP` e imposta il `cloudIdentity` utilizzando la variabile di ambiente `$ANNOTATION`:

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

## `tridentctl`

Imposta i valori dei flag **cloud provider** e **cloud identity** utilizzando le seguenti variabili d'ambiente:

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

Il seguente esempio installa Trident e imposta il `cloud-provider` flag su `$CP`, e `cloud-identity` su `$ANNOTATION`:

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

## Prepararsi a configurare un backend Google Cloud NetApp Volumes

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

### Prerequisiti per i volumi NFS

Se si utilizza Google Cloud NetApp Volumes per la prima volta o in una nuova posizione, è necessaria una configurazione iniziale per configurare Google Cloud NetApp Volumes e creare un volume NFS. Consultare ["Prima di iniziare"](#).

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

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

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

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

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

### Opzioni di configurazione del backend

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

Parametro	Descrizione	Predefinito
version		Sempre 1
storageDriverName	Nome del driver di archiviazione	Il valore di <code>storageDriverName</code> deve essere specificato come <code>"google-cloud-netapp-volumes"</code> .

Parametro	Descrizione	Predefinito
backendName	(Facoltativo) Nome personalizzato dello storage backend	Nome driver + "_" + parte della chiave API
storagePools	Parametro opzionale usato per specificare i pool di storage per la creazione dei volumi.	
projectNumber	Numero di progetto dell'account Google Cloud. Il valore si trova nella home page del portale Google Cloud.	
location	La posizione di Google Cloud in cui Trident crea i volumi GCNV. Quando si creano cluster Kubernetes cross-region, i volumi creati in un location possono essere utilizzati nei carichi di lavoro pianificati sui nodi di più regioni di Google Cloud. Il traffico cross-region comporta un costo aggiuntivo.	
apiKey	Chiave API per l'account del servizio Google Cloud con il netapp.admin ruolo. Include il contenuto in formato JSON del file della chiave privata di un account del servizio Google Cloud (copiato testualmente nel file di configurazione del backend). Il apiKey deve includere coppie chiave-valore per le seguenti chiavi: type, project_id, client_email, client_id, auth_uri, token_uri, auth_provider_x509_cert_url e client_x509_cert_url.	
nfsMountOptions	Controllo granulare delle opzioni di mount NFS.	"nfsvers=3"
limitVolumeSize	Il provisioning fallisce se la dimensione del volume richiesta è superiore a questo valore.	"" (non applicato di default)
serviceLevel	Il livello di servizio di un pool di storage e dei suoi volumi. I valori sono flex, standard, premium, o extreme.	
labels	Set di etichette arbitrarie in formato JSON da applicare ai volumi	""
network	Rete Google Cloud utilizzata per i volumi GCNV.	
debugTraceFlags	Flag di debug da usare per la risoluzione dei problemi. Esempio, {"api":false, "method":true}. Non usare questo a meno che non si stia eseguendo una risoluzione dei problemi e sia necessario un dump dettagliato del registro.	null
nasType	Configura la creazione di volumi NFS o SMB. Le opzioni sono nfs, smb o null. Impostando su null, vengono creati di default volumi NFS.	nfs

Parametro	Descrizione	Predefinito
supportedTopologies	Rappresenta un elenco di regioni e zone supportate da questo backend. Per ulteriori informazioni, fare riferimento a <a href="#">"Usa la topologia CSI"</a> . Ad esempio: supportedTopologies: - topology.kubernetes.io/region: asia-east1 topology.kubernetes.io/zone: asia-east1-a	

## Opzioni di provisioning del volume

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

Parametro	Descrizione	Predefinito
exportRule	Le regole di esportazione per i nuovi volumi. Deve essere un elenco separato da virgole di qualsiasi combinazione di indirizzi IPv4.	"0.0.0.0/0"
snapshotDir	Accesso alla <code>.snapshot</code> directory	"true" per NFSv4 "false" per NFSv3
snapshotReserve	Percentuale di volume riservata alle snapshot	"" (accetta il valore predefinito di 0)
unixPermissions	I permessi unix dei nuovi volumi (4 cifre ottali).	""

## Esempi di configurazione

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

## Configurazione minima

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

Questa configurazione è ideale quando si è agli inizi con Google Cloud NetApp Volumes e si stanno facendo delle prove, ma in pratica è molto probabile che sia necessario fornire uno scoping aggiuntivo per i volumi che si forniscono.

```

---
apiVersion: v1
kind: Secret
metadata:
  name: backend-tbc-gcnv-secret
type: Opaque
stringData:
  private_key_id: f2cb6ed6d7cc10c453f7d3406fc700c5df0ab9ec
  private_key: |
    -----BEGIN PRIVATE KEY-----
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/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
  apiKey:
    type: service_account
    project_id: my-gcnv-project
    client_email: myproject-prod@my-gcnv-
project.iam.gserviceaccount.com
    client_id: "103346282737811234567"
    auth_uri: https://accounts.google.com/o/oauth2/auth
    token_uri: https://oauth2.googleapis.com/token
    auth_provider_x509_cert_url:
https://www.googleapis.com/oauth2/v1/certs
    client_x509_cert_url:
https://www.googleapis.com/robot/v1/metadata/x509/myproject-prod%40my-
gcnv-project.iam.gserviceaccount.com
  credentials:
    name: backend-tbc-gcnv-secret

```

## Configurazione per i volumi SMB

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

**Configurazione con filtro StoragePools**



```

---
apiVersion: v1
kind: Secret
metadata:
  name: backend-tbc-gcnv-secret
type: Opaque
stringData:
  private_key_id: f2cb6ed6d7cc10c453f7d3406fc700c5df0ab9ec
  private_key: |
    -----BEGIN PRIVATE KEY-----
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/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 di backend definisce più pool virtuali in un unico file. I pool virtuali sono definiti nella `storage` sezione. Sono utili quando si dispone di più pool di storage che supportano diversi livelli di servizio e si desidera creare classi di storage in Kubernetes che li rappresentano. Le etichette dei pool virtuali sono utilizzate per differenziare i pool. Ad esempio, nell'esempio seguente `performance label` e `serviceLevel type` sono utilizzati per differenziare i pool virtuali.

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

Per ulteriori informazioni, 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-----
    znHczZsrtrHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrtrHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrtrHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrtrHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m
    XsYg6gyxy4zq70lwWgLwGa==
    -----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 facilita il provisioning dei volumi per i carichi di lavoro in base alle regioni e alle zone di disponibilità. Il `supportedTopologies` blocco in questa configurazione di backend viene utilizzato per fornire un elenco di regioni e zone per backend. I valori di regione e zona specificati qui devono corrispondere ai valori di regione e zona delle etichette su ciascun nodo del cluster Kubernetes. Queste regioni e zone rappresentano l'elenco dei valori consentiti che possono essere forniti in una classe di storage. Per le classi di storage che contengono un sottoinsieme delle regioni e delle zone fornite in un backend, Trident crea i volumi nella regione e nella zona menzionate. Per ulteriori informazioni, consultare "[Usa la topologia CSI](#)".

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

## E ora?

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

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

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

```
kubectl get tridentbackendconfig
```

NAME	BACKEND NAME	BACKEND UUID
backend-tbc-gcgv	backend-tbc-gcgv	b2fd1ff9-b234-477e-88fd-713913294f65
Bound	Success	

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

```
tridentctl logs
```

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

### Definizioni delle classi di storage

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

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

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

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

```

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

---

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

---

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

```

Per ulteriori dettagli sulle storage class, consultare ["Creare una storage class"](#).

### Esempi di definizioni per volumi SMB

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

## Configurazione di base sul namespace predefinito

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

## Utilizzo di segreti diversi per ogni namespace

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

## Utilizzo di segreti diversi per volume

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



Driver	Protocollo	VolumeMode	Modalità di accesso supportate	File system supportati
solidfire-san	iSCSI	Blocco	RWO, ROX, RWX, RWOP	Nessun filesystem. Dispositivo a blocchi raw.
solidfire-san	iSCSI	Filesystem	RWO, RWOP	xf <sub>s</sub> , ext3, ext4

## Prima di iniziare

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

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

## Opzioni di configurazione del backend

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

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

Parametro	Descrizione	Predefinito
limitVolumeSize	Il provisioning non riesce se la dimensione del volume richiesta è superiore a questo valore	"" (non applicato di default)
debugTraceFlags	Flag di debug da utilizzare quando si esegue la risoluzione dei problemi. Esempio, {"api":false, "method":true}	null



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

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

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

```

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

```

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

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

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

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

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

```

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

```

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

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

```

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

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

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

```

```

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

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

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

```

## Trova ulteriori informazioni

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

## Driver SAN ONTAP

### Panoramica del driver ONTAP SAN

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

### Dettagli del driver ONTAP SAN

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

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



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

## Permessi utente

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



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

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

## Considerazioni aggiuntive per NVMe/TCP

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

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

Trident non supporta:

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



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

## Prepararsi a configurare il backend con i driver ONTAP SAN

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

### Requisiti

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



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

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

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

### Autenticare il backend ONTAP

Trident offre due modalità di autenticazione per un backend ONTAP.

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

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



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

### Abilitare l'autenticazione basata su credenziali

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

Un esempio di definizione di backend sarà simile al seguente:

## YAML

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

## JSON

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

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

### Abilita l'autenticazione basata sul certificato

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

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

Un tipico flusso di lavoro prevede i seguenti passaggi.

### Passaggi

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

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

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

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

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

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



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

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

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

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

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

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

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

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

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

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

### Aggiorna i metodi di autenticazione o ruota le credenziali

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

```

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

#Update backend with tridentctl
tridentctl update backend SanBackend -f cert-backend-updated.json -n
trident
+-----+-----+-----+
+-----+-----+
|   NAME   | STORAGE DRIVER |          UUID          |
STATE | VOLUMES |
+-----+-----+-----+
+-----+-----+
| SanBackend | ontap-san      | 586b1cd5-8cf8-428d-a76c-2872713612c1 |
online |      9 |
+-----+-----+-----+
+-----+-----+

```



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

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

### Crea un ruolo personalizzato ONTAP per Trident

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

Fare riferimento a ["Generatore di ruoli personalizzati Trident"](#) per ulteriori informazioni sulla creazione di ruoli personalizzati Trident.

## Utilizzo di ONTAP CLI

1. Crea un nuovo ruolo utilizzando il seguente comando:

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

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

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

3. Assegna il ruolo all'utente:

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

## Utilizzo di System Manager

Esegui i seguenti passaggi in ONTAP System Manager:

1. **Crea un ruolo personalizzato:**

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

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

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

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

- a. Selezionare l'icona Aggiungi + sotto **Utenti**.
- b. Selezionare il nome utente richiesto e selezionare un ruolo nel menu a discesa per **Role**.
- c. Fare clic su **Save**.

Per maggiori informazioni, consultare le seguenti pagine:

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

## Autenticare le connessioni con CHAP bidirezionale

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

```
---  
version: 1  
storageDriverName: ontap-san  
backendName: ontap_san_chap  
managementLIF: 192.168.0.135  
svm: ontap_iscsi_svm  
useCHAP: true  
username: vsadmin  
password: password  
chapInitiatorSecret: cl9qxIm36DKyawxy  
chapTargetInitiatorSecret: rqxigXgkesIpwxyz  
chapTargetUsername: iJF4heBRT0TCwxyz  
chapUsername: uh2aNCLSD6cNwxyz
```



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

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

### Come funziona

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

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

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

### Ruota le credenziali e aggiorna i backend

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



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

```
cat backend-san.json
{
  "version": 1,
  "storageDriverName": "ontap-san",
  "backendName": "ontap_san_chap",
  "managementLIF": "192.168.0.135",
  "svm": "ontap_iscsi_svm",
  "useCHAP": true,
  "username": "vsadmin",
  "password": "password",
  "chapInitiatorSecret": "cl9qxUpDaTeD",
  "chapTargetInitiatorSecret": "rqxigXgkeUpDaTeD",
  "chapTargetUsername": "iJF4heBRT0TCwxyz",
  "chapUsername": "uh2aNCLSD6cNwxyz",
}

./tridentctl update backend ontap_san_chap -f backend-san.json -n trident
+-----+-----+-----+-----+
+-----+-----+
| NAME          | STORAGE DRIVER |          UUID          |
STATE | VOLUMES |
+-----+-----+-----+-----+
+-----+-----+
| ontap_san_chap | ontap-san      | aa458f3b-ad2d-4378-8a33-1a472ffbeb5c |
online |      7 |
+-----+-----+-----+-----+
+-----+-----+
```

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

## Opzioni di configurazione SAN ONTAP ed esempi

Scopri come creare e utilizzare i driver SAN ONTAP con la tua installazione Trident. Questa sezione fornisce esempi di configurazione del backend e dettagli per il mapping dei backend a StorageClasses.

["Sistemi ASA r2"](#) differiscono dagli altri sistemi ONTAP (ASA, AFF e FAS) nell'implementazione del loro livello di storage. Queste variazioni influiscono sull'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` driver (con protocolli iSCSI, NVMe/TCP e FC) è supportato per i sistemi ASA r2.


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


## Opzioni di configurazione del backend

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

Parametro	Descrizione	Predefinito
<code>version</code>		Sempre 1
<code>storageDriverName</code>	Nome del driver di archiviazione	<code>ontap-san</code> o <code>ontap-san-economy</code>
<code>backendName</code>	Nome personalizzato o lo storage backend	Nome driver + "_" + <code>dataLIF</code>
<code>managementLIF</code>	<p>Indirizzo IP di un cluster o di una LIF di gestione SVM.</p> <p>È possibile specificare un domain name pienamente qualificato (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 di MetroCluster, vedere <a href="#">Esempio di MetroCluster</a>.</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p> Se si utilizzano le credenziali "vsadmin", <code>managementLIF</code> deve essere quella dell'SVM; se si utilizzano le credenziali "admin", <code>managementLIF</code> deve essere quella 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]. <b>Non specificare per iSCSI.</b> Trident utilizza <a href="#">"ONTAP Selective LUN Map"</a> per rilevare i LIF iSCSI necessari per stabilire una sessione multipath. Viene generato un avviso se dataLIF è definito esplicitamente. <b>Omettere per MetroCluster.</b> Vedere il <a href="#">Esempio di MetroCluster</a> .	Derivato dall'SVM
svm	Macchina virtuale di storage da utilizzare <b>Ometti per MetroCluster.</b> Vedi la <a href="#">Esempio di MetroCluster</a> .	Derivato se viene specificato un SVM managementLIF
useCHAP	Utilizzare CHAP per autenticare iSCSI per i driver SAN ONTAP [Booleano]. Impostare su true per consentire a Trident di configurare e utilizzare CHAP bidirezionale come autenticazione predefinita per l'SVM specificato nel backend. Fare riferimento a <a href="#">"Prepararsi a configurare il backend con i driver ONTAP SAN"</a> per i dettagli. <b>Non supportato per FCP o NVMe/TCP.</b>	false
chapInitiatorSecret	Segreto dell'iniziatore CHAP. Obbligatorio se useCHAP=true	""
labels	Set di etichette arbitrarie in formato JSON da applicare ai volumi	""
chapTargetInitiatorSecret	Segreto dell'iniziatore di destinazione CHAP. Obbligatorio se useCHAP=true	""
chapUsername	Nome utente in entrata. Obbligatorio se useCHAP=true	""
chapTargetUsername	Nome utente di destinazione. Obbligatorio se useCHAP=true	""
clientCertificate	Valore codificato in Base64 del certificato client. Utilizzato per l'autenticazione basata su certificato	""
clientPrivateKey	Valore codificato in Base64 della chiave privata del client. Utilizzato per l'autenticazione basata su certificato	""
trustedCACertificate	Valore codificato in Base64 del certificato CA attendibile. Facoltativo. Utilizzato per l'autenticazione basata su certificato.	""
username	Nome utente necessario per comunicare con il cluster ONTAP. Utilizzato per l'autenticazione basata su credenziali. Per l'autenticazione Active Directory, vedere <a href="#">"Autenticare Trident a un backend SVM utilizzando le credenziali di Active Directory"</a> .	""

Parametro	Descrizione	Predefinito
password	Password necessaria per comunicare con il cluster ONTAP. Utilizzata per l'autenticazione basata sulle credenziali. Per l'autenticazione Active Directory, vedere <a href="#">"Autenticare Trident a un backend SVM utilizzando le credenziali di Active Directory"</a> .	""
svm	Macchina virtuale di storage da utilizzare	Derivato se viene specificato un SVM managementLIF
storagePrefix	Prefisso utilizzato durante il provisioning di nuovi volumi nell'SVM. Non può essere modificato in seguito. Per aggiornare questo parametro, sarà necessario creare un nuovo backend.	trident
aggregate	<p>Aggregato per il provisioning (facoltativo; se impostato, deve essere assegnato alla SVM). Per il <code>ontap-nas-flexgroup</code> driver, questa opzione viene ignorata. Se non assegnato, qualsiasi degli aggregati disponibili può essere utilizzato per il provisioning di un FlexGroup volume.</p> <div style="border: 1px solid gray; padding: 10px; margin: 10px 0;"> <p> Quando l'aggregato viene aggiornato in SVM, viene aggiornato automaticamente in Trident interrogando SVM senza dover riavviare il Trident Controller. Quando hai configurato un aggregato specifico in Trident per il provisioning dei volumi, se l'aggregato viene rinominato o spostato fuori dalla SVM, il backend passerà allo stato di errore in Trident durante l'interrogazione dell'aggregato SVM. Devi modificare l'aggregato con uno presente sulla SVM o rimuoverlo completamente per riportare online il backend.</p> </div> <p><b>Non specificare per i sistemi ASA r2.</b></p>	""
limitAggregateUsage	Il provisioning fallisce se l'utilizzo supera questa percentuale. Se si utilizza un Amazon FSx for NetApp ONTAP backend, non specificare <code>limitAggregateUsage</code> . I valori forniti <code>fsxadmin</code> e <code>vsadmin</code> non contengono le autorizzazioni necessarie per recuperare l'utilizzo aggregato e limitarlo tramite Trident. <b>Non specificare per sistemi ASA r2.</b>	"" (non applicato di default)
limitVolumeSize	Il provisioning fallisce se la dimensione del volume richiesto è superiore a questo valore. Limita anche la dimensione massima dei volumi che gestisce per LUN.	"" (non applicato per impostazione predefinita)

Parametro	Descrizione	Predefinito
lunsPerFlexvol	Numero massimo di LUN per FlexVol, deve essere nell'intervallo [50, 200]	100
debugTraceFlags	Flag di debug da usare per la risoluzione dei problemi. Esempio, {"api":false, "method":true} non usare a meno che non si stia eseguendo una risoluzione dei problemi e sia necessario un dump dettagliato del registro.	null
useREST	<p>Parametro booleano per utilizzare le ONTAP REST API.</p> <div style="border: 1px solid gray; padding: 10px; margin: 10px 0;"> <pre>`useREST` Quando è impostato su `true`, Trident utilizza le ONTAP REST API per comunicare con il backend; quando è impostato su `false`, Trident utilizza le chiamate ONTAPI (ZAPI) per comunicare con il backend. Questa funzione richiede ONTAP 9.11.1 e versioni successive. Inoltre, il ruolo di login ONTAP utilizzato deve avere accesso all'applicazione `ontapi`. Questo è soddisfatto dai ruoli predefiniti `vsadmin` e `cluster-admin`. A partire dalla release Trident 24.06 e ONTAP 9.15.1 o versioni successive, `useREST` è impostato su `true` per impostazione predefinita; cambiare `useREST` in `false` per utilizzare le chiamate ONTAPI (ZAPI).</pre> </div> <p>useREST è pienamente qualificato per NVMe/TCP.</p> <div style="display: flex; align-items: center; margin: 10px 0;">  <div> <p>NVMe è supportato solo con le API REST di ONTAP e non è supportato con ONTAPI (ZAPI).</p> </div> </div> <p><b>Se specificato, impostare sempre su <code>true</code> per i sistemi ASA r2.</b></p>	true per ONTAP 9.15.1 o versioni successive, altrimenti false.
sanType	Utilizzare per selezionare <code>iscsi</code> per iSCSI, <code>nvme</code> per NVMe/TCP o <code>fc</code> per SCSI over Fibre Channel (FC).	iscsi se vuoto

Parametro	Descrizione	Predefinito
formatOptions	<p>Usa <code>formatOptions</code> per specificare gli argomenti della riga di comando per il comando <code>mkfs</code>, che saranno applicati ogni volta che un volume viene formattato. Questo consente di formattare il volume secondo le tue preferenze. Assicurati di specificare i <code>formatOptions</code> simili a quelli delle opzioni del comando <code>mkfs</code>, escludendo il percorso del dispositivo. Esempio: "-E nodiscard"</p> <p><b>Supportato per <code>ontap-san</code> e <code>ontap-san-economy</code> driver con protocollo iSCSI. Inoltre, supportato per sistemi ASA r2 quando si utilizzano i protocolli iSCSI e NVMe/TCP.</b></p>	
limitVolumePoolSize	Dimensione massima richiedibile FlexVol quando si usano LUN nel backend <code>ontap-san-economy</code> .	"" (non applicato di default)
denyNewVolumePools	Restringe i backend dal <code>ontap-san-economy</code> creare nuovi volumi FlexVol per contenere le loro LUN. Solo i FlexVol preesistenti vengono utilizzati per il provisioning di nuovi PV.	

### Raccomandazioni per l'utilizzo di `formatOptions`

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

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

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

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

#### passi

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

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

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

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

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

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

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

### Opzioni di configurazione del backend per il provisioning dei volumi

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

Parametro	Descrizione	Predefinito
<code>spaceAllocation</code>	Allocazione dello spazio per le LUN	"true" <b>Se specificato, impostare su true per sistemi ASA r2.</b>
<code>spaceReserve</code>	Modalità di prenotazione dello spazio; "none" (con thin provisioning) o "volume" (con thick provisioning). <b>Impostare su none per sistemi ASA r2.</b>	"none"
<code>snapshotPolicy</code>	Policy di Snapshot da utilizzare. <b>Impostare su none per sistemi ASA r2.</b>	"none"
<code>qosPolicy</code>	Gruppo di policy QoS da assegnare ai volumi creati. Scegliere uno tra <code>qosPolicy</code> o <code>adaptiveQosPolicy</code> per pool di storage/backend. L'utilizzo dei gruppi di policy QoS con Trident richiede ONTAP 9.8 o versioni successive. Si dovrebbe utilizzare un gruppo di policy QoS non condiviso e garantire che il gruppo di policy venga applicato a ciascun costituente individualmente. Un gruppo di policy QoS condiviso impone il limite massimo per il throughput totale di tutti i carichi di lavoro.	""
<code>adaptiveQosPolicy</code>	Gruppo di policy Adaptive QoS da assegnare ai volumi creati. Scegliere uno tra <code>qosPolicy</code> o <code>adaptiveQosPolicy</code> per storage pool/backend	""
<code>snapshotReserve</code>	Percentuale di volume riservata alle snapshot. <b>Non specificare per i sistemi ASA r2.</b>	"0" se <code>snapshotPolicy</code> è "none", altrimenti ""
<code>splitOnClone</code>	Dividere un clone dal suo genitore al momento della creazione	"false"
<code>encryption</code>	Abilita NetApp Volume Encryption (NVE) sul nuovo volume; l'impostazione predefinita è <code>false</code> . NVE deve essere concesso in licenza e abilitato sul cluster per utilizzare questa opzione. Se NAE è abilitato sul backend, qualsiasi volume fornito in Trident sarà abilitato per NAE. Per ulteriori informazioni, fare riferimento a: <a href="#">"Come funziona Trident con NVE e NAE"</a> .	"false" <b>Se specificato, impostare su true per i sistemi ASA r2.</b>

Parametro	Descrizione	Predefinito
luksEncryption	Abilitare la crittografia LUKS. Fare riferimento a <a href="#">"Usa Linux Unified Key Setup (LUKS)"</a> .	"" <b>Impostato su false per i sistemi ASA r2.</b>
tieringPolicy	Criterio di tiering da utilizzare "none" <b>Non specificare per i sistemi ASA r2.</b>	
nameTemplate	Modello per creare nomi di volume personalizzati.	""

### Esempi di provisioning del volume

Ecco un esempio con i valori predefiniti:

```

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

```



Per tutti i volumi creati utilizzando il driver `ontap-san`, Trident aggiunge un ulteriore 10 per cento di capacità alla FlexVol per ospitare i metadati della LUN. La LUN verrà fornita con la dimensione esatta richiesta dall'utente nel PVC. Trident aggiunge il 10 per cento alla FlexVol (visualizzato come Available size in ONTAP). Gli utenti otterranno ora la quantità di capacità utilizzabile richiesta. Questa modifica impedisce anche che le LUN diventino di sola lettura a meno che lo spazio disponibile non sia completamente utilizzato. Questo non si applica a `ontap-san-economy`.

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

$$\text{Total volume size} = [(\text{PVC requested size}) / (1 - (\text{snapshotReserve percentage}) / 100)] * 1.1$$

L'1,1 è il 10 per cento in più che Trident aggiunge a FlexVol per ospitare i metadati della LUN. Per `snapshotReserve = 5%` e `PVC request = 5 GiB`, la dimensione totale del volume è 5,79 GiB e la dimensione disponibile è 5,5 GiB. Il comando `volume show` dovrebbe mostrare risultati simili a questo esempio:

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 ai valori 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 invece degli indirizzi IP.

### Esempio SAN ONTAP

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

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

## Esempio di MetroCluster

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

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

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

## Esempio di economia SAN ONTAP

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

## Esempio di autenticazione basata su certificato

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

```
---
version: 1
storageDriverName: ontap-san
backendName: DefaultSANBackend
managementLIF: 10.0.0.1
svm: svm_iscsi
useCHAP: true
chapInitiatorSecret: cl9qxIm36DKyawxy
chapTargetInitiatorSecret: rqxigXgkesIpwxyz
chapTargetUsername: iJF4heBRT0TCwxyz
chapUsername: uh2aNCLSD6cNwxyz
clientCertificate: ZXR0ZXJwYXB...ICMgJ3BhcGVyc2
clientPrivateKey: vciwKIyAgZG...0cnksIGRlc2NyaX
trustedCACertificate: zcyBbaG...b3Igb3duIGNsYXNz
```

## Esempi di CHAP bidirezionale

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

### Esempio ONTAP SAN CHAP

```
---  
version: 1  
storageDriverName: ontap-san  
managementLIF: 10.0.0.1  
svm: svm_iscsi  
labels:  
  k8scluster: test-cluster-1  
  backend: testcluster1-sanbackend  
useCHAP: true  
chapInitiatorSecret: cl9qxIm36DKyawxy  
chapTargetInitiatorSecret: rqxigXgkesIpwxyz  
chapTargetUsername: iJF4heBRT0TCwxyz  
chapUsername: uh2aNCLSD6cNwxyz  
username: vsadmin  
password: <password>
```

### Esempio di CHAP economy ONTAP SAN

```
---  
version: 1  
storageDriverName: ontap-san-economy  
managementLIF: 10.0.0.1  
svm: svm_iscsi_eco  
useCHAP: true  
chapInitiatorSecret: cl9qxIm36DKyawxy  
chapTargetInitiatorSecret: rqxigXgkesIpwxyz  
chapTargetUsername: iJF4heBRT0TCwxyz  
chapUsername: uh2aNCLSD6cNwxyz  
username: vsadmin  
password: <password>
```

## Esempio NVMe/TCP

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

```
---  
version: 1  
backendName: NVMeBackend  
storageDriverName: ontap-san  
managementLIF: 10.0.0.1  
svm: svm_nvme  
username: vsadmin  
password: password  
sanType: nvme  
useREST: true
```

## Esempio di SCSI su FC (FCP)

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

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

## Esempio di configurazione del backend con nameTemplate

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

## formatOptions example per il driver ontap-san-economy

```
---
version: 1
storageDriverName: ontap-san-economy
managementLIF: ""
svm: svm1
username: ""
password: "!"
storagePrefix: whelk_
debugTraceFlags:
  method: true
  api: true
defaults:
  formatOptions: -E nodiscard
```

## Esempi di backend con pool virtuali

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

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

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

**Esempio SAN ONTAP**



```
---
version: 1
storageDriverName: ontap-san
managementLIF: 10.0.0.1
svm: svm_iscsi
useCHAP: true
chapInitiatorSecret: cl9qxIm36DKyawxy
chapTargetInitiatorSecret: rqxigXgkesIpwxyz
chapTargetUsername: iJF4heBRT0TCwxyz
chapUsername: uh2aNCLSD6cNwxyz
username: vsadmin
password: <password>
defaults:
  spaceAllocation: "false"
  encryption: "false"
  qosPolicy: standard
labels:
  store: san_store
  kubernetes-cluster: prod-cluster-1
region: us_east_1
storage:
  - labels:
    protection: gold
    creditpoints: "40000"
    zone: us_east_1a
    defaults:
      spaceAllocation: "true"
      encryption: "true"
      adaptiveQosPolicy: adaptive-extreme
  - labels:
    protection: silver
    creditpoints: "20000"
    zone: us_east_1b
    defaults:
      spaceAllocation: "false"
      encryption: "true"
      qosPolicy: premium
  - labels:
    protection: bronze
    creditpoints: "5000"
    zone: us_east_1c
    defaults:
      spaceAllocation: "true"
      encryption: "false"
```

## Esempio di economia SAN ONTAP

```
---
version: 1
storageDriverName: ontap-san-economy
managementLIF: 10.0.0.1
svm: svm_iscsi_eco
useCHAP: true
chapInitiatorSecret: cl9qxIm36DKyawxy
chapTargetInitiatorSecret: rqxigXgkesIpwxyz
chapTargetUsername: iJF4heBRT0TCwxyz
chapUsername: uh2aNCLSD6cNwxyz
username: vsadmin
password: <password>
defaults:
  spaceAllocation: "false"
  encryption: "false"
labels:
  store: san_economy_store
  region: us_east_1
storage:
  - labels:
    app: oracledb
    cost: "30"
    zone: us_east_1a
    defaults:
      spaceAllocation: "true"
      encryption: "true"
  - labels:
    app: postgresdb
    cost: "20"
    zone: us_east_1b
    defaults:
      spaceAllocation: "false"
      encryption: "true"
  - labels:
    app: mysqldb
    cost: "10"
    zone: us_east_1c
    defaults:
      spaceAllocation: "true"
      encryption: "false"
  - labels:
    department: legal
    creditpoints: "5000"
    zone: us_east_1c
```

```
defaults:
  spaceAllocation: "true"
  encryption: "false"
```

## Esempio NVMe/TCP

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

## Mappa i backend a StorageClasses

Le seguenti definizioni di StorageClass si riferiscono a [Esempi di backend con pool virtuali](#). Utilizzando il campo `parameters.selector`, ciascuna StorageClass indica 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. Questo è l'unico pool che offre 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 nel `ontap-san` backend. Questi sono gli unici pool che offrono un livello di protezione diverso da `gold`.

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

- Il `app-mysqldb` StorageClass verrà mappato sul terzo pool virtuale nel `ontap-san-economy` backend. Questo è l'unico pool che offre la configurazione dello storage pool 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 nel `ontap-san` backend. Questo è l'unico pool che offre protezione di livello `silver` e 20000 creditpoints.

```
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 nel `ontap-san` backend e sul quarto pool virtuale nel `ontap-san-economy` backend. Queste sono le uniche offerte di pool con 5000 creditpoints.

```

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

```

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

```

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

```

Trident deciderà quale pool virtuale selezionare e garantirà che il requisito di storage sia soddisfatto.

## Driver NAS ONTAP

### Panoramica del driver ONTAP NAS

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

### Dettagli del driver ONTAP NAS

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

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

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



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

### Permessi utente

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

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



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

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

### Prepararsi a configurare un backend con i driver ONTAP NAS

Comprendere i requisiti, le opzioni di autenticazione e le policy di esportazione per la configurazione di un backend ONTAP con driver ONTAP NAS.

A partire dalla release 25.10, NetApp Trident supporta "[NetApp AFX sistema storage](#)". NetApp AFX storage systems differiscono dagli altri sistemi ONTAP (ASA, AFF e FAS) nell'implementazione del loro livello di storage.



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

Nella configurazione del backend Trident non è necessario specificare che il sistema è AFX. Quando si

seleziona `ontap-nas` come `storageDriverName`, Trident rileva automaticamente i sistemi AFX.

## Requisiti

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

## Autenticare il backend ONTAP

Trident offre due modalità di autenticazione per un backend ONTAP.

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

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



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

### Abilitare l'autenticazione basata su credenziali

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

Un esempio di definizione di backend sarà simile al seguente:

## YAML

```
---
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"
  }
}
```

Si tenga presente che la definizione del backend è l'unico luogo in cui le credenziali vengono memorizzate in testo normale. Dopo la creazione del backend, i nomi utente/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. Pertanto, si tratta di un'operazione riservata all'amministratore, da eseguire dall'amministratore Kubernetes/storage.

### Abilita l'autenticazione basata su certificati

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

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

Un tipico flusso di lavoro prevede i seguenti passaggi.

### Passaggi

1. Generare un certificato e una chiave client. Durante la generazione, impostare Common Name (CN)

sull'utente ONTAP con cui autenticarsi.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

### Aggiorna i metodi di autenticazione o ruota le credenziali

È possibile aggiornare un backend esistente per utilizzare un diverso metodo di autenticazione o per ruotare le proprie credenziali. Questo funziona in entrambi i modi: i backend che fanno uso di username/password possono essere aggiornati per utilizzare i certificati; i backend che utilizzano i certificati possono essere aggiornati per basarsi su username/password. Per farlo, occorre 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
online	9	



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

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

### Crea un ruolo personalizzato ONTAP per Trident

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

Fare riferimento a ["Generatore di ruoli personalizzati Trident"](#) per ulteriori informazioni sulla creazione di ruoli personalizzati Trident.

## Utilizzo di ONTAP CLI

1. Crea un nuovo ruolo utilizzando il seguente comando:

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

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

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

3. Assegna il ruolo all'utente:

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

## Utilizzo di System Manager

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 > Settings**.

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

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

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

- a. Selezionare l'icona Aggiungi + sotto **Utenti**.
- b. Selezionare il nome utente richiesto e selezionare un ruolo nel menu a discesa per **Role**.
- c. Fare clic su **Save**.

Per maggiori informazioni, consultare le seguenti pagine:

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

## Gestisci le policy di esportazione NFS

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

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

- Trident può gestire dinamicamente la policy di esportazione stessa; in questa modalità operativa, l'amministratore dello storage specifica un elenco di blocchi CIDR che rappresentano gli indirizzi IP

ammissibili. Trident aggiunge automaticamente gli IP dei nodi che rientrano in questi intervalli alla policy di esportazione al momento della pubblicazione. In alternativa, quando non vengono specificati CIDR, tutti gli IP unicast a livello globale trovati sul nodo a cui si sta pubblicando il volume verranno aggiunti alla policy di esportazione.

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

### Gestisci dinamicamente le policy di esportazione

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



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

### Esempio

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

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



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

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

- `autoExportPolicy` è impostato su `true`. Questo indica che Trident crea una policy di esportazione per ogni volume fornito con questo backend per la SVM `svm1` e gestisce l'aggiunta e l'eliminazione delle

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

- Ad esempio:

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

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

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

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

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

```

./tridentctl get backends ontap_nas_auto_export -n trident -o yaml
items:
- backendUUID: 403b5326-8482-40db-96d0-d83fb3f4daec
  config:
    aggregate: ""
    autoExportCIDRs:
    - 192.168.0.0/24
    autoExportPolicy: true
    backendName: ontap_nas_auto_export
    chapInitiatorSecret: ""
    chapTargetInitiatorSecret: ""
    chapTargetUsername: ""
    chapUsername: ""
    dataLIF: 192.168.0.135
    debug: false
    debugTraceFlags: null
    defaults:
      encryption: "false"
      exportPolicy: <automatic>
      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 montaggi non autorizzati, a meno che questo IP non venga riutilizzato da un nuovo nodo nel cluster.

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



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

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

### Prepararsi al provisioning dei volumi SMB

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



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



autoExportPolicy non è supportato per i volumi SMB.

## Prima di iniziare

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

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

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

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

## Passaggi

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



Le condivisioni SMB sono necessarie per Amazon FSx per ONTAP.

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

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

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

- b. Crea una condivisione SMB associata alla SVM specificata:

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

- c. Verificare che la condivisione sia stata creata:

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



Consultare ["Creare una condivisione SMB"](#) per tutti i dettagli.

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

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

#### Abilita SMB sicuro

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

#### Punti da ricordare

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

#### Passaggi

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

```

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

```

## 2. Aggiungi l'annotazione nella storage class.

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

```

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

```

## 1. Crea un PVC.

Il seguente esempio crea un PVC:

```

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

```

## Opzioni ed esempi di configurazione NAS ONTAP

Impara a creare e utilizzare i driver NAS ONTAP con la tua installazione Trident. Questa sezione fornisce esempi di configurazione del backend e dettagli per il mapping dei backend su StorageClasses.

A partire dalla versione 25.10, NetApp Trident supporta ["Sistemi storage NetApp AFX"](#). NetApp AFX storage systems differiscono dagli altri sistemi basati su ONTAP (ASA, AFF e FAS) nell'implementazione del loro livello di storage.




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


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


### Opzioni di configurazione del backend


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

Parametro	Descrizione	Predefinito
<code>version</code>		Sempre 1

Parametro	Descrizione	Predefinito
storageDriverName	<p>Nome del driver di archiviazione</p> <p> Per i sistemi NetApp AFX, è supportato solo ontap-nas.</p>	ontap-nas, ontap-nas-economy, 0 ontap-nas-flexgroup
backendName	Nome personalizzato o lo storage backend	Nome driver + "_" + dataLIF
managementLIF	<p>Indirizzo IP di un cluster o di un LIF di gestione SVM. È possibile specificare un fully-qualified domain name (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 di MetroCluster, vedere <a href="#">Esempio di MetroCluster</a>.</p>	"10.0.0.1", "[2001:1234:abcd::fefe]"
dataLIF	<p>Indirizzo IP del protocollo LIF. NetApp consiglia di specificare dataLIF. Se non specificato, Trident recupera i dataLIF dall'SVM. È possibile specificare un fully-qualified domain name (FQDN) da utilizzare per le operazioni di montaggio NFS, consentendo di creare un round-robin DNS per il bilanciamento del 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]. <b>Omettere per MetroCluster.</b> Vedere il <a href="#">Esempio di MetroCluster</a>.</p>	Indirizzo specificato o derivato da SVM, se non specificato (non consigliato)
svm	Macchina virtuale di storage da utilizzare <b>Ometti per MetroCluster.</b> Vedi la <a href="#">Esempio di MetroCluster</a> .	Derivato se viene specificato un SVM managementLIF
autoExportPolicy	Abilita la creazione e l'aggiornamento automatici delle policy di esportazione [Boolean]. Utilizzando le opzioni autoExportPolicy e autoExportCIDRs, Trident può gestire automaticamente le policy di esportazione.	falso
autoExportCIDRs	Elenco di CIDR in base ai quali filtrare gli IP dei nodi Kubernetes quando autoExportPolicy è abilitato. Utilizzando le autoExportPolicy e autoExportCIDRs opzioni, Trident può gestire automaticamente le policy di esportazione.	["0.0.0.0/0", ":::0"]
labels	Set di etichette arbitrarie in formato JSON da applicare ai volumi	""
clientCertificate	Valore codificato in Base64 del certificato client. Utilizzato per l'autenticazione basata su certificato	""

Parametro	Descrizione	Predefinito
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. Facoltativo. Utilizzato per l'autenticazione basata su certificato	""
username	Nome utente per connettersi al cluster/SVM. Utilizzato per l'autenticazione basata su credenziali. Per l'autenticazione di Active Directory, vedere <a href="#">"Autenticare Trident a un backend SVM utilizzando le credenziali di Active Directory"</a> .	
password	Password per connettersi al cluster/SVM. Utilizzata per l'autenticazione basata su credenziali. Per l'autenticazione di Active Directory, vedere <a href="#">"Autenticare Trident a un backend SVM utilizzando le credenziali di Active Directory"</a> .	
storagePrefix	<p>Prefisso utilizzato durante il provisioning di nuovi volumi nell'SVM. Non può essere aggiornato dopo averlo impostato</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;">  <p>Quando si utilizza ontap-nas-economy e un storagePrefix che è di 24 o più caratteri, i qtree non avranno il prefisso di archiviazione incorporato, anche se sarà presente nel nome del volume.</p> </div>	"Trident"

Parametro	Descrizione	Predefinito
aggregate	<p>Aggregato per il provisioning (facoltativo; se impostato, deve essere assegnato alla SVM). Per il <code>ontap-nas-flexgroup</code> driver, questa opzione viene ignorata. Se non assegnato, qualsiasi degli aggregati disponibili può essere utilizzato per il provisioning di un FlexGroup volume.</p> <div style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;">  <p>Quando l'aggregato viene aggiornato in SVM, viene aggiornato automaticamente in Trident interrogando SVM senza dover riavviare il Trident Controller. Quando hai configurato un aggregato specifico in Trident per il provisioning dei volumi, se l'aggregato viene rinominato o spostato fuori dalla SVM, il backend passerà allo stato di errore in Trident durante l'interrogazione dell'aggregato SVM. Devi modificare l'aggregato con uno presente sulla SVM o rimuoverlo completamente per riportare online il backend.</p> </div> <p><b>Non specificare per i sistemi di storage AFX.</b></p>	""
limitAggregateUsage	<p>Il provisioning fallisce se l'utilizzo supera questa percentuale. <b>Non si applica ad Amazon FSx per ONTAP. Non specificare per i sistemi di storage AFX.</b></p>	"" (non applicato di default)

Parametro	Descrizione	Predefinito
flexgroupAggregateList	<p>Elenco di aggregati per il provisioning (facoltativo; se impostato, deve essere assegnato alla SVM). Tutti gli aggregati assegnati alla SVM vengono utilizzati per il provisioning di un FlexGroup volume. Supportato per il driver di storage <b>ontap-nas-flexgroup</b>.</p> <p> Quando l'elenco degli aggregati viene aggiornato in SVM, l'elenco viene aggiornato automaticamente in Trident interrogando SVM senza dover riavviare il Trident Controller. Quando hai configurato un elenco di aggregati specifico in Trident per il provisioning dei volumi, se l'elenco degli aggregati viene rinominato o spostato fuori da SVM, il backend passerà allo stato di errore in Trident durante l'interrogazione dell'aggregato SVM. Devi modificare l'elenco degli aggregati con uno presente su SVM o rimuoverlo completamente per riportare online il backend.</p>	""
limitVolumeSize	Il provisioning non riesce se la dimensione del volume richiesto è superiore a questo valore.	"" (non applicato per impostazione predefinita)
debugTraceFlags	Flag di debug da utilizzare durante la risoluzione dei problemi. Ad esempio, {"api":false, "method":true} non utilizzare debugTraceFlags a meno che non si stia risolvendo un problema e si richieda un dump dettagliato del log.	null
nasType	Configura la creazione di volumi NFS o SMB. Le opzioni sono <code>nfs</code> , <code>smb</code> o <code>null</code> . Impostando su <code>null</code> , i volumi NFS vengono impostati come predefiniti. <b>Se specificato, impostare sempre su <code>nfs</code> per i sistemi di storage AFX.</b>	<code>nfs</code>
nfsMountOptions	Elenco separato da virgole delle opzioni di montaggio NFS. Le opzioni di montaggio per i volumi persistenti Kubernetes sono normalmente specificate nelle classi di storage, ma se non vengono specificate opzioni di montaggio in una classe di storage, Trident utilizzerà le opzioni di montaggio specificate nel file di configurazione del backend di storage. Se non vengono specificate opzioni di montaggio nella classe di storage o nel file di configurazione, Trident non imposterà alcuna opzione di montaggio su un volume persistente associato.	""
qtreesPerFlexvol	Numero massimo di Qtree per FlexVol, deve essere nell'intervallo [50, 300]	"200"

Parametro	Descrizione	Predefinito
smbShare	È possibile specificare uno dei seguenti valori: il nome di una condivisione SMB creata tramite Microsoft Management Console o ONTAP CLI; un nome per consentire a Trident di creare la condivisione SMB; oppure è possibile lasciare il parametro vuoto per impedire l'accesso condiviso ai volumi. Questo parametro è facoltativo per ONTAP on-premises. Questo parametro è obbligatorio per Amazon FSx for ONTAP backends e non può essere vuoto.	smb-share
useREST	Parametro booleano per utilizzare le ONTAP REST API. useREST`Quando impostato su `true, Trident utilizza le ONTAP REST API per comunicare con il backend; quando impostato su false, Trident utilizza le chiamate ONTAPI (ZAPI) per comunicare con il backend. Questa funzionalità richiede ONTAP 9.11.1 e versioni successive. Inoltre, il ruolo di login ONTAP utilizzato deve avere accesso all'applicazione ontapi. Questo è soddisfatto dai ruoli predefiniti vsadmin e cluster-admin. A partire dalla release Trident 24.06 e ONTAP 9.15.1 o versioni successive, useREST`è impostato su `true per impostazione predefinita; modificare useREST su false per utilizzare le chiamate ONTAPI (ZAPI). <b>Se specificato, impostare sempre su true per i sistemi di storage AFX.</b>	true per ONTAP 9.15.1 o versioni successive, altrimenti false.
limitVolumePoolSize	Dimensione massima richiedibile FlexVol quando si utilizzano Qtrees nel backend ontap-nas-economy.	"" (non applicato di default)
denyNewVolumePools	Restringe ontap-nas-economy i backend dal creare nuovi volumi FlexVol per contenere i loro Qtree. Solo i FlexVol preesistenti vengono utilizzati per il provisioning di nuovi PV.	
adAdminUser	Utente o gruppo di utenti amministratore di Active Directory con accesso completo alle condivisioni SMB. Utilizzare questo parametro per fornire diritti di amministratore alla condivisione SMB con controllo completo.	

### Opzioni di configurazione del backend per il provisioning dei volumi

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

Parametro	Descrizione	Predefinito
spaceAllocation	Allocazione dello spazio per Qtrees	"true"
spaceReserve	Modalità di prenotazione dello spazio; "none" (thin) o "volume" (thick)	"none"

Parametro	Descrizione	Predefinito
snapshotPolicy	policy di Snapshot da utilizzare	"none"
qosPolicy	Gruppo di policy QoS da assegnare ai volumi creati. Scegli uno tra qosPolicy o adaptiveQosPolicy per ogni pool di storage/backend	""
adaptiveQosPolicy	Gruppo di policy QoS adattivo da assegnare ai volumi creati. Scegli uno tra qosPolicy o adaptiveQosPolicy per ogni pool di storage/backend. Non supportato da ontap-nas-economy.	""
snapshotReserve	Percentuale di volume riservata alle snapshot	"0" se snapshotPolicy è "none", altrimenti ""
splitOnClone	Dividere un clone dal suo genitore al momento della creazione	"false"
encryption	Abilita NetApp Volume Encryption (NVE) sul nuovo volume; l'impostazione predefinita è false. NVE deve essere concesso in licenza e abilitato sul cluster per utilizzare questa opzione. Se NAE è abilitato sul backend, qualsiasi volume fornito in Trident sarà abilitato per NAE. Per ulteriori informazioni, fare riferimento a: <a href="#">"Come funziona Trident con NVE e NAE"</a> .	"false"
tieringPolicy	Policy di tiering da utilizzare "none"	
unixPermissions	Modalità per nuovi volumi	"777" per i volumi NFS; vuoto (non applicabile) per i volumi SMB
snapshotDir	Controlla l'accesso alla .snapshot directory	"true" per NFSv4 "false" per NFSv3
exportPolicy	Policy di esportazione da utilizzare	"default"
securityStyle	Stile di sicurezza per i nuovi volumi. NFS supporta mixed e unix stili di sicurezza. SMB supporta mixed e ntfs stili di sicurezza.	L'impostazione predefinita di NFS è unix. L'impostazione predefinita di SMB è ntfs.
nameTemplate	Modello per creare nomi di volume personalizzati.	""



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

### Esempi di provisioning del volume

Ecco un esempio con i valori predefiniti:

```

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

```

Per `ontap-nas` e `ontap-nas-flexgroups`, Trident ora utilizza un nuovo calcolo per garantire che il FlexVol sia dimensionato correttamente con la percentuale di `snapshotReserve` e il PVC. Quando l'utente richiede un PVC, Trident crea il FlexVol originale con più spazio utilizzando il nuovo calcolo. Questo calcolo garantisce che l'utente riceva lo spazio scrivibile richiesto nel PVC e non meno spazio di quanto richiesto. Prima della v21.07, quando l'utente richiedeva un PVC (ad esempio, 5 GiB), con la `snapshotReserve` al 50 per cento, otteneva solo 2,5 GiB di spazio scrivibile. Questo perché ciò che l'utente richiedeva era l'intero volume e `snapshotReserve` era una percentuale di quello. Con Trident 21.07, ciò che l'utente richiede è lo spazio scrivibile e Trident definisce il numero di `snapshotReserve` come percentuale dell'intero volume. Questo non si applica a `ontap-nas-economy`. Vedi il seguente esempio per capire come funziona:

Il calcolo è il seguente:

```

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

```

Per `snapshotReserve = 50%` e richiesta PVC = 5 GiB, la dimensione totale del volume è  $5/0,5 = 10$  GiB e la dimensione disponibile è 5 GiB, che è quanto richiesto dall'utente nella richiesta PVC. Il `volume show` 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 che hai creato prima dell'aggiornamento, dovresti ridimensionare i loro volumi affinché la modifica venga rilevata. Ad esempio, un PVC da 2 GiB con `snapshotReserve=50` in precedenza generava un volume che forniva 1 GiB di spazio scrivibile. Ridimensionando il volume a 3 GiB, ad esempio, si forniscono all'applicazione 3 GiB di spazio scrivibile su un volume da 6 GiB.

## Esempi di configurazione minima

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



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 ONTAP NAS economy

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

### Esempio di ONTAP NAS FlexGroup

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

## Esempio di MetroCluster

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

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

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

## Esempio di volumi SMB

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

## Esempio di autenticazione basata su certificato

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

```
---
version: 1
backendName: DefaultNASBackend
storageDriverName: ontap-nas
managementLIF: 10.0.0.1
dataLIF: 10.0.0.15
svm: nfs_svm
clientCertificate: ZXR0ZXJwYXB...ICMgJ3BhcGVyc2
clientPrivateKey: vciwKIyAgZG...0cnksIGRlc2NyaX
trustedCACertificate: zcyBbaG...b3Igb3duIGNsYXNz
storagePrefix: myPrefix_
```

## Esempio di policy di esportazione automatica

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

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

## Esempio di indirizzi IPv6

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

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

## Esempio di Amazon FSx for ONTAP che utilizza volumi SMB

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

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

## Esempio di configurazione del backend con nameTemplate

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

## Esempi di backend con pool virtuali

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

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

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

## Esempio di 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"
```

## Esempio di ONTAP NAS economy

```
---
version: 1
storageDriverName: ontap-nas-economy
managementLIF: 10.0.0.1
svm: svm_nfs
username: vsadmin
password: <password>
defaults:
  spaceReserve: none
  encryption: "false"
labels:
  store: nas_economy_store
  region: us_east_1
storage:
  - labels:
    department: finance
    creditpoints: "6000"
    zone: us_east_1a
    defaults:
      spaceReserve: volume
      encryption: "true"
      unixPermissions: "0755"
  - labels:
    protection: bronze
    creditpoints: "5000"
    zone: us_east_1b
    defaults:
      spaceReserve: none
      encryption: "true"
      unixPermissions: "0755"
  - labels:
    department: engineering
    creditpoints: "3000"
    zone: us_east_1c
    defaults:
      spaceReserve: none
      encryption: "true"
      unixPermissions: "0775"
  - labels:
    department: humanresource
    creditpoints: "2000"
    zone: us_east_1d
    defaults:
      spaceReserve: volume
```

```
encryption: "false"
unixPermissions: "0775"
```

## Mappa i backend a StorageClasses

Le seguenti definizioni di StorageClass si riferiscono a [Esempi di backend con pool virtuali](#). Utilizzando il campo `parameters.selector`, ciascuna StorageClass indica quali pool virtuali 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 nel `ontap-nas-flexgroup` backend. Questi sono gli unici pool che offrono 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 nel `ontap-nas-flexgroup` backend. Questi sono gli unici pool che offrono un livello di protezione diverso da gold.

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

- 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 storage per app di tipo `mysqldb`.

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

- The protection-silver-creditpoints-20k StorageClass verrà mappato sul terzo pool virtuale nel ontap-nas-flexgroup backend. Questo è l'unico pool che offre protezione di livello silver e 20000 creditpoints.

```
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 nel ontap-nas backend e sul secondo pool virtuale nel ontap-nas-economy backend. Queste sono le uniche offerte di pool con 5000 creditpoints.

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

Trident deciderà quale pool virtuale selezionare e garantirà che il requisito di storage sia soddisfatto.

### Aggiorna dataLIF dopo la configurazione iniziale

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

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



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

## Esempi di SMB sicuro

### Configurazione del backend con driver ontap-nas

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

### Configurazione backend con driver ontap-nas-economy

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

## Configurazione backend con pool di storage

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

## Esempio di storage class con driver ontap-nas

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



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

### Esempio di storage class con driver ontap-nas-economy

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

### Esempio di PVC con un singolo utente AD

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

### Esempio di PVC con più utenti AD

```

apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: my-test-pvc
  annotations:
    trident.netapp.io/smbShareAccessControl: |
      full_control:
        - tridentTestuser
        - tridentuser
        - tridentTestuser1
        - tridentuser1
      change:
        - tridentADuser
        - tridentADuser1
        - tridentADuser4
        - tridentTestuser2
      read:
        - tridentTestuser2
        - tridentTestuser3
        - tridentADuser2
        - tridentADuser3
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi

```

## Amazon FSx for NetApp ONTAP

### Usa Trident con Amazon FSx for NetApp ONTAP

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

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

Un file system è la risorsa principale in Amazon FSx, analoga a un ONTAP cluster on premises. All'interno di

ogni SVM puoi creare uno o più volumi, che sono contenitori di dati che memorizzano i file e le cartelle nel tuo file system. Con Amazon FSx for NetApp ONTAP verrà fornito come file system gestito nel cloud. Il nuovo tipo di file system è chiamato **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 eseguire il provisioning di volumi persistenti a blocchi e file supportati da ONTAP.

## Requisiti

Oltre a "[Requisiti di Trident](#)", per integrare FSx for ONTAP con Trident, è necessario:

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



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

## Considerazioni

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



Se si specifica un ruolo IAM per Trident, è possibile omettere di specificare i campi `apiRegion`, `apiKey` e `secretKey` a Trident esplicitamente. Per ulteriori informazioni, fare riferimento a "[Opzioni ed esempi di configurazione di FSx per ONTAP](#)".

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

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

```

defaults {
    find_multipaths no
}
blacklist {
    device {
        vendor "NVME"
        product "Amazon Elastic Block Store"
    }
}

```

## Autenticazione

Trident offre due modalità di autenticazione.

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



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

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

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

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

## Amazon Machine Images (AMI) testate

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

AMI	NAS	NAS-economy	iSCSI	iSCSI-economy
AL2023_x86_64_ST ANDARD	Sì	Sì	Sì	Sì
AL2_x86_64	Sì	Sì	Sì*	Sì*
BOTTLEROCKET_x 86_64	Sì**	Sì	N/A	N/A
AL2023_ARM_64_S TANDARD	Sì	Sì	Sì	Sì
AL2_ARM_64	Sì	Sì	Sì*	Sì*
BOTTLEROCKET_A RM_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 le AMI di cui è noto il funzionamento.

### Test eseguiti con:

- EKS versione: 1.32
- Metodo di installazione: Helm 25.06 e come AWS add-On 25.06
- Per NAS sono stati testati sia NFSv3 che NFSv4.1.
- Per SAN è stato testato solo iSCSI, non NVMe-oF.

### Test eseguiti:

- Crea: Storage Class, pvc, pod
- Elimina: pod, pvc (normale, qtree/lun – economy, NAS con backup AWS)

### Trova ulteriori informazioni

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

## Crea un ruolo IAM e un AWS Secret

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



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

### Crea un secret di AWS Secrets Manager

Poiché Trident emetterà API contro un FSx vserver per gestire lo storage per te, avrà bisogno di credenziali per farlo. Il modo sicuro per trasmettere tali credenziali è tramite un segreto AWS Secrets Manager. Pertanto, se non ne hai già uno, dovrai creare un segreto AWS Secrets Manager che contenga le credenziali per l'account vsadmin.

Questo esempio crea un segreto AWS Secrets Manager per archiviare le credenziali Trident CSI:

```
aws secretsmanager create-secret --name trident-secret --description
"Trident CSI credentials" \
  --secret-string
"{\"username\": \"vsadmin\", \"password\": \"<svmpassword>\"}"
```

## Crea policy IAM

Anche Trident necessita delle autorizzazioni AWS per funzionare correttamente. Pertanto, è necessario creare una policy che dia a Trident le autorizzazioni di cui ha bisogno.

I seguenti esempi creano una policy IAM utilizzando l'AWS CLI:

```
aws iam create-policy --policy-name AmazonFSxNCSIDriverPolicy --policy-
-document file://policy.json
  --description "This policy grants access to Trident CSI to FSxN and
  Secrets manager"
```

## Esempio di policy JSON:

```
{
  "Statement": [
    {
      "Action": [
        "fsx:DescribeFileSystems",
        "fsx:DescribeVolumes",
        "fsx:CreateVolume",
        "fsx:RestoreVolumeFromSnapshot",
        "fsx:DescribeStorageVirtualMachines",
        "fsx:UntagResource",
        "fsx:UpdateVolume",
        "fsx:TagResource",
        "fsx>DeleteVolume"
      ],
      "Effect": "Allow",
      "Resource": "*"
    },
    {
      "Action": "secretsmanager:GetSecretValue",
      "Effect": "Allow",
      "Resource": "arn:aws:secretsmanager:<aws-region>:<aws-account-
      id>:secret:<aws-secret-manager-name>*"
    }
  ],
  "Version": "2012-10-17"
}
```

## Crea Pod Identity o ruolo IAM per l'associazione dell'account di servizio (IRSA)

È possibile configurare un account di servizio Kubernetes per assumere un ruolo AWS Identity and Access Management (IAM) con EKS Pod Identity o IAM role for Service account association (IRSA). Tutti i Pod configurati per utilizzare l'account di servizio possono quindi accedere a qualsiasi servizio AWS a cui il ruolo ha

permessi di accesso.

## Identità del pod

Le associazioni Amazon EKS Pod Identity consentono di gestire le credenziali per le tue applicazioni, in modo simile a come i profili delle istanze Amazon EC2 forniscono credenziali alle istanze Amazon EC2.

### Installa Pod Identity sul tuo cluster EKS:

Puoi creare l'identità del Pod tramite la console AWS o utilizzando il seguente comando AWS CLI:

```
aws eks create-addon --cluster-name <EKS_CLUSTER_NAME> --addon-name
eks-pod-identity-agent
```

Per ulteriori informazioni, fare riferimento a ["Configura l'agente di identità del pod Amazon EKS"](#).

### Crea trust-relationship.json:

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

```
aws iam create-role \
  --role-name fsxn-csi-role --assume-role-policy-document file://trust-
relationship.json \
  --description "fsxn csi pod identity role"
```

### file trust-relationship.json:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "pods.eks.amazonaws.com"
      },
      "Action": [
        "sts:AssumeRole",
        "sts:TagSession"
      ]
    }
  ]
}
```

### Associare la policy del ruolo al ruolo IAM:

Associa il criterio di ruolo del passaggio precedente al ruolo IAM che è stato creato:

```
aws iam attach-role-policy \  
  --policy-arn arn:aws:iam::aws:111122223333:policy/fsxn-csi-policy \  
  --role-name fsxn-csi-role
```

### **Crea un'associazione di identità pod:**

Crea un'associazione di identità pod tra il ruolo IAM e il service account Trident (trident-controller)

```
aws eks create-pod-identity-association \  
  --cluster-name <EKS_CLUSTER_NAME> \  
  --role-arn arn:aws:iam::111122223333:role/fsxn-csi-role \  
  --namespace trident --service-account trident-controller
```

### **Ruolo IAM per l'associazione dell'account di servizio (IRSA)**

Utilizzando l'AWS CLI:

```
aws iam create-role --role-name AmazonEKS_FSxN_CSI_DriverRole \  
  --assume-role-policy-document file://trust-relationship.json
```

### **file trust-relationship.json:**

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Principal": {  
        "Federated": "arn:aws:iam::<account_id>:oidc-  
provider/<oidc_provider>"  
      },  
      "Action": "sts:AssumeRoleWithWebIdentity",  
      "Condition": {  
        "StringEquals": {  
          "<oidc_provider>:aud": "sts.amazonaws.com",  
          "<oidc_provider>:sub":  
"system:serviceaccount:trident:trident-controller"  
        }  
      }  
    }  
  ]  
}
```

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

- **<account\_id>** - ID del tuo account AWS
- **<oidc\_provider>** - L'OIDC del tuo cluster EKS. Puoi ottenere l'`oidc_provider` eseguendo:

```
aws eks describe-cluster --name my-cluster --query
"cluster.identity.oidc.issuer" \
  --output text | sed -e "s/^https://\///"
```

### Associare il ruolo IAM alla policy IAM:

Una volta creato il ruolo, associare il criterio (che è stato creato nel passaggio sopra) al ruolo utilizzando questo comando:

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

### Verifica che il provider OICD sia associato:

Verifica che il tuo provider OIDC sia associato al tuo cluster. Puoi verificarlo utilizzando questo comando:

```
aws iam list-open-id-connect-providers | grep $oidc_id | cut -d "/" -f4
```

Se l'output è vuoto, utilizzare il seguente comando per associare IAM OIDC al cluster:

```
eksctl utils associate-iam-oidc-provider --cluster $cluster_name
--approve
```

Se si utilizza `eksctl`, utilizzare il seguente esempio per creare un ruolo IAM per account di servizio in EKS:

```
eksctl create iamserviceaccount --name trident-controller --namespace
trident \
  --cluster <my-cluster> --role-name AmazonEKS_FSxN_CSI_DriverRole
--role-only \
  --attach-policy-arn <IAM-Policy ARN> --approve
```

## Installare Trident

Trident semplifica la gestione dello storage Amazon FSx for NetApp ONTAP in Kubernetes, consentendo a sviluppatori e amministratori di concentrarsi sulla distribuzione delle applicazioni.

Puoi installare Trident utilizzando uno dei seguenti metodi:

- Helm
- Componente aggiuntivo EKS

Se desideri utilizzare la funzionalità di snapshot, installa il componente aggiuntivo CSI snapshot controller. Consulta ["Abilita la funzionalità snapshot per i volumi CSI"](#) per ulteriori informazioni.

### **Installa Trident tramite helm**

## Identità del pod

### 1. Aggiungi il repository Trident Helm:

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

### 2. Installa Trident utilizzando il seguente esempio:

```
helm install trident-operator netapp-trident/trident-operator --version 100.2502.1 --namespace trident --create-namespace
```

Puoi utilizzare il comando `helm list` per rivedere i dettagli dell'installazione come nome, namespace, chart, stato, versione dell'app e numero di revisione.

```
helm list -n trident
```

NAME	NAMESPACE	REVISION	UPDATED
STATUS	CHART		APP VERSION
trident-operator	trident	1	2024-10-14
14:31:22.463122 +0300	IDT	deployed	trident-operator-100.2502.0
25.02.0			

## Associazione account di servizio (IRSA)

### 1. Aggiungi il repository Trident Helm:

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

### 2. Imposta i valori per **cloud provider** e **cloud identity**:

```
helm install trident-operator netapp-trident/trident-operator --version 100.2502.1 \ --set cloudProvider="AWS" \ --set cloudIdentity="'eks.amazonaws.com/role-arn:arn:aws:iam::<accountID>:role/<AmazonEKS_FSxN_CSI_DriverRole>'" \ --namespace trident \ --create-namespace
```

Puoi utilizzare il comando `helm list` per rivedere i dettagli dell'installazione come nome, namespace, chart, stato, versione dell'app e numero di revisione.

```
helm list -n trident
```

NAME	NAMESPACE	REVISION	UPDATED
STATUS	CHART		APP VERSION
trident-operator	trident	1	2024-10-14
14:31:22.463122	+0300 IDT	deployed	trident-operator-
100.2510.0	25.10.0		

Se si prevede di utilizzare iSCSI, assicurarsi che iSCSI sia abilitato sul computer client. Se si utilizza il sistema operativo AL2023 Worker node, è possibile automatizzare l'installazione del client iSCSI aggiungendo il parametro `node prep` nell'installazione di helm:



```
helm install trident-operator netapp-trident/trident-operator  
--version 100.2502.1 --namespace trident --create-namespace --  
set nodePrep={iscsi}
```

## Installa Trident tramite il componente aggiuntivo EKS

Il componente aggiuntivo Trident EKS include le patch di sicurezza più recenti, le correzioni di bug ed è convalidato da AWS per funzionare con Amazon EKS. Il componente aggiuntivo EKS consente di garantire costantemente che i cluster Amazon EKS siano sicuri e stabili e di ridurre la quantità di lavoro necessaria per installare, configurare e aggiornare i componenti aggiuntivi.

### Prerequisiti

Assicurati di avere quanto segue prima di configurare il componente aggiuntivo Trident per AWS EKS:

- Un account cluster Amazon EKS con abbonamento aggiuntivo
- Autorizzazioni AWS per l'AWS marketplace:  
"aws-marketplace:ViewSubscriptions",  
"aws-marketplace:Subscribe",  
"aws-marketplace:Unsubscribe
- Tipo AMI: Amazon Linux 2 (AL2\_x86\_64) o Amazon Linux 2 Arm(AL2\_ARM\_64)
- Tipo di nodo: AMD o ARM
- Un file system Amazon FSx per NetApp ONTAP esistente

### Abilita il componente aggiuntivo Trident per AWS

## Console di gestione

1. Apri la console Amazon EKS su <https://console.aws.amazon.com/eks/home#/clusters>.
2. Nel riquadro di navigazione a sinistra, seleziona **Clusters**.
3. Seleziona il nome del cluster per cui desideri configurare il componente aggiuntivo NetApp Trident CSI.
4. Seleziona **Componenti aggiuntivi** e poi seleziona **Ottieni altri componenti aggiuntivi**.
5. Seguire questi passaggi per selezionare il software add-on:
  - a. Scorri verso il basso fino alla sezione **AWS Marketplace add-ons** e digita **"Trident"** nella casella di ricerca.
  - b. Selezionare la check box nell'angolo in alto a destra della casella Trident by NetApp.
  - c. Seleziona **Next**.
6. Nella pagina delle impostazioni **Configura i componenti aggiuntivi selezionati**, eseguire le seguenti operazioni:



**Salta questi passaggi se utilizzi l'associazione Pod Identity.**

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

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

+

Se si seleziona **Override** per il metodo di risoluzione dei conflitti, una o più impostazioni del componente aggiuntivo esistente possono essere sovrascritte con le impostazioni dell'add-on Amazon EKS. Se non si abilita questa opzione e si verifica un conflitto con le impostazioni esistenti, l'operazione non riesce. È possibile utilizzare il messaggio di errore risultante per risolvere il conflitto. Prima di selezionare questa opzione, assicurarsi che l'add-on Amazon EKS non gestisca impostazioni che è necessario autogestire.

7. Scegli **Next**.
8. Nella pagina **Revisione e aggiunta**, scegliere **Crea**.

Al termine dell'installazione del software add-on, viene visualizzato il software add-on installato.

## AWS CLI

## 1. Crea il add-on.json file:

Per Pod Identity, utilizzare il seguente formato:



Utilizzare il

```
{
  "clusterName": "<eks-cluster>",
  "addonName": "netapp_trident-operator",
  "addonVersion": "v25.6.0-eksbuild.1",
}
```

Per l'autenticazione IRSA, utilizzare il seguente formato:

```
{
  "clusterName": "<eks-cluster>",
  "addonName": "netapp_trident-operator",
  "addonVersion": "v25.6.0-eksbuild.1",
  "serviceAccountRoleArn": "<role ARN>",
  "configurationValues": {
    "cloudIdentity": "'eks.amazonaws.com/role-arn: <role ARN>'",
    "cloudProvider": "AWS"
  }
}
```



Sostituisci <role ARN> con l'ARN del ruolo che è stato creato nel passaggio precedente.

## 2. Installare il Trident EKS add-on.

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

### eksctl

Il seguente esempio di comando installa il Trident EKS add-on:

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

## Aggiornare il software add-on Trident EKS

## Console di gestione

1. Apri la console Amazon EKS <https://console.aws.amazon.com/eks/home#/clusters>.
2. Nel riquadro di navigazione a sinistra, seleziona **Clusters**.
3. Selezionare il nome del cluster per cui si desidera aggiornare il software add-on NetApp Trident CSI.
4. Selezionare la scheda **Add-ons**.
5. Seleziona **Trident by NetApp** e poi seleziona **Modifica**.
6. Nella pagina **Configura Trident by NetApp**, procedere come segue:
  - a. Seleziona la **Version** che desideri utilizzare.
  - b. Espandi le **Impostazioni di configurazione opzionali** e modificalo secondo necessità.
  - c. Seleziona **Salva modifiche**.

## AWS CLI

Il seguente esempio aggiorna l'add-on EKS:

```
aws eks update-addon --cluster-name <eks_cluster_name> --addon-name
netapp_trident-operator --addon-version v25.6.0-eksbuild.1 \
  --service-account-role-arn <role-ARN> --resolve-conflict preserve \
  --configuration-values "{\"cloudIdentity\":
\"'eks.amazonaws.com/role-arn: <role ARN>'\"}"
```

## eksctl

- Controlla la versione corrente del tuo software add-on FSxN Trident CSI. Sostituisci `my-cluster` con il nome del tuo cluster.

```
eksctl get addon --name netapp_trident-operator --cluster my-cluster
```

### Esempio di output:

NAME	VERSION	STATUS	ISSUES
IAMROLE	UPDATE AVAILABLE	CONFIGURATION VALUES	
netapp_trident-operator	v25.6.0-eksbuild.1	ACTIVE	0
{"cloudIdentity":"'eks.amazonaws.com/role-arn: arn:aws:iam::139763910815:role/AmazonEKS_FSXN_CSI_DriverRole'"}			

- Aggiornare il software add-on alla versione riportata sotto UPDATE AVAILABLE nell'output del passaggio precedente.

```
eksctl update addon --name netapp_trident-operator --version
v25.6.0-eksbuild.1 --cluster my-cluster --force
```

Se si rimuove l' `--force` opzione e una qualsiasi delle impostazioni Amazon EKS add-on è in conflitto con le impostazioni esistenti, l'aggiornamento dell'Amazon EKS add-on non riesce; viene visualizzato un messaggio di errore per aiutarti a risolvere il conflitto. Prima di specificare questa opzione, assicurati che l'Amazon EKS add-on non gestisca impostazioni che devi gestire, perché tali impostazioni vengono sovrascritte con questa opzione. Per ulteriori informazioni su altre opzioni per questa impostazione, vedi ["Componenti aggiuntivi"](#). Per ulteriori informazioni sulla gestione dei campi Amazon EKS Kubernetes, vedi ["Gestione dei campi Kubernetes"](#).

### Disinstallare/rimuovere il Trident EKS add-on

Hai due opzioni per rimuovere un add-on di Amazon EKS:

- **Conserva il software add-on sul tuo cluster** – Questa opzione rimuove la gestione di qualsiasi impostazione da parte di Amazon EKS. Rimuove anche la possibilità per Amazon EKS di notificarti gli aggiornamenti e di aggiornare automaticamente l'add-on Amazon EKS dopo che hai avviato un aggiornamento. Tuttavia, conserva il software add-on sul tuo cluster. Questa opzione rende l'add-on un'installazione autogestita, invece che un add-on Amazon EKS. Con questa opzione, non c'è alcun downtime per l'add-on. Mantieni l' `--preserve` opzione nel comando per conservare l'add-on.
- **Rimuovere il software add-on interamente dal cluster** – NetApp consiglia di rimuovere l'add-on Amazon EKS dal cluster solo se non ci sono risorse sul cluster che dipendono da esso. Rimuovere l'opzione `--preserve` dal comando `delete` per rimuovere l'add-on.



Se al software add-on è associato un account IAM, l'account IAM non viene rimosso.

## Console di gestione

1. Apri la console Amazon EKS su <https://console.aws.amazon.com/eks/home#/clusters>.
2. Nel riquadro di navigazione sinistro, selezionare **Clusters**.
3. Selezionare il nome del cluster dal quale si desidera rimuovere il software add-on NetApp Trident CSI.
4. Selezionare la scheda **Add-ons** e poi scegliere **Trident by NetApp**.\*
5. Seleziona **Rimuovi**.
6. Nella finestra di dialogo **Remove netapp\_trident-operator confirmation**, procedere come segue:
  - a. Se si desidera che Amazon EKS smetta di gestire le impostazioni del software add-on, selezionare **Preserva sul cluster**. Eseguire questa operazione se si desidera conservare il software add-on sul cluster in modo da poter gestire autonomamente tutte le impostazioni del software add-on.
  - b. Immettere **netapp\_trident-operator**.
  - c. Seleziona **Rimuovi**.

## AWS CLI

Sostituire `my-cluster` con il nome del cluster, quindi eseguire il seguente comando.

```
aws eks delete-addon --cluster-name my-cluster --addon-name
netapp_trident-operator --preserve
```

## eksctl

Il seguente comando disinstalla il componente aggiuntivo Trident EKS:

```
eksctl delete addon --cluster K8s-arm --name netapp_trident-operator
```

## Configura il backend di storage

### Integrazione dei driver ONTAP SAN e NAS

Per creare un backend di storage, 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 con esso. Il seguente esempio mostra come definire uno storage basato su NAS e utilizzare un AWS secret per memorizzare le credenziali dell'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 Trident Backend Configuration (TBC):

- Crea la configurazione del backend Trident (TBC) dal file yaml ed esegui il seguente comando:

```
kubectl create -f backendconfig.yaml -n trident
```

```
tridentbackendconfig.trident.netapp.io/backend-tbc-ontap-nas created
```

- Verificare che la configurazione del backend Trident (TBC) sia stata creata correttamente:

```
Kubectl get tbc -n trident
```

NAME	BACKEND NAME	BACKEND UUID
PHASE	STATUS	
backend-tbc-ontap-nas	tbc-ontap-nas	933e0071-66ce-4324-
b9ff-f96d916ac5e9	Bound	Success

### 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 storage a blocchi.
- `ontap-nas`: Ogni PV fornito è un volume completo Amazon FSx for NetApp ONTAP. Consigliato per NFS e SMB.
- `ontap-san-economy`: Ogni PV fornito è una LUN con un numero configurabile di LUN per Amazon FSx for NetApp ONTAP volume.
- `ontap-nas-economy`: Ogni PV fornito è un qtree, con un numero configurabile di qtree per Amazon FSx for NetApp ONTAP volume.
- `ontap-nas-flexgroup`: Ogni PV fornito è un volume completo Amazon FSx for NetApp ONTAP FlexGroup.

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

Una volta creato il file di configurazione, eseguire 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

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

Parametro	Descrizione	Esempio
version		Sempre 1
storageDriverName	Nome del driver di archiviazione	ontap-nas, ontap-nas-economy, ontap-nas-flexgroup, ontap-san, ontap-san-economy
backendName	Nome personalizzato o lo storage backend	Nome driver + "_" + dataLIF
managementLIF	Indirizzo IP di un cluster o di una LIF di gestione SVM. È possibile specificare un fully-qualified domain name (FQDN). Può essere impostato per 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 si fornisce fsxFilesystemID sotto il campo aws, non è necessario fornire managementLIF perché Trident recupera le informazioni SVM managementLIF da AWS. Quindi, è necessario fornire le credenziali per un utente sotto l'SVM (ad esempio: vsadmin) e l'utente deve avere il ruolo vsadmin.	"10.0.0.1", "[2001:1234:abcd::fefe]"

Parametro	Descrizione	Esempio
dataLIF	<p>Indirizzo IP del protocollo LIF.</p> <p><b>ONTAP NAS drivers:</b> NetApp consiglia di specificare dataLIF. Se non fornito, Trident recupera i dataLIF dall'SVM. È possibile specificare un fully-qualified domain name (FQDN) da utilizzare per le operazioni di montaggio NFS, consentendo di creare un round-robin DNS per il bilanciamento del carico su più dataLIF. Può essere modificato dopo l'impostazione iniziale. Fare riferimento a <b>ONTAP SAN drivers:</b> non specificare per iSCSI. Trident utilizza ONTAP Selective LUN Map per individuare i LIF iSCSI necessari per stabilire una sessione multipath. Viene generato un avviso se dataLIF è definito esplicitamente. 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>	
autoExportPolicy	<p>Abilita la creazione e l'aggiornamento automatici delle policy di esportazione [Boolean]. Utilizzando le opzioni <code>autoExportPolicy</code> e <code>autoExportCIDRs</code>, Trident può gestire automaticamente le policy di esportazione.</p>	false
autoExportCIDRs	<p>Elenco di CIDR in base ai quali filtrare gli IP dei nodi Kubernetes quando <code>autoExportPolicy</code> è abilitato. Utilizzando le <code>autoExportPolicy</code> e <code>autoExportCIDRs</code> opzioni, Trident può gestire automaticamente le policy di esportazione.</p>	"["0.0.0.0/0", ":::/0"]"
labels	<p>Set di etichette arbitrarie in formato JSON da applicare ai volumi</p>	""
clientCertificate	<p>Valore codificato in Base64 del certificato client. Utilizzato per l'autenticazione basata su certificato</p>	""

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. Facoltativo. Utilizzato per l'autenticazione basata su certificato.	""
username	Nome utente per connettersi al cluster o alla SVM. Utilizzato per l'autenticazione basata su credenziali. Ad esempio, vsadmin.	
password	Password per connettersi al cluster o alla SVM. Utilizzata per l'autenticazione basata sulle credenziali.	
svm	Macchina virtuale di storage da utilizzare	Derivato se viene specificato un SVM managementLIF.
storagePrefix	Prefisso utilizzato durante il provisioning di nuovi volumi nell'SVM. Non può essere modificato dopo la creazione. Per aggiornare questo parametro, sarà necessario creare un nuovo backend.	trident
limitAggregateUsage	<b>Non specificare per Amazon FSx per NetApp ONTAP.</b> I <code>fsxadmin</code> e <code>vsadmin</code> forniti non contengono le autorizzazioni necessarie per recuperare l'utilizzo aggregato e limitarlo tramite Trident.	Non utilizzare.
limitVolumeSize	Il provisioning fallisce se la dimensione del volume richiesto supera questo valore. Limita inoltre la dimensione massima dei volumi che gestisce per qtree e LUN, e l'`qtreesPerFlexvol` opzione consente di personalizzare il numero massimo di qtree per FlexVol volume	"" (non applicato di default)
lunsPerFlexvol	Numero massimo di LUN per FlexVol volume, deve essere compreso nell'intervallo [50, 200]. Solo SAN.	"100"

Parametro	Descrizione	Esempio
debugTraceFlags	Flag di debug da utilizzare durante la risoluzione dei problemi. Ad esempio, {"api":false, "method":true} non utilizzare debugTraceFlags a meno che non si stia risolvendo un problema e si richieda un dump dettagliato del log.	null
nfsMountOptions	Elenco separato da virgole delle opzioni di montaggio NFS. Le opzioni di montaggio per i volumi persistenti Kubernetes sono normalmente specificate nelle classi di storage, ma se non vengono specificate opzioni di montaggio in una classe di storage, Trident utilizzerà le opzioni di montaggio specificate nel file di configurazione del backend di storage. Se non vengono specificate opzioni di montaggio nella classe di storage o nel file di configurazione, Trident non imposterà alcuna opzione di montaggio su un volume persistente associato.	""
nasType	Configura la creazione di volumi NFS o SMB. Le opzioni sono <code>nfs</code> , <code>smb</code> o <code>null</code> . <b>Deve essere impostato su <code>smb</code> per i volumi SMB.</b> Impostando su <code>null</code> , vengono creati di default volumi NFS.	<code>nfs</code>
qtreesPerFlexvol	Numero massimo di qtree per volume FlexVol, deve essere compreso nell'intervallo [50, 300]	"200"
smbShare	È possibile specificare uno dei seguenti parametri: il nome di una condivisione SMB creata tramite Microsoft Management Console o ONTAP CLI oppure un nome che consenta a Trident di creare la condivisione SMB. Questo parametro è obbligatorio per Amazon FSx for ONTAP backends.	<code>smb-share</code>

Parametro	Descrizione	Esempio
useREST	Parametro booleano per utilizzare le API REST di ONTAP. Se impostato su <code>true</code> , Trident utilizzerà le API REST di ONTAP per comunicare con il backend. Questa funzione richiede ONTAP 9.11.1 e versioni successive. Inoltre, il ruolo di login ONTAP utilizzato deve avere accesso all'applicazione <code>ontap</code> . Questo è soddisfatto dai ruoli predefiniti <code>vsadmin</code> e <code>cluster-admin</code> .	<code>false</code>
aws	È possibile specificare quanto segue nel file di configurazione per AWS FSx for 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.	"" "" ""
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> : Impostare su <code>awsarn</code> . Fare riferimento a <a href="#">"Crea un segreto AWS Secrets Manager"</a> per ulteriori informazioni.	

## Opzioni di configurazione del backend per il provisioning dei volumi

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

Parametro	Descrizione	Predefinito
<code>spaceAllocation</code>	Allocazione dello spazio per le LUN	<code>true</code>
<code>spaceReserve</code>	Modalità di prenotazione dello spazio; "none" (thin) o "volume" (thick)	<code>none</code>
<code>snapshotPolicy</code>	policy di Snapshot da utilizzare	<code>none</code>

Parametro	Descrizione	Predefinito
qosPolicy	Gruppo di policy QoS da assegnare ai volumi creati. Scegliere uno tra qosPolicy o adaptiveQosPolicy per ogni pool di storage o backend. L'utilizzo di gruppi di policy QoS con Trident richiede ONTAP 9.8 o versioni successive. È consigliabile utilizzare un gruppo di policy QoS non condiviso e assicurarsi che il gruppo di policy venga applicato a ciascun componente singolarmente. Un gruppo di policy QoS condiviso impone il limite massimo per il throughput di tutti i carichi di lavoro.	""
adaptiveQosPolicy	Gruppo di policy QoS adattivo da assegnare ai volumi creati. Scegli uno tra qosPolicy o adaptiveQosPolicy per ogni pool di storage o backend. Non supportato da ontap-nas-economy.	""
snapshotReserve	Percentuale di volume riservata per gli snapshot "0"	Se snapshotPolicy è none, else ""
splitOnClone	Dividere un clone dal suo genitore al momento della creazione	false
encryption	Abilita NetApp Volume Encryption (NVE) sul nuovo volume; l'impostazione predefinita è false. NVE deve essere concesso in licenza e abilitato sul cluster per utilizzare questa opzione. Se NAE è abilitato sul backend, qualsiasi volume fornito in Trident sarà abilitato per NAE. Per ulteriori informazioni, fare riferimento a: <a href="#">"Come funziona Trident con NVE e NAE"</a> .	false
luksEncryption	Abilita la crittografia LUKS. Fare riferimento a <a href="#">"Usa Linux Unified Key Setup (LUKS)"</a> . Solo SAN.	""
tieringPolicy	Criterio di tiering da utilizzare none	
unixPermissions	Modalità per nuovi volumi. <b>Lasciare vuoto per volumi SMB.</b>	""
securityStyle	Stile di sicurezza per i nuovi volumi. NFS supporta mixed e unix stili di sicurezza. SMB supporta mixed e ntfs stili di sicurezza.	L'impostazione predefinita di NFS è unix. L'impostazione predefinita di SMB è ntfs.

## Effettuare il provisioning dei volumi SMB

È possibile eseguire il provisioning dei volumi SMB utilizzando il `ontap-nas` driver. Prima di completare [Integrazione dei driver ONTAP SAN e NAS](#) completare questi passaggi: "[Prepararsi al provisioning dei volumi SMB](#)".

## Configura una storage class e un PVC

Configura un oggetto Kubernetes StorageClass e crea la storage class per istruire Trident su come effettuare il provisioning dei volumi. Crea un PersistentVolumeClaim (PVC) che utilizza il Kubernetes StorageClass configurato per richiedere l'accesso al PV. Puoi quindi montare il PV su un pod.

### Creare una storage class

#### Configura un oggetto Kubernetes StorageClass

L' "[Oggetto Kubernetes StorageClass](#)" oggetto identifica Trident come il provisioner utilizzato per quella classe e istruisce Trident su come effettuare il provisioning di un volume. Usa questo esempio per configurare la Storageclass per i volumi che utilizzano NFS (consulta la sezione [Attributi di Trident](#) qui sotto 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"
```

Utilizza questo esempio per configurare Storageclass per 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 eseguire il provisioning di volumi NFSv3 su AWS Bottlerocket, aggiungete il necessario `mountOptions` alla storage class:

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontap-gold
provisioner: csi.trident.netapp.io
parameters:
  backendType: "ontap-nas"
  media: "ssd"
  provisioningType: "thin"
  snapshots: "true"
mountOptions:
  - nfsvers=3
  - nolock
```

Fate riferimento a ["Oggetti Kubernetes e Trident"](#) per i dettagli su come le classi di storage interagiscono con PersistentVolumeClaim e sui parametri per controllare come Trident effettua il provisioning dei volumi.

### Creare una storage class

#### Passaggi

1. Si tratta di un oggetto Kubernetes, quindi usa `kubectl` per crearlo in Kubernetes.

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

2. Ora dovresti vedere una classe di storage **basic-csi** sia in Kubernetes che in Trident, e Trident dovrebbe aver rilevato i pool sul backend.

```
kubectl get sc basic-csi
```

NAME	PROVISIONER	AGE
basic-csi	csi.trident.netapp.io	15h

### Crea il PVC

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

Il PVC può essere configurato per richiedere storage di una certa dimensione o modalità di accesso. Utilizzando il StorageClass associato, l'amministratore del cluster può controllare più della sola dimensione e modalità di accesso della PersistentVolume, come ad esempio le prestazioni o il livello di servizio.

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

## Esempi di manifest

### PersistentVolumeClaim manifesti di esempio

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
```

#### Esempio di PVC utilizzando iSCSI

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

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

## Crea PVC

### Passaggi

1. Crea il PVC.

```
kubectl create -f pvc.yaml
```

## 2. Verificare lo stato del PVC.

```
kubectl get pvc
```

```
NAME          STATUS  VOLUME      CAPACITY  ACCESS  MODES  STORAGECLASS  AGE
pvc-storage  Bound  pv-name  2Gi      RWO                               5m
```

Fate riferimento a ["Oggetti Kubernetes e Trident"](#) per i dettagli su come le classi di storage interagiscono con `PersistentVolumeClaim` e sui parametri per controllare come Trident effettua il provisioning dei volumi.

### Attributi di Trident

Questi parametri determinano quali pool di storage gestiti da Trident devono essere utilizzati per effettuare il provisioning dei volumi di un determinato tipo.

Attributo	Tipo	Valori	Offerta	Richiesta	Supportato da
media <sup>1</sup>	stringa	hdd, hybrid, ssd	Il pool contiene supporti di questo tipo; ibrido significa entrambi	Tipo di media specificato	ontap-nas, ontap-nas-economy, ontap-nas-flexgroup, ontap-san, solidfire-san
provisioningType	stringa	sottile, spesso	Il pool supporta questo metodo di provisioning	Metodo di provisioning specificato	spesso: tutti ontap; sottile: tutti ontap & solidfire-san
backendType	stringa	ontap-nas, ontap-nas-economy, ontap-nas-flexgroup, ontap-san, solidfire-san, azure-netapp-files, ontap-san-economy	Il pool appartiene a questo tipo di backend	Backend specificato	Tutti i driver
istantanee	bool	vero, falso	Il pool supporta volumi con snapshot	Volume con snapshot abilitato	ontap-nas, ontap-san, solidfire-san
cloni	bool	vero, falso	Il pool supporta la clonazione dei volumi	Volume con cloni abilitati	ontap-nas, ontap-san, solidfire-san

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

<sup>1</sup>: Non supportato dai sistemi ONTAP Select

## Distribuisci l'applicazione di esempio

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

### Passaggi

1. Monta il volume in un pod.

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

Questi esempi mostrano configurazioni di base per collegare il PVC a un pod: **Configurazione di base:**

```
kind: Pod
apiVersion: v1
metadata:
  name: pv-pod
spec:
  volumes:
  - name: pv-storage
    persistentVolumeClaim:
      claimName: basic
  containers:
  - name: pv-container
    image: nginx
    ports:
    - containerPort: 80
      name: "http-server"
    volumeMounts:
    - mountPath: "/my/mount/path"
      name: pv-storage
```



Puoi monitorare l'avanzamento usando `kubectl get pod --watch`.

2. Verificare che il volume sia montato su `/my/mount/path`.

```
kubectl exec -it pv-pod -- df -h /my/mount/path
```

```
Filesystem                                Size
Used Avail Use% Mounted on
192.168.188.78:/trident_pvc_ae45ed05_3ace_4e7c_9080_d2a83ae03d06 1.1G
320K 1.0G 1% /my/mount/path
```

Ora puoi eliminare il Pod. L'applicazione Pod non esisterà più, ma il volume rimarrà.

```
kubectl delete pod pv-pod
```

## Configura il componente aggiuntivo Trident EKS su un cluster EKS

NetApp Trident semplifica la gestione dello storage Amazon FSx for NetApp ONTAP in Kubernetes, consentendo a sviluppatori e amministratori di concentrarsi sulla distribuzione delle applicazioni. Il componente aggiuntivo NetApp Trident EKS include le patch di sicurezza più recenti, le correzioni di bug ed è convalidato da AWS per funzionare con Amazon EKS. Il componente aggiuntivo EKS consente di garantire costantemente che i cluster Amazon EKS siano sicuri e stabili e di ridurre la quantità di lavoro necessaria per installare, configurare e aggiornare i componenti aggiuntivi.

### Prerequisiti

Assicurati di avere quanto segue prima di configurare il componente aggiuntivo Trident per AWS EKS:

- Un account cluster Amazon EKS con autorizzazioni per utilizzare i componenti aggiuntivi. Fare riferimento a ["Componenti aggiuntivi Amazon EKS"](#).
- Autorizzazioni AWS per l'AWS marketplace:  
"aws-marketplace:ViewSubscriptions",  
"aws-marketplace:Subscribe",  
"aws-marketplace:Unsubscribe"
- Tipo AMI: Amazon Linux 2 (AL2\_x86\_64) o Amazon Linux 2 Arm(AL2\_ARM\_64)
- Tipo di nodo: AMD o ARM
- Un file system Amazon FSx per NetApp ONTAP esistente

### Passaggi

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

2. Nel tuo 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 right, 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 [link].' Below this is a 'Cluster info' section with a table:

<b>Status</b> Active	<b>Kubernetes version</b> 1.30	<b>Support period</b> Standard support until July 28, 2025	<b>Provider</b> EKS
-------------------------	-----------------------------------	---	------------------------

Below the table are 'Cluster health issues' (0) and 'Upgrade insights' (0). A navigation bar includes 'Overview', 'Resources', 'Compute', 'Networking', 'Add-ons 1', 'Access', 'Observability', 'Update history', and 'Tags'. A notification says 'New versions are available for 1 add-on.' The 'Add-ons (3)' section has a search bar and filters for 'Any category', 'Any status', and '3 matches'.

3. Vai su **AWS Marketplace add-ons** e scegli la categoria *storage*.

The screenshot shows the 'AWS Marketplace add-ons (1)' page. It includes a search bar, filtering options for 'Any category', 'NetApp, Inc.', and 'Any pricing model', and a 'Clear filters' button. A 'NetApp, Inc.' filter is active. The main content area displays the 'NetApp Trident' add-on with a checkbox for selection. The add-on details are as follows:

<b>Category</b> storage	<b>Listed by</b> NetApp, Inc. [link]	<b>Supported versions</b> 1.31, 1.30, 1.29, 1.28, 1.27, 1.26, 1.25, 1.24, 1.23	<b>Pricing starting at</b> View pricing details [link]
----------------------------	---	---	---

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 dell'add-on.

## Configure selected add-ons settings

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

### NetApp Trident

Listed by **NetApp** | Category storage | Status Ready to install [Remove add-on](#)

**You're subscribed to this software** [View subscription](#) ×  
You can view the terms and pricing details for this product or choose another offer if one is available.

Version  
Select the version for this add-on.  
v25.6.0-eksbuild.1 ▾

► Optional configuration settings

[Cancel](#) [Previous](#) [Next](#)

6. Configura 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	<span>✓</span> 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 (IAM roles for service account), fare riferimento ai passaggi di configurazione aggiuntivi "qui".

8. Seleziona **Create**.

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

The screenshot shows the AWS Marketplace 'Add-ons' page. At the top, there's a search bar with 'netapp' entered and a '1 match' indicator. Below the search bar, the 'NetApp Trident' add-on is listed. It has a status of 'Active' (indicated by a green checkmark icon). The category is 'storage', the version is 'v24.10.0-eksbuild.1', and the IAM role for service account (IRSA) is 'Not set'. There are buttons for 'View details', 'Edit', 'Remove', and 'Get more add-ons'. A 'View subscription' button is also visible at the bottom right of the add-on card.

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

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

11. Continua la configurazione e configura il backend di storage. Per informazioni, vedi "[Configura il backend di storage](#)".

## Installa/disinstalla il componente aggiuntivo Trident EKS tramite CLI

### Installa il componente aggiuntivo Trident EKS di NetApp utilizzando la CLI:

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

```
eksctl create addon --cluster clusterName --name netapp_trident-operator  
--version v25.6.0-eksbuild.1 (con una versione dedicata)
```

Il seguente comando di esempio installa il componente aggiuntivo Trident EKS versione 25.6.1:

```
eksctl create addon --cluster clusterName --name netapp_trident-operator  
--version v25.6.1-eksbuild.1 (con una versione dedicata)
```

Il seguente comando di esempio installa il componente aggiuntivo Trident EKS versione 25.6.2:

```
eksctl create addon --cluster clusterName --name netapp_trident-operator  
--version v25.6.2-eksbuild.1 (con una versione dedicata)
```

### Disinstalla il componente aggiuntivo NetApp Trident EKS utilizzando la CLI:

Il seguente comando disinstalla il componente aggiuntivo Trident EKS:

```
eksctl delete addon --cluster K8s-arm --name netapp_trident-operator
```

## Crea backend con kubectl

Un backend definisce la relazione tra Trident e un sistema storage. Indica a Trident come comunicare con quel sistema storage e come Trident deve eseguire il provisioning dei volumi da esso. Dopo l'installazione di Trident, il passo successivo è creare un backend. La `TridentBackendConfig` Custom Resource Definition (CRD) consente di creare e gestire i backend di Trident direttamente attraverso l'interfaccia di Kubernetes. Puoi farlo

utilizzando `kubectl` o lo strumento CLI equivalente per la tua distribuzione Kubernetes.

## TridentBackendConfig

`TridentBackendConfig` (`tbc`, `tbconfig`, `tbackendconfig`) è un CRD frontend e namespaced che consente di gestire i backend Trident utilizzando `kubectl`. Gli amministratori di Kubernetes e dello storage possono ora creare e gestire i backend direttamente attraverso la CLI di Kubernetes senza richiedere un'utility a riga di comando dedicata (`tridentctl`).

Alla creazione di un `TridentBackendConfig` oggetto, avviene quanto segue:

- Un backend viene creato automaticamente da Trident in base alla configurazione che fornisci. Questo è rappresentato internamente come un `TridentBackend` (`tbe`, `tridentbackend`) CR.
- Il `TridentBackendConfig` è legato in modo univoco a un `TridentBackend` che è stato creato da Trident.

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



`TridentBackend` I CR sono creati automaticamente da Trident. Non dovresti modificarli. Se vuoi apportare aggiornamenti ai backend, fallo modificando l'oggetto `TridentBackendConfig`.

Vedere il seguente esempio per il formato del `TridentBackendConfig` CR:

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-ontap-san
spec:
  version: 1
  backendName: ontap-san-backend
  storageDriverName: ontap-san
  managementLIF: 10.0.0.1
  dataLIF: 10.0.0.2
  svm: trident_svm
  credentials:
    name: backend-tbc-ontap-san-secret
```

È anche possibile dare un'occhiata agli esempi nella directory "[trident-installer](#)" per configurazioni di esempio per la piattaforma di storage/servizio desiderato.

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

La `spec` sezione include anche `credentials` e `deletionPolicy` campi, che sono stati introdotti di recente nella `TridentBackendConfig` CR:

- `credentials`: Questo parametro è un campo obbligatorio e contiene le credenziali utilizzate per l'autenticazione con il sistema storage/servizio. Questo è impostato su un Kubernetes Secret creato dall'utente. Le credenziali non possono essere passate in testo normale e genereranno un errore.
- `deletionPolicy`: Questo campo definisce cosa dovrebbe accadere quando `TridentBackendConfig` viene eliminato. Può assumere uno dei due valori possibili:
  - `delete`: Ciò comporta l'eliminazione sia di `TridentBackendConfig` CR che del backend associato. Questo è il valore predefinito.
  - `retain`: Quando un `TridentBackendConfig` CR viene eliminato, la definizione del backend sarà ancora presente e potrà essere gestita con `tridentctl`. Impostando la policy di eliminazione su `retain`, gli utenti potranno effettuare il downgrade a una versione precedente (pre-21.04) e mantenere i backend creati. Il valore di questo campo può essere aggiornato dopo che un `TridentBackendConfig` è stato creato.



Il nome di un backend viene impostato usando `spec.backendName`. Se non specificato, il nome del backend viene impostato sul nome dell' `TridentBackendConfig` oggetto (`metadata.name`). Si consiglia di impostare esplicitamente i nomi dei backend usando `spec.backendName`.



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

## Panoramica dei passaggi

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

1. Crea un "[Kubernetes Secret](#)". Il secret contiene le credenziali di cui Trident ha bisogno per comunicare con il cluster/servizio di storage.
2. Crea un `TridentBackendConfig` oggetto. Questo contiene informazioni specifiche sul cluster/servizio di storage e fa riferimento al secret creato nel passaggio precedente.

Dopo aver creato un backend, puoi osservarne lo stato utilizzando `kubectl get tbc <tbc-name> -n <trident-namespace>` e raccogliere ulteriori dettagli.

## Passaggio 1: crea un Kubernetes Secret

Crea un Secret che contiene le credenziali di accesso per il backend. Questo è univoco per ogni servizio/piattaforma di storage. Ecco un esempio:

```
kubectl -n trident create -f backend-tbc-ontap-san-secret.yaml
```

```

apiVersion: v1
kind: Secret
metadata:
  name: backend-tbc-ontap-san-secret
type: Opaque
stringData:
  username: cluster-admin
  password: password

```

Questa tabella riassume i campi che devono essere inclusi nel Secret per ogni storage platform:

Descrizione dei campi segreti della storage platform	Segreto	Descrizione dei campi
Azure NetApp Files	clientID	L'ID client da una registrazione dell'app
Element (NetApp HCI/SolidFire)	Punto finale	MVIP per il SolidFire cluster con credenziali tenant
ONTAP	nome utente	Nome utente per connettersi al cluster/SVM. Utilizzato per l'autenticazione basata sulle credenziali
ONTAP	password	Password per connettersi al cluster/SVM. Utilizzata per l'autenticazione basata sulle credenziali
ONTAP	clientPrivateKey	Valore codificato in Base64 della chiave privata del client. Utilizzato per l'autenticazione basata su certificato
ONTAP	chapUsername	Nome utente in entrata. Obbligatorio se useCHAP=true. Per ontap-san e ontap-san-economy
ONTAP	chapInitiatorSecret	Segreto initiator CHAP. Richiesto se useCHAP=true. Per ontap-san e ontap-san-economy

Descrizione dei campi segreti della storage platform	Segreto	Descrizione dei campi
ONTAP	chapTargetUsername	Nome utente di destinazione. Obbligatorio se useCHAP=true. Per ontap-san e ontap-san-economy
ONTAP	chapTargetInitiatorSecret	Segreto dell'iniziatore di destinazione CHAP. Obbligatorio se useCHAP=true. Per ontap-san e ontap-san-economy

Il Secret creato in questo passaggio verrà referenziato nel campo `spec.credentials` dell'oggetto `TridentBackendConfig` che viene creato nel passaggio successivo.

## Passaggio 2: crea la `TridentBackendConfig` CR

Ora sei pronto per creare il tuo `TridentBackendConfig` CR. In questo esempio, un backend che utilizza il `ontap-san` driver viene creato utilizzando l'oggetto `TridentBackendConfig` mostrato di seguito:

```
kubectl -n trident create -f backend-tbc-ontap-san.yaml
```

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-ontap-san
spec:
  version: 1
  backendName: ontap-san-backend
  storageDriverName: ontap-san
  managementLIF: 10.0.0.1
  dataLIF: 10.0.0.2
  svm: trident_svm
  credentials:
    name: backend-tbc-ontap-san-secret
```

## Fase 3: verificare lo stato del `TridentBackendConfig` CR

Ora che hai creato la `TridentBackendConfig` CR, puoi verificarne lo stato. Guarda il seguente esempio:

```
kubectl -n trident get tbc backend-tbc-ontap-san
NAME                                BACKEND NAME                BACKEND UUID
PHASE    STATUS
backend-tbc-ontap-san    ontap-san-backend          8d24fce7-6f60-4d4a-8ef6-
bab2699e6ab8    Bound    Success
```

Un backend è stato creato correttamente e associato al `TridentBackendConfig` CR.

La fase può assumere uno dei seguenti valori:

- **Bound:** Il `TridentBackendConfig` CR è associato a un backend e tale backend contiene `configRef` impostato sull' `TridentBackendConfig` uid del CR.
- **Unbound:** Rappresentato usando `""`. L' `TridentBackendConfig` oggetto non è vincolato a un backend. Tutte le nuove `TridentBackendConfig` CR sono in questa fase per impostazione predefinita. Dopo il cambio di fase, non può tornare a Unbound.
- **Deleting:** Il `TridentBackendConfig` CR `deletionPolicy` è stato impostato per l'eliminazione. Quando il `TridentBackendConfig` CR viene eliminato, passa allo stato Eliminazione in corso.
  - Se non esistono persistent volume claims (PVC) sul backend, l'eliminazione del `TridentBackendConfig` comporterà che Trident elimini sia il backend sia il `TridentBackendConfig` CR.
  - Se uno o più PVC sono presenti sul backend, questo entra in uno stato di eliminazione. Il `TridentBackendConfig` CR successivamente entra anch'esso in fase di eliminazione. Il backend e `TridentBackendConfig` vengono eliminati solo dopo che tutti i PVC sono stati eliminati.
- **Lost:** Il backend associato al `TridentBackendConfig` CR è stato eliminato accidentalmente o deliberatamente e il `TridentBackendConfig` CR contiene ancora un riferimento al backend eliminato. Il `TridentBackendConfig` CR può comunque essere eliminato indipendentemente dal `deletionPolicy` valore.
- **Unknown:** Trident non è in grado di determinare lo stato o l'esistenza del backend associato al `TridentBackendConfig` CR. Ad esempio, se il server API non risponde o se il `tridentbackends.trident.netapp.io` CRD è mancante. Ciò potrebbe richiedere un intervento.

A questo punto, un backend è stato creato con successo! Ci sono diverse operazioni che possono essere gestite in aggiunta, come ["aggiornamenti del backend ed eliminazioni del backend"](#).

## (Facoltativo) Passaggio 4: Ottieni maggiori dettagli

Puoi eseguire il seguente comando per ottenere maggiori informazioni sul tuo backend:

```
kubectl -n trident get tbc backend-tbc-ontap-san -o wide
```

NAME	BACKEND NAME	BACKEND UUID		
PHASE	STATUS	STORAGE DRIVER	DELETION POLICY	
backend-tbc-ontap-san	ontap-san-backend	8d24fce7-6f60-4d4a-8ef6-		
bab2699e6ab8	Bound	Success	ontap-san	delete

Inoltre, è possibile ottenere anche un dump YAML/JSON di `TridentBackendConfig`.

```
kubectl -n trident get tbc backend-tbc-ontap-san -o yaml
```

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  creationTimestamp: 2021-04-21T20:45:11Z
  finalizers:
    - trident.netapp.io
  generation: 1
  name: backend-tbc-ontap-san
  namespace: trident
  resourceVersion: "947143"
  uid: 35b9d777-109f-43d5-8077-c74a4559d09c
spec:
  backendName: ontap-san-backend
  credentials:
    name: backend-tbc-ontap-san-secret
  managementLIF: 10.0.0.1
  dataLIF: 10.0.0.2
  storageDriverName: ontap-san
  svm: trident_svm
  version: 1
status:
  backendInfo:
    backendName: ontap-san-backend
    backendUUID: 8d24fce7-6f60-4d4a-8ef6-bab2699e6ab8
  deletionPolicy: delete
  lastOperationStatus: Success
  message: Backend 'ontap-san-backend' created
  phase: Bound
```

`backendInfo` contiene il `backendName` e il `backendUUID` del backend che è stato creato in risposta al `TridentBackendConfig` CR. Il `lastOperationStatus` campo rappresenta lo stato dell'ultima operazione del `TridentBackendConfig` CR, che può essere attivata dall'utente (ad esempio, l'utente ha modificato qualcosa in `spec`) o attivata da Trident (ad esempio, durante i riavvii di Trident). Può essere `Success` o `Failed`. `phase` rappresenta lo stato della relazione tra il `TridentBackendConfig` CR e il backend. Nell'esempio

sopra, `phase` ha il valore `Bound`, il che significa che il `TridentBackendConfig` CR è associato al backend.

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



Non è possibile aggiornare o eliminare un backend che contiene un oggetto associato `TridentBackendConfig` utilizzando `tridentctl`. Per comprendere i passaggi necessari per passare tra `tridentctl` e `TridentBackendConfig`, ["vedi qui"](#).

## Gestisci i backend

### Esegui la gestione del backend con kubectl

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

#### Elimina un backend

Eliminando un `TridentBackendConfig`, si indica a Trident di eliminare/mantenere i backend (in base a `deletionPolicy`). Per eliminare un backend, assicurarsi che `deletionPolicy` sia impostato su `delete`. Per eliminare solo `TridentBackendConfig`, assicurarsi che `deletionPolicy` sia impostato su `retain`. Questo garantisce che il backend sia ancora presente e possa essere gestito utilizzando `tridentctl`.

Eseguire il seguente comando:

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

Trident non elimina i Kubernetes Secrets che erano in uso da `TridentBackendConfig`. L'utente Kubernetes è responsabile della pulizia dei secrets. È necessario prestare attenzione quando si eliminano i secrets. Dovresti eliminare i secrets solo se non sono in uso dai backends.

#### Visualizza i backend esistenti

Eseguire il seguente comando:

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

È anche possibile eseguire `tridentctl get backend -n trident` o `tridentctl get backend -o yaml -n trident` per ottenere un elenco di tutti i backend esistenti. Questo elenco includerà anche i backend creati con `tridentctl`.

#### Aggiorna un backend

Possono esserci molteplici motivi per aggiornare un backend:

- Le credenziali per il sistema storage sono cambiate. Per aggiornare le credenziali, il Kubernetes Secret utilizzato nell' `TridentBackendConfig` oggetto deve essere aggiornato. Trident aggiornerà automaticamente il backend con le credenziali più recenti fornite. Eseguire il seguente comando per aggiornare il Kubernetes Secret:

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

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

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

- In alternativa, puoi apportare modifiche al CR `TridentBackendConfig` esistente utilizzando il seguente comando:

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



- Se un aggiornamento del backend fallisce, il backend continua a mantenere l'ultima configurazione nota. Puoi visualizzare i log per determinarne la causa eseguendo `kubectl get tbc <tbc-name> -o yaml -n trident` o `kubectl describe tbc <tbc-name> -n trident`.
- Dopo aver identificato e corretto il problema con il file di configurazione, puoi rieseguire il comando di aggiornamento.

## Esegui 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 backend](#)", esegui il seguente comando:

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

Se la creazione del backend fallisce, si è verificato un errore nella configurazione del backend. Puoi visualizzare i log per determinarne la causa eseguendo il seguente comando:

```
tridentctl logs -n trident
```

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

### Elimina un backend

Per eliminare un backend da Trident, procedere come segue:

1. Recupera il nome del backend:

```
tridentctl get backend -n trident
```

2. Elimina il backend:

```
tridentctl delete backend <backend-name> -n trident
```



Se Trident ha eseguito il provisioning di volumi e snapshot da questo backend che sono ancora esistenti, l'eliminazione del backend impedisce il provisioning di nuovi volumi da parte sua. Il backend continuerà a esistere in uno stato di "Eliminazione".

### Visualizza i backend esistenti

Per visualizzare i backend di cui Trident è a conoscenza, procedere come segue:

- Per ottenere un riepilogo, eseguire il seguente comando:

```
tridentctl get backend -n trident
```

- Per ottenere tutti i dettagli, eseguire il seguente comando:

```
tridentctl get backend -o json -n trident
```

### Aggiorna un backend

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

```
tridentctl update backend <backend-name> -f <backend-file> -n trident
```

Se l'aggiornamento del backend fallisce, si è verificato un errore nella configurazione del backend o hai tentato un aggiornamento non valido. Puoi visualizzare i log per determinarne la causa eseguendo il seguente comando:

```
tridentctl logs -n trident
```

Dopo aver identificato e corretto il problema con il file di configurazione, puoi semplicemente eseguire il `update` comando nuovamente.

### Identificare le classi di storage che utilizzano un backend

Questo è un esempio del tipo di domande a cui è possibile rispondere con il JSON che `tridentctl`

restituisce per gli oggetti backend. Questo utilizza l'`jq` utility, che è necessario installare.

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

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

## Spostarsi tra le opzioni di gestione del backend

Scopri i diversi modi per gestire i backend in Trident.

### Opzioni per la gestione dei backend

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

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

### Gestisci `tridentctl` backend utilizzando `TridentBackendConfig`

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

Questo si applicherà ai seguenti scenari:

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

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

- Continua a utilizzare `tridentctl` per gestire i backend creati utilizzandolo.
- Associa i backend creati usando `tridentctl` a un nuovo `TridentBackendConfig` oggetto. Così facendo, i backend saranno gestiti usando `kubectl` e non `tridentctl`.

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

1. Crea un segreto Kubernetes. Il segreto contiene le credenziali Trident necessarie per comunicare con il cluster/servizio di storage.
2. Crea un `TridentBackendConfig` oggetto. Questo contiene informazioni specifiche sul cluster/servizio di storage e fa riferimento al segreto creato nel passaggio precedente. È necessario prestare attenzione a specificare parametri di configurazione identici (ad esempio, `spec.backendName`, `spec.storagePrefix`, `spec.storageDriverName` e così via). `spec.backendName` deve essere impostato sul nome del backend esistente.

## Passaggio 0: Identificare il backend

Per creare un `TridentBackendConfig` che si leghi a un backend esistente, è necessario ottenere la configurazione del backend. In questo esempio, supponiamo che un backend sia stato creato utilizzando la seguente definizione JSON:

```
tridentctl get backend ontap-nas-backend -n trident
+-----+-----+
+-----+-----+-----+-----+
|          NAME          | STORAGE DRIVER |          UUID          |
| STATE  | VOLUMES |          |
+-----+-----+-----+-----+
+-----+-----+-----+-----+
| ontap-nas-backend    | ontap-nas      | 52f2eb10-e4c6-4160-99fc-
96b3be5ab5d7 | online |          25 |
+-----+-----+-----+-----+
+-----+-----+-----+-----+
```

```
cat ontap-nas-backend.json
```

```

{
  "version": 1,
  "storageDriverName": "ontap-nas",
  "managementLIF": "10.10.10.1",
  "dataLIF": "10.10.10.2",
  "backendName": "ontap-nas-backend",
  "svm": "trident_svm",
  "username": "cluster-admin",
  "password": "admin-password",
  "defaults": {
    "spaceReserve": "none",
    "encryption": "false"
  },
  "labels": {
    "store": "nas_store"
  },
  "region": "us_east_1",
  "storage": [
    {
      "labels": {
        "app": "msoffice",
        "cost": "100"
      },
      "zone": "us_east_1a",
      "defaults": {
        "spaceReserve": "volume",
        "encryption": "true",
        "unixPermissions": "0755"
      }
    },
    {
      "labels": {
        "app": "mysqldb",
        "cost": "25"
      },
      "zone": "us_east_1d",
      "defaults": {
        "spaceReserve": "volume",
        "encryption": "false",
        "unixPermissions": "0775"
      }
    }
  ]
}

```

## Passaggio 1: crea un Kubernetes Secret

Crea un Secret che contiene le credenziali per il backend, come mostrato in questo esempio:

```
cat tbc-ontap-nas-backend-secret.yaml
```

```
apiVersion: v1
kind: Secret
metadata:
  name: ontap-nas-backend-secret
type: Opaque
stringData:
  username: cluster-admin
  password: admin-password
```

```
kubectl create -f tbc-ontap-nas-backend-secret.yaml -n trident
secret/backend-tbc-ontap-san-secret created
```

## Passaggio 2: crea un TridentBackendConfig CR

Il passaggio successivo consiste nel creare una `TridentBackendConfig` CR che si associ automaticamente a quella preesistente `ontap-nas-backend` (come in questo esempio). Assicurarsi che siano soddisfatti i seguenti requisiti:

- Lo stesso nome del backend è definito in `spec.backendName`.
- I parametri di configurazione sono identici al backend originale.
- I pool virtuali (se presenti) devono mantenere lo stesso ordine come nel backend originale.
- Le credenziali vengono fornite tramite un Kubernetes Secret e non in testo normale.

In questo caso, il `TridentBackendConfig` apparirà così:

```
cat backend-tbc-ontap-nas.yaml
```

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: tbc-ontap-nas-backend
spec:
  version: 1
  storageDriverName: ontap-nas
  managementLIF: 10.10.10.1
  dataLIF: 10.10.10.2
  backendName: ontap-nas-backend
  svm: trident_svm
  credentials:
    name: mysecret
  defaults:
    spaceReserve: none
    encryption: 'false'
  labels:
    store: nas_store
    region: us_east_1
  storage:
  - labels:
    app: msoffice
    cost: '100'
    zone: us_east_1a
    defaults:
      spaceReserve: volume
      encryption: 'true'
      unixPermissions: '0755'
  - labels:
    app: mysqlpdb
    cost: '25'
    zone: us_east_1d
    defaults:
      spaceReserve: volume
      encryption: 'false'
      unixPermissions: '0775'
```

```
kubectl create -f backend-tbc-ontap-nas.yaml -n trident
tridentbackendconfig.trident.netapp.io/tbc-ontap-nas-backend created
```

### Fase 3: verificare lo stato del TridentBackendConfig CR

Dopo che la TridentBackendConfig è stata creata, la sua fase deve essere Bound. Dovrebbe inoltre riflettere lo stesso nome backend e UUID del backend esistente.

```

kubect1 get tbc tbc-ontap-nas-backend -n trident
NAME                                BACKEND NAME                BACKEND UUID
PHASE    STATUS
tbc-ontap-nas-backend  ontap-nas-backend          52f2eb10-e4c6-4160-99fc-
96b3be5ab5d7    Bound    Success

#confirm that no new backends were created (i.e., TridentBackendConfig did
not end up creating a new backend)
tridentctl get backend -n trident
+-----+-----+
+-----+-----+-----+-----+
|          NAME          | STORAGE DRIVER |          UUID
| STATE  | VOLUMES |
+-----+-----+-----+-----+
| ontap-nas-backend     | ontap-nas      | 52f2eb10-e4c6-4160-99fc-
96b3be5ab5d7 | online |          25 |
+-----+-----+-----+-----+
+-----+-----+-----+-----+

```

Il backend sarà ora completamente gestito utilizzando l' `tbc-ontap-nas-backend` `TridentBackendConfig` oggetto.

### Gestisci `TridentBackendConfig` backend utilizzando `tridentctl`

`tridentctl` può essere utilizzato per elencare i backend che sono stati creati usando `TridentBackendConfig`. Inoltre, gli amministratori possono anche scegliere di gestire completamente tali backend tramite `tridentctl` eliminando `TridentBackendConfig` e assicurandosi che `spec.deletionPolicy` sia impostato su `retain`.

#### Passaggio 0: Identificare il backend

Ad esempio, supponiamo che il seguente backend sia stato creato utilizzando `TridentBackendConfig`:

```
kubectl get tbc backend-tbc-ontap-san -n trident -o wide
NAME                                BACKEND NAME                BACKEND UUID
PHASE  STATUS  STORAGE DRIVER  DELETION POLICY
backend-tbc-ontap-san  ontap-san-backend  81abcb27-ea63-49bb-b606-
0a5315ac5f82  Bound  Success  ontap-san  delete

tridentctl get backend ontap-san-backend -n trident
+-----+-----+
+-----+-----+-----+-----+
|          NAME          | STORAGE DRIVER |          UUID
| STATE  | VOLUMES |
+-----+-----+
+-----+-----+-----+-----+
| ontap-san-backend | ontap-san      | 81abcb27-ea63-49bb-b606-
0a5315ac5f82 | online |      33 |
+-----+-----+
+-----+-----+-----+-----+
```

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

**Passaggio 1: confermare** `deletionPolicy` è impostato su `retain`

Diamo un'occhiata al valore di `deletionPolicy`. Questo deve essere impostato su `retain`. Questo garantisce che quando un `TridentBackendConfig` CR viene eliminato, la definizione del backend sarà ancora presente e potrà essere gestita con `tridentctl`.

```
kubectl get tbc backend-tbc-ontap-san -n trident -o wide
NAME                                BACKEND NAME                BACKEND UUID
PHASE  STATUS  STORAGE DRIVER  DELETION POLICY
backend-tbc-ontap-san  ontap-san-backend  81abcb27-ea63-49bb-b606-
0a5315ac5f82  Bound  Success  ontap-san  delete

# Patch value of deletionPolicy to retain
kubectl patch tbc backend-tbc-ontap-san --type=merge -p
'{"spec":{"deletionPolicy":"retain"}}' -n trident
tridentbackendconfig.trident.netapp.io/backend-tbc-ontap-san patched

#Confirm the value of deletionPolicy
kubectl get tbc backend-tbc-ontap-san -n trident -o wide
NAME                                BACKEND NAME                BACKEND UUID
PHASE  STATUS  STORAGE DRIVER  DELETION POLICY
backend-tbc-ontap-san  ontap-san-backend  81abcb27-ea63-49bb-b606-
0a5315ac5f82  Bound  Success  ontap-san  retain
```



Non procedere al passaggio successivo a meno che `deletionPolicy` non sia impostato su `retain`.

## Passaggio 2: Eliminare il `TridentBackendConfig` CR

Il passaggio finale consiste nell'eliminare il `TridentBackendConfig` CR. Dopo aver verificato che `deletionPolicy` è impostato su `retain`, è possibile procedere con l'eliminazione:

```
kubectl delete tbc backend-tbc-ontap-san -n trident
tridentbackendconfig.trident.netapp.io "backend-tbc-ontap-san" deleted

tridentctl get backend ontap-san-backend -n trident
+-----+-----+
+-----+-----+-----+-----+
|          NAME          | STORAGE DRIVER |                      UUID
| STATE  | VOLUMES |
+-----+-----+-----+-----+
+-----+-----+-----+-----+
| ontap-san-backend | ontap-san      | 81abcb27-ea63-49bb-b606-
0a5315ac5f82 | online |      33 |
+-----+-----+-----+-----+
+-----+-----+-----+-----+
```

Dopo l'eliminazione dell'oggetto `TridentBackendConfig`, Trident lo rimuove semplicemente senza eliminare effettivamente il backend stesso.

## 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.