Sicurezza

Trident

NetApp
January 15, 2026

Sommario

Sicurezza . . . 1
- Sicurezza . . . 1
  - Esegui Trident nel suo namespace . . . 1
  - Utilizzare l'autenticazione CHAP con i backend ONTAP SAN . . . 1
  - Utilizzare l'autenticazione CHAP con i backend NetApp HCI e SolidFire . . . 1
  - Utilizzare Trident con NVE e NAE . . . 1
- Configurazione della chiave unificata Linux (LUKS) . . . 2
  - Abilita la crittografia LUKS . . . 2
  - Configurazione backend per l'importazione di volumi LUKS . . . 4
  - Configurazione PVC per l'importazione di volumi LUKS . . . 4
  - Ruota una passphrase LUKS . . . 5
  - Abilita l'espansione del volume . . . 7
- Crittografia Kerberos in volo . . . 8
  - Configurare la crittografia Kerberos in-flight con volumi ONTAP on-premise . . . 8
  - Configurare la crittografia Kerberos in-flight con volumi Azure NetApp Files . . . 12

Sicurezza

Sicurezza

Per garantire la sicurezza dell'installazione Trident , attenersi alle raccomandazioni elencate qui.

Esegui Trident nel suo namespace

È importante impedire alle applicazioni, agli amministratori delle applicazioni, agli utenti e alle applicazioni di gestione di accedere alle definizioni degli oggetti Trident o ai pod per garantire un'archiviazione affidabile e bloccare potenziali attività dannose.

Per separare le altre applicazioni e gli utenti da Trident, installare sempre Trident nel proprio namespace Kubernetes(`trident`). Inserire Trident nel proprio namespace garantisce che solo il personale amministrativo di Kubernetes abbia accesso al pod Trident e agli artefatti (come i segreti backend e CHAP, se applicabile) archiviati negli oggetti CRD con namespace. Dovresti assicurarti di consentire solo agli amministratori l'accesso allo spazio dei nomi Trident e quindi l'accesso a `tridentctl` applicazione.

Utilizzare l'autenticazione CHAP con i backend ONTAP SAN

Trident supporta l'autenticazione basata su CHAP per carichi di lavoro ONTAP SAN (utilizzando `ontap-san` E `ontap-san-economy` conducenti). NetApp consiglia di utilizzare CHAP bidirezionale con Trident per l'autenticazione tra un host e il backend di storage.

Per i backend ONTAP che utilizzano i driver di archiviazione SAN, Trident può impostare CHAP bidirezionale e gestire i nomi utente e i segreti CHAP tramite `tridentctl` . Fare riferimento a"Prepararsi a configurare il backend con i driver ONTAP SAN" per capire come Trident configura CHAP sui backend ONTAP .

Utilizzare l'autenticazione CHAP con i backend NetApp HCI e SolidFire

NetApp consiglia di implementare CHAP bidirezionale per garantire l'autenticazione tra un host e i backend NetApp HCI e SolidFire . Trident utilizza un oggetto segreto che include due password CHAP per tenant. Quando Trident è installato, gestisce i segreti CHAP e li memorizza in un `tridentvolume` Oggetto CR per il rispettivo PV. Quando si crea un PV, Trident utilizza i segreti CHAP per avviare una sessione iSCSI e comunicare con il sistema NetApp HCI e SolidFire tramite CHAP.

I volumi creati da Trident non sono associati ad alcun Volume Access Group.

Utilizzare Trident con NVE e NAE

NetApp ONTAP fornisce la crittografia dei dati a riposo per proteggere i dati sensibili nel caso in cui un disco venga rubato, restituito o riutilizzato. Per i dettagli, fare riferimento a"Panoramica sulla configurazione della crittografia del volume NetApp" .

- Se NAE è abilitato sul backend, qualsiasi volume fornito in Trident sarà abilitato per NAE.
  - È possibile impostare il flag di crittografia NVE su `""` per creare volumi abilitati per NAE.
- Se NAE non è abilitato sul backend, qualsiasi volume fornito in Trident sarà abilitato per NVE a meno che il flag di crittografia NVE non sia impostato su `false` (valore predefinito) nella configurazione del backend.

I volumi creati in Trident su un backend abilitato NAE devono essere crittografati tramite NVE o NAE.

- È possibile impostare il flag di crittografia NVE su `true` nella configurazione del backend Trident per ignorare la crittografia NAE e utilizzare una chiave di crittografia specifica per ogni volume.
- Impostazione del flag di crittografia NVE su `false` su un backend abilitato NAE crea un volume abilitato NAE. Non è possibile disabilitare la crittografia NAE impostando il flag di crittografia NVE su `false` .

- È possibile creare manualmente un volume NVE in Trident impostando esplicitamente il flag di crittografia NVE su `true` .

Per maggiori informazioni sulle opzioni di configurazione del backend, fare riferimento a:

- "Opzioni di configurazione SAN ONTAP"
- "Opzioni di configurazione NAS ONTAP"

Configurazione della chiave unificata Linux (LUKS)

È possibile abilitare Linux Unified Key Setup (LUKS) per crittografare i volumi ONTAP SAN e ONTAP SAN ECONOMY su Trident. Trident supporta la rotazione delle passphrase e l'espansione del volume per i volumi crittografati con LUKS.

In Trident, i volumi crittografati LUKS utilizzano la cifratura e la modalità aes-xts-plain64, come raccomandato da"NIST" .

La crittografia LUKS non è supportata per i sistemi ASA r2. Per informazioni sui sistemi ASA r2, vedere"Scopri di più sui sistemi di archiviazione ASA r2" .

Prima di iniziare

- Sui nodi worker deve essere installato cryptsetup 2.1 o versione successiva (ma inferiore a 3.0). Per maggiori informazioni, visita"Gitlab: cryptsetup" .
- Per motivi di prestazioni, NetApp consiglia che i nodi worker supportino Advanced Encryption Standard New Instructions (AES-NI). Per verificare il supporto AES-NI, eseguire il seguente comando:

```
grep "aes" /proc/cpuinfo
```

Se non viene restituito nulla, il processore non supporta AES-NI. Per maggiori informazioni su AES-NI, visitare:"Intel: Istruzioni per lo standard di crittografia avanzata (AES-NI)" .

Abilita la crittografia LUKS

È possibile abilitare la crittografia lato host per volume utilizzando Linux Unified Key Setup (LUKS) per i volumi ONTAP SAN e ONTAP SAN ECONOMY.

Passi

1. Definire gli attributi di crittografia LUKS nella configurazione del backend. Per ulteriori informazioni sulle opzioni di configurazione backend per ONTAP SAN, fare riferimento a"Opzioni di configurazione SAN ONTAP" .

```
{
  "storage": [
    {
      "labels": {
        "luks": "true"
      },
      "zone": "us_east_1a",
      "defaults": {
        "luksEncryption": "true"
      }
    },
    {
      "labels": {
        "luks": "false"
      },
      "zone": "us_east_1a",
      "defaults": {
        "luksEncryption": "false"
      }
    }
  ]
}
```

2. Utilizzo `parameters.selector` per definire i pool di archiviazione utilizzando la crittografia LUKS. Per esempio:

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: luks
provisioner: csi.trident.netapp.io
parameters:
  selector: "luks=true"
  csi.storage.k8s.io/node-stage-secret-name: luks-${pvc.name}
  csi.storage.k8s.io/node-stage-secret-namespace: ${pvc.namespace}
```

3. Crea un segreto che contenga la passphrase LUKS. Per esempio:

```
kubectl -n trident create -f luks-pvc1.yaml
apiVersion: v1
kind: Secret
metadata:
  name: luks-pvc1
stringData:
  luks-passphrase-name: A
  luks-passphrase: secretA
```

Limitazioni

I volumi crittografati con LUKS non possono sfruttare la deduplicazione e la compressione ONTAP .

Configurazione backend per l'importazione di volumi LUKS

Per importare un volume LUKS, è necessario impostare `luksEncryption` A(`true` sul backend. IL `luksEncryption` l'opzione indica a Trident se il volume è conforme a LUKS(`true`) o non conforme a LUKS(`false`) come mostrato nell'esempio seguente.

```
version: 1
storageDriverName: ontap-san
managementLIF: 10.0.0.1
dataLIF: 10.0.0.2
svm: trident_svm
username: admin
password: password
defaults:
  luksEncryption: 'true'
  spaceAllocation: 'false'
  snapshotPolicy: default
  snapshotReserve: '10'
```

Configurazione PVC per l'importazione di volumi LUKS

Per importare dinamicamente i volumi LUKS, impostare l'annotazione `trident.netapp.io/luksEncryption` A `true` e includere una classe di archiviazione abilitata LUKS nel PVC come mostrato in questo esempio.

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: luks-pvc
  namespace: trident
  annotations:
    trident.netapp.io/luksEncryption: "true"
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: luks-sc
```

Ruota una passphrase LUKS

È possibile ruotare la passphrase LUKS e confermare la rotazione.

> Non dimenticare una passphrase finché non hai verificato che non sia più referenziata da alcun volume, snapshot o segreto. Se si perde una passphrase di riferimento, potrebbe non essere possibile montare il volume e i dati rimarranno crittografati e inaccessibili.

Informazioni su questo compito

La rotazione della passphrase LUKS si verifica quando viene creato un pod che monta il volume dopo aver specificato una nuova passphrase LUKS. Quando viene creato un nuovo pod, Trident confronta la passphrase LUKS sul volume con la passphrase attiva nel segreto.

- Se la passphrase sul volume non corrisponde alla passphrase attiva nel segreto, si verifica la rotazione.
- Se la passphrase sul volume corrisponde alla passphrase attiva nel segreto, il `previous-luks-passphrase` il parametro viene ignorato.

Passi

1. Aggiungi il `node-publish-secret-name` E `node-publish-secret-namespace` Parametri StorageClass. Per esempio:

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: csi-san
provisioner: csi.trident.netapp.io
parameters:
  trident.netapp.io/backendType: "ontap-san"
  csi.storage.k8s.io/node-stage-secret-name: luks
  csi.storage.k8s.io/node-stage-secret-namespace: ${pvc.namespace}
  csi.storage.k8s.io/node-publish-secret-name: luks
  csi.storage.k8s.io/node-publish-secret-namespace: ${pvc.namespace}
```

2. Identificare le passphrase esistenti sul volume o sullo snapshot.

Volume

```
tridentctl -d get volume luks-pvc1
GET http://127.0.0.1:8000/trident/v1/volume/<volumeID>

...luksPassphraseNames:["A"]
```

Istantanea

```
tridentctl -d get snapshot luks-pvc1
GET http://127.0.0.1:8000/trident/v1/volume/<volumeID>/<snapshotID>

...luksPassphraseNames:["A"]
```

3. Aggiornare il segreto LUKS per il volume per specificare le passphrase nuove e precedenti. Garantire `previous-luke-passphrase-name` E `previous-luks-passphrase` corrisponde alla passphrase precedente.

```
apiVersion: v1
kind: Secret
metadata:
  name: luks-pvc1
stringData:
  luks-passphrase-name: B
  luks-passphrase: secretB
  previous-luks-passphrase-name: A
  previous-luks-passphrase: secretA
```

4. Crea un nuovo pod montando il volume. Ciò è necessario per avviare la rotazione.
5. Verificare che la passphrase sia stata ruotata.

Volume

```
tridentctl -d get volume luks-pvc1
GET http://127.0.0.1:8000/trident/v1/volume/<volumeID>

...luksPassphraseNames:["B"]
```

Istantanea

```
tridentctl -d get snapshot luks-pvc1
GET http://127.0.0.1:8000/trident/v1/volume/<volumeID>/<snapshotID>

...luksPassphraseNames:["B"]
```

Risultati

La passphrase è stata ruotata quando sul volume e sullo snapshot è stata restituita solo la nuova passphrase.

> Se vengono restituite due passphrase, ad esempio `luksPassphraseNames: ["B", "A"]`, la rotazione è incompleta. È possibile attivare un nuovo pod per tentare di completare la rotazione.

Abilita l'espansione del volume

È possibile abilitare l'espansione del volume su un volume crittografato con LUKS.

Passi

1. Abilita il `CSINodeExpandSecret` feature gate (beta 1.25+). Fare riferimento a "Kubernetes 1.25: utilizzare i segreti per l'espansione dei volumi CSI basata sui nodi" per i dettagli.
2. Aggiungi il `node-expand-secret-name` E `node-expand-secret-namespace` Parametri StorageClass. Per esempio:

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: luks
provisioner: csi.trident.netapp.io
parameters:
  selector: "luks=true"
  csi.storage.k8s.io/node-stage-secret-name: luks-${pvc.name}
  csi.storage.k8s.io/node-stage-secret-namespace: ${pvc.namespace}
  csi.storage.k8s.io/node-expand-secret-name: luks-${pvc.name}
  csi.storage.k8s.io/node-expand-secret-namespace: ${pvc.namespace}
allowVolumeExpansion: true
```

Risultati

Quando si avvia l'espansione dello storage online, kubelet passa le credenziali appropriate al driver.

Crittografia Kerberos in volo

Utilizzando la crittografia in-flight Kerberos, puoi migliorare la sicurezza dell'accesso ai dati abilitando la crittografia per il traffico tra il cluster gestito e il backend di archiviazione.

Trident supporta la crittografia Kerberos per ONTAP come backend di archiviazione:

- * ONTAP on-premise* - Trident supporta la crittografia Kerberos su connessioni NFSv3 e NFSv4 da Red Hat OpenShift e cluster Kubernetes upstream a volumi ONTAP on-premise.

È possibile creare, eliminare, ridimensionare, creare snapshot, clonare, clonare in sola lettura e importare volumi che utilizzano la crittografia NFS.

Configurare la crittografia Kerberos in-flight con volumi ONTAP on-premise

È possibile abilitare la crittografia Kerberos sul traffico di archiviazione tra il cluster gestito e un backend di archiviazione ONTAP locale.

> La crittografia Kerberos per il traffico NFS con backend di archiviazione ONTAP on-premise è supportata solo utilizzando `ontap-nas` driver di archiviazione.

Prima di iniziare

- Assicurati di avere accesso a `tridentctl` utilità.
- Assicurati di avere accesso come amministratore al backend di archiviazione ONTAP .
- Assicurati di conoscere il nome del volume o dei volumi che condividerai dal backend di archiviazione ONTAP .
- Assicurarsi di aver preparato la VM di archiviazione ONTAP per supportare la crittografia Kerberos per i volumi NFS. Fare riferimento a "Abilita Kerberos su un dataLIF" per istruzioni.
- Assicurarsi che tutti i volumi NFSv4 utilizzati con la crittografia Kerberos siano configurati correttamente. Fare riferimento alla sezione Configurazione del dominio NetApp NFSv4 (pagina 13) del manuale "Guida ai miglioramenti e alle best practice NetApp NFSv4" .

Aggiungere o modificare le policy di esportazione ONTAP

È necessario aggiungere regole alle policy di esportazione ONTAP esistenti o creare nuove policy di esportazione che supportino la crittografia Kerberos per il volume radice della VM di archiviazione ONTAP e per tutti i volumi ONTAP condivisi con il cluster Kubernetes upstream. Le regole dei criteri di esportazione aggiunti o i nuovi criteri di esportazione creati devono supportare i seguenti protocolli di accesso e autorizzazioni di accesso:

Protocolli di accesso

Configurare la policy di esportazione con i protocolli di accesso NFS, NFSv3 e NFSv4.

Dettagli di accesso

È possibile configurare una delle tre diverse versioni della crittografia Kerberos, a seconda delle esigenze del volume:

- **Kerberos 5** - (autenticazione e crittografia)

- **Kerberos 5i** - (autenticazione e crittografia con protezione dell'identità)
- **Kerberos 5p** - (autenticazione e crittografia con protezione dell'identità e della privacy)

Configurare la regola di policy di esportazione ONTAP con le autorizzazioni di accesso appropriate. Ad esempio, se i cluster monteranno i volumi NFS con una combinazione di crittografia Kerberos 5i e Kerberos 5p, utilizzare le seguenti impostazioni di accesso:

Tipo	Accesso di sola lettura	Accesso in lettura/scrittura	Accesso superutente
UNIX	Abilitato	Abilitato	Abilitato
Kerberos 5i	Abilitato	Abilitato	Abilitato
Kerberos 5p	Abilitato	Abilitato	Abilitato

Per informazioni su come creare policy di esportazione ONTAP e regole di policy di esportazione, fare riferimento alla seguente documentazione:

- "Creare una politica di esportazione"
- "Aggiungere una regola a un criterio di esportazione"

Creare un backend di archiviazione

È possibile creare una configurazione backend di archiviazione Trident che includa la funzionalità di crittografia Kerberos.

Informazioni su questo compito

Quando si crea un file di configurazione del backend di archiviazione che configura la crittografia Kerberos, è possibile specificare una delle tre diverse versioni della crittografia Kerberos utilizzando `spec.nfsMountOptions` parametro:

- `spec.nfsMountOptions: sec=krb5`(autenticazione e crittografia)
- `spec.nfsMountOptions: sec=krb5i`(autenticazione e crittografia con protezione dell'identità)
- `spec.nfsMountOptions: sec=krb5p`(autenticazione e crittografia con protezione dell'identità e della privacy)

Specificare un solo livello Kerberos. Se si specifica più di un livello di crittografia Kerberos nell'elenco dei parametri, verrà utilizzata solo la prima opzione.

Passi

1. Nel cluster gestito, creare un file di configurazione del backend di archiviazione utilizzando il seguente esempio. Sostituisci i valori tra parentesi <> con le informazioni del tuo ambiente:

```
apiVersion: v1
kind: Secret
metadata:
  name: backend-ontap-nas-secret
type: Opaque
stringData:
  clientID: <CLIENT_ID>
  clientSecret: <CLIENT_SECRET>
---
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-ontap-nas
spec:
  version: 1
  storageDriverName: "ontap-nas"
  managementLIF: <STORAGE_VM_MGMT_LIF_IP_ADDRESS>
  dataLIF: <PROTOCOL_LIF_FQDN_OR_IP_ADDRESS>
  svm: <STORAGE_VM_NAME>
  username: <STORAGE_VM_USERNAME_CREDENTIAL>
  password: <STORAGE_VM_PASSWORD_CREDENTIAL>
  nasType: nfs
  nfsMountOptions: ["sec=krb5i"] #can be krb5, krb5i, or krb5p
  qtreesPerFlexvol:
  credentials:
    name: backend-ontap-nas-secret
```

2. Utilizzare il file di configurazione creato nel passaggio precedente per creare il backend:

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

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

```
tridentctl logs
```

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

Creare una classe di archiviazione

È possibile creare una classe di archiviazione per eseguire il provisioning dei volumi con crittografia Kerberos.

Informazioni su questo compito

Quando si crea un oggetto di classe di archiviazione, è possibile specificare una delle tre diverse versioni della crittografia Kerberos utilizzando `mountOptions` parametro:

- `mountOptions: sec=krb5`(autenticazione e crittografia)
- `mountOptions: sec=krb5i`(autenticazione e crittografia con protezione dell'identità)
- `mountOptions: sec=krb5p`(autenticazione e crittografia con protezione dell'identità e della privacy)

Specificare un solo livello Kerberos. Se si specifica più di un livello di crittografia Kerberos nell'elenco dei parametri, verrà utilizzata solo la prima opzione. Se il livello di crittografia specificato nella configurazione del backend di archiviazione è diverso dal livello specificato nell'oggetto della classe di archiviazione, l'oggetto della classe di archiviazione ha la precedenza.

Passi

1. Creare un oggetto StorageClass Kubernetes, utilizzando il seguente esempio:

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontap-nas-sc
provisioner: csi.trident.netapp.io
mountOptions:
  - sec=krb5i #can be krb5, krb5i, or krb5p
parameters:
  backendType: ontap-nas
  storagePools: ontapnas_pool
  trident.netapp.io/nasType: nfs
allowVolumeExpansion: true
```

2. Creare la classe di archiviazione:

```
kubectl create -f sample-input/storage-class-ontap-nas-sc.yaml
```

3. Assicurarsi che la classe di archiviazione sia stata creata:

```
kubectl get sc ontap-nas-sc
```

   Dovresti vedere un output simile al seguente:

```
NAME          PROVISIONER             AGE
ontap-nas-sc    csi.trident.netapp.io   15h
```

Volumi di fornitura

Dopo aver creato un backend di archiviazione e una classe di archiviazione, è possibile effettuare il provisioning di un volume. Per le istruzioni, fare riferimento a "Fornire un volume" .

Configurare la crittografia Kerberos in-flight con volumi Azure NetApp Files

È possibile abilitare la crittografia Kerberos sul traffico di archiviazione tra il cluster gestito e un singolo backend di archiviazione Azure NetApp Files o un pool virtuale di backend di archiviazione di Azure NetApp Files .

Prima di iniziare

- Assicurati di aver abilitato Trident sul cluster Red Hat OpenShift gestito.
- Assicurati di avere accesso a `tridentctl` utilità.
- Assicurarsi di aver preparato il backend di archiviazione Azure NetApp Files per la crittografia Kerberos prendendo nota dei requisiti e seguendo le istruzioni in "Documentazione Azure NetApp Files" .
- Assicurarsi che tutti i volumi NFSv4 utilizzati con la crittografia Kerberos siano configurati correttamente. Fare riferimento alla sezione Configurazione del dominio NetApp NFSv4 (pagina 13) del manuale "Guida ai miglioramenti e alle best practice NetApp NFSv4" .

Creare un backend di archiviazione

È possibile creare una configurazione back-end di archiviazione di Azure NetApp Files che includa la funzionalità di crittografia Kerberos.

Informazioni su questo compito

Quando si crea un file di configurazione del backend di archiviazione che configura la crittografia Kerberos, è possibile definirlo in modo che venga applicato a uno dei due livelli possibili:

- Il **livello di backend di archiviazione** che utilizza `spec.kerberos` campo
- Il **livello della piscina virtuale** utilizzando il `spec.storage.kerberos` campo

Quando si definisce la configurazione a livello di pool virtuale, il pool viene selezionato utilizzando l'etichetta nella classe di archiviazione.

A entrambi i livelli è possibile specificare una delle tre diverse versioni della crittografia Kerberos:

- `kerberos: sec=krb5`(autenticazione e crittografia)
- `kerberos: sec=krb5i`(autenticazione e crittografia con protezione dell'identità)
- `kerberos: sec=krb5p`(autenticazione e crittografia con protezione dell'identità e della privacy)

Passi

1. Nel cluster gestito, creare un file di configurazione del backend di archiviazione utilizzando uno degli esempi seguenti, a seconda di dove è necessario definire il backend di archiviazione (livello di backend di archiviazione o livello di pool virtuale). Sostituisci i valori tra parentesi <> con le informazioni del tuo ambiente:

Esempio di livello di backend di archiviazione

```
apiVersion: v1
kind: Secret
metadata:
  name: backend-tbc-secret
type: Opaque
stringData:
  clientID: <CLIENT_ID>
  clientSecret: <CLIENT_SECRET>

---
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc
spec:
  version: 1
  storageDriverName: azure-netapp-files
  subscriptionID: <SUBSCRIPTION_ID>
  tenantID: <TENANT_ID>
  location: <AZURE_REGION_LOCATION>
  serviceLevel: Standard
  networkFeatures: Standard
  capacityPools: <CAPACITY_POOL>
  resourceGroups: <RESOURCE_GROUP>
  netappAccounts: <NETAPP_ACCOUNT>
  virtualNetwork: <VIRTUAL_NETWORK>
  subnet: <SUBNET>
  nasType: nfs
  kerberos: sec=krb5i #can be krb5, krb5i, or krb5p
  credentials:
    name: backend-tbc-secret
```

Esempio di livello di piscina virtuale

```
---
apiVersion: v1
kind: Secret
metadata:
  name: backend-tbc-secret
type: Opaque
stringData:
  clientID: <CLIENT_ID>
  clientSecret: <CLIENT_SECRET>

---
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc
spec:
  version: 1
  storageDriverName: azure-netapp-files
  subscriptionID: <SUBSCRIPTION_ID>
  tenantID: <TENANT_ID>
  location: <AZURE_REGION_LOCATION>
  serviceLevel: Standard
  networkFeatures: Standard
  capacityPools: <CAPACITY_POOL>
  resourceGroups: <RESOURCE_GROUP>
  netappAccounts: <NETAPP_ACCOUNT>
  virtualNetwork: <VIRTUAL_NETWORK>
  subnet: <SUBNET>
  nasType: nfs
  storage:
    - labels:
        type: encryption
      kerberos: sec=krb5i #can be krb5, krb5i, or krb5p
  credentials:
    name: backend-tbc-secret
```

2. Utilizzare il file di configurazione creato nel passaggio precedente per creare il backend:

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

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

```
tridentctl logs
```

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

Creare una classe di archiviazione

È possibile creare una classe di archiviazione per eseguire il provisioning dei volumi con crittografia Kerberos.

Passi

1. Creare un oggetto StorageClass Kubernetes, utilizzando il seguente esempio:

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: sc-nfs
provisioner: csi.trident.netapp.io
parameters:
  backendType: azure-netapp-files
  trident.netapp.io/nasType: nfs
  selector: type=encryption
```

2. Creare la classe di archiviazione:

```
kubectl create -f sample-input/storage-class-sc-nfs.yaml
```

3. Assicurarsi che la classe di archiviazione sia stata creata:

```
kubectl get sc -sc-nfs
```

Dovresti vedere un output simile al seguente:

```
NAME          PROVISIONER             AGE
sc-nfs        csi.trident.netapp.io   15h
```

Volumi di fornitura

Dopo aver creato un backend di archiviazione e una classe di archiviazione, è possibile effettuare il provisioning di un volume. Per le istruzioni, fare riferimento a "Fornire un volume" .

Informazioni sul copyright

Informazioni sul marchio commerciale