Sicherheit

Trident

NetApp
January 15, 2026

Inhalt

Sicherheit

Sicherheit

Befolgen Sie die hier aufgeführten Empfehlungen, um eine sichere Installation Ihres Trident zu gewährleisten.

Trident in einem eigenen Namensraum ausführen

Um eine zuverlässige Speicherung zu gewährleisten und potenziell schädliche Aktivitäten zu verhindern, ist es wichtig, Anwendungen, Anwendungsadministratoren, Benutzer und Verwaltungsanwendungen den Zugriff auf Trident -Objektdefinitionen oder die Pods zu verwehren.

Um andere Anwendungen und Benutzer von Trident zu trennen, installieren Sie Trident immer in einem eigenen Kubernetes-Namespace.(`trident`). Durch die Platzierung von Trident in einem eigenen Namespace wird sichergestellt, dass nur das Kubernetes-Administratorpersonal Zugriff auf den Trident -Pod und die in den Namespace-CRD-Objekten gespeicherten Artefakte (wie z. B. Backend- und CHAP-Secrets, falls zutreffend) hat. Sie sollten sicherstellen, dass nur Administratoren Zugriff auf den Trident -Namensraum und somit auf die entsprechenden Funktionen haben. `tridentctl` Anwendung.

CHAP-Authentifizierung mit ONTAP SAN-Backends verwenden

Trident unterstützt die CHAP-basierte Authentifizierung für ONTAP -SAN-Workloads (unter Verwendung der `ontap-san` Und `ontap-san-economy` Fahrer). NetApp empfiehlt die Verwendung von bidirektionalem CHAP mit Trident zur Authentifizierung zwischen einem Host und dem Speicher-Backend.

Für ONTAP Backends, die SAN-Speichertreiber verwenden, kann Trident bidirektionales CHAP einrichten und CHAP-Benutzernamen und -Geheimnisse verwalten. `tridentctl` . Siehe"Bereiten Sie die Konfiguration des Backends mit ONTAP SAN-Treibern vor." um zu verstehen, wie Trident CHAP auf ONTAP Backends konfiguriert.

CHAP-Authentifizierung mit NetApp HCI und SolidFire -Backends verwenden

NetApp empfiehlt den Einsatz von bidirektionalem CHAP, um die Authentifizierung zwischen einem Host und den NetApp HCI und SolidFire -Backends sicherzustellen. Trident verwendet ein geheimes Objekt, das zwei CHAP-Passwörter pro Mandant enthält. Nach der Installation von Trident verwaltet es die CHAP-Geheimnisse und speichert sie in einem `tridentvolume` CR-Objekt für das jeweilige PV. Wenn Sie ein PV erstellen, verwendet Trident die CHAP-Secrets, um eine iSCSI-Sitzung zu initiieren und über CHAP mit dem NetApp HCI und SolidFire -System zu kommunizieren.

Die von Trident erstellten Volumes sind keiner Volume-Zugriffsgruppe zugeordnet.

Verwenden Sie Trident mit NVE und NAE

NetApp ONTAP bietet Datenverschlüsselung im Ruhezustand, um sensible Daten für den Fall zu schützen, dass eine Festplatte gestohlen, zurückgegeben oder anderweitig verwendet wird. Weitere Einzelheiten finden Sie unter"Übersicht über die Konfiguration der NetApp Volume-Verschlüsselung" .

- Wenn NAE im Backend aktiviert ist, wird jedes in Trident bereitgestellte Volume NAE-fähig sein.
  - Sie können das NVE-Verschlüsselungsflag auf „“ setzen. `""` um NAE-fähige Volumes zu erstellen.

- Wenn NAE im Backend nicht aktiviert ist, wird jedes in Trident bereitgestellte Volume NVE-fähig sein, es sei denn, das NVE-Verschlüsselungsflag ist auf „true“ gesetzt. `false` (der Standardwert) in der Backend-Konfiguration.

> In Trident auf einem NAE-fähigen Backend erstellte Volumes müssen NVE- oder NAE-verschlüsselt sein.
>
> - Sie können das NVE-Verschlüsselungsflag auf „“ setzen. `true` in der Trident -Backend -Konfiguration, um die NAE-Verschlüsselung zu überschreiben und einen spezifischen Verschlüsselungsschlüssel pro Volume zu verwenden.
> - Setzen des NVE-Verschlüsselungsflags auf `false` Auf einem NAE-fähigen Backend wird ein NAE-fähiges Volume erstellt. Die NAE-Verschlüsselung kann nicht durch Setzen des NVE-Verschlüsselungsflags deaktiviert werden. `false` .

- Sie können in Trident manuell ein NVE-Volume erstellen, indem Sie das NVE-Verschlüsselungsflag explizit setzen. `true` .

Weitere Informationen zu den Backend-Konfigurationsoptionen finden Sie unter:

- "ONTAP SAN-Konfigurationsoptionen"
- "ONTAP NAS-Konfigurationsoptionen"

Linux Unified Key Setup (LUKS)

Sie können Linux Unified Key Setup (LUKS) aktivieren, um ONTAP SAN- und ONTAP SAN ECONOMY-Volumes auf Trident zu verschlüsseln. Trident unterstützt die Rotation von Passphrasen und die Volumenerweiterung für LUKS-verschlüsselte Volumes.

In Trident verwenden LUKS-verschlüsselte Volumes die aes-xts-plain64-Verschlüsselung und den entsprechenden Modus, wie von empfohlen."NIST" .

> LUKS-Verschlüsselung wird für ASA r2-Systeme nicht unterstützt. Informationen zu ASA r2-Systemen finden Sie unter"Erfahren Sie mehr über ASA R2-Speichersysteme" .

Bevor Sie beginnen

- Auf den Worker-Knoten muss cryptsetup 2.1 oder höher (aber niedriger als 3.0) installiert sein. Weitere Informationen finden Sie unter"GitLab: cryptsetup" .
- Aus Performancegründen empfiehlt NetApp , dass Worker-Knoten den Advanced Encryption Standard New Instructions (AES-NI) unterstützen. Um die AES-NI-Unterstützung zu überprüfen, führen Sie folgenden Befehl aus:

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

Wenn keine Antwort zurückgegeben wird, unterstützt Ihr Prozessor AES-NI nicht. Weitere Informationen zu AES-NI finden Sie unter:"Intel: Advanced Encryption Standard Instructions (AES-NI)" .

LUKS-Verschlüsselung aktivieren

Sie können die volumenbezogene, hostseitige Verschlüsselung mithilfe von Linux Unified Key Setup (LUKS) für ONTAP SAN- und ONTAP SAN ECONOMY-Volumes aktivieren.

Schritte

1. Definieren Sie die LUKS-Verschlüsselungsattribute in der Backend-Konfiguration. Weitere Informationen zu den Backend-Konfigurationsoptionen für ONTAP SAN finden Sie unter"ONTAP SAN-Konfigurationsoptionen" .

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

2. Verwenden `parameters.selector` Die Speicherpools werden mithilfe der LUKS-Verschlüsselung definiert. Beispiel:

```
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. Erstellen Sie ein Geheimnis, das die LUKS-Passphrase enthält. Beispiel:

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

Einschränkungen

LUKS-verschlüsselte Datenträger können die Deduplizierung und Komprimierung von ONTAP nicht nutzen.

Backend-Konfiguration für den Import von LUKS-Volumes

Um ein LUKS-Volume zu importieren, müssen Sie Folgendes einstellen: `luksEncryption` Zu(`true` im Backend. Der `luksEncryption` Diese Option teilt Trident mit, ob das Volume LUKS-kompatibel ist.(`true` oder nicht LUKS-konform(`false`) wie im folgenden Beispiel gezeigt.

```
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'
```

PVC-Konfiguration für den Import von LUKS-Volumes

Um LUKS-Volumes dynamisch zu importieren, legen Sie die Annotation fest. `trident.netapp.io/luksEncryption` Zu `true` und eine LUKS-fähige Speicherklasse in die PVC einbinden, wie in diesem Beispiel gezeigt.

```
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
```

LUKS-Passphrase rotieren

Sie können die LUKS-Passphrase ändern und die Änderung bestätigen.

> Vergessen Sie eine Passphrase erst dann, wenn Sie sichergestellt haben, dass sie von keinem Volume, Snapshot oder Geheimnis mehr referenziert wird. Geht die angegebene Passphrase verloren, kann das Volume möglicherweise nicht eingebunden werden und die Daten bleiben verschlüsselt und unzugänglich.

Informationen zu diesem Vorgang

Eine LUKS-Passphrase-Rotation tritt auf, wenn ein Pod, der das Volume einbindet, erstellt wird, nachdem eine neue LUKS-Passphrase angegeben wurde. Wenn ein neuer Pod erstellt wird, vergleicht Trident die LUKS-Passphrase des Volumes mit der aktiven Passphrase im Secret.

- Stimmt die Passphrase des Volumes nicht mit der aktiven Passphrase im Geheimnis überein, findet eine Rotation statt.
- Wenn die Passphrase des Volumes mit der aktiven Passphrase im Geheimnis übereinstimmt, `previous-luks-passphrase` Der Parameter wird ignoriert.

Schritte

1. Füge die `node-publish-secret-name` Und `node-publish-secret-namespace` StorageClass-Parameter. Beispiel:

```
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. Identifizieren Sie vorhandene Passphrasen auf dem Volume oder Snapshot.

Volumen

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

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

Schnappschuss

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

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

3. Aktualisieren Sie das LUKS-Geheimnis für das Volume, um die neue und die vorherige Passphrase anzugeben. Sicherstellen `previous-luke-passphrase-name` Und `previous-luks-passphrase` Die vorherige Passphrase muss übereinstimmen.

```
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. Erstelle einen neuen Pod, der das Volume einbindet. Dies ist erforderlich, um die Rotation einzuleiten.
5. Überprüfen Sie, ob die Passphrase geändert wurde.

Volumen

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

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

Schnappschuss

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

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

Ergebnisse

Die Passphrase wurde geändert, wenn auf dem Volume und im Snapshot nur noch die neue Passphrase angezeigt wird.

> Wenn zwei Passphrasen zurückgegeben werden, zum Beispiel `luksPassphraseNames: ["B", "A"]` Die Rotation ist unvollständig. Sie können eine neue Kapsel auslösen, um zu versuchen, die Rotation abzuschließen.

Volumenerweiterung aktivieren

Sie können die Volumenerweiterung auf einem LUKS-verschlüsselten Volumen aktivieren.

Schritte

1. Aktivieren Sie die `CSINodeExpandSecret` Feature Gate (Beta 1,25+). Siehe "Kubernetes 1.25: Verwendung von Secrets zur knotengesteuerten Erweiterung von CSI-Volumes" für Details.
2. Füge die `node-expand-secret-name` Und `node-expand-secret-namespace` StorageClass-Parameter. Beispiel:

```
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
```

Ergebnisse

Wenn Sie die Online-Speichererweiterung starten, übergibt das Kubelet die entsprechenden Anmeldeinformationen an den Treiber.

Kerberos-Verschlüsselung während des Fluges

Durch die Verwendung von Kerberos-Verschlüsselung während der Datenübertragung können Sie die Datensicherheit verbessern, indem Sie die Verschlüsselung für den Datenverkehr zwischen Ihrem verwalteten Cluster und dem Speicher-Backend aktivieren.

Trident unterstützt Kerberos-Verschlüsselung für ONTAP als Speicher-Backend:

- **On-Premise ONTAP** - Trident unterstützt die Kerberos-Verschlüsselung über NFSv3- und NFSv4-Verbindungen von Red Hat OpenShift und Upstream-Kubernetes-Clustern zu On-Premise ONTAP Volumes.

Sie können Volumes erstellen, löschen, ihre Größe ändern, Snapshots erstellen, klonen, schreibgeschützte Klone erstellen und importieren, die NFS-Verschlüsselung verwenden.

Konfigurieren der Kerberos-Verschlüsselung während der Übertragung mit lokalen ONTAP -Volumes

Sie können die Kerberos-Verschlüsselung für den Speicherdatenverkehr zwischen Ihrem verwalteten Cluster und einem lokalen ONTAP -Speicher-Backend aktivieren.

> Die Kerberos-Verschlüsselung für NFS-Datenverkehr mit lokalen ONTAP -Speicher-Backends wird nur mit folgender Konfiguration unterstützt: `ontap-nas` Speichertreiber.

Bevor Sie beginnen

- Stellen Sie sicher, dass Sie Zugriff auf die `tridentctl` Dienstprogramm.
- Stellen Sie sicher, dass Sie über Administratorzugriff auf das ONTAP -Speicher-Backend verfügen.
- Stellen Sie sicher, dass Sie den Namen des oder der Volumes kennen, die Sie vom ONTAP -Speicher -Backend freigeben werden.
- Stellen Sie sicher, dass Sie die ONTAP -Speicher-VM für die Unterstützung der Kerberos-Verschlüsselung für NFS-Volumes vorbereitet haben. Siehe "Kerberos auf einem dataLIF aktivieren" für Anweisungen.
- Stellen Sie sicher, dass alle NFSv4-Volumes, die Sie mit Kerberos-Verschlüsselung verwenden, korrekt konfiguriert sind. Siehe Abschnitt „NetApp NFSv4-Domänenkonfiguration“ (Seite 13) des "NetApp NFSv4-Erweiterungen und Best Practices-Leitfaden" .

ONTAP Exportrichtlinien hinzufügen oder ändern

Sie müssen bestehenden ONTAP Exportrichtlinien Regeln hinzufügen oder neue Exportrichtlinien erstellen, die die Kerberos-Verschlüsselung für das Root-Volume der ONTAP Speicher-VM sowie für alle ONTAP -Volumes unterstützen, die mit dem Upstream-Kubernetes-Cluster geteilt werden. Die von Ihnen hinzugefügten Exportrichtlinienregeln oder die von Ihnen erstellten neuen Exportrichtlinien müssen die folgenden Zugriffsprotokolle und Zugriffsberechtigungen unterstützen:

Zugriffsprotokolle

Konfigurieren Sie die Exportrichtlinie mit den Zugriffsprotokollen NFS, NFSv3 und NFSv4.

Zugangsdaten

Je nach Ihren Anforderungen an das Volume können Sie eine von drei verschiedenen Versionen der Kerberos-Verschlüsselung konfigurieren:

- **Kerberos 5** - (Authentifizierung und Verschlüsselung)
- **Kerberos 5i** - (Authentifizierung und Verschlüsselung mit Identitätsschutz)
- **Kerberos 5p** - (Authentifizierung und Verschlüsselung mit Identitäts- und Datenschutz)

Konfigurieren Sie die ONTAP Exportrichtlinienregel mit den entsprechenden Zugriffsberechtigungen. Wenn Cluster beispielsweise die NFS-Volumes mit einer Mischung aus Kerberos 5i- und Kerberos 5p-Verschlüsselung einbinden, verwenden Sie die folgenden Zugriffseinstellungen:

Typ	Nur-Lese-Zugriff	Lese-/Schreibzugriff	Superuser-Zugriff
UNIX	Ermöglicht	Ermöglicht	Ermöglicht
Kerberos 5i	Ermöglicht	Ermöglicht	Ermöglicht
Kerberos 5p	Ermöglicht	Ermöglicht	Ermöglicht

In der folgenden Dokumentation finden Sie Informationen zum Erstellen von ONTAP Exportrichtlinien und Exportrichtlinienregeln:

- "Erstellen einer Exportrichtlinie"
- "Hinzufügen einer Regel zu einer Exportrichtlinie"

Erstellen Sie ein Speicher-Backend

Sie können eine Trident -Speicher-Backend-Konfiguration erstellen, die die Kerberos-Verschlüsselungsfunktion beinhaltet.

Informationen zu diesem Vorgang

Wenn Sie eine Konfigurationsdatei für das Speicher-Backend erstellen, die die Kerberos-Verschlüsselung konfiguriert, können Sie mithilfe der folgenden Optionen eine von drei verschiedenen Versionen der Kerberos-Verschlüsselung angeben: `spec.nfsMountOptions` Parameter:

- `spec.nfsMountOptions: sec=krb5`(Authentifizierung und Verschlüsselung)
- `spec.nfsMountOptions: sec=krb5i`(Authentifizierung und Verschlüsselung mit Identitätsschutz)
- `spec.nfsMountOptions: sec=krb5p`(Authentifizierung und Verschlüsselung mit Identitäts- und Datenschutz)

Geben Sie nur eine Kerberos-Ebene an. Wenn Sie in der Parameterliste mehr als eine Kerberos-Verschlüsselungsstufe angeben, wird nur die erste Option verwendet.

Schritte

1. Erstellen Sie auf dem verwalteten Cluster eine Speicher-Backend-Konfigurationsdatei anhand des folgenden Beispiels. Ersetzen Sie die Werte in eckigen Klammern <> durch Informationen aus Ihrer Umgebung:

```
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. Verwenden Sie die im vorherigen Schritt erstellte Konfigurationsdatei, um das Backend zu erstellen:

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

Wenn die Backend-Erstellung fehlschlägt, stimmt etwas mit der Backend-Konfiguration nicht. Sie können die Protokolle einsehen, um die Ursache zu ermitteln, indem Sie folgenden Befehl ausführen:

```
tridentctl logs
```

Nachdem Sie das Problem mit der Konfigurationsdatei identifiziert und behoben haben, können Sie den Befehl zum Erstellen erneut ausführen.

Erstellen einer Speicherklasse

Sie können eine Speicherklasse erstellen, um Volumes mit Kerberos-Verschlüsselung bereitzustellen.

Informationen zu diesem Vorgang

Beim Erstellen eines Speicherklassenobjekts können Sie mithilfe der Kerberos-Verschlüsselungsmethode eine von drei verschiedenen Versionen der Kerberos-Verschlüsselung angeben. `mountOptions` Parameter:

- `mountOptions: sec=krb5`(Authentifizierung und Verschlüsselung)
- `mountOptions: sec=krb5i`(Authentifizierung und Verschlüsselung mit Identitätsschutz)
- `mountOptions: sec=krb5p`(Authentifizierung und Verschlüsselung mit Identitäts- und Datenschutz)

Geben Sie nur eine Kerberos-Ebene an. Wenn Sie in der Parameterliste mehr als eine Kerberos-Verschlüsselungsstufe angeben, wird nur die erste Option verwendet. Wenn der in der Speicher-Backend-Konfiguration angegebene Verschlüsselungsgrad von dem im Speicherklassenobjekt angegebenen Grad abweicht, hat das Speicherklassenobjekt Vorrang.

Schritte

1. Erstellen Sie ein StorageClass-Kubernetes-Objekt anhand des folgenden Beispiels:

```
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. Erstellen Sie die Speicherklasse:

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

3. Stellen Sie sicher, dass die Speicherklasse erstellt wurde:

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

   Die Ausgabe sollte in etwa wie folgt aussehen:

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

Bereitstellungsmengen

Nachdem Sie ein Speicher-Backend und eine Speicherklasse erstellt haben, können Sie nun ein Volume bereitstellen. Anweisungen hierzu finden Sie unter "Bereitstellung eines Volumens" .

Konfigurieren der Kerberos-Verschlüsselung während der Übertragung mit Azure NetApp Files -Volumes

Sie können die Kerberos-Verschlüsselung für den Speicherdatenverkehr zwischen Ihrem verwalteten Cluster und einem einzelnen Azure NetApp Files -Speicher-Backend oder einem virtuellen Pool von Azure NetApp Files -Speicher-Backends aktivieren.

Bevor Sie beginnen

- Stellen Sie sicher, dass Sie Trident auf dem verwalteten Red Hat OpenShift-Cluster aktiviert haben.
- Stellen Sie sicher, dass Sie Zugriff auf die `tridentctl` Dienstprogramm.
- Stellen Sie sicher, dass Sie das Azure NetApp Files -Speicher-Backend für die Kerberos-Verschlüsselung vorbereitet haben, indem Sie die Anforderungen beachten und die Anweisungen in [Link einfügen] befolgen. "Azure NetApp Files Dokumentation" .
- Stellen Sie sicher, dass alle NFSv4-Volumes, die Sie mit Kerberos-Verschlüsselung verwenden, korrekt konfiguriert sind. Siehe Abschnitt „NetApp NFSv4-Domänenkonfiguration“ (Seite 13) des "NetApp NFSv4-Erweiterungen und Best Practices-Leitfaden" .

Erstellen Sie ein Speicher-Backend

Sie können eine Azure NetApp Files -Speicher-Backend-Konfiguration erstellen, die die Kerberos-Verschlüsselungsfunktion beinhaltet.

Informationen zu diesem Vorgang

Wenn Sie eine Konfigurationsdatei für das Speicher-Backend erstellen, die die Kerberos-Verschlüsselung konfiguriert, können Sie festlegen, dass diese auf einer von zwei möglichen Ebenen angewendet werden soll:

- Die **Speicher-Backend-Ebene** unter Verwendung der `spec.kerberos` Feld
- Die **virtuelle Poolebene** unter Verwendung der `spec.storage.kerberos` Feld

Wenn Sie die Konfiguration auf Ebene des virtuellen Pools definieren, wird der Pool anhand der Bezeichnung in der Speicherklasse ausgewählt.

Auf beiden Ebenen können Sie eine von drei verschiedenen Versionen der Kerberos-Verschlüsselung angeben:

- `kerberos: sec=krb5`(Authentifizierung und Verschlüsselung)
- `kerberos: sec=krb5i`(Authentifizierung und Verschlüsselung mit Identitätsschutz)
- `kerberos: sec=krb5p`(Authentifizierung und Verschlüsselung mit Identitäts- und Datenschutz)

Schritte

1. Erstellen Sie auf dem verwalteten Cluster eine Storage-Backend-Konfigurationsdatei anhand eines der folgenden Beispiele, je nachdem, wo Sie das Storage-Backend definieren müssen (Storage-Backend-Ebene oder Virtual-Pool-Ebene). Ersetzen Sie die Werte in eckigen Klammern <> durch Informationen aus Ihrer Umgebung:

Beispiel auf Speicher-Backend-Ebene

```
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
```

Beispiel für ein virtuelles Schwimmbecken

```
---
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. Verwenden Sie die im vorherigen Schritt erstellte Konfigurationsdatei, um das Backend zu erstellen:

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

Wenn die Backend-Erstellung fehlschlägt, stimmt etwas mit der Backend-Konfiguration nicht. Sie können die Protokolle einsehen, um die Ursache zu ermitteln, indem Sie folgenden Befehl ausführen:

```
tridentctl logs
```

Nachdem Sie das Problem mit der Konfigurationsdatei identifiziert und behoben haben, können Sie den Befehl zum Erstellen erneut ausführen.

Erstellen einer Speicherklasse

Sie können eine Speicherklasse erstellen, um Volumes mit Kerberos-Verschlüsselung bereitzustellen.

Schritte

1. Erstellen Sie ein StorageClass-Kubernetes-Objekt anhand des folgenden Beispiels:

```
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. Erstellen Sie die Speicherklasse:

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

3. Stellen Sie sicher, dass die Speicherklasse erstellt wurde:

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

Die Ausgabe sollte in etwa wie folgt aussehen:

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

Bereitstellungsmengen

Nachdem Sie ein Speicher-Backend und eine Speicherklasse erstellt haben, können Sie nun ein Volume bereitstellen. Anweisungen hierzu finden Sie unter "Bereitstellung eines Volumens" .