Backends verwalten

Trident

NetApp
January 15, 2026

Inhalt

Backends verwalten . . . 1
- Führen Sie die Backend-Verwaltung mit kubectl durch. . . . 1
  - Backend löschen . . . 1
  - Die vorhandenen Backends ansehen . . . 1
  - Aktualisieren Sie ein Backend . . . 1
- Führen Sie die Backend-Verwaltung mit tridentctl durch. . . . 2
  - Erstelle ein Backend . . . 2
  - Backend löschen . . . 2
  - Die vorhandenen Backends ansehen . . . 3
  - Aktualisieren Sie ein Backend . . . 3
  - Identifizieren Sie die Speicherklassen, die ein Backend verwenden. . . . 3
- Wechseln Sie zwischen verschiedenen Backend-Verwaltungsoptionen. . . . 4
  - Optionen zur Verwaltung von Backends . . . 4
  - Verwalten `tridentctl` Backends verwenden `TridentBackendConfig` . . . 4
  - Verwalten `TridentBackendConfig` Backends verwenden `tridentctl` . . . 9

Backends verwalten

Führen Sie die Backend-Verwaltung mit kubectl durch.

Erfahren Sie, wie Sie Backend-Verwaltungsvorgänge durchführen, indem Sie `kubectl` .

Backend löschen

Durch das Löschen eines `TridentBackendConfig` Sie weisen Trident an, Backends zu löschen/beizubehalten (basierend auf `deletionPolicy`). Um ein Backend zu löschen, stellen Sie sicher, dass `deletionPolicy` ist auf Löschen eingestellt. Nur das löschen `TridentBackendConfig` , sicherstellen, dass `deletionPolicy` ist auf Beibehaltung eingestellt. Dadurch wird sichergestellt, dass das Backend weiterhin vorhanden ist und mithilfe von verwaltet werden kann. `tridentctl` .

Führen Sie den folgenden Befehl aus:

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

Trident löscht die Kubernetes-Secrets nicht, die von diesem Unternehmen verwendet wurden. `TridentBackendConfig` . Der Kubernetes-Benutzer ist für das Bereinigen von Geheimnissen verantwortlich. Beim Löschen von Geheimnissen ist Vorsicht geboten. Geheimnisse sollten nur dann gelöscht werden, wenn sie von den Backends nicht verwendet werden.

Die vorhandenen Backends ansehen

Führen Sie den folgenden Befehl aus:

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

Sie können auch ausführen `tridentctl get backend -n trident` oder `tridentctl get backend -o yaml -n trident` um eine Liste aller existierenden Backends zu erhalten. Diese Liste wird auch Backends enthalten, die mit `tridentctl` .

Aktualisieren Sie ein Backend

Es kann mehrere Gründe für ein Backend-Update geben:

- Die Zugangsdaten zum Speichersystem haben sich geändert. Um die Anmeldeinformationen zu aktualisieren, muss das Kubernetes-Secret, das im `TridentBackendConfig` Das Objekt muss aktualisiert werden. Trident aktualisiert das Backend automatisch mit den zuletzt angegebenen Zugangsdaten. Führen Sie folgenden Befehl aus, um das Kubernetes-Secret zu aktualisieren:

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

- Parameter (wie zum Beispiel der Name der verwendeten ONTAP SVM) müssen aktualisiert werden.

- Sie können aktualisieren `TridentBackendConfig` Objekte direkt über Kubernetes mit folgendem Befehl:

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

- Alternativ können Sie Änderungen an der bestehenden Datei vornehmen. `TridentBackendConfig` CR mit folgendem Befehl:

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

- Wenn ein Backend-Update fehlschlägt, bleibt das Backend in seiner zuletzt bekannten Konfiguration. Sie können die Protokolle einsehen, um die Ursache zu ermitteln, indem Sie Folgendes ausführen: `kubectl get tbc <tbc-name> -o yaml -n trident` oder `kubectl describe tbc <tbc-name> -n trident`.
- Nachdem Sie das Problem mit der Konfigurationsdatei identifiziert und behoben haben, können Sie den Aktualisierungsbefehl erneut ausführen.

Führen Sie die Backend-Verwaltung mit tridentctl durch.

Erfahren Sie, wie Sie Backend-Verwaltungsvorgänge durchführen, indem Sie `tridentctl`.

Erstelle ein Backend

Nachdem Sie ein"Backend-Konfigurationsdatei" Führen Sie folgenden Befehl aus:

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

Wenn die Backend-Erstellung fehlschlägt, gab es ein Problem mit der Backend-Konfiguration. Sie können die Protokolle einsehen, um die Ursache zu ermitteln, indem Sie folgenden Befehl ausführen:

```
tridentctl logs -n trident
```

Nachdem Sie das Problem mit der Konfigurationsdatei identifiziert und behoben haben, können Sie es einfach ausführen. `create` Den Befehl erneut geben.

Backend löschen

Um ein Backend aus Trident zu löschen, gehen Sie wie folgt vor:

1. Den Backend-Namen abrufen:

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

2. Backend löschen:

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

> Falls Trident über dieses Backend Volumes und Snapshots bereitgestellt hat, die noch existieren, verhindert das Löschen des Backends, dass neue Volumes über dieses Backend bereitgestellt werden. Das Backend wird weiterhin im Status „Löschen“ verbleiben.

Die vorhandenen Backends ansehen

Um die von Trident bekannten Backends anzuzeigen, gehen Sie wie folgt vor:

- Um eine Zusammenfassung zu erhalten, führen Sie folgenden Befehl aus:

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

- Um alle Details zu erhalten, führen Sie folgenden Befehl aus:

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

Aktualisieren Sie ein Backend

Nachdem Sie eine neue Backend-Konfigurationsdatei erstellt haben, führen Sie folgenden Befehl aus:

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

Wenn das Backend-Update fehlschlägt, lag entweder ein Fehler in der Backend-Konfiguration vor oder Sie haben versucht, ein ungültiges Update durchzuführen. Sie können die Protokolle einsehen, um die Ursache zu ermitteln, indem Sie folgenden Befehl ausführen:

```
tridentctl logs -n trident
```

Nachdem Sie das Problem mit der Konfigurationsdatei identifiziert und behoben haben, können Sie es einfach ausführen. `update` Den Befehl erneut geben.

Identifizieren Sie die Speicherklassen, die ein Backend verwenden.

Dies ist ein Beispiel für die Art von Fragen, die Sie mit JSON beantworten können. `tridentctl` Ausgaben für Backend-Objekte. Dies verwendet die `jq` Dienstprogramm, das Sie installieren müssen.

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

Dies gilt auch für Backends, die durch die Verwendung von [fehlende Information] erstellt wurden. `TridentBackendConfig` .

Wechseln Sie zwischen verschiedenen Backend-Verwaltungsoptionen.

Erfahren Sie mehr über die verschiedenen Möglichkeiten zur Backend-Verwaltung in Trident.

Optionen zur Verwaltung von Backends

Mit der Einführung von `TridentBackendConfig` Administratoren haben nun zwei unterschiedliche Möglichkeiten, Backends zu verwalten. Dies wirft folgende Fragen auf:

- Können Backends erstellt werden mit `tridentctl` verwaltet werden mit `TridentBackendConfig` ?
- Können Backends erstellt werden mit `TridentBackendConfig` verwaltet werden mit `tridentctl` ?

Verwalten `tridentctl` Backends verwenden `TridentBackendConfig`

Dieser Abschnitt beschreibt die erforderlichen Schritte zur Verwaltung von Backends, die mit folgendem Werkzeug erstellt wurden: `tridentctl` direkt über die Kubernetes-Schnittstelle durch Erstellen `TridentBackendConfig` Objekte.

Dies gilt für folgende Szenarien:

- Vorhandene Backends, die keine haben `TridentBackendConfig` weil sie mit `tridentctl` .
- Neue Backends, die mit `tridentctl` während andere `TridentBackendConfig` Objekte existieren.

In beiden Szenarien bleiben die Backend-Systeme weiterhin vorhanden, über die Trident die Datenmengen plant und verarbeitet. Administratoren haben hier zwei Möglichkeiten:

- Weiter verwenden `tridentctl` zur Verwaltung von Backends, die damit erstellt wurden.
- Bind-Backends, die mit `tridentctl` zu einem neuen `TridentBackendConfig` Objekt. Dies würde bedeuten, dass die Backends wie folgt verwaltet werden: `kubectl` und nicht `tridentctl` .

Um ein bereits bestehendes Backend zu verwalten `kubectl` Sie müssen einen erstellen `TridentBackendConfig` das an das bestehende Backend angebunden wird. Hier ist eine Übersicht, wie das funktioniert:

1. Erstelle ein Kubernetes-Secret. Das Geheimnis enthält die Anmeldeinformationen, die Trident für die Kommunikation mit dem Speichercluster/-dienst benötigt.
2. Erstellen Sie ein `TridentBackendConfig` Objekt. Dieser Eintrag enthält spezifische Informationen zum Speichercluster/Speicherdienst und verweist auf das im vorherigen Schritt erstellte Geheimnis. Es ist darauf zu achten, dass identische Konfigurationsparameter angegeben werden (wie z. B.

`spec.backendName` , `spec.storagePrefix` , `spec.storageDriverName` , und so weiter). `spec.backendName` muss auf den Namen des bestehenden Backends eingestellt werden.

Schritt 0: Backend identifizieren

Um ein `TridentBackendConfig` Um eine Verbindung zu einem bestehenden Backend herzustellen, müssen Sie die Backend-Konfiguration abrufen. In diesem Beispiel gehen wir davon aus, dass ein Backend mit der folgenden JSON-Definition erstellt wurde:

```
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"
      }
    }
  ]
}
```

Schritt 1: Erstellen Sie ein Kubernetes-Secret.

Erstellen Sie ein Geheimnis, das die Anmeldeinformationen für das Backend enthält, wie in diesem Beispiel gezeigt:

```
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
```

Schritt 2: Erstellen Sie ein `TridentBackendConfig` CR

Der nächste Schritt besteht darin, ein/e zu erstellen `TridentBackendConfig` CR, das automatisch an das bereits vorhandene gebunden wird `ontap-nas-backend` (wie in diesem Beispiel). Stellen Sie sicher, dass die folgenden Anforderungen erfüllt sind:

- Derselbe Backend-Name ist definiert in `spec.backendName` .
- Die Konfigurationsparameter sind identisch mit denen des ursprünglichen Backends.
- Virtuelle Pools (sofern vorhanden) müssen die gleiche Reihenfolge wie im ursprünglichen Backend beibehalten.
- Die Zugangsdaten werden über ein Kubernetes-Secret und nicht im Klartext bereitgestellt.

In diesem Fall `TridentBackendConfig` wird folgendermaßen aussehen:

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

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

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

Schritt 3: Überprüfen Sie den Status des `TridentBackendConfig` CR

Nach dem `TridentBackendConfig` wurde erstellt, seine Phase muss sein `Bound` . Es sollte außerdem denselben Backend-Namen und dieselbe UUID wie das bestehende Backend aufweisen.

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

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

Das Backend wird nun vollständig über die Software verwaltet. `tbc-ontap-nas-backend` `TridentBackendConfig` Objekt.

Verwalten `TridentBackendConfig` Backends verwenden `tridentctl`

```
`tridentctl`kann verwendet werden, um Backends aufzulisten, die mit
folgendem Werkzeug erstellt wurden: `TridentBackendConfig` .  Darüber
hinaus können Administratoren diese Backends auch vollständig selbst
verwalten durch `tridentctl` durch Löschen `TridentBackendConfig` und
sicherstellen `spec.deletionPolicy` ist eingestellt auf `retain` .
```

Schritt 0: Backend identifizieren

Nehmen wir beispielsweise an, das folgende Backend wurde erstellt mit `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 |
+-------------------+----------------
+--------------------------------------+--------+---------+
```

Aus der Ausgabe geht hervor, dass `TridentBackendConfig` wurde erfolgreich erstellt und ist an ein Backend gebunden [siehe die UUID des Backends].

Schritt 1: Bestätigen `deletionPolicy` **ist eingestellt auf** `retain`

Werfen wir einen Blick auf den Wert von `deletionPolicy` . Dies muss auf eingestellt werden `retain` . Dies stellt sicher, dass, wenn ein `TridentBackendConfig` Wenn CR gelöscht wird, bleibt die Backend-Definition weiterhin vorhanden und kann verwaltet werden mit `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
```

Fahren Sie nicht mit dem nächsten Schritt fort, es sei denn `deletionPolicy` ist eingestellt auf `retain` .

Schritt 2: Löschen Sie die `TridentBackendConfig` CR

Der letzte Schritt besteht darin, die `TridentBackendConfig` CR. Nach Bestätigung der `deletionPolicy` ist eingestellt auf `retain` Sie können die Löschung nun durchführen:

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

Nach der Löschung der `TridentBackendConfig` Trident entfernt das Objekt einfach, ohne das Backend selbst zu löschen.