Backends verwalten

Trident

NetApp
July 01, 2026

Inhalt

Backends verwalten

Führen Sie die Backend-Verwaltung mit kubectl durch

Erfahren Sie, wie Sie Backend-Verwaltungsvorgänge mithilfe von `kubectl` durchführen.

Ein Backend löschen

Durch das Löschen eines `TridentBackendConfig` weisen Sie Trident an, Backends zu löschen oder beizubehalten (basierend auf `deletionPolicy`). Um ein Backend zu löschen, stellen Sie sicher, dass `deletionPolicy` auf „delete“ gesetzt ist. Um nur das `TridentBackendConfig` zu löschen, stellen Sie sicher, dass `deletionPolicy` auf „retain“ gesetzt ist. Dadurch wird sichergestellt, dass das Backend weiterhin vorhanden ist und mit `tridentctl` verwaltet werden kann.

Führen Sie den folgenden Befehl aus:

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

Trident löscht die Kubernetes-Secrets, die von `TridentBackendConfig` verwendet wurden, nicht. Der Kubernetes-Benutzer ist für die Bereinigung der Secrets verantwortlich. Beim Löschen von Secrets ist Vorsicht geboten. Sie sollten Secrets nur löschen, wenn sie von den Backends nicht verwendet werden.

Vorhandene Backends anzeigen

Führen Sie den folgenden Befehl aus:

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

Sie können auch `tridentctl get backend -n trident` oder `tridentctl get backend -o yaml -n trident` ausführen, um eine Liste aller vorhandenen Backends zu erhalten. Diese Liste enthält auch Backends, die mit `tridentctl` erstellt wurden.

Aktualisieren Sie ein Backend

Es kann mehrere Gründe geben, ein Backend zu aktualisieren:

- Die Zugangsdaten für das Speichersystem haben sich geändert. Um die Zugangsdaten zu aktualisieren, muss das Kubernetes-Secret, das im `TridentBackendConfig` Objekt verwendet wird, aktualisiert werden. Trident aktualisiert das Backend automatisch mit den neuesten Zugangsdaten. Führen Sie den 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 `TridentBackendConfig` Objekte direkt über Kubernetes mit dem folgenden Befehl aktualisieren:

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

- Alternativ können Sie Änderungen an der bestehenden `TridentBackendConfig` CR mit dem folgenden Befehl vornehmen:

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

- Schlägt ein Backend-Update fehl, bleibt das Backend in seiner zuletzt bekannten Konfiguration. Sie können die Protokolle einsehen, um die Ursache zu ermitteln, indem Sie `kubectl get tbc <tbc-name> -o yaml -n trident` oder `kubectl describe tbc <tbc-name> -n trident` ausführen.
- 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 mithilfe von `tridentctl` durchführen.

Erstelle ein Backend

Nachdem Sie eine "Konfigurationsdatei für das Backend" erstellt haben, führen Sie den 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 den folgenden Befehl ausführen:

```
tridentctl logs -n trident
```

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

Ein 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 Volumes und Snapshots von diesem Backend bereitgestellt hat, die noch existieren, verhindert das Löschen des Backends, dass neue Volumes von ihm bereitgestellt werden. Das Backend verbleibt im Status „Wird gelöscht“.

Vorhandene Backends anzeigen

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

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

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

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

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

Aktualisieren Sie ein Backend

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

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

Wenn die Backend-Aktualisierung fehlschlägt, lag ein Fehler in der Backend-Konfiguration vor oder Sie haben eine ungültige Aktualisierung versucht. Sie können die Protokolle einsehen, um die Ursache zu ermitteln, indem Sie den folgenden Befehl ausführen:

```
tridentctl logs -n trident
```

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

Identifizieren Sie die Speicherklassen, die ein Backend verwenden

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

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

Dies gilt auch für Backends, die mit `TridentBackendConfig` erstellt wurden.

Wechseln Sie zwischen Backend-Verwaltungsoptionen

Erfahren Sie mehr über die verschiedenen Möglichkeiten, Backends in Trident zu verwalten.

Optionen zur Verwaltung von Backends

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

- Können Backends, die mit `tridentctl` erstellt wurden, mit `TridentBackendConfig` verwaltet werden?
- Können Backends, die mit `TridentBackendConfig` erstellt wurden, mit `tridentctl` verwaltet werden?

Backends mit `tridentctl` verwalten unter Verwendung von `TridentBackendConfig`

Dieser Abschnitt beschreibt die Schritte, die erforderlich sind, um Backends zu verwalten, die mit `tridentctl` direkt über die Kubernetes-Schnittstelle durch das Erstellen von `TridentBackendConfig` Objekten erstellt wurden.

Dies gilt für die folgenden Szenarien:

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

In beiden Szenarien bleiben die Backends weiterhin vorhanden, wobei Trident die Volumes plant und auf ihnen arbeitet. Administratoren haben hier zwei Möglichkeiten:

- Verwenden Sie `tridentctl` weiterhin, um Backends zu verwalten, die damit erstellt wurden.
- Binden Sie Backends, die mit `tridentctl` erstellt wurden, an ein neues `TridentBackendConfig` Objekt. Dadurch werden die Backends mit `kubectl` und nicht mit `tridentctl` verwaltet.

Um ein bestehendes Backend mit `kubectl` zu verwalten, müssen Sie eine `TridentBackendConfig` erstellen, die an das bestehende Backend gebunden ist. Hier ist eine Übersicht, wie das funktioniert:

1. Erstellen Sie ein Kubernetes-Secret. Das Secret enthält die Anmeldeinformationen, die Trident für die Kommunikation mit dem Speichercluster bzw. -dienst benötigt.
2. Erstellen Sie ein `TridentBackendConfig` Objekt. Dieses enthält spezifische Informationen zum Speicher-Cluster/-Dienst und verweist auf das im vorherigen Schritt erstellte Geheimnis. Achten Sie darauf, identische Konfigurationsparameter anzugeben (wie `spec.backendName`, `spec.storagePrefix`, `spec.storageDriverName` usw.). `spec.backendName` muss auf den Namen des vorhandenen Backends gesetzt werden.

Schritt 0: Backend identifizieren

Um ein `TridentBackendConfig` zu erstellen, das an ein bestehendes Backend gebunden ist, müssen Sie die Backend-Konfiguration abrufen. In diesem Beispiel nehmen wir an, 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 Secret, 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 einen `TridentBackendConfig`CR

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

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

In diesem Fall wird die `TridentBackendConfig`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 der `TridentBackendConfig` CR

Nachdem das `TridentBackendConfig` erstellt wurde, muss seine Phase `Bound` sein. 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 das `tbc-ontap-nas-backend` `TridentBackendConfig` Objekt verwaltet.

Backends mit `TridentBackendConfig` verwalten unter Verwendung von `tridentctl`

```
`tridentctl` kann verwendet werden, um Backends aufzulisten, die mit
`TridentBackendConfig` erstellt wurden. Darüber hinaus können
Administratoren solche Backends auch vollständig über  `tridentctl`
verwalten, indem sie  `TridentBackendConfig` löschen und sicherstellen,
dass  `spec.deletionPolicy` auf  `retain` gesetzt ist.
```

Schritt 0: Backend identifizieren

Nehmen wir beispielsweise an, das folgende Backend wurde mit `TridentBackendConfig` erstellt:

```
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` erfolgreich erstellt wurde und mit einem Backend verbunden ist [beachten Sie die UUID des Backends].

Schritt 1: Bestätigen Sie, dass `deletionPolicy` auf `retain` gesetzt ist

Betrachten wir den Wert von `deletionPolicy`. Dieser muss auf `retain`gesetzt werden. Dadurch wird sichergestellt, dass beim Löschen eines `TridentBackendConfig` CR die Backend-Definition weiterhin vorhanden ist und mit `tridentctl`verwaltet werden kann.

```
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 auf `retain` gesetzt.

Schritt 2: Löschen Sie das `TridentBackendConfig` CR

Der letzte Schritt besteht darin, die `TridentBackendConfig` CR zu löschen. Nachdem Sie bestätigt haben, dass das `deletionPolicy` auf `retain` gesetzt ist, können Sie mit dem Löschen fortfahren:

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

Beim Löschen des `TridentBackendConfig` Objekts entfernt Trident dieses einfach, ohne das Backend selbst zu löschen.