백엔드 관리

Trident

NetApp
September 26, 2025

This PDF was generated from https://docs.netapp.com/ko-kr/trident-2410/trident-use/backend_ops_kubectl.html on September 26, 2025. Always check docs.netapp.com for the latest.

목차

백엔드 관리 . . . 1
kubeck을 사용하여 백엔드 관리 수행 . . . 1
백엔드를 삭제합니다 . . . 1
기존 백엔드를 봅니다 . . . 1
백엔드를 업데이트합니다 . . . 1
tridentctl을 사용하여 백엔드 관리를 수행합니다 . . . 2
백엔드를 생성합니다 . . . 2
백엔드를 삭제합니다 . . . 2
기존 백엔드를 봅니다 . . . 3
백엔드를 업데이트합니다 . . . 3
백엔드를 사용하는 스토리지 클래스를 식별합니다 . . . 3
백엔드 관리 옵션 간 이동 . . . 4
백엔드 관리 옵션 . . . 4
을 사용하여 백엔드 `TridentBackendConfig` 관리 `tridentctl` . . . 4
을 사용하여 백엔드 `tridentctl` 관리 `TridentBackendConfig` . . . 8

백엔드 관리

kubeck을 사용하여 백엔드 관리 수행

을 사용하여 백엔드 관리 작업을 수행하는 방법에 대해 `kubectl` 알아봅니다.

백엔드를 삭제합니다

를 삭제하면 `TridentBackendConfig` Trident가 백엔드를 삭제/유지하도록 지시합니다(기준 `deletionPolicy`). 백엔드를 삭제하려면 가 `deletionPolicy` DELETE로 설정되어 있는지 확인합니다. 만 삭제하려면 `TridentBackendConfig` 가 유지 로 설정되어 있는지 `deletionPolicy` 확인합니다. 이렇게 하면 백엔드가 계속 존재하고 을 사용하여 관리할 수 `tridentctl` 있습니다.

다음 명령을 실행합니다.

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

Trident은 에서 사용 중인 Kubernetes 비밀을 삭제하지 `TridentBackendConfig` 않습니다. Kubernetes 사용자는 기밀을 정해야 합니다. 비밀 정보를 삭제할 때는 주의해야 합니다. 암호는 백엔드에서 사용하지 않는 경우에만 삭제해야 합니다.

기존 백엔드를 봅니다

다음 명령을 실행합니다.

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

또는 를 `tridentctl get backend -o yaml -n trident` 실행하여 존재하는 모든 백엔드의 목록을 가져올 수도 `tridentctl get backend -n trident` 있습니다. 이 목록에는 로 만든 백엔드도 `tridentctl` 포함됩니다.

백엔드를 업데이트합니다

백엔드를 업데이트해야 하는 이유는 여러 가지가 있을 수 있습니다.

- 스토리지 시스템에 대한 자격 증명이 변경되었습니다. 자격 증명을 업데이트하려면 개체에서 사용되는 Kubernetes 암호를 `TridentBackendConfig` 업데이트해야 합니다. Trident는 제공된 최신 자격 증명으로 백엔드를 자동으로 업데이트합니다. 다음 명령을 실행하여 Kubernetes Secret를 업데이트하십시오.

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

- 매개 변수(예: 사용 중인 ONTAP SVM의 이름)를 업데이트해야 합니다.
  - 다음 명령을 사용하여 Kubernetes를 통해 오브젝트를 직접 업데이트할 수 `TridentBackendConfig` 있습니다.

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

- 또는 다음 명령을 사용하여 기존 CR을 변경할 수 `TridentBackendConfig` 있습니다.

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

- 백엔드 업데이트에 실패하면 백엔드는 마지막으로 알려진 구성으로 계속 유지됩니다. 또는 `kubectl describe tbc <tbc-name> -n trident` 를 실행하여 로그를 보고 원인을 확인할 수 `kubectl get tbc <tbc-name> -o yaml -n trident` 있습니다.
- 구성 파일의 문제를 확인하고 수정한 후 update 명령을 다시 실행할 수 있습니다.

tridentctl을 사용하여 백엔드 관리를 수행합니다

을 사용하여 백엔드 관리 작업을 수행하는 방법에 대해 `tridentctl` 알아봅니다.

백엔드를 생성합니다

를 생성한 후 "백엔드 구성 파일"다음 명령을 실행합니다.

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

백엔드 생성에 실패하면 백엔드 구성에 문제가 있는 것입니다. 다음 명령을 실행하여 로그를 보고 원인을 확인할 수 있습니다.

```
tridentctl logs -n trident
```

구성 파일에서 문제를 확인하고 수정한 후에는 명령을 다시 실행하면 `create` 됩니다.

백엔드를 삭제합니다

Trident에서 백엔드를 삭제하려면 다음을 수행합니다.

1. 백엔드 이름 검색:

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

2. 백엔드를 삭제합니다.

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

Trident가 이 백엔드에서 아직 존재하는 볼륨 및 스냅샷을 프로비저닝한 경우 백엔드를 삭제하면 새 볼륨이 프로비저닝되지 않습니다. 백엔드는 계속해서 "삭제" 상태에 있으며, Trident는 삭제될 때까지 해당 볼륨 및 스냅샷을 계속 관리합니다.

기존 백엔드를 봅니다

Trident가 알고 있는 백엔드를 보려면 다음을 실행합니다.

- 요약을 보려면 다음 명령을 실행합니다.

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

- 모든 세부 정보를 보려면 다음 명령을 실행합니다.

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

백엔드를 업데이트합니다

새 백엔드 구성 파일을 생성한 후 다음 명령을 실행합니다.

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

백엔드 업데이트에 실패하면 백엔드 구성에 문제가 있거나 잘못된 업데이트를 시도했습니다. 다음 명령을 실행하여 로그를 보고 원인을 확인할 수 있습니다.

```
tridentctl logs -n trident
```

구성 파일에서 문제를 확인하고 수정한 후에는 명령을 다시 실행하면 `update` 됩니다.

백엔드를 사용하는 스토리지 클래스를 식별합니다

다음은 백엔드 객체에 대해 출력하는 JSON으로 답변할 수 있는 질문의 `tridentctl` 예입니다. 이 `jq` 유틸리티는 설치해야 하는 유틸리티를 사용합니다.

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

이는 을 사용하여 만든 백엔드에도 `TridentBackendConfig` 적용됩니다.

백엔드 관리 옵션 간 이동

Trident에서 백엔드를 관리하는 다양한 방법에 대해 알아보십시오.

백엔드 관리 옵션

의 도입으로 `TridentBackendConfig` 관리자는 이제 두 가지 고유한 방식으로 백엔드를 관리할 수 있습니다. 이 질문은 다음과 같습니다.

- 을 사용하여 만든 백엔드를 와 함께 관리할 `TridentBackendConfig` 수 `tridentctl` 있습니까?
- 을 사용하여 만든 백엔드를 사용하여 관리할 `tridentctl` 수 `TridentBackendConfig` 있습니까?

을 사용하여 백엔드 `TridentBackendConfig` 관리 `tridentctl`

이 섹션에서는 오브젝트를 생성하여 Kubernetes 인터페이스를 통해 직접 `TridentBackendConfig` 생성된 백엔드를 관리하는 데 필요한 단계를 설명합니다 `tridentctl`.

이 내용은 다음 시나리오에 적용됩니다.

- 에서 만들어졌기 때문에 `tridentctl` 가 없는 기존 백엔드가 `TridentBackendConfig` 있습니다.
- 다른 오브젝트가 있는 동안 `TridentBackendConfig` 로 만들어진 새 백엔드가 `tridentctl` 있습니다.

두 시나리오 모두에서 Trident의 볼륨 예약 및 운영 기능과 함께 백엔드가 계속 표시됩니다. 관리자는 다음 두 가지 옵션 중 하나를 선택할 수 있습니다.

- 를 계속 사용하여 `tridentctl` 만든 백엔드를 관리합니다.
- 을 사용하여 만든 백엔드를 새 `TridentBackendConfig` 개체에 바인딩합니다. `tridentctl` 이렇게 하면 백엔드가 을 사용하여 관리되고, 그렇지 않습니다 `kubectl` `tridentctl`.

을 사용하여 기존 백엔드를 관리하려면 `kubectl` 기존 백엔드에 바인딩되는 을 `TridentBackendConfig` 생성해야 합니다. 작동 방식에 대한 개요는 다음과 같습니다.

1. Kubernetes 암호를 생성하십시오. 비밀에는 Trident가 스토리지 클러스터/서비스와 통신하는 데 필요한 자격 증명이 포함되어 있습니다.
2. 개체를 만듭니다. `TridentBackendConfig` 스토리지 클러스터/서비스에 대한 자세한 내용과 이전 단계에서 생성한 암호를 참조하십시오. 동일한 구성 매개 변수(`spec.storagePrefix`,, `spec.storageDriverName` 등)를 지정할 때 주의를 기울여야 `spec.backendName` 합니다. `spec.backendName` 기존 백엔드의 이름으로 설정해야 합니다.

단계 **0:** 백엔드를 식별합니다

기존 백엔드에 바인딩되는 를 생성하려면 `TridentBackendConfig` 백엔드 구성을 얻어야 합니다. 이 예에서는 다음과 같은 JSON 정의를 사용하여 백엔드를 생성했다고 가정합니다.

```
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"
            }
        }
```

```
    ]
}
```

1단계: **Kubernetes Secret** 생성

이 예에 표시된 것처럼 백엔드에 대한 자격 증명이 포함된 암호를 생성합니다.

```
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
```

2단계: `TridentBackendConfig` **CR**을 생성합니다

다음 단계는 기존 시스템에 자동으로 바인딩되는 CR을 `ontap-nas-backend` 만드는 `TridentBackendConfig` 것입니다(이 예제와 같이). 다음 요구 사항이 충족되는지 확인합니다.

- 동일한 백엔드 이름이 에 정의되어 `spec.backendName` 있습니다.
- 구성 매개 변수는 원래 백엔드와 동일합니다.
- 가상 풀(있는 경우)은 원래 백엔드와 동일한 순서를 유지해야 합니다.
- 자격 증명은 일반 텍스트가 아닌 Kubernetes Secret을 통해 제공됩니다.

이 경우 는 `TridentBackendConfig` 다음과 같이 표시됩니다.

```
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
```

3단계: CR의 상태를 확인합니다 `TridentBackendConfig`

를 생성한 후에는 `TridentBackendConfig` 해당 단계가 되어야 `Bound`합니다. 또한 기존 백엔드의 백엔드 이름과 UUID도 동일하게 반영되어야 합니다.

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

이제 백엔드가 객체를 사용하여 완전히 `tbc-ontap-nas-backend` `TridentBackendConfig` 관리됩니다.

을 사용하여 백엔드 `tridentctl` 관리 `TridentBackendConfig`

`tridentctl` 을 사용하여 만든 백엔드를 나열하는 데 사용할 수 `TridentBackendConfig` 있습니다. 또한 관리자는 `spec.deletionPolicy` 를 삭제하고 `TridentBackendConfig` 로 설정하여 `retain` 이러한 백엔드를 완전히 관리하도록 선택할 수도 `tridentctl` 있습니다.

단계 **0:** 백엔드를 식별합니다

예를 들어 다음 백엔드가 을 사용하여 생성된 것으로 `TridentBackendConfig` 가정합니다.

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

출력에서 이 성공적으로 생성되었으며 백엔드에 바인딩되어 있는 것으로 `TridentBackendConfig` 보입니다 [백엔드의 UUID 관찰].

1단계: 확인이 `deletionPolicy` 로 설정되어 있습니다 `retain`

의 가치를 `deletionPolicy` 살펴보겠습니다. 이 설정은 로 `retain`**설정해야 합니다. 이렇게 하면** `CR`**이 삭제되어도** `TridentBackendConfig` 백엔드 정의가 계속 존재하며 로 관리할 수 `tridentctl` 있습니다.

```
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
```

가 로 설정되지 `retain` 않은 경우 다음 단계를 진행하지 마십시오. `deletionPolicy`

2단계: `TridentBackendConfig` **CR**을 삭제합니다

마지막 단계는 CR을 삭제하는 `TridentBackendConfig` 것입니다. 가 로 설정되어 `retain` 있는지 확인한 후 `deletionPolicy` 다음 작업을 계속 진행할 수 있습니다.

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

오브젝트가 삭제되면 `TridentBackendConfig` Trident은 백엔드 자체를 실제로 삭제하지 않고 기존 오브젝트를 단순히 제거합니다.

저작권 정보

Copyright © 2025 NetApp, Inc. All Rights Reserved. 미국에서 인쇄된 본 문서의 어떠한 부분도 저작권 소유자의 사전 서면 승인 없이는 어떠한 형식이나 수단(복사, 녹음, 녹화 또는 전자 검색 시스템에 저장하는 것을 비롯한 그래픽, 전자적 또는 기계적 방법)으로도 복제될 수 없습니다.

NetApp이 저작권을 가진 자료에 있는 소프트웨어에는 아래의 라이센스와 고지사항이 적용됩니다.

본 소프트웨어는 NetApp에 의해 '있는 그대로' 제공되며 상품성 및 특정 목적에의 적합성에 대한 명시적 또는 묵시적 보증을 포함하여(이에 제한되지 않음) 어떠한 보증도 하지 않습니다. NetApp은 대체품 또는 대체 서비스의 조달, 사용 불능, 데이터 손실, 이익 손실, 영업 중단을 포함하여(이에 국한되지 않음), 이 소프트웨어의 사용으로 인해 발생하는 모든 직접 및 간접 손해, 우발적 손해, 특별 손해, 징벌적 손해, 결과적 손해의 발생에 대하여 그 발생 이유, 책임론, 계약 여부, 엄격한 책임, 불법 행위(과실 또는 그렇지 않은 경우)와 관계없이 어떠한 책임도 지지 않으며, 이와 같은 손실의 발생 가능성이 통지되었다 하더라도 마찬가지입니다.

NetApp은 본 문서에 설명된 제품을 언제든지 예고 없이 변경할 권리를 보유합니다. NetApp은 NetApp의 명시적인 서면 동의를 받은 경우를 제외하고 본 문서에 설명된 제품을 사용하여 발생하는 어떠한 문제에도 책임을 지지 않습니다. 본 제품의 사용 또는 구매의 경우 NetApp에서는 어떠한 특허권, 상표권 또는 기타 지적 재산권이 적용되는 라이센스도 제공하지 않습니다.

본 설명서에 설명된 제품은 하나 이상의 미국 특허, 해외 특허 또는 출원 중인 특허로 보호됩니다.

제한적 권리 표시: 정부에 의한 사용, 복제 또는 공개에는 DFARS 252.227-7013(2014년 2월) 및 FAR 52.227-19(2007년 12월)의 기술 데이터-비상업적 품목에 대한 권리(Rights in Technical Data -Noncommercial Items) 조항의 하위 조항 (b)(3)에 설명된 제한사항이 적용됩니다.

여기에 포함된 데이터는 상업용 제품 및/또는 상업용 서비스(FAR 2.101에 정의)에 해당하며 NetApp, Inc.의 독점 자산입니다. 본 계약에 따라 제공되는 모든 NetApp 기술 데이터 및 컴퓨터 소프트웨어는 본질적으로 상업용이며 개인 비용만으로 개발되었습니다. 미국 정부는 데이터가 제공된 미국 계약과 관련하여 해당 계약을 지원하는 데에만 데이터에 대한 전 세계적으로 비독점적이고 양도할 수 없으며 재사용이 불가능하며 취소 불가능한 라이센스를 제한적으로 가집니다. 여기에 제공된 경우를 제외하고 NetApp, Inc.의 사전 서면 승인 없이는 이 데이터를 사용, 공개, 재생산, 수정, 수행 또는 표시할 수 없습니다. 미국 국방부에 대한 정부 라이센스는 DFARS 조항 252.227-7015(b)(2014년 2월)에 명시된 권한으로 제한됩니다.

상표 정보

NETAPP, NETAPP 로고 및 http://www.netapp.com/TM에 나열된 마크는 NetApp, Inc.의 상표입니다. 기타 회사 및 제품 이름은 해당 소유자의 상표일 수 있습니다.