管理後端

Astra Trident

NetApp
April 18, 2024

目錄

管理後端

以KECBECVL執行後端管理

瞭解如何使用「kubecll」來執行後端管理作業。

刪除後端

刪除「TridentBackendConfig」（TridentBackendConfig）之後、即指示Astra Trident刪除/保留後端（根據「刪除原則」）。若要刪除後端、請確定「刪除原則」已設定為刪除。如果只要刪除「TridentBackendConfig」、請確定「刪除原則」已設定為保留。這可確保後端仍存在、並可使用「tridentctl」進行管理。

執行下列命令：

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

Astra Trident並不會刪除「TridentBackendConfig」所使用的Kubernetes Secrets。Kubernetes使用者負責清除機密。刪除機密時必須小心。只有在後端未使用機密時、才應刪除這些機密。

檢視現有的後端

執行下列命令：

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

您也可以執行「tridentctl Get backend -n trident」或「tridentctl Get backend -o yaml -n trident」、以取得所有後端的清單。這份清單也會包含以「tridentctl」建立的後端。

更新後端

更新後端可能有多種原因：

- 儲存系統的認證資料已變更。若要更新認證資料、必須更新「TridentBackendConfig」物件中使用的Kubernetes Secret。Astra Trident會自動以提供的最新認證資料更新後端。執行下列命令以更新Kubernetes Secret：

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

- 需要ONTAP 更新參數（例如使用的SVM名稱）。
  - 您可以更新 `TridentBackendConfig` 使用下列命令直接透過 Kubernetes 執行物件：

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

- 或者、您也可以變更現有的 `TridentBackendConfig` 使用下列命令的 CR ：

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

> - 如果後端更新失敗、後端仍會繼續維持其最後已知的組態。您可以執行「kubecl Get tbc <tbc-name>-o yaml -n trident」或「kubecl描述tbc <tbc-name>-n trident」來檢視記錄以判斷原因。
> - 識別並修正組態檔的問題之後、即可重新執行update命令。

使用tridentctl執行後端管理

瞭解如何使用「tridentctl」來執行後端管理作業。

建立後端

建立之後 "後端組態檔"，執行下列命令：

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

如果後端建立失敗、表示後端組態有問題。您可以執行下列命令來檢視記錄、以判斷原因：

```
tridentctl logs -n trident
```

識別並修正組態檔的問題之後、您只需再次執行「create」命令即可。

刪除後端

若要從Astra Trident刪除後端、請執行下列步驟：

1. 擷取後端名稱：

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

2. 刪除後端：

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

> 如果Astra Trident已從這個後端配置磁碟區和快照、但該後端仍存在、則刪除後端會使新的磁碟區無法由其進行資源配置。後端將繼續處於「刪除」狀態、而Trident將繼續管理這些磁碟區和快照、直到它們被刪除為止。

檢視現有的後端

若要檢視Trident知道的後端、請執行下列步驟：

- 若要取得摘要、請執行下列命令：

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

- 若要取得所有詳細資料、請執行下列命令：

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

更新後端

建立新的後端組態檔之後、請執行下列命令：

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

如果後端更新失敗、表示後端組態有問題、或是您嘗試了無效的更新。您可以執行下列命令來檢視記錄、以判斷原因：

```
tridentctl logs -n trident
```

識別並修正組態檔的問題之後、您只需再次執行「update」命令即可。

識別使用後端的儲存類別

這是您可以用Json回答的問題類型範例、其中的「tridentctl」會輸出後端物件。這會使用您需要安裝的「jq」公用程式。

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

這也適用於使用「TridentBackendConfig」建立的後端。

在後端管理選項之間切換

瞭解Astra Trident管理後端的不同方法。

管理後端的選項

隨之推出 `TridentBackendConfig`管理員現在有兩種獨特的後端管理方法。這會提出下列問題：

- 使用「tridentctl」建立的後端、是否能以「TridentBackendConfig」來管理？
- 使用「TridentBackendConfig」建立的後端、是否可以使用「tridentctl」來管理？

管理 `tridentctl` 後端使用 `TridentBackendConfig`

本節說明透過Kubernetes介面建立「TridentBackendConfig」物件、直接透過「tridentctl」建立的後端管理所需的步驟。

這將適用於下列案例：

- 沒有的既有後端 `TridentBackendConfig` 因為它們是使用建立的 `tridentctl`。
- 使用「tridentctl」建立的新後端、而其他「TridentBackendConfig」物件則存在。

在這兩種情況下、後端仍會繼續存在、Astra Trident排程磁碟區會繼續運作。系統管理員有兩種選擇之一：

- 繼續使用「tridentctl」來管理使用它建立的後端。
- 將使用「tridentctl」建立的後端連結至新的「TridentBackendConfig」物件。這樣做將意味着後端將使用“kubedl”而不是“tridentctl”來管理。

若要使用「kubecll」管理預先存在的後端、您需要建立連結至現有後端的「TridentBackendConfig」。以下是如何運作的總覽：

1. 建立Kubernetes機密。此機密包含Astra Trident與儲存叢集/服務通訊所需的認證資料。
2. 建立「TridentBackendConfig」物件。其中包含有關儲存叢集/服務的詳細資訊、並參考上一步建立的機密。必須謹慎指定相同的組態參數（例如「s.pec.backendName」、「sec.storagePrefix」、「sPEec.storageDriverName」等）。必須將「Pec.backendName」設定為現有後端的名稱。

步驟**0**：識別後端

以建立 `TridentBackendConfig` 若要連結至現有的後端、您必須取得後端組態。在此範例中、假設使用下列Json定義建立後端：

```
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"
            }
        }
    ]
}
```

步驟1：建立Kubernetes機密

建立包含後端認證的秘密、如以下範例所示：

```
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
```

步驟2：建立 `TridentBackendConfig` CR

下一步是建立一個「TridentBackendConfig」（TridentBackendConfig）CR、它會自動連結至現有的「ONTAP-NAS-backend」（如本範例所示）。確保符合下列要求：

- 相同的後端名稱是在「s.pec.backendName」中定義。
- 組態參數與原始後端相同。
- 虛擬資源池（若有）必須維持與原始後端相同的順序。
- 認證資料是透過Kubernetes Secret提供、而非以純文字提供。

在這種情況下、「TridentBackendConfig」將會如下所示：

```
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
```

步驟3：確認的狀態 `TridentBackendConfig` CR

在建立「TridentBackendConfig」之後、其階段必須是「綁定」。它也應反映與現有後端相同的後端名稱和UUID。

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

現在可以使用「tbc-ontap-nas-backend」「TridentBackendConfig」物件來完全管理後端。

管理 `TridentBackendConfig` 後端使用 `tridentctl`

可以使用「tridentctl」來列出使用「TridentBackendConfig」建立的後端。此外、系統管理員也可以刪除「TridentBackendConfig」、並確定「pec.deletionPolicy`」設為「效能」、藉此選擇透過「tridentctl」來完全管理此類後端。

步驟0：識別後端

例如、假設使用「TridentBackendConfig」建立下列後端：

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

從輸出中可以看出這一點 `TridentBackendConfig` 已成功建立並繫結至後端 [觀察後端的 UUID] 。

步驟1：確認 `deletionPolicy` **設為** `retain`

讓我們來看看「改革政策」的價值。這需要設定為「維護」。這將確保刪除「TridentBackendConfig」（TridentBackendConfig）的CR時、後端定義仍會存在、而且可以使用「tridentctl」進行管理。

```
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
```

除非將「刪除原則」設定為「需要」、否則請勿繼續下一步。

步驟2：刪除 `TridentBackendConfig` CR

最後一個步驟是刪除「TridentBackendConfig」（TridentBackendConfig）。確認「刪除原則」設為「保留」之後、您可以繼續刪除：

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

刪除「TridentBackendConfig」物件之後、Astra Trident便會移除該物件、而不會實際刪除後端本身。