管理與監控 Trident

Trident

NetApp
July 01, 2026

目錄

管理與監控 Trident

升級 Trident

升級 Trident

從 24.02 版本開始，Trident 遵循四個月的發布節奏，每年發布三個主要版本。每個新版本都基於先前的版本，並提供新功能、效能提升、錯誤修復和改進。我們建議您至少每年升級一次，以充分利用 Trident 的新功能。

升級前的注意事項

升級至最新版 Trident 時，請注意以下事項：

- 在給定的 Kubernetes 叢集中，所有命名空間應該只安裝一個 Trident 執行個體。
- Trident 23.07 及更高版本需要 v1 Volume Snapshot，不再支援 alpha 或 beta Snapshot。
- 升級時，請務必提供 `parameter.fsType` 在 `StorageClasses` 中由 Trident 使用。您可以刪除並重新建立 `StorageClasses` 而不會影響現有磁碟區。
  - 這是對 SAN 磁碟區進行 "安全情境"強制執行的要求。
  - 範例輸入 目錄包含範例，例如 `storage-class-basic.yaml.templ` 和 `storage-class-bronze-default.yaml`。
  - 如需更多資訊，請參閱 "已知問題"。

步驟 1：選擇版本

Trident 版本遵循基於日期的 `YY.MM`命名規則，其中「YY」代表年份的後兩位數字，「MM」代表月份。小版本更新遵循 `YY.MM.X`規則，其中「X」代表補丁級別。您將根據要升級的版本選擇要升級到的版本。

- 您可以將目前版本直接升級到與其相差不超過四個版本號的目標版本。例如，您可以直接從 24.06（或任何 24.06 的小版本）升級到 25.06。
- 如果您要從超出四版本視窗期的版本升級，請執行多步驟升級。使用您要升級的 "早期版本" 版本對應的升級說明，升級到符合四版本視窗期的最新版本。例如，如果您目前運行的是 23.07 版本，並且想要升級到 25.06 版本：
  a. 首次從 23.07 升級到 24.06。
  b. 然後從 24.06 升級到 25.06 。

> 在 OpenShift Container Platform 上使用 Trident Operator 進行升級時,應升級至 Trident 21.01.1 或更高版本。21.01.0 版本發布的 Trident Operator 包含一個已知問題,該問題已在 21.01.1 版本中修復。如需更多詳細資訊,請參閱 "GitHub 上的問題詳情"。

步驟 2：確定原始安裝方法

若要確定您最初用於安裝 Trident 的版本：

1. 使用 `kubectl get pods -n trident`檢查 Pod。
    - 如果沒有操作員 pod，Trident 是使用 `tridentctl`安裝的。
    - 如果存在 operator pod，則 Trident 是透過 Trident operator 手動安裝的，或使用 Helm 安裝的。
2. 如果有操作員 pod，請使用 `kubectl describe torc`來判斷 Trident 是否使用 Helm 安裝。
    - 如果有 Helm 標籤，Trident 是使用 Helm 安裝的。
    - 如果沒有 Helm 標籤、則 Trident 是使用 Trident 操作員手動安裝。

步驟 **3**：選擇升級方法

通常情況下，您應該使用與初始安裝相同的方法進行升級，但您也可以"在安裝方法之間移動"。有兩種升級 Trident 的選項。

- "使用 Trident 操作員進行升級"

我們建議您在使用 operator 升級之前先檢閱"了解 operator 升級工作流程"。

*

使用 **operator** 進行升級

了解 **operator** 升級工作流程

在使用 Trident operator 升級 Trident 之前，您應該先了解升級過程中發生的背景程序。這包括對 Trident 控制器、控制器 Pod 和節點 Pod 的變更，以及啟用滾動更新的節點 DaemonSet。

Trident 操作員升級處理

安裝和升級 Trident 的眾多"使用 Trident 運算子的好處"之一是能夠自動處理 Trident 和 Kubernetes 物件，而不會中斷現有掛載的磁碟區。這樣，Trident 可以支援零停機升級，或"滾動更新"。具體而言，Trident operator 與 Kubernetes 叢集通訊以：

- 刪除並重新建立 Trident Controller 部署和節點 DaemonSet。
- 將 Trident Controller Pod 和 Trident Node Pod 替換為新版本。
    - 如果某個節點沒有更新，並不妨礙其他節點的更新。
    - 只有運行了 Trident Node Pod 的節點才能掛載磁碟區。

有關 Kubernetes 叢集上 Trident 架構的更多資訊，請參閱 "Trident 架構"。

Operator 升級工作流程

當您使用 Trident 運算子啟動升級時：

1. **Trident** 運算子：
    a. 偵測目前安裝的 Trident 版本（版本 *n*）。

b. 更新所有 Kubernetes 物件，包括 CRD、RBAC 和 Trident SVC。

c. 刪除版本 *n* 的 Trident Controller 部署。

d. 建立版本 *n+1* 的 Trident Controller 部署。

2. **Kubernetes** 為 *n+1* 建立 Trident Controller Pod。

3. **Trident** 運算子：

a. 刪除 *n* 的 Trident Node DaemonSet。該操作符不會等待 Node Pod 終止。

b. 為 *n+1* 建立 Trident 節點守護程序集。

4. **Kubernetes** 會在未執行 Trident Node Pod *n* 的節點上建立 Trident Node Pod。這可確保每個節點上永遠不會存在多個 Trident Node Pod，無論版本為何。

使用 **Trident Operator** 或 **Helm** 升級 **Trident** 安裝

您可以使用 Trident Operator 手動或透過 Helm 升級 Trident。您可以從一個 Trident Operator 安裝升級到另一個 Trident Operator 安裝，也可以從 `tridentctl`安裝升級到 Trident Operator 版本。在升級 Trident Operator 安裝之前，請先檢閱"選擇升級方法"。

升級手動安裝

您可以將叢集範圍的 Trident 操作員安裝升級到另一個叢集範圍的 Trident 操作員安裝。所有 Trident 版本都使用叢集範圍的操作員。

若要從使用命名空間範圍運算子安裝的 Trident （版本 20.07 至 20.10）進行升級，請使用 "您已安裝的版本" 的 Trident 升級說明。

關於此任務

Trident 提供了一個捆綁文件，您可以使用該文件安裝 Operator 並為您的 Kubernetes 版本建立關聯物件。

- 對於運行 Kubernetes 1.25 或更高版本的集群，請使用 "bundle_post_1_25.yaml"。

開始之前

請確保您使用的是正在運行 "支援的 Kubernetes 版本" 的 Kubernetes 叢集。

步驟

1. 驗證您的 Trident 版本：

```
./tridentctl -n trident version
```

2. 使用要升級到的版本（例如 25.06）的登錄和映像路徑以及正確的金鑰更新 `operator.yaml`、`tridentorchestrator_cr.yaml`和 `post_1_25_bundle.yaml`。

3. 刪除用於安裝目前 Trident 實例的 Trident 操作符。例如，如果您是從 25.02 版本升級，請執行以下命令：

```
kubectl delete -f 25.02.0/trident-installer/deploy/<bundle.yaml> -n
trident
```

4. 如果您使用 `TridentOrchestrator`屬性自訂了初始安裝，則可以編輯該 `TridentOrchestrator`物件來修改安裝參數。這可能包括為離線模式指定鏡像 Trident 和 CSI 映像登錄、啟用偵錯日誌或指定映像拉取金鑰等變更。
5. 使用適用於您環境的正確 bundle YAML 檔案安裝 Trident，其中 *<bundle.yaml>* 是 `bundle_pre_1_25.yaml`或 `bundle_post_1_25.yaml`，取決於您的 Kubernetes 版本。例如，如果您要安裝 Trident 25.06.0，請執行下列命令：

```
kubectl create -f 25.06.0/trident-installer/deploy/<bundle.yaml> -n
trident
```

6. 編輯 Trident torc，使其包含映像 25.06.0。

升級 **Helm** 安裝

您可以升級 Trident Helm 安裝。

> 當將安裝了 Trident 的 Kubernetes 叢集從 1.24 升級到 1.25 或更高版本時，必須先更新 values.yaml 將 `excludePodSecurityPolicy`設定為 `true`或新增 `--set excludePodSecurityPolicy=true`至 `helm upgrade`命令中，然後才能升級叢集。

如果您已將 Kubernetes 叢集從 1.24 升級到 1.25，但未升級 Trident helm，則 helm 升級將會失敗。若要成功完成 helm 升級，請先執行以下步驟：

1. 從 https://github.com/helm/helm-mapkubeapis安裝 helm-mapkubeapis 外掛程式。
2. 在安裝 Trident 的命名空間中對 Trident 版本執行預運行。這將列出要清理的資源。

```
helm mapkubeapis --dry-run trident  --namespace trident
```

3. 使用 Helm 執行完整運行以進行清理。

```
helm mapkubeapis trident  --namespace trident
```

步驟

1. 如果您"使用 Helm 安裝了 Trident"，可以使用 `helm upgrade trident netapp-trident/trident-operator --version 100.2602.0`一步升級。如果您沒有新增 Helm 倉庫或無法使用它進行升級：
   a. 從 "GitHub 上的_資產_部分" 下載最新版的 Trident。
   b. 使用 `helm upgrade` 命令，其中 `trident-operator-26.02.0.tgz` 反映您要升級到的版本。

```
helm upgrade <name> trident-operator-26.02.0.tgz
```

如果您在初始安裝期間設定了自訂選項（例如為 Trident 和 CSI 映像指定私有、鏡像註冊表），請在 `helm upgrade` 指令後附加 `--set`，以確保這些選項包含在升級指令中，否則這些值將會重設為預設值。

2. 運行 `helm list` 以驗證圖表和應用程式版本是否均已升級。運行 `tridentctl logs` 以查看任何調試訊息。

從 `tridentctl` 安裝程式升級到 **Trident operator**

您可以從 `tridentctl`安裝升級到最新版本的 Trident 操作器。現有的後端和 PVC 將自動可用。

在切換安裝方法之前，請先查看 "在安裝方法之間移動"。

步驟

1. 下載最新的 Trident 版本。

```
# Download the release required [26.02.0]
mkdir 26.02.0
cd 26.02.0
wget
https://github.com/NetApp/trident/releases/download/v26.02.0/trident-
installer-26.02.0.tar.gz
tar -xf trident-installer-26.02.0.tar.gz
cd trident-installer
```

2. 根據清單檔案建立 `tridentorchestrator` CRD。

```
kubectl create -f
deploy/crds/trident.netapp.io_tridentorchestrators_crd_post1.16.yaml
```

3. 在相同命名空間中部署叢集範圍的 operator。

```
kubectl create -f deploy/<bundle-name.yaml>

serviceaccount/trident-operator created
clusterrole.rbac.authorization.k8s.io/trident-operator created
clusterrolebinding.rbac.authorization.k8s.io/trident-operator created
deployment.apps/trident-operator created
podsecuritypolicy.policy/tridentoperatorpods created

#Examine the pods in the Trident namespace
NAME                                  READY   STATUS    RESTARTS   AGE
trident-controller-79df798bdc-m79dc   6/6     Running   0          150d
trident-node-linux-xrst8              2/2     Running   0          150d
trident-operator-5574dbbc68-nthjv     1/1     Running   0          1m30s
```

4. 建立 `TridentOrchestrator` CR 以安裝 Trident。

```
cat deploy/crds/tridentorchestrator_cr.yaml
apiVersion: trident.netapp.io/v1
kind: TridentOrchestrator
metadata:
  name: trident
spec:
  debug: true
  namespace: trident

kubectl create -f deploy/crds/tridentorchestrator_cr.yaml

#Examine the pods in the Trident namespace
NAME                                READY   STATUS    RESTARTS   AGE
trident-csi-79df798bdc-m79dc        6/6     Running   0          1m
trident-csi-xrst8                   2/2     Running   0          1m
trident-operator-5574dbbc68-nthjv   1/1     Running   0          5m41s
```

5. 確認 Trident 已升級至預期版本。

```
kubectl describe torc trident | grep Message -A 3

Message:                Trident installed
Namespace:              trident
Status:                 Installed
Version:                v26.02.0
```

使用 **tridentctl** 進行升級

您可以使用 `tridentctl`輕鬆升級現有的 Trident 安裝。

關於此任務

解除安裝並重新安裝 Trident 相當於一次升級。卸載 Trident 時，Trident 部署使用的持久卷聲明（PVC）和持久卷（PV）不會被刪除。已配置的 PV 在 Trident 離線期間仍可用，Trident 將在恢復在線後為在此期間創建的任何 PVC 配置卷。

開始之前

在使用 `tridentctl`進行升級之前，請先檢查"選擇升級方法"。

步驟

1. 在 `tridentctl`中執行卸載指令，以移除與 Trident 相關的所有資源，但 CRD 和相關物件除外。

```
./tridentctl uninstall -n <namespace>
```

2. 重新安裝 Trident 。請參閱"使用 tridentctl 安裝 Trident"。

> 請勿中斷升級程序。確保安裝程式執行完成。

使用 **tridentctl** 管理 **Trident**

該軟體包 "Trident 安裝程式套件"包含 `tridentctl`命令列實用程序，方便用戶輕鬆存取 Trident。擁有足夠權限的 Kubernetes 使用者可以使用它來安裝 Trident 或管理包含 Trident Pod 的命名空間。

命令和全域標誌

您可以執行 `tridentctl help`以取得 `tridentctl`的可用命令清單，或將 `--help`旗標附加至任何命令，以取得該特定命令的選項和旗標清單。

```
tridentctl [command] [--optional-flag]
```

Trident `tridentctl` 工具支援以下指令和全域標誌。

命令

create
將資源新增至 Trident。

delete
從 Trident 移除一個或多個資源。

get
從 Trident 取得一或多個資源。

help
關於任何命令的說明。

images
列印一份 Trident 所需容器映像的表格。

import
將現有資源匯入 Trident。

install
安裝 Trident。

logs
列印 Trident 的日誌。

send
從 Trident 傳送資源。

uninstall
卸載 Trident。

update
在 Trident 中修改資源。

update backend state
暫時停止後端作業。

upgrade
在 Trident 中升級資源。

version
列印 Trident 的版本。

全域旗標

-d, --debug

偵錯輸出。

-h, --help

的說明 `tridentctl`。

-k, --kubeconfig string

指定 `KUBECONFIG` 路徑，以便在本機或從一個 Kubernetes 叢集到另一個 Kubernetes 叢集運行命令。

或者，您可以匯出 `KUBECONFIG` 變數以指向特定的 Kubernetes 叢集，並向該叢集發出 `tridentctl` 命令。

-n, --namespace string

Trident 部署的命名空間。

-o, --output string

輸出格式。json|yaml|name|wide|ps（預設）其中之一。

-s, --server string

Trident REST 介面的位址 / 連接埠。

Trident REST 介面可以設定為僅監聽和提供服務於 127.0.0.1 （適用於 IPv4）或 [::1] （適用於 IPv6）。

命令選項和標誌

建立

使用 `create` 指令將資源新增到 Trident。

```
tridentctl create [option]
```

選項

`backend`：將後端新增至 Trident。

刪除

使用 `delete` 指令從 Trident 移除一個或多個資源。

```
tridentctl delete [option]
```

選項

`backend`：從 Trident 刪除一個或多個儲存後端。
`snapshot`：從 Trident 刪除一個或多個 Volume 快照。

`storageclass`：從 Trident 刪除一個或多個儲存類別。
`volume`：從 Trident 刪除一個或多個儲存 Volume 。

取得

使用 `get` 指令從 Trident 取得一個或多個資源。

```
tridentctl get [option]
```

選項

`backend`：從 Trident 取得一個或多個儲存後端。
`snapshot`：從 Trident 取得一個或多個快照。
`storageclass`：從 Trident 取得一個或多個儲存類別。
`volume`：從 Trident 取得一個或多個磁碟區。

旗標

`-h, --help`：磁碟區的說明。
`--parentOfSubordinate string`：將查詢限制為從屬來源磁碟區。
`--subordinateOf string`：將查詢限制為磁碟區的從屬磁碟區。

映像

使用 `images` 標誌列印 Trident 所需的容器映像表格。

```
tridentctl images [flags]
```

旗標

`-h, --help`:圖像幫助。
`-v, --k8s-version string`:Kubernetes 叢集的語意版本。

匯入磁碟區

使用 `import volume` 指令將現有磁碟區導入 Trident。

```
tridentctl import volume <backendName> <volumeName> [flags]
```

別名

`volume, v`

旗標

`-f, --filename string`: YAML 或 JSON PVC 檔案的路徑。
`-h, --help`: 卷的幫助資訊。
`--no-manage`: 僅建立 PV/PVC。不進行磁碟區生命週期管理。

安裝

使用 `install` 標誌安裝 Trident。

```
tridentctl install [flags]
```

旗標

`--autosupport-image string`：自動支援遙測的容器映像（預設值為「netapp/trident autosupport:<current-version>」）。
`--autosupport-proxy string`：用於發送自動支援遙測的代理程式的位址/連接埠。
`--enable-node-prep`：嘗試在節點上安裝所需的軟體包。
`--generate-custom-yaml`：產生 YAML 檔案而不進行任何安裝。
`-h, --help`：安裝幫助。
`--http-request-timeout`：覆蓋 Trident 控制器 REST API 的 HTTP 請求逾時時間（預設值為 1m30s）。
`--image-registry string`：內部鏡像倉庫的位址/連接埠。
`--k8s-timeout duration`：所有 Kubernetes 操作的逾時時間（預設值為 3m0s）。
`--kubelet-dir string`：kubelet 內部狀態的主機位置（預設值為「/var/lib/kubelet」）。
`--log-format string`：Trident 日誌格式（text、json）（預設值為「text」）。
`--node-prep`：啟用 Trident 使 Kubernetes 叢集的節點能夠使用指定的資料儲存協定管理磁碟區。目前，**`iscsi`是唯一支援的值。從 `OpenShift 4.19` 版本開始，此功能支援的最低 `Trident` 版本為 `25.06.1`。**
`--pv string`：Trident 使用的舊版 PV 名稱，確保該 PV 不存在（預設值為「trident」）。
`--pvc string`：Trident 使用的舊版 PVC 名稱，確保該 PVC 不存在（預設值為「trident」）。
`--silence-autosupport`：不自動發送自動支援包至 NetApp（預設值為 true）。
`--silent`：安裝期間停用大部分輸出。
`--trident-image string`：要安裝的 Trident 鏡像。
`--k8s-api-qps`：Kubernetes API 請求的每秒查詢數（QPS）限制（預設值為 100；可選）。
`--use-custom-yaml`：使用 setup 目錄中存在的任何 YAML 檔案。
`--use-ipv6`：Trident 通訊使用 IPv6。

日誌

使用 `logs` 標誌列印 Trident 的日誌。

```
tridentctl logs [flags]
```

旗標

`-a, --archive`：除非另有指定，否則建立包含所有日誌的支援存檔。
`-h, --help`：日誌說明。
`-l, --log string`：要顯示的 Trident 日誌。可選值包含 trident|auto|trident-operator|all（預設值為 "auto"）。
`--node string`：要從中收集節點 Pod 日誌的 Kubernetes 節點名稱。
`-p, --previous`：如果存在，則取得上一個容器執行個體的日誌。
`--sidecars`：取得 Sidecar 容器的日誌。

傳送

使用 `send` 指令從 Trident 傳送資源。

```
tridentctl send [option]
```

選項

`autosupport`：將 Autosupport 歸檔傳送至 NetApp。

解除安裝

使用 `uninstall` 標誌卸載 Trident 。

```
tridentctl uninstall [flags]
```

旗標

`-h, --help`：卸載幫助。
`--silent`：卸載過程中停用大部分輸出。

更新

使用 `update` 指令修改 Trident 中的資源。

```
tridentctl update [option]
```

選項

`backend`：在 Trident 中更新後端。

更新後端狀態

使用 `update backend state` 命令可以暫停或恢復後端操作。

```
tridentctl update backend state <backend-name> [flag]
```

需要考慮的要點

- 如果後端是使用 TridentBackendConfig（tbc）建立的，則無法使用 `backend.json`檔案更新後端。
- 如果 `userState`已在 tbc 中設置，則無法使用 `tridentctl update backend state <backend-name> --user -state suspended/normal`命令對其進行修改。
- 若要恢復透過 tridentctl 設定 `userState` 的功能（此欄位已透過 tbc 設定），必須先從 tbc 移除 `userState` 欄位。這可以透過 `kubectl edit tbc` 命令完成。移除 `userState` 欄位後，您可以使用 `tridentctl update backend state` 命令變更後端的 `userState`。
- 使用 `tridentctl update backend state`來更改 `userState`。你也可以使用 `TridentBackendConfig`或 `backend.json`檔案來更新 `userState`；這會觸發後端的完整重新初始化，並且可能會花費較多時間。

  旗標

  `-h, --help`:後端狀態說明。
  `--user-state`:設定為 `suspended`可暫停後端操作。設定為 `normal`可恢復後端操作。設定為 `suspended`時:

- `AddVolume` 和 `Import Volume` 已暫停。
- `CloneVolume`, `ResizeVolume`, `PublishVolume`, `UnPublishVolume`, `CreateSnapshot`, `GetSnapshot`, `RestoreSnapshot`, `DeleteSnapshot`, `RemoveVolume`, `GetVolumeExternal`, `ReconcileNodeAccess` 仍然可用。

您也可以使用 `userState`欄位在後端組態檔 `TridentBackendConfig`或 `backend.json`中更新後端狀態。如需更多資訊，請參閱"管理後端的選項"和"使用 kubectl 執行後端管理"。

- 範例： *

JSON

請按照以下步驟，使用 `userState`檔案來更新 `backend.json`：

1. 編輯 `backend.json` 檔案以包含 `userState` 欄位，並將其值設為「suspended」。
2. 使用 `tridentctl update backend` 命令和更新的 `backend.json` 檔案路徑來更新後端。

   範例：`tridentctl update backend -f /<path to backend JSON file>/backend.json -n trident`

```
{
  "version": 1,
  "storageDriverName": "ontap-nas",
  "managementLIF": "<redacted>",
  "svm": "nas-svm",
  "backendName": "customBackend",
  "username": "<redacted>",
  "password": "<redacted>",
  "userState": "suspended"
}
```

YAML

您可以使用 `kubectl edit <tbc-name> -n <namespace>` 指令在套用 tbc 後對其進行編輯。以下範例使用 `userState: suspended` 選項將後端狀態更新為暫停：

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-ontap-nas
spec:
  version: 1
  backendName: customBackend
  storageDriverName: ontap-nas
  managementLIF: <redacted>
  svm: nas-svm
  userState: suspended
  credentials:
    name: backend-tbc-ontap-nas-secret
```

版本

使用 `version`標誌列印 `tridentctl`和正在執行的 Trident 服務的版本。

```
tridentctl version [flags]
```

旗標

`--client`：僅限客戶端版本（無需伺服器）。
`-h, --help`：版本說明。

外掛程式支援

Tridentctl 支援類似 kubectl 的插件。如果外掛程式二進位檔案名稱遵循「tridentctl-<plugin>」格式，且該二進位檔案位於 PATH 環境變數指定的資料夾中，Tridentctl 就會偵測到該外掛程式。所有偵測到的外掛程式都會列在 tridentctl help 的插件部分。此外，您也可以透過在環境變數 TRIDENTCTL_PLUGIN_PATH 中指定插件資料夾來縮小搜尋範圍（例如： `TRIDENTCTL_PLUGIN_PATH=~/tridentctl-plugins/`）。如果使用此變數，tridentctl 將僅在指定的資料夾中搜尋。

監控 Trident

Trident 提供了一組 Prometheus 指標端點、您可以使用這些端點來監控 Trident 效能。

概況

Trident 提供的指標可讓您執行以下操作：

- 密切注意 Trident 的運作狀況和組態。您可以檢查作業的成功程度，以及它是否能如預期般與後端通訊。
- 檢查後端使用資訊，了解後端配置了多少磁碟區、消耗的空間量等等。
- 維護可用後端已配置磁碟區數量的映射表。
- 追蹤效能。您可以查看 Trident 與後端通訊以及執行作業所需的時間。

> ⓘ 預設情況下， Trident 的指標會在目標連接埠 `8001`的 `/metrics`端點上公開。安裝 Trident 時，這些指標*預設為啟用*。您也可以設定為透過 HTTPS 在連接埠 `8444`上使用 Trident 指標。

您需要準備的項目

- 已安裝 Trident 的 Kubernetes 叢集。
- 一個 Prometheus 執行個體。這可以是 "容器化 Prometheus 部署"，或者您可以選擇將 Prometheus 作為 "原生應用程式"執行。

步驟 1：定義 Prometheus 目標

您應該定義一個 Prometheus 目標來收集指標並獲取有關 Trident 管理的後端、它創建的磁碟區等資訊。請參閱 "Prometheus Operator 說明文件"。

步驟 2 ：建立 Prometheus ServiceMonitor

要使用 Trident 指標，您應該建立一個 Prometheus ServiceMonitor 來監視 `trident-csi` 服務並監聽

metrics 連接埠。範例 ServiceMonitor 如下：

```
apiVersion: monitoring.coreos.com/v1
kind: ServiceMonitor
metadata:
  name: trident-sm
  namespace: monitoring
  labels:
    release: prom-operator
spec:
  jobLabel: trident
  selector:
    matchLabels:
      app: controller.csi.trident.netapp.io
  namespaceSelector:
    matchNames:
      - trident
  endpoints:
    - port: metrics
      interval: 15s
```

此 ServiceMonitor 定義會擷取 trident-csi 服務傳回的指標，並專門尋找服務的 metrics 端點。因此，Prometheus 現在已設定為瞭解 Trident 的指標。

除了直接從 Trident 取得的指標之外、 kubelet 還透過自己的指標端點公開了許多 `kubelet_volume_*`指標。Kubelet 可以提供有關已掛載磁碟區、 Pod 以及它處理的其他內部操作的資訊。請參閱 "這裡"。

透過 **HTTPS** 使用 **Trident** 指標

若要透過 HTTPS（連接埠 8444）使用 Trident 指標，您必須修改 ServiceMonitor 定義以包含 TLS 組態。您還需要將 trident-csi 密碼從 trident 命名空間複製到 Prometheus 執行的命名空間。您可以使用以下命令執行此操作：

```
kubectl get secret trident-csi -n trident -o yaml | sed 's/namespace:
trident/namespace: monitoring/' | kubectl apply -f -
```

HTTPS 指標的 ServiceMonitor 範例如下所示：

```
apiVersion: monitoring.coreos.com/v1
kind: ServiceMonitor
metadata:
  name: trident-sm
  namespace: monitoring
  labels:
    release: prom-operator
spec:
  jobLabel: trident
  selector:
    matchLabels:
      app: controller.csi.trident.netapp.io
  namespaceSelector:
    matchNames:
      - trident
  endpoints:
    - interval: 15s
      path: /metrics
      port: https-metrics
      scheme: https
      tlsConfig:
        ca:
          secret:
            key: caCert
            name: trident-csi
        cert:
          secret:
            key: clientCert
            name: trident-csi
        keySecret:
          key: clientKey
          name: trident-csi
        serverName: trident-csi
```

Trident 支援所有安裝方法中的 HTTPS 指標：tridentctl、Helm chart 和 Operator：

- 如果您使用 `tridentctl install` 指令，可以傳遞 `--https-metrics` 標誌來啟用 HTTPS 指標。
- 如果您使用的是 Helm chart，則可以設定 `httpsMetrics` 參數以啟用 HTTPS 指標。
- 如果您正在使用 YAML 檔案，您可以在 `trident-deployment.yaml` 檔案中的 `trident-main` container 新增 `--https_metrics` flag。

步驟 3：使用 PromQL 查詢 Trident 指標

PromQL 非常適合建立傳回時間序列或表格資料的運算式。

以下是您可以使用的 PromQL 查詢：

取得 **Trident** 健全狀況資訊

- **Trident HTTP 2XX** 回應的百分比

```
(sum (trident_rest_ops_seconds_total_count{status_code=~"2.."} OR on()
vector(0)) / sum (trident_rest_ops_seconds_total_count)) * 100
```

- 透過狀態碼從 **Trident** 取得的 **REST** 回應百分比

```
(sum (trident_rest_ops_seconds_total_count) by (status_code)  / scalar
(sum (trident_rest_ops_seconds_total_count))) * 100
```

- **Trident** 執行作業的平均持續時間（毫秒）

```
sum by (operation)
(trident_operation_duration_milliseconds_sum{success="true"}) / sum by
(operation)
(trident_operation_duration_milliseconds_count{success="true"})
```

獲取 **Trident** 使用資訊

- 平均磁碟區大小

```
trident_volume_allocated_bytes/trident_volume_count
```

- 各後端配置的總磁碟區空間

```
sum (trident_volume_allocated_bytes) by (backend_uuid)
```

取得個別磁碟區使用量

只有同時收集 kubelet 指標時，此功能才會啟用。

- 每個磁碟區已使用空間的百分比

```
kubelet_volume_stats_used_bytes / kubelet_volume_stats_capacity_bytes *
100
```

了解 Trident AutoSupport 遙測技術

預設情況下， Trident 每天向 NetApp 發送 Prometheus 指標和基本後端資訊。

- 若要阻止 Trident 向 NetApp 發送 Prometheus 指標和基本後端訊息，請在 Trident 安裝期間傳遞 `--silence-autosupport` 標誌。
- Trident 也可以透過 `tridentctl send autosupport`按需向 NetApp 支援部門發送容器日誌。您需要觸發 Trident 上傳日誌。提交日誌之前，您應該接受 NetApp 的https://www.netapp.com/company/legal/privacy-policy/["隱私權政策"^]。
- 除非另有說明，Trident 會取得過去 24 小時的日誌。
- 您可以使用 `--since`旗標指定日誌保留時間範圍。例如： `tridentctl send autosupport --since=1h`此資訊會透過與 Trident 一起安裝的 `trident-autosupport`容器收集並傳送。您可以在 "Trident AutoSupport"取得容器映像。
- Trident AutoSupport 不會收集或傳輸個人識別資訊（PII）或個人資訊。它附帶 "最終用戶授權協議"不適用於 Trident 容器映像本身。您可以深入瞭解 NetApp 對資料安全性和信任的承諾 "這裡"。

Trident 傳送的承載範例如下所示：

```
---
items:
  - backendUUID: ff3852e1-18a5-4df4-b2d3-f59f829627ed
    protocol: file
    config:
      version: 1
      storageDriverName: ontap-nas
      debug: false
      debugTraceFlags: null
      disableDelete: false
      serialNumbers:
        - nwkvzfanek_SN
      limitVolumeSize: ""
    state: online
    online: true
```

- AutoSupport 訊息會傳送到 NetApp 的 AutoSupport 端點。如果您使用私有登錄來儲存容器映像，可以使用 `--image-registry` 旗標。
- 您也可以透過產生安裝 YAML 檔案來設定 proxy URL。這可以使用 `tridentctl install --generate-custom-yaml`來建立 `YAML` 檔案，並為 `trident-autosupport` container 在 `trident-deployment.yaml`中新增 `--proxy-url`參數來完成。

停用 Trident 指標

若要停用指標回報，您應該產生自訂 YAML（使用 `--generate-custom-yaml` 旗標），並編輯它們以移除 `--metrics` 旗標，使其不會在 `trident-main` 容器中被呼叫。

解除安裝 Trident

您應該使用與安裝 Trident 相同的方法來卸載 Trident。

關於此任務

- 如果升級後出現錯誤、依賴項問題或升級失敗 / 不完整，需要修復，則應卸載 Trident 並按照相應的說明重新安裝早期版本 "版本"。這是_降級_到早期版本的唯一推薦方法。
- 為了方便升級和重新安裝，解除安裝 Trident 不會移除 Trident 所建立的 CRD 或相關物件。如果您需要徹底移除 Trident 及其所有資料，請參閱 "徹底移除 Trident 和 CRDs"。

開始之前

如果您要停用 Kubernetes 集群，則必須在卸載之前刪除所有使用 Trident 建立的磁碟區的應用程式。這可以確保在刪除 Kubernetes 節點之前，PVC 已從這些節點上取消發布。

確定原始安裝方法

您應該使用與安裝 Trident 時相同的方法來卸載 Trident。解除安裝前，請確認您最初用於安裝 Trident 的版本。

1. 使用 `kubectl get pods -n trident`檢查 Pod。
   - 如果沒有操作員 pod，Trident 是使用 `tridentctl`安裝的。
   - 如果存在 operator pod，則 Trident 是透過 Trident operator 手動安裝的，或使用 Helm 安裝的。
2. 如果有操作員 pod，請使用 `kubectl describe tproc trident`來判斷 Trident 是否使用 Helm 安裝。
   - 如果有 Helm 標籤，Trident 是使用 Helm 安裝的。
   - 如果沒有 Helm 標籤、則 Trident 是使用 Trident 操作員手動安裝。

卸載 Trident Operator 安裝

您可以手動解除安裝 Trident Operator 安裝，也可以使用 Helm 解除安裝。

解除安裝手動安裝

如果您使用操作員安裝了 Trident，則可以透過執行下列其中一項操作來解除安裝它：

1. 編輯 **`TridentOrchestrator` CR** 並設定卸載標誌：

   ```
   kubectl patch torc <trident-orchestrator-name> --type=merge -p
   '{"spec":{"uninstall":true}}'
   ```

   當 `uninstall`標誌設為 `true`時，Trident 運算元會卸載 Trident，但不會刪除 TridentOrchestrator 本身。如果您想再次安裝 Trident，則需要清理 TridentOrchestrator 並建立新的設定檔。

2. 刪除 **`TridentOrchestrator`**：透過移除 `TridentOrchestrator` 用於部署 Trident 的 CR，您可以指示操作員卸載 Trident。操作員將處理 `TridentOrchestrator` 的移除，並繼續移除 Trident 部署和守護程序集，同時刪除其在安裝過程中建立的 Trident Pod。

```
kubectl delete -f deploy/<bundle.yaml> -n <namespace>
```

解除安裝 **Helm**

如果您使用 Helm 安裝了 Trident，則可以使用 `helm uninstall`卸載它。

```
#List the Helm release corresponding to the Trident install.
helm ls -n trident
NAME          NAMESPACE       REVISION        UPDATED
STATUS          CHART                           APP VERSION
trident       trident         1               2021-04-20
00:26:42.417764794 +0000 UTC deployed        trident-operator-21.07.1
21.07.1

#Uninstall Helm release to remove Trident
helm uninstall trident -n trident
release "trident" uninstalled
```

解除安裝 `tridentctl`

使用 `uninstall` 命令在 `tridentctl` 中刪除與 Trident 相關的所有資源，但 CRD 和相關物件除外：

```
./tridentctl uninstall -n <namespace>
```

版權資訊

商標資訊