參考資料
Trident

NetApp
March 05, 2026

目錄

參考資料

Trident 連接埠

深入瞭解 Trident 用於通訊的連接埠。

Trident 連接埠

Trident 透過下列連接埠進行通訊：

連接埠	目的
8443	後端通道HTTPS
8001	Prometheus指標端點
8000	Trident REST伺服器
17546	Trident取消安裝套件所使用的活動/整備度探針連接埠

在安裝期間、可使用旗標變更活性 / 整備性探查連接埠 `--probe-port`。請務必確認工作節點上的其他程序並未使用此連接埠。

Trident REST API

雖然是與 Trident REST API 互動最簡單的方法、但"tridentctl命令和選項"您可以視需要直接使用其餘端點。

何時使用REST API

REST API 適用於在非 Kubernetes 部署中使用 Trident 做為獨立二進位檔的進階安裝。

為了獲得更好的安全性、在 Pod 內執行時、 Trident `REST API` 預設會限制為 localhost 。若要變更此行為、您需要在其 Pod 組態中設定 Trident 的 `-address`引數。

使用REST API

有關如何調用這些 API 的示例，請傳遞 debug (`-d`標誌。如需詳細資訊、請 "使用 tridentctl 管理 Trident"參閱。

API的運作方式如下：

取得

`GET <trident-address>/trident/v1/<object-type>`

列出該類型的所有物件。

`GET <trident-address>/trident/v1/<object-type>/<object-name>`

取得命名物件的詳細資料。

貼文

POST <trident-address>/trident/v1/<object-type>

建立指定類型的物件。

- 需要Json組態才能建立物件。有關每種物件類型的規格、請"使用 tridentctl 管理 Trident"參閱。
- 如果物件已經存在、行為會有所不同：後端會更新現有物件、而其他所有物件類型都會使作業失敗。

刪除

DELETE <trident-address>/trident/v1/<object-type>/<object-name>

刪除命名資源。

與後端或儲存類別相關聯的磁碟區將繼續存在、必須分別刪除。如需詳細資訊、請 "使用 tridentctl 管理 Trident"參閱。

命令列選項

Trident 為 Trident Orchestrator 提供數個命令列選項。您可以使用這些選項來修改部署。

記錄

-debug

啟用除錯輸出。

-loglevel <level>

設定記錄層級（偵錯、資訊、警告、錯誤、嚴重）。預設為資訊。

Kubernetes

-k8s_pod

使用此選項或 `-k8s_api_server`啟用 Kubernetes 支援。設定此選項會使Trident使用內含Pod的Kubernetes服務帳戶認證、來聯絡API伺服器。這只有當Trident在Kubernetes叢集中以Pod形式執行、且已啟用服務帳戶時才會運作。

-k8s_api_server <insecure-address:insecure-port>

使用此選項或 `-k8s_pod`啟用 Kubernetes 支援。如果指定、Trident會使用提供的不安全位址和連接埠、連線至Kubernetes API伺服器。如此一來、 Trident 就能部署在 Pod 之外、但它只支援與 API 伺服器的不安全連線。若要安全連線、請在具有選項的 Pod 中部署 Trident `-k8s_pod 。

Docker

-volume_driver <name>

登錄 Docker 外掛程式時使用的驅動程式名稱。預設為 netapp。

-driver_port <port-number>

聆聽此連接埠、而非 UNIX 網域通訊端。

-config <file>

必要；您必須指定後端組態檔案的路徑。

休息

-address <ip-or-host>

指定 Trident 的 REST 伺服器應接聽的位址。預設為localhost。當偵聽localhost並在Kubernetes Pod內部執行時、無法從Pod外部直接存取REST介面。用於 `-address ""`讓 REST 介面可從 Pod IP 位址存取。

> ⚠ Trident REST介面可設定為偵聽、僅適用於127.0.0.1（適用於IPV4）或[:1]（適用於IPv6）。

-port <port-number>

指定 Trident 的 REST 伺服器應接聽的連接埠。預設為8000.

-rest

啟用 REST 介面。預設為true。

Kubernetes和Trident物件

您可以透過讀取和寫入資源物件、使用REST API與Kubernetes和Trident互動。Kubernetes與Trident、Trident與Storage、Kubernetes與儲存設備之間有幾個資源物件、分別是它們之間的關係。其中有些物件是透過Kubernetes進行管理、其他物件則是透過Trident進行管理。

物件如何彼此互動？

瞭解物件、物件的適用範圍及其互動方式、最簡單的方法可能是遵循Kubernetes使用者的單一儲存要求：

1. 使用者會從先前由系統管理員設定的 Kubernetes 建立 `PersistentVolumeClaim`要求新的 `PersistentVolume`特定大小 `StorageClass`。
2. Kubernetes `StorageClass` 會將 Trident 識別為其置備程式、並包含一些參數、告訴 Trident 如何為要求的類別佈建磁碟區。
3. Trident 本身的 `StorageClass`名稱與識別相符項目的名稱相同 `Backends`、`StoragePools`可用於為類別佈建 Volume 。
4. Trident 在相符的後端配置儲存設備、並建立兩個物件： A `PersistentVolume` in Kubernetes 、告知 Kubernetes 如何尋找、裝載及處理磁碟區、以及 Trident 中保留與實際儲存設備之間關係的磁碟區 `PersistentVolume`。
5. Kubernetes 將綁定 `PersistentVolumeClaim`到新 `PersistentVolume`的 .包含 `PersistentVolume` 掛載的 `Pod` `PersistentVolumeClaim`、位於其執行的任何主機上。
6. 使用者使用指向 Trident 的建立 `VolumeSnapshot`現有 `PVC` 的 `VolumeSnapshotClass`。
7. Trident會識別與該PVc相關聯的磁碟區、並在其後端建立磁碟區快照。它也會建立

`VolumeSnapshotContent`指示 Kubernetes 如何識別快照的。

8. 使用者可以使用 `VolumeSnapshot`做為來源建立 `PersistentVolumeClaim`。
9. Trident 會識別所需的快照、並執行與建立和所 `Volume`需的相同步驟集 `PersistentVolume`。

如需進一步瞭解 Kubernetes 物件、我們強烈建議您閱讀 "持續磁碟區" Kubernetes 文件的一節。

Kubernetes `PersistentVolumeClaim` 物件

Kubernetes `PersistentVolumeClaim` 物件是 Kubernetes 叢集使用者提出的儲存要求。

除了標準規格之外、Trident還可讓使用者指定下列Volume專屬附註、以覆寫您在後端組態中設定的預設值：

註釋	**Volume**選項	支援的驅動程式
trident.netapp.io/fileSystem	檔案系統	ONTAP-SAN、solidfire-san、ONTAP-san經濟型
trident.netapp.io/cloneFromPVC	cloneSourceVolume	ONTAP-NAS、ONTAP-SAN、solidfire-san、azure-NetApp-Files、GCP-CVS、 ONTAP-san經濟型
trident.netapp.io/splitOnClone	分岔OnClone	ONTAP-NAS、ONTAP-SAN
trident.netapp.io/protocol	傳輸協定	任何
trident.netapp.io/exportPolicy	匯出原則	ONTAP-NAS、ONTAP-NAS-經濟型、ONTAP-NAS- Flexgroup
trident.netapp.io/snapshotPolicy	Snapshot原則	ONTAP-NAS、ONTAP-NAS-經濟型、ONTAP-NAS-flexgroup、ONTAP-SAN
trident.netapp.io/snapshotReserve	Snapshot保留區	ONTAP-NAS、ONTAP-NAs-flexgroup、ONTAP-SAN、GCP-CVS
trident.netapp.io/snapshotDirectory	Snapshot目錄	ONTAP-NAS、ONTAP-NAS-經濟型、ONTAP-NAS- Flexgroup
trident.netapp.io/unixPermissions	unix權限	ONTAP-NAS、ONTAP-NAS-經濟型、ONTAP-NAS- Flexgroup
trident.netapp.io/blockSize	區塊大小	solidfire-san

如果建立的 PV 具有 `Delete`回收原則、 `Trident` 會在 `PV` 發行時（亦即使用者刪除 `PVC` 時）同時刪除 `PV` 和備份 `Volume` 。如果刪除動作失敗、`Trident`會將`PV`標示為這樣、並定期重試該作業、直到成功或手動刪除`PV`為止。如果 `PV` 使用該 `Retain`原則、 `Trident` 會忽略該原則、並假設系統管理員會從 `Kubernetes` 和後端清理該原則、以便在磁碟區移除之前備份或檢查該磁碟區。請注意、刪除`PV`並不會導致`Trident`刪除背板`Volume`。您應該使用 `REST API` 將其移除(`tridentctl`) 。

Trident支援使用csi規格建立Volume Snapshot：您可以建立Volume Snapshot、並將其作為資料來源來複製現有的PVCS。如此一來、PV的時間點複本就能以快照形式呈現給Kubernetes。快照可用來建立新的PV。請看一`On-Demand Volume Snapshots`下、瞭解這項功能的運作方式。

Trident 也提供 `cloneFromPVC`建立複本的和 `splitOnClone`註釋。您可以使用這些註釋來複製 PVC 、而無需使用 CSI 實作。

以下是一個範例：如果使用者已經有一個稱為的 PVC 、則 `mysql`使用者可以使用附註建立新的 PVC `mysqlclone 、例如 trident.netapp.io/cloneFromPVC: mysql。使用此註釋集、Trident會複製對應於mySQL PVc的磁碟區、而非從頭開始配置磁碟區。

請考量以下幾點：

- NetApp 建議複製閒置磁碟區。
- 一個PVc及其複本應位於相同的Kubernetes命名空間中、且具有相同的儲存類別。
- 使用 `ontap-nas`和 `ontap-san`驅動程式時、最好將 PVC 註釋與一起 `trident.netapp.io/cloneFromPVC`設定 `trident.netapp.io/splitOnClone。`trident.netapp.io/splitOnClone`設為 `true`時、 Trident 會將複製的磁碟區與父磁碟區分離、因此會完全將複製磁碟區的生命週期與父磁碟區分離、而犧牲一些儲存效率。如果不將其設定 `trident.netapp.io/splitOnClone`或設定為 `false`可減少後端的空間使用量、而會犧牲父磁碟區和複製磁碟區之間的相依性、因此除非先刪除複製、否則無法刪除父磁碟區。分割實體複製是合理的做法、是將空的資料庫磁碟區複製到磁碟區及其實體複製環境、以大幅分散差異、而非ONTAP 受益於由NetApp提供的儲存效率。

此 `sample-input`目錄包含用於 Trident 的 PVC 定義範例。如需與 Trident Volume 相關的參數和設定的完整說明、請參閱。

Kubernetes `PersistentVolume` 物件

Kubernetes `PersistentVolume` 物件代表可供 Kubernetes 叢集使用的儲存區。它的生命週期與使用它的Pod無關。

Trident 會根據它所配置的磁碟區、自動建立 `PersistentVolume`物件並在 Kubernetes 叢集上登錄。您不需要自行管理。

當您建立的 PVC 參照 Trident 型 `StorageClass`時、 Trident 會使用對應的儲存類別來配置新的磁碟區、並為該磁碟區登錄新的 PV 。在設定已配置的Volume和對應的PV時、Trident遵循下列規則：

- Trident會產生Kubernetes的PV名稱、以及用來配置儲存設備的內部名稱。在這兩種情況下、都是確保名稱在其範圍內是唯一的。
- 磁碟區的大小會盡可能接近在室早中所要求的大小、不過視平台而定、磁碟區可能會四捨五入至最接近的可分配數量。

Kubernetes `StorageClass` 物件

Kubernetes `StorageClass` 物件是以中的名稱來指定 `PersistentVolumeClaims`、以提供一組內容的儲存空間。儲存類別本身會識別要使用的資源配置程式、並根據資源配置程式所瞭解的方式來定義該組內容。

這是需要由系統管理員建立及管理的兩個基本物件之一。另一個是Trident後端物件。

使用 Trident 的 Kubernetes `StorageClass` 物件看起來像這樣：

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: <Name>
provisioner: csi.trident.netapp.io
mountOptions: <Mount Options>
parameters: <Trident Parameters>
allowVolumeExpansion: true
volumeBindingMode: Immediate
```

這些參數是Trident專屬的、可告訴Trident如何為類別配置Volume。

儲存類別參數包括：

屬性	類型	必要	說明
屬性	map[stric]字串	否	請參閱以下「屬性」一節
storagePools	map[stringList	否	將後端名稱對應至中的儲存資源池清單
其他StoragePools	map[stringList	否	將後端名稱對應至中的儲存資源池清單
排除StoragePools	map[stringList	否	將後端名稱對應至中的儲存資源池清單

儲存屬性及其可能值可分類為儲存資源池選擇屬性和Kubernetes屬性。

儲存資源池選擇屬性

這些參數決定應使用哪些Trident託管儲存資源池來配置特定類型的磁碟區。

屬性	類型	價值	優惠	申請	支援者
媒體1^	字串	HDD、混合式、SSD	資源池包含此類型的媒體、混合式表示兩者	指定的媒體類型	ONTAP-NAS、ONTAP-NAS-經濟型、ONTAP-NAS-flexgroup、ONTAP-SAN、solidfire-san
資源配置類型	字串	纖薄、厚實	Pool支援此資源配置方法	指定的資源配置方法	厚：全ONTAP 是邊、薄：全ONTAP 是邊、邊、邊、邊、邊、邊、邊、邊、邊、邊、邊

屬性	類型	價值	優惠	申請	支援者
後端類型	字串	ONTAP-NAS、ONTAP-NAS-經濟型、ONTAP-NAS-flexgroup、ONTAP-SAN、solidfire-san、GCP-CVS、azure-NetApp-Files、ONTAP-san經濟	集區屬於此類型的後端	指定後端	所有驅動程式
快照	布爾	對、錯	集區支援具有快照的磁碟區	已啟用快照的Volume	ONTAP-NAS、ONTAP-SAN、Solidfire-SAN、GCP-CVS
複製	布爾	對、錯	資源池支援複製磁碟區	已啟用複本的Volume	ONTAP-NAS、ONTAP-SAN、Solidfire-SAN、GCP-CVS
加密	布爾	對、錯	資源池支援加密磁碟區	已啟用加密的Volume	ONTAP-NAS、ONTAP-NAS-經濟型、ONTAP-NAS-FlexGroups、ONTAP-SAN
IOPS	內部	正整數	集區能夠保證此範圍內的IOPS	Volume保證這些IOPS	solidfire-san

[1]：ONTAP Select 不受支援

在大多數情況下、所要求的值會直接影響資源配置、例如、要求完整資源配置會導致資源配置較為密集的Volume。不過、元素儲存資源池會使用其提供的IOPS下限和上限來設定QoS值、而非所要求的值。在此情況下、要求的值僅用於選取儲存資源池。

理想情況下、您可以單獨使用 `attributes`來建構滿足特定類別需求所需的儲存設備品質。`Trident` 會自動探索並選取符合您指定之 *`all`* 的儲存資源池 `attributes`。

如果您發現自己無法 `attributes`自動為類別選取適當的集區、您可以使用 `storagePools`和 `additionalStoragePools`參數進一步調整集區、甚至是選取一組特定的集區。

您可以使用此 `storagePools`參數進一步限制與任何指定相匹配的池集 `attributes`。換句話說、 Trident 會使用和 `storagePools`參數所識別的集區交集來進行資源 `attributes`配置。您可以單獨使用參數、也可以同時使用兩者。

您可以使用此 `additionalStoragePools`參數來擴充 Trident 用於資源配置的集區集區集 , 而不受和 `storagePools`參數所選取的任何集區 `attributes`限制。

您可以使用此 `excludeStoragePools`參數來篩選 Trident 用於資源配置的資源池集。使用此參數會移除任何相符

的集區。

在和 additionalStoragePools`參數中， `storagePools`每個項目都採用格式 `<backend>:<storagePoolList>`，其中 <storagePoolList>`是指定後端的儲存集區清單，以逗號分隔。例如，的值 `additionalStoragePools`可能看起來像 `ontapnas_192.168.1.100:aggr1,aggr2;solidfire_192.168.1.101:bronze`。這些清單接受後端值和清單值的regex值。您可以使用 `tridentctl get backend`取得後端及其集區的清單。

Kubernetes屬性

這些屬性在動態資源配置期間、不會影響Trident選擇儲存資源池/後端。相反地、這些屬性只會提供Kubernetes持續磁碟區所支援的參數。工作節點負責檔案系統建立作業、可能需要檔案系統公用程式、例如xfsprogs。

屬性	類型	價值	說明	相關驅動因素	**Kubernetes**版本
FSType	字串	ext4 、 ext3 、 xfs	區塊磁碟區的檔案系統類型	solidfire-san 、ontap、nap、nap、nas經濟、ontap、nas 、flexgroup、ontap、san、ONTAP-san經濟型	全部
owVolume擴充	布林值	對、錯	啟用或停用對增加PVc大小的支援	ONTAP-NAS 、ONTAP-NAS-經濟型、ONTAP-NAS-flexgroup 、ONTAP-SAN、ONTAP-san經濟型、solidfire-san 、gcp-CVS 、azure-netapp檔案	超過 1.11 個
Volume BindingMode	字串	立即、WaitForFirst消費者	選擇何時進行磁碟區繫結和動態資源配置	全部	1.19 - 1.26

- 此 `fsType`參數用於控制 `SAN LUN` 所需的檔案系統類型。此外、 `Kubernetes` 也會使用儲存類別中的存在來表示檔案系統存在 `fsType`。只有在設定時、才能使用 Pod 的安全內容 `fsType`來控制 `Volume` 擁有權 `fsGroup`。如需使用內容設定 Volume 擁有權的概述 `fsGroup`、請參閱"Kubernetes：設定Pod或Container的安全內容"。Kubernetes 只有在下列情況下才會套用此 `fsGroup`值：

  - `fsType`在儲存類別中設定。
  - PVc存取模式為rwo。

  對於NFS儲存驅動程式、檔案系統已存在做為NFS匯出的一部分。若要使用 `fsGroup`儲存類別，仍需指定 `fsType`。您可以將其設定為或任何非 null 值。 `nfs`

- 如需有關 Volume 擴充的詳細資訊、請參閱"展開Volume"。
- Trident 安裝程式套件提供多個儲存類別定義範例`sample-input/storage-class-*.yaml`、可與中的 Trident 搭配使用。刪除Kubernetes儲存類別也會刪除對應的Trident儲存類別。

Kubernetes `VolumeSnapshotClass` 物件

Kubernetes `VolumeSnapshotClass` 物件類似於 `StorageClasses`。它們有助於定義多種儲存類別、並由Volume Snapshot參考、以將快照與所需的Snapshot類別建立關聯。每個Volume Snapshot都與單一Volume Snapshot類別相關聯。

`VolumeSnapshotClass`應由管理員定義、以建立快照。建立具有下列定義的`Volume Snapshot`類別：

```
apiVersion: snapshot.storage.k8s.io/v1
kind: VolumeSnapshotClass
metadata:
  name: csi-snapclass
driver: csi.trident.netapp.io
deletionPolicy: Delete
```

`driver`指定 Kubernetes 、 Trident 會處理該類別的磁碟區快照要求 `csi-snapclass`。 `deletionPolicy`指定必須刪除快照時要採取的動作。當`deletionPolicy`設為`Delete`時、磁碟區快照物件以及儲存叢集上的基礎快照會在刪除快照時移除。或者、將其設定為`Retain`表示 `VolumeSnapshotContent`保留實體快照。

Kubernetes `VolumeSnapshot` 物件

Kubernetes `VolumeSnapshot` 物件是建立磁碟區快照的要求。就像使用者針對磁碟區所提出的要求一樣、磁碟區快照是使用者建立現有虛擬磁碟快照的要求。

當磁碟區快照要求進入時、 Trident 會自動管理後端磁碟區的快照建立、並透過建立唯一物件來公開快照 `VolumeSnapshotContent`。您可以從現有的PVCS建立快照、並在建立新的PVCS時、將快照作為DataSource使用。

Volume Snapshot的生命週期與來源PVCs無關：即使刪除來源PVCs、快照仍會持續存在。刪除具有相關快照的永久虛擬磁碟時、Trident會將此永久虛擬磁碟的備份磁碟區標示為*刪除*狀態、但不會將其完全移除。刪除所有相關的快照時、即會移除該磁碟區。

Kubernetes `VolumeSnapshotContent` 物件

Kubernetes `VolumeSnapshotContent` 物件代表從已佈建的磁碟區擷取的快照。它類似於 `PersistentVolume`、表示儲存叢集上的已佈建快照。與和 `PersistentVolume`物件類似 `PersistentVolumeClaim`、建立快照時、 `VolumeSnapshotContent`物件會維護對物件的一對一對應 `VolumeSnapshot`、而物件已要求建立快照。

`VolumeSnapshotContent`物件包含可唯一識別快照的詳細資料、例如 `snapshotHandle`。這 `snapshotHandle`是 `PV` 名稱與物件名稱的獨特組合 `VolumeSnapshotContent`。

當快照要求出現時、Trident會在後端建立快照。建立快照之後、 Trident 會設定 `VolumeSnapshotContent`物件、將快照公開給 Kubernetes API 。

一般而言、您不需要管理 `VolumeSnapshotContent`物件。例外情況是您想要"匯入 Volume 快照"在 Trident 之外建立。

Kubernetes `CustomResourceDefinition` 物件

Kubernetes自訂資源是Kubernetes API中由系統管理員定義的端點、用於將類似物件分組。Kubernetes支援建立自訂資源來儲存物件集合。您可以執行來取得這些資源定義 `kubectl get crds`。

自訂資源定義（CRD）及其相關的物件中繼資料會由Kubernetes儲存在其中繼資料儲存區中。如此一來、您就不需要另外建立Trident的儲存區。

Trident 使用 `CustomResourceDefinition`物件來保留 Trident 物件的身分識別、例如 Trident 後端、 Trident 儲存類別和 Trident Volume 。這些物件由Trident管理。此外、「csi Volume Snapshot」架構也引進了定義Volume快照所需的部分CRD。

CRD是Kubernetes建構。上述資源的物件是由Trident所建立。例如、使用建立後端時 `tridentctl`、Kubernetes 會建立對應的 `tridentbackends` CRD 物件以供使用。

以下是Trident客戶需求日的幾點重點：

- 安裝Trident時、會建立一組客戶需求日、並可像使用任何其他資源類型一樣使用。
- 使用命令解除安裝 Trident 時 `tridentctl uninstall`、會刪除 Trident Pod 、但不會清除建立的客戶需求日。請參閱"解除安裝Trident"以瞭解如何從頭開始完全移除和重新設定 Trident 。

Trident 物件 StorageClass

Trident 會為 Kubernetes 物件建立相符的儲存類別 StorageClass、這些物件在其置備程式欄位中指定 csi.trident.netapp.io。儲存類別名稱符合其所代表的 Kubernetes 物件名稱 StorageClass。

> 使用 Kubernetes 時、這些物件會在登錄使用 Trident 做為置備程式的 Kubernetes 時自動建立 StorageClass。

儲存類別包含一組磁碟區需求。Trident會將這些需求與每個儲存資源池中的屬性相符；如果符合、則該儲存資源池是使用該儲存類別來配置磁碟區的有效目標。

您可以使用REST API建立儲存類別組態、以直接定義儲存類別。不過、對於 Kubernetes 部署、我們預期在登錄新的 Kubernetes 物件時會建立這些部署 StorageClass。

Trident後端物件

後端代表儲存供應商、其中Trident會配置磁碟區；單一Trident執行個體可管理任何數量的後端。

> 這是您自己建立和管理的兩種物件類型之一。另一個是 Kubernetes StorageClass 物件。

有關如何構造這些對象的詳細信息，請參閱"設定後端"。

Trident 物件 StoragePool

儲存資源池代表可在每個後端上進行資源配置的不同位置。就支援而言ONTAP 、這些項目對應於SVM中的集合體。對於NetApp HCI / SolidFire、這些服務會對應到系統管理員指定的QoS頻段。就架構而言、這些項目對應於雲端供應商所在的地區。Cloud Volumes Service每個儲存資源池都有一組獨特的儲存屬性、可定義其效能特性和資料保護特性。

與此處的其他物件不同、儲存資源池候選項目一律會自動探索及管理。

Trident 物件 Volume

Volume 是資源配置的基本單位，包括 NFS 共用， iSCSI 和 FC LUN 等後端端點。在 Kubernetes 中、這些直接對應到 PersistentVolumes。建立磁碟區時、請確定它有一個儲存類別、決定該磁碟區可以配置的位置及大小。

> - 在Kubernetes中、會自動管理這些物件。您可以檢視這些資源、以查看資源配置的Trident內容。
> - 刪除具有相關快照的PV時、對應的Trident Volume會更新為*刪除*狀態。若要刪除Trident磁碟區、您應該移除該磁碟區的快照。

Volume組態會定義已配置磁碟區應具備的內容。

屬性	類型	必要	說明
版本	字串	否	Trident API版本（「1」）
名稱	字串	是的	要建立的Volume名稱

屬性	類型	必要	說明
storageClass	字串	是的	配置Volume時使用的儲存類別
尺寸	字串	是的	要配置的磁碟區大小（以位元組為單位）
傳輸協定	字串	否	要使用的傳輸協定類型；「檔案」或「區塊」
內部名稱	字串	否	儲存系統上的物件名稱；由Trident產生
cloneSourceVolume	字串	否	Sname（NAS、SAN）& S--*：要複製的磁碟區名稱ONTAP SolidFire
分岔OnClone	字串	否	例（NAS、SAN）：從父實體分割複本ONTAP
Snapshot原則	字串	否	S--*：快照原則ONTAP
Snapshot保留區	字串	否	Sing-*：保留給快照的磁碟區百分比ONTAP
匯出原則	字串	否	ONTAP-NAS*：要使用的匯出原則
Snapshot目錄	布爾	否	ONTAP-NAS*：快照目錄是否可見
unix權限	字串	否	ONTAP-NAS*：初始UNIX權限
區塊大小	字串	否	S--*：區塊/區段大小SolidFire
檔案系統	字串	否	檔案系統類型

Trident 會在建立磁碟區時產生 `internalName`。這包括兩個步驟。首先，它會將儲存前置詞（即後端組態中的預設或前置詞）預先加 `trident`到磁碟區名稱，從而產生表單的名稱 `<prefix>-<volume-name>`。然後、它會繼續清理名稱、取代後端不允許的字元。對於 ONTAP 後端、它會以底線取代連字號（因此內部名稱會變成 `<prefix>_<volume-name>`）。對於元素後端、它會以連字號取代底線。

您可以使用 Volume 組態、使用 REST API 直接配置磁碟區、但在 Kubernetes 部署中、我們預期大多數使用者都會使用標準 Kubernetes `PersistentVolumeClaim` 方法。Trident會自動建立此Volume物件、做為資源配置程序的一部分。

Trident 物件 `Snapshot`

快照是磁碟區的時間點複本、可用來配置新的磁碟區或還原狀態。在 Kubernetes 中、這些項目會直接對應到 `VolumeSnapshotContent`物件。每個快照都與一個Volume相關聯、該磁碟區是快照資料的來源。

每個物件都 `Snapshot`包含下列內容：

屬性	類型	必要	說明
版本	字串	是的	Trident API版本（「1」）
名稱	字串	是的	Trident Snapshot物件的名稱
內部名稱	字串	是的	儲存系統上Trident Snapshot物件的名稱
Volume名稱	字串	是的	為其建立快照的持續Volume名稱
Volume內部名稱	字串	是的	儲存系統上相關Trident Volume物件的名稱

在Kubernetes中、會自動管理這些物件。您可以檢視這些資源、以查看資源配置的Trident內容。

建立 Kubernetes 物件要求時 `VolumeSnapshot`、 Trident 會在備份儲存系統上建立快照物件。 `internalName`此快照物件的是藉由將前置詞與 `UID`物件的 `VolumeSnapshot`結合而產生 `snapshot-`（例如 `snapshot-e8d8a0ca-9826-11e9-9807-525400f3f660`）。 `volumeName`並 `volumeInternalName`透過取得備份磁碟區的詳細資料來填入。

Trident 物件 `ResourceQuota`

Trident 去除會使用優先順序類別（ Kubernetes 中可用的最高優先順序類別）、以確保 Trident 能在正常節點關機期間識別及清理磁碟區、並允許 Trident 去 `system-node-critical`除設定群組在資源壓力較大的叢集中、以較低的優先順序來搶佔工作負載。

為達成此目標、 Trident 採用 `ResourceQuota`物件來確保 Trident 標章集上的「系統節點關鍵」優先順序類別獲得滿足。在建立部署和取消設定集之前、 Trident 會先尋找物件、如果未發現、則會套用該 `ResourceQuota`物件。

如果您需要更多控制預設資源配額和優先順序類別、可以使用 Helm 圖表來產生 `custom.yaml`或設定 `ResourceQuota`物件。

以下是「資源配額」物件優先處理Trident的範例。

```
apiVersion: <version>
kind: ResourceQuota
metadata:
  name: trident-csi
  labels:
    app: node.csi.trident.netapp.io
spec:
  scopeSelector:
    matchExpressions:
      - operator: In
        scopeName: PriorityClass
        values:
          - system-node-critical
```

有關資源配額的詳細信息，請參閱"Kubernetes：資源配額"。

如果安裝失敗、請進行清理 `ResourceQuota`

在建立物件後安裝失敗的罕見情況 `ResourceQuota`下、請先嘗試"正在解除安裝"再重新安裝。

如果無法正常運作、請手動移除 `ResourceQuota`物件。

移除 `ResourceQuota`

如果您偏好控制自己的資源配置、可以使用下列命令移除 Trident `ResourceQuota` 物件：

```
kubectl delete quota trident-csi -n trident
```

Pod安全標準（PSS）與安全內容限制（SCC）

Kubernetes Pod安全標準（Ps）和Pod安全政策（Ps）定義權限等級、並限制Pod的行為。OpenShift Security內容限制（SCC）同樣定義OpenShift Kubernetes Engine特有的Pod限制。為了提供此自訂功能、 Trident 會在安裝期間啟用特定權限。下列各節詳細說明 Trident 所設定的權限。

PSS-取代Pod安全性原則（PSP）。在Kubernetes v1.21中、已不再使用PSP、將在v1.25中移除。如需詳細資訊"Kubernetes：安全性"、請參閱。

必要的**Kubernetes**安全內容和相關欄位

權限	說明
權限	SCSI需要雙向裝載點、這表示Trident節點Pod必須執行特殊權限容器。如需詳細資訊、請 "Kubernetes：掛載傳播"參閱。
主機網路	iSCSI 常駐程式所需。 `iscsiadm`管理 iSCSI 掛載、並使用主機網路與 iSCSI 常駐程式通訊。
主機 IPC	NFS使用程序間通訊（IPC）與nfsd通訊。
主機 PID	啟動 NFS 所需 `rpc-statd`。Trident 會查詢主機處理程序、以判斷在掛載 NFS 磁碟區之前是否 `rpc-statd`正在執行。

功能	此 `SYS_ADMIN`功能是權限容器預設功能的一部分。例如、 `Docker` 會為權限容器設定以下功能：
	`CapPrm: 0000003fffffffff`
	`CapEff: 0000003fffffffff`

Seccomp	Seccomp 設定檔在特殊權限的容器中一律為「未限制」、因此無法在 Trident 中啟用。
SELinux	在 OpenShift 上、權限容器會在（「超級貴賓 Container 」）網域中執行 `spc_t`、而非權限容器則會在網域中執行 `container_t`。在上 `containerd`、安裝後 `container-selinux`、所有容器都會在網域中執行 `spc_t`、這會有效停用 SELinux 。因此、Trident 不會新增 `seLinuxOptions`至容器。
DAC	權限容器必須以root身分執行。非權限容器會以root身分執行、以存取csi所需的UNIX通訊端。

Pod安全標準（PSS）

標籤	說明	預設
`pod-security.kubernetes.io/enforce pod-security.kubernetes.io/enforce-version`	允許Trident控制器和節點進入安裝命名空間。請勿變更命名空間標籤。	`enforce: privileged enforce-version: <version of the current cluster or highest version of PSS tested.>`

變更命名空間標籤可能會導致無法排程Pod、「建立錯誤：...」或「警告：Trident：Cig-...」。如果發生這種情況、請檢查的命名空間標籤是否 `privileged`已變更。如果是、請重新安裝Trident。

Pod安全原則（PSP）

欄位	說明	預設
`allowPrivilegeEscalation`	特殊權限容器必須允許權限提高。	`true`
`allowedCSIDrivers`	Trident不使用即時的csi暫時性磁碟區。	空白

欄位	說明	預設
allowedCapabilities	非權限Trident容器不需要比預設集更多的功能、而且會將所有可能的功能授予權限容器。	空白
allowedFlexVolumes	Trident 不使用"FlexVolume驅動程式"，因此它們不包括在允許的卷列表中。	空白
allowedHostPaths	Trident節點Pod會掛載節點的根檔案系統、因此設定此清單沒有任何好處。	空白
allowedProcMountTypes	Trident 不使用任何 ProcMountTypes。	空白
allowedUnsafeSysctls	Trident 不需要任何不安全的 sysctls。	空白
defaultAddCapabilities	不需要將任何功能新增至權限容器。	空白
defaultAllowPrivilegeEscalation	每個Trident Pod都會處理允許權限提高的問題。	false
forbiddenSysctls	不 `sysctls`允許。	空白
fsGroup	Trident容器以root執行。	RunAsAny
hostIPC	掛載 NFS 磁碟區需要主機 IPC 才能與進行通訊 nfsd	true
hostNetwork	iscsiadm要求主機網路與iSCSI精靈進行通訊。	true
hostPID	需要主機 PID 才能檢查節點上是否 `rpc-statd`正在執行。	true
hostPorts	Trident不使用任何主機連接埠。	空白
privileged	Trident節點Pod必須執行特殊權限容器、才能掛載磁碟區。	true
readOnlyRootFilesystem	Trident節點Pod必須寫入節點檔案系統。	false
requiredDropCapabilities	Trident節點Pod執行特殊權限容器、無法丟棄功能。	none
runAsGroup	Trident容器以root執行。	RunAsAny
runAsUser	Trident容器以root執行。	runAsAny
runtimeClass	Trident 不使用 RuntimeClasses。	空白
seLinux	Trident 並未設定 seLinuxOptions、因為目前容器執行時間和 Kubernetes 套裝作業系統處理 SELinux 的方式有差異。	空白

欄位	說明	預設
supplementalGroups	Trident容器以root執行。	RunAsAny
volumes	Trident Pod需要這些Volume外掛程式。	hostPath, projected, emptyDir

安全內容限制（SCC）

標籤	說明	預設
allowHostDirVolumePlugin	Trident節點Pod會掛載節點的根檔案系統。	true
allowHostIPC	掛載 NFS 磁碟區需要主機 IPC 才能與通訊 nfsd。	true
allowHostNetwork	iscsiadm要求主機網路與iSCSI精靈進行通訊。	true
allowHostPID	需要主機 PID 才能檢查節點上是否 `rpc-statd`正在執行。	true
allowHostPorts	Trident不使用任何主機連接埠。	false
allowPrivilegeEscalation	特殊權限容器必須允許權限提高。	true
allowPrivilegedContainer	Trident節點Pod必須執行特殊權限容器、才能掛載磁碟區。	true
allowedUnsafeSysctls	Trident 不需要任何不安全的 sysctls。	none
allowedCapabilities	非權限Trident容器不需要比預設集更多的功能、而且會將所有可能的功能授予權限容器。	空白
defaultAddCapabilities	不需要將任何功能新增至權限容器。	空白
fsGroup	Trident容器以root執行。	RunAsAny
groups	此SCC僅適用於Trident、並與其使用者有關。	空白
readOnlyRootFilesystem	Trident節點Pod必須寫入節點檔案系統。	false
requiredDropCapabilities	Trident節點Pod執行特殊權限容器、無法丟棄功能。	none
runAsUser	Trident容器以root執行。	RunAsAny
seLinuxContext	Trident 並未設定 seLinuxOptions、因為目前容器執行時間和 Kubernetes 套裝作業系統處理 SELinux 的方式有差異。	空白
seccompProfiles	特殊權限容器永遠都會執行「未限制」。	空白

標籤	說明	預設
`supplementalGroups`	Trident容器以root執行。	`RunAsAny`
`users`	提供一個項目來將此SCC繫結至Trident命名空間中的Trident使用者。	不適用
`volumes`	Trident Pod需要這些Volume外掛程式。	`hostPath, downwardAPI, projected, emptyDir`