參考資料
Trident

NetApp
March 02, 2026

目錄

參考資料

Trident 連接埠

深入瞭解 Trident 用於通訊的連接埠。

總覽

Trident 使用各種連接埠與 Kubernetes 叢集內部以及儲存後端進行通訊。以下概述了關鍵連接埠、其用途和安全注意事項。

- 出站流量控制重點：Kubernetes 節點（控制器和工作節點）主要啟動對儲存 LIF/IP 的流量，因此 iptables 規則應允許節點 IP 透過這些連接埠向特定儲存 IP 發起出站流量。避免使用過於寬泛的「任意到任意」規則。
- 入站限制：將內部 Trident 連接埠限制為叢集內部流量（例如，使用 Calico 等 CNI）。主機防火牆上無不必要的入站暴露。
- 協定安全性：
  - 盡可能使用 TCP （更可靠）。
  - 如果敏感，請為 iSCSI 啟用 CHAP/IPsec；為管理啟用 TLS/HTTPS（連接埠 443/8443）。
  - 對於 NFSv4（Trident 中的預設值），如果不需要，請剪除 UDP/ 較舊的 NFSv3 連接埠（例如 4045-4049）。
  - 限制在受信任的子網路內；使用 Prometheus 等工具進行監控（選用連接埠 8001）。

控制器節點的連接埠

這些連接埠主要用於 Trident 操作員（後端管理）。所有內部連接埠均為 Pod 層級；僅當主機防火牆干擾 CNI 時才允許在節點上使用。

連接埠 / 協定	方向	目的	驅動程式 / 通訊協定	安全注意事項
TCP 8000	入站 / 出站（叢集內部）	Trident REST 伺服器（operator-controller 通訊）	全部	僅限使用 pod CIDR；不得暴露於外部環境。
TCP 8443	入站 / 出站（叢集內部）	反向通道 HTTPS （安全內部 API）	全部	採用 TLS 加密；若使用，則僅限於 Kubernetes 服務網格。
TCP 8001	入站（叢集內部、選用）	Prometheus 指標	全部	僅對監控工具開放（例如，使用 RBAC）；如果未使用則停用。
TCP 443	傳出	HTTPS 至 ONTAP SVM/ 叢集管理 LIF	ONTAP（全部）、ANF	需要進行 TLS 憑證驗證；僅限管理 LIF IP 位址。
TCP 8443	傳出	HTTPS 至 E 系列 Web Services Proxy	E 系列（iSCSI）	預設 REST API；使用憑證；可在後端 YAML 中設定。

工作節點的連接埠

這些連接埠用於 CSI 節點守護程序集和 pod 掛載。資料連接埠用於出站連接到儲存資料 LIF；如果使用 NFSv3 ，則包含 NFSv3 附加資訊（NFSv4 為選用）。

連接埠 / 協定	方向	目的	驅動程式 / 通訊協定	安全注意事項
TCP 17546	傳入（本機至 Pod）	CSI 節點存活 / 就緒偵測	全部	可設定（--probe-port）；確保無主機衝突；僅限本機。
TCP 8000	入站 / 出站（叢集內部）	Trident REST伺服器	全部	如上所述；Pod 內部。
TCP 8443	入站 / 出站（叢集內部）	後端通道HTTPS	全部	如上所述。
TCP 8001	入站（叢集內部、選用）	Prometheus 指標	全部	如上所述。
TCP 443	傳出	HTTPS 至 ONTAP SVM/ 叢集管理 LIF	ONTAP（全部）、ANF	如上所述；用於探索。
TCP 8443	傳出	HTTPS 至 E 系列 Web Services Proxy	E 系列（iSCSI）	如上所述。
TCP/UDP 111	傳出	RPCBIND/portmapper	ONTAP-NAS（NFSv3/v4）、ANF（NFS）	v3 版本必需；v4 版本可選（防火牆卸載）；如果僅使用 NFSv4 ，則限制使用。
TCP/UDP 2049	傳出	NFS 精靈程式	ONTAP-NAS（NFSv3/v4）、ANF（NFS）	核心資料；眾所周知；使用 TCP 以確保可靠性。
TCP/UDP 635	傳出	掛載精靈程式	ONTAP-NAS（NFSv3/v4）、ANF（NFS）	掛載；可進行雙向回呼（如有需要，允許傳入臨時連線）。
UDP 4045	傳出	NFS 鎖定管理器（nlockmgr）	ONTAP-NAS（NFSv3）	檔案鎖定；跳過 v4 （ pNFS 處理）；僅限 UDP 。
UDP 4046	傳出	NFS 狀態監視器（statd）	ONTAP-NAS（NFSv3）	通知；可能需要入站臨時連接埠 (1024-65535) 進行回調。
UDP 4049	傳出	NFS 配額守護程式（rquotad）	ONTAP-NAS（NFSv3）	配額；v4 版本跳過。
TCP 3260	傳出	iSCSI 目標（探索 / 資料 / CHAP）	ONTAP-SAN（iSCSI）、E-Series（iSCSI）	眾所周知；透過此連接埠進行 CHAP 驗證；啟用雙向 CHAP 以確保安全。
TCP 445	傳出	SMB/CIFS	ONTAP-NAS (SMB)、ANF (SMB)	眾所周知；使用具有加密功能的 SMB3（Trident 註釋 netapp.io/smb-encryption=true）。

連接埠 / 協定	方向	目的	驅動程式 / 通訊協定	安全注意事項
TCP/UDP 88（選用）	傳出	Kerberos 驗證	ONTAP （NFS/SMB/iSCSI with Kerb）	如果使用 Kerberos（非預設）；連接至 AD 伺服器，而非儲存設備。
TCP/UDP 389（選用）	傳出	LDAP	ONTAP （NFS/SMB 搭配 LDAP ）	類似；用於名稱解析 / 驗證；限制為 AD。

您可以在安裝期間使用變更活動力/整備度探針連接埠 `--probe-port` 旗標。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` 以啟用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。使用 `-address ""` 可讓REST介面從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. 使用者會建立一個「PeristentVolume Claim」、要求系統管理員先前設定的Kubernetes「torageClass」中的特定大小的新「PersistentVolume」。
2. Kubernetes「torageClass」可將Trident識別為其資源配置程式、並包含可告知Trident如何為所要求的類別資源配置Volume的參數。
3. Trident查看自己的「儲存類」、其名稱與用來為類別配置磁碟區的「後端」和「儲存類」相符。
4. Trident會在相符的後端上配置儲存設備、並建立兩個物件：Kubernetes的「PeristentVolume」、告訴Kubernetes如何尋找、掛載及處理Volume、以及Trident中保留「PeristentVolume」與實際儲存設備之間關係的Volume。
5. Kubernetes將「PeristentVolume Claim」連結到新的「PeristentVolume」。在執行的任何主機上、包含PeristentVolume的「PeristentVolume Claim」掛載的Pod。
6. 使用者使用指向Trident的「Volume SnapshotClass」建立現有的永久虛擬磁碟的「Volume Snapshot」。
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「Volume SnapshotContent」、指示Kubernetes如何識別快照。
8. 使用者可以使用「Volume Snapshot」作為來源來建立「PeristentVolume Claim」。
9. Trident會識別所需的快照、並執行建立「PeristentVolume」和「Volume」所需的相同步驟。

> 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「PeristentVolume Claim」物件是Kubernetes叢集使用者所提出的儲存要求。

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、ontap-san-economy
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
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
trident.netapp.io/skipRecoveryQueue	跳過恢復隊列	ontap-nas、ontap-nas-economy、ontap-nas-flexgroup、ontap-san、ontap-san-economy

如果建立的PV具有「刪除」回收原則、則當PV釋出時（亦即使用者刪除PVc時）、Trident會同時刪除PV和備用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「+Retain +」原則、Trident會忽略它、並假設系統管理員會從Kubernetes和後端進行清理、以便在移除之前備份或檢查磁碟區。請注意、刪除PV並不會導致Trident刪除背板Volume。您應該使用REST API（「tridentctl」）將其移除。

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。請參閱「隨需磁碟區快照」、瞭解這項功能的運作方式。

Trident也提供 `cloneFromPVC` 和 `splitOnClone` 建立複本的附註。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名為「mysql"的PVC,則使用者可以使用「trident.netapp.io/cloneFromPVC: mySQL」之類的註解來建立一個名為「mysqlclone」的新PVC.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。
- 有了「ONTAP-NAS」和「ONTAP-SAN」驅動程式、可能需要將「trident.netapp.io/splitOnClone`」標註與「trident.netapp.io/cloneFromPVC`」一起設定。Trident將trident.netapp.io/splitOnClone`設為「true」、將複製的磁碟區從父磁碟區分割出來、因此將複製的磁碟區的生命週期與其父磁碟區完全分離、而犧牲部分儲存效率。如果不將「trident.netapp.io/splitOnClone`」設定為「假」、則會減少後端的空間使用量、而犧牲父磁碟區與複製磁碟區之間的相依性、使父磁碟區無法刪除、除非先刪除複本。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。

。 `sample-input` 目錄包含用於Trident的PVc定義範例。請參閱 以取得與 Trident Volume 相關的參數和設定的完整說明。

Kubernetes `PersistentVolume` 物件

Kubernetes「PersistentVolume」物件代表Kubernetes叢集可用的儲存設備。它的生命週期與使用它的Pod無關。

Trident會建立「PeristentVolume」物件、並根據其所配置的磁碟區、自動在Kubernetes叢集上登錄。您不需要自行管理。

當您建立參照Trident型「TorageClass」的PVc時、Trident會使用對應的儲存類別來配置新的Volume、並針對該Volume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的「torageClass」物件是以名稱在「PeristentVolume Claims」中指定、以一組內容來配置儲存設備。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「torageClass」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-economy、ontap-nas-flexgroup 、ontap-san 、solidfire-san 、azure-netapp-files、ontap-san-economy	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
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
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「屬性」來建構儲存設備的品質、以滿足特定類別的需求。Trident會自動探索並選取符合您指定「屬性」的_all_儲存集區。

如果您發現自己無法使用「屬性」來自動選取適合某個類別的資源池、您可以使用「儲存池」和「其他儲存池」參數來進一步精簡資源池、甚至選取特定的資源池集區。

您可以使用「儲存池」參數、進一步限制符合任何指定「屬性」的集區集區集區。換句話說、Trident會使用由「屬性」和「儲存庫」參數所識別的資源池交會來進行資源配置。您可以單獨使用參數、也可以同時使用兩者。

您可以使用「addionalStoragePools」參數來擴充Trident用來資源配置的資源池集區集區集區、而不論「attributes」和「scoragePools」參數所選取的任何資源池為何。

您可以使用「排除StoragePools」參數來篩選Trident用於資源配置的資源池集區集區。使用此參數會移除任何相符的集區。

在「儲存池」和「其他儲存池」參數中、每個項目的格式均為「<backender>:<storagePoollist>'」、其中「<storagePoollist>'」是以逗號分隔的儲存池清單、用於指定的後端。例如、「addionalStoragePools」的值可能會像是「ontapnas_192.168.1.100:solidgr1、aggr2、aggrfire、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」來取得後端及其資源池的清單。

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-economy、ontap-nas-flexgroup 、ontap-san 、ontap-san-economy、solidfire-san、azure-netapp-files	1.11+
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

- 。 `fsType` 參數用於控制SAN LUN所需的檔案系統類型。此外、Kubernetes也會使用的 `fsType` 在儲存類別中、表示檔案系統存在。您可以使用來控制Volume擁有權 `fsGroup` 只有在下列情況下、Pod的安全內容才會出現 `fsType` 已設定。請參閱 "Kubernetes：設定Pod或Container的安全內容" 如需使用設定Volume擁有權的總覽 `fsGroup` 背景。Kubernetes將套用 `fsGroup` 只有在下列情況下才會有

  - 「FSType」是在儲存類別中設定的。
  - 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。為了使用「fsGroup」、儲存類別仍需指定「FSType」。您可以將其設定為「NFS」或任何非null值。

- 請參閱 "展開Volume" 如需磁碟區擴充的詳細資料、
- Trident安裝程式套裝組合提供多個範例儲存類別定義、可與Trident搭配使用、位於「sham-INPUT /儲存設備類別-*。yaml」。刪除Kubernetes儲存類別也會刪除對應的Trident儲存類別。

Kubernetes `VolumeSnapshotClass` 物件

Kubernetes的「Volume SnapshotClass」物件類似於「儲存類別」。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。

系統管理員應定義「Volume SnapshotClass」、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
```

對Kubernetes而言、「driver」是指Trident處理「Cig-snapClass」類別的Volume快照要求。「刪除原則」指定必須刪除快照時要採取的動作。當「刪除原則」設定為「刪除」時、刪除快照時、就會移除儲存叢集上的Volume Snapshot物件和基礎Snapshot。或者、將其設為「保留」、表示保留「Volume SnapshotContent」和實體快照。

Kubernetes `VolumeSnapshot` 物件

Kubernetes「Volume Snapshot」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。

當磁碟區快照要求出現時、Trident會自動管理後端磁碟區的快照建立、並建立獨特的「Volume SnapshotContent」物件來公開快照。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。

> VolumeSnapshot 的生命週期與來源 PVC 無關：即使來源 PVC 被刪除，快照仍然存在。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「Volume SnapshotContent」物件代表從已配置的磁碟區擷取的快照。它類似於「PersistentVolume」、代表儲存叢集上已配置的快照。與「PeristentVolume Claim」和「PeristentVolume」物件類似、建立快照時、「Volume SnapshotContent」物件會維持一對一的對應、以對應「Volume Snapshot」物件、該物件已要求建立快照。

「Volume SnapshotContent」物件包含可唯一識別快照的詳細資料、例如「快照資料」。此「快照處理」是PV名稱與「Volume SnapshotContent」物件名稱的獨特組合。

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「Volume SnapshotContent」物件、並將快照公開給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 `VolumeGroupSnapshotClass` 物件

Kubernetes `VolumeGroupSnapshotClass` 物件類似於 `VolumeSnapshotClass`。它們有助於定義多種儲存類別，並被磁碟區組快照引用，以將快照與所需的快照類別關聯。每個磁碟區組快照都與單一磁碟區組快照類別相關聯。

一個 `VolumeGroupSnapshotClass`應由管理員定義，以便建立快照群組。卷冊組快照類別使用以下定義建立：

```
apiVersion: groupsnapshot.storage.k8s.io/v1beta1
kind: VolumeGroupSnapshotClass
metadata:
  name: csi-group-snap-class
  annotations:
    kubernetes.io/description: "Trident group snapshot class"
driver: csi.trident.netapp.io
deletionPolicy: Delete
```

由 Trident 處理。 `deletionPolicy`指定必須刪除群組快照時要採取的動作。當 `deletionPolicy`設定為 `Delete`，刪除快照時，磁碟區組快照物件以及儲存叢集上的底層快照也將被刪除。或者、將其設定為`Retain`表示`VolumeGroupSnapshotContent`保留實體快照。

Kubernetes `VolumeGroupSnapshot` 物件

Kubernetes `VolumeGroupSnapshot`物件是建立多個磁碟區快照的請求。正如 PVC 代表使用者對磁碟區的請求一樣，磁碟區組快照是使用者為現有 PVC 建立快照的請求。

當磁碟區組快照請求到達時，Trident 會自動管理後端磁碟區的群組快照的創建，並透過建立唯一的`VolumeGroupSnapshotContent`目的。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。

> ⓘ VolumeGroupSnapshot 的生命週期與來源 PVC 無關：即使來源 PVC 被刪除，快照仍然有效。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。當所有關聯的快照都被刪除時，磁碟區組快照也會被移除。

Kubernetes `VolumeGroupSnapshotContent` 物件

Kubernetes `VolumeGroupSnapshotContent`物件表示從已配置的磁碟區中取得的群組快照。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。

這`VolumeGroupSnapshotContent`物件包含識別快照群組的詳細信息，例如`volumeGroupSnapshotHandle`以及儲存系統上現有的各個volumeSnapshotHandles。

當快照請求到達時，Trident 會在後端建立磁碟區組快照。建立卷宗組快照後，Trident 會配置一個`VolumeGroupSnapshotContent`對象，從而將快照公開給 Kubernetes API。

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「kubecl Get crds」來取得這些資源定義。

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、當使用「tridentctl」建立後端時、Kubernetes會建立一個對應的「tridentbackend」CRD物件供其使用。

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。
- 使用解除安裝Trident時 `tridentctl uninstall` 命令、Trident Pod會刪除、但建立的客戶需求日不會清除。請參閱 "解除安裝Trident" 瞭解如何徹底移除Trident並從頭重新設定。

Trident 物件 `StorageClass`

Trident為Kubernetes建立相符的儲存類別 `StorageClass` 指定的物件 `csi.trident.netapp.io` 在他們的資源配置工具欄位中。儲存類別名稱與Kubernetes名稱相符 `StorageClass` 所代表的物件。

使用Kubernetes、當Kubernetes「torageClass」以Trident做為資源配置程式登錄時、就會自動建立這些物件。

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、在Kubernetes部署中、我們預期在登錄新的Kubernetes「torageClass」物件時、會建立這些物件。

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的「torageClass」物件。

如需如何建構這些物件的詳細資訊、請參閱 "設定後端"。

Trident 物件 `StoragePool`

儲存池代表每個後端可用於配置的不同位置。對於ONTAP而言，這些對應於 SVM 中的聚合。對於NetApp HCI/SolidFire，這些對應於管理員指定的 QoS 頻段。每個儲存池都有一組獨特的儲存屬性，這些屬性定義了其效能特徵和資料保護特徵。

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
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
跳過恢復隊列	字串	否	刪除磁碟區時，繞過儲存中的復原佇列，立即刪除磁碟區。

Trident在建立磁碟區時會產生「內部名稱」。這包括兩個步驟。首先、它會將儲存前置詞（預設的「Trident」或後端組態中的前置詞）預先加上磁碟區名稱、以「<prefix>-<volume名稱>」格式命名。然後、它會繼續清理名稱、取代後端不允許的字元。對於後端、它會以底線取代連字號（因此內部名稱會變成「<prefix>_<volume名稱>」）ONTAP 。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的Kubernetes「PeristentVolume Claim」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、這些物件會直接對應到「Volume SnapshotContent」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「napshot」物件都包含下列內容：

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。

當Kubernetes「Volume Snapshot」物件要求建立時、Trident會在備份儲存系統上建立Snapshot物件。此快照物件的「內部名稱」是將前置詞「sfapshot-」與「Volume Snapshot」物件的「UID」（例如、「sfapshot-e8d8a0ca-9826-11e9-9807-525400f3f660」）結合在一起產生的。「Volume Name」（Volume名稱）和「Volume InternalName」（磁碟區內部名稱）會透過取得備用磁碟區的詳細資料來填入資料。

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。

如果您需要對預設資源配額和優先順序類別的更多控制權、可以產生「custustry.yaml」、或使用Helm圖表來設定「資源配額」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
```

如需資源配額的詳細資訊、請參閱 "Kubernetes：資源配額"。

清理 `ResourceQuota` 如果安裝失敗

在極少數情況下、如果在建立「資源配額」物件之後安裝失敗、請先嘗試 "正在解除安裝" 然後重新安裝。

如果這不管用、請手動移除「資源配額」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、請參閱 "Kubernetes：安全性"。

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、請參閱 "Kubernetes：掛載傳播"。
主機網路	iSCSI精靈所需。「iscsiadm'管理iSCSI掛載、並使用主機網路來與iSCSI精靈通訊。
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	「SYS-ADMIN」功能是專為特殊權限容器提供的預設功能之一。例如、Docker為特殊權限容器設定了這些功能：「CapPrm：0000003fffffffff'、「CapEff: 0000003fffffff」
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安全性.Kubernetes.io/enforce（pod安全性）。Kubernetes.io/enforce版本	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：特權」的「enforce version：<目前叢集的版本或通過測試的最高版本的PSS>.」

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「特殊權限」的命名空間標籤是否已變更。如果是、請重新安裝Trident。

Pod安全原則（PSP）

欄位	說明	預設
「允許升級」	特殊權限容器必須允許權限提高。	"真的"
《分配CSIDriver》	Trident不使用即時的csi暫時性磁碟區。	空白

欄位	說明	預設
《分配能力》	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
《分配FlexVolumes》	Trident並未使用 "FlexVolume驅動程式"因此，它們不會包含在允許的磁碟區清單中。	空白
《主機路徑》	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
《處理器類型》	Trident不使用任何「ProctMountTypes」。	空白
《非安全性系統》	Trident不需要任何不安全的「縮圖」。	空白
'資料錯誤附加功能'	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。	「假」
《ForbiddenSysctls》	不允許使用"sysctls"。	空白
「fsGroup」	Trident容器以root執行。	《RunAsAny》
《hostipc》	掛載NFS磁碟區需要主機iPC與"nfsd"通訊	"真的"
「主機網路」	iscsiadm要求主機網路與iSCSI精靈進行通訊。	"真的"
"hostPID"	需要主機PID來檢查節點上是否正在執行「rps-statd」。	"真的"
"hostPortes"	Trident不使用任何主機連接埠。	空白
"特權"	Trident節點Pod必須執行特殊權限容器、才能掛載磁碟區。	"真的"
《ReadOnlyRootFilesystem》	Trident節點Pod必須寫入節點檔案系統。	「假」
《requireddropCapabilities》	Trident節點Pod執行特殊權限容器、無法丟棄功能。	無
《RunAsGroup》（《RunAsGroup》）	Trident容器以root執行。	《RunAsAny》
「RunAsUser」	Trident容器以root執行。	「RunAsAny」
《RuntimeClass》	Trident不使用「RuntimeClass」。	空白
「eLinux」	Trident並未設定「最新Linux選項」、因為目前容器執行時間與Kubernetes發行版本處理SELinux的方式有所不同。	空白
《支援團體》	Trident容器以root執行。	《RunAsAny》

欄位	說明	預設
《Volume》（Volume）	Trident Pod需要這些Volume外掛程式。	《hostPath》、《Project預計》、《empityDir.》

安全內容限制（SCC）

標籤	說明	預設
"owHostDirVolume Plugin"	Trident節點Pod會掛載節點的根檔案系統。	"真的"
"owhostipc"	掛載NFS磁碟區需要主機iPC與"nfsd"通訊。	"真的"
「允許主機網路」	iscsiadm要求主機網路與iSCSI精靈進行通訊。	"真的"
"owhostpID"	需要主機PID來檢查節點上是否正在執行「rps-statd」。	"真的"
"allowHostPort"	Trident不使用任何主機連接埠。	「假」
「允許升級」	特殊權限容器必須允許權限提高。	"真的"
《允許使用容器》	Trident節點Pod必須執行特殊權限容器、才能掛載磁碟區。	"真的"
《非安全性系統》	Trident不需要任何不安全的「縮圖」。	無
《分配能力》	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
'資料錯誤附加功能'	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》
《團體》	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。	「假」
《requireddropCapabilities》	Trident節點Pod執行特殊權限容器、無法丟棄功能。	無
「RunAsUser」	Trident容器以root執行。	《RunAsAny》
「Linux轉換」	Trident並未設定「最新Linux選項」、因為目前容器執行時間與Kubernetes發行版本處理SELinux的方式有所不同。	空白
「eccompProfiles」	特殊權限容器永遠都會執行「未限制」。	空白
《支援團體》	Trident容器以root執行。	《RunAsAny》

標籤	說明	預設
《使用者》	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
《Volume》（Volume）	Trident Pod需要這些Volume外掛程式。	《hostPath、DownwardAPI、Project預計、empyDir'》

版權資訊

商標資訊