參考資料

Astra Trident

NetApp
June 28, 2024

目錄

參考資料

Astra Trident連接埠

深入瞭解Astra Trident用於通訊的連接埠。

Astra Trident連接埠

Astra 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

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。

Astra Trident REST API

而 "tridentctl命令和選項" 這是與Astra Trident REST API互動最簡單的方法、您可以視需要直接使用REST端點。

何時使用REST API

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

為了提升安全性、Astra Trident `REST API` 在Pod內部執行時、預設限制為localhost。若要變更此行為、您必須設定Astra Trident的 `-address` 其Pod組態中的引數。

使用REST API

如需如何呼叫這些API的範例、請通過偵錯 (`-d`）旗標。如需詳細資訊、請參閱 "使用 Tridentctl 管理 Astra Trident"。

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。如需每種物件類型的規格、請參閱 "使用 Tridentctl 管理 Astra Trident"。
- 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、請參閱 "使用 Tridentctl 管理 Astra Trident"。

命令列選項

Astra Trident提供Trident Orchestrator的數個命令列選項。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` 以啟用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。這可讓Trident部署在Pod外部、但它只支援不安全的API伺服器連線。若要安全地連線、請將Trident部署在Pod中 `-k8s_pod` 選項。

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 `PersistentVolumeClaim` 正在申請新的 `PersistentVolume` Kubernetes的特定尺寸 `StorageClass` 這是先前由系統管理員設定的。
2. Kubernetes `StorageClass` 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會自行決定 `StorageClass` 使用識別相符項目的相同名稱 `Backends` 和 `StoragePools` 可用來為類別配置磁碟區。
4. Trident會在相符的後端上配置儲存設備、並建立兩個物件：a `PersistentVolume` Kubernetes告訴Kubernetes如何尋找、掛載及處理Volume、以及Trident中保留兩者關係的Volume `PersistentVolume` 以及實際儲存設備。
5. Kubernetes會連結 `PersistentVolumeClaim` 新的 `PersistentVolume`。包含的Pod `PersistentVolumeClaim` 將該PeristentVolume掛載到其執行的任何主機上。

6. 使用者會建立 `VolumeSnapshot` 現有的PVC,使用 `VolumeSnapshotClass` 這是Trident的重點。
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。也會建立 `VolumeSnapshotContent` 這會指示Kubernetes如何識別快照。
8. 使用者可以建立 `PersistentVolumeClaim` 使用 `VolumeSnapshot` 來源：
9. Trident會識別所需的快照、並執行建立所需的相同步驟集 `PersistentVolume` 和答 `Volume`。

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 、 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 彈性群組
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 、 gcp 、 CVS
trident.netapp.io/snapshotDirectory	Snapshot目錄	ONTAP NAS 、 ONTAP NAS 經濟型、 ONTAP NAS 彈性群組
trident.netapp.io/unixPermissions	unix權限	ONTAP NAS 、 ONTAP NAS 經濟型、 ONTAP NAS 彈性群組
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、以便在移除磁碟區之前、先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。請看一下 `On-Demand Volume Snapshots` 以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的PVc `mysql`、使用者可以建立名為的新永久虛擬環境 `mysqlclone` 使用註釋、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：

- 我們建議您複製閒置的Volume。
- 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、可能需要設定PVc註釋 `trident.netapp.io/splitOnClone` 與搭配使用 `trident.netapp.io/cloneFromPVC`。與 `trident.netapp.io/splitOnClone` 設定為 `true`、Trident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生命週期與父磁碟區完全分離、而犧牲部分儲存效率。未設定 `trident.netapp.io/splitOnClone` 或設定為 `false` 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、因此除非先刪除複本、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。

。 `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叢集可用的一部分儲存設備。它的生命週期與使用它的Pod無關。

Trident會建立 `PersistentVolume` 根據資源配置的磁碟區、自動在Kubernetes叢集中登錄物件。您不需要自行管理。

當您建立參照Trident型的PVc時 `StorageClass`、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 `StorageClass` 物件是以中的名稱來指定 `PersistentVolumeClaims` 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。

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

```
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	將後端名稱對應至清單的儲存資源池
其他StoragePools	map[stringList	否	後端名稱對應至中的儲存集區清單
排除StoragePools	map[stringList	否	後端名稱對應至中的儲存集區清單

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

屬性	類型	價值	優惠	申請	支援者
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
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、您可以使用 `attributes` 只有模型、才能建立儲存設備的品質、滿足特定類別的需求。Trident會自動探索並選取符合_all_的儲存集區 `attributes` 您指定的。

如果您發現自己無法使用 `attributes` 若要自動為類別選取適當的資源池、您可以使用 `storagePools` 和 `additionalStoragePools` 用於進一步精簡集區或甚至選取特定集區集區的參數。

您可以使用 `storagePools` 參數以進一步限制符合任何指定之集區的集合 `attributes`。換句話說、Trident

會使用由所識別的資源池交會 `attributes` 和 `storagePools` 資源配置參數。您可以單獨使用參數、也可以同時使用兩者。

您可以使用 `additionalStoragePools` 此參數可延伸Trident用於資源配置的集區集區集區集區集區集區、無論所選取的任何集區為何 `attributes` 和 `storagePools` 參數。

您可以使用 `excludeStoragePools` 篩選Trident用於資源配置的資源池集區集合的參數。使用此參數會移除任何相符的集區。

在中 `storagePools` 和 `additionalStoragePools` 參數、每個項目都採用格式 `<backend>:<storagePoolList>`、其中 `<storagePoolList>` 是指定後端的儲存資源池清單、以英文分隔。例如、的值 `additionalStoragePools` 看起來可能是這樣 `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` 以取得後端及其資源池清單。

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等	區塊的檔案系統類型 磁碟區	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	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` 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。以供使用 `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安裝程式套件提供數個範例儲存類別定義、可與中的Trident搭配使用 `sample-input/storage-class-*.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 `VolumeSnapshotClass` 物件類似 `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。

答 `VolumeSnapshotClass` 應由系統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` 指定要要求的Kubernetes磁碟區快照 `csi-snapclass` 類別由Trident處理。。 `deletionPolicy` 指定必須刪除快照時要採取的動作。何時 `deletionPolicy` 設為 `Delete`、刪除快照時、會移除儲存叢集上的Volume Snapshot物件及基礎快照。或者、將其設定為 `Retain` 也就是說 `VolumeSnapshotContent` 並保留實體快照。

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。

當磁碟區快照要求出現時、Trident會在後端自動管理磁碟區的快照建立、並建立唯一的快照來公開快照 `VolumeSnapshotContent` 物件：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` 物件代表從已配置的磁碟區擷取的快照。類似於 `PersistentVolume` 並表示儲存叢集上已配置的快照。類似 `PersistentVolumeClaim` 和 `PersistentVolume` 建立快照時的物件 `VolumeSnapshotContent` 物件會將一對一的對應維持在上 `VolumeSnapshot` 物件、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名稱與名稱的獨特組合 `VolumeSnapshotContent` 物件：

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` 然後將快照公開給Kubernetes API。

一般而言、您不需要管理 `VolumeSnapshotContent` 物件：這是您想要的例外情況 "匯入 Volume 快照" 在 Astra Trident 外部建立。

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。

Astra Trident的用途 `CustomResourceDefinition` 保留Trident物件身分的物件、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`、對應的 `tridentbackends` CRD物件是由Kubernetes所建立、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。

Astra Trident `StorageClass` 物件

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時、這些物件會在Kubernetes時自動建立 `StorageClass` 使用Trident做為資源配置程式的功能已登錄。

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Kubernetes時會建立這些部署 `StorageClass` 物件：

Astra 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：

如需如何建構這些物件的詳細資訊、請參閱 "設定後端"。

Astra Trident `StoragePool` 物件

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。

Astra Trident `Volume` 物件

Volume是資源配置的基本單位、包含NFS共用和iSCSI LUN等後端端點。在Kubernetes中、這些項目會直接對應至 `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

屬性	類型	必要	說明
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會產生 `internalName` 建立Volume時。這包括兩個步驟。首先、它會預先加上儲存前置詞（預設值之一 `trident` 或是後端組態中的前置字元）到磁碟區名稱、產生表單名稱 `<prefix>-<volume-name>`。然後、它會繼續清理名稱、取代後端不允許的字元。對於後端、它會以底線取代連字號（因此內部名稱會變成ONTAP `<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、作為資源配置的一部分
流程。

Astra Trident `Snapshot` 物件

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至 `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 `Snapshot` 物件包含下列內容：

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

屬性	類型	必要	說明
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時 `VolumeSnapshot` 物件要求已建立、Trident可在備份儲存系統上建立Snapshot物件。。 `internalName` 此快照物件的產生方式為結合前置詞 `snapshot-` 使用 `UID` 的 `VolumeSnapshot` 物件（例如、 `snapshot-e8d8a0ca-9826-11e9-9807-525400f3f660`）。 `volumeName` 和 `volumeInternalName` 會透過取得支援的詳細資料來填入資料
Volume：

Astra Trident `ResourceQuota` 物件

Trident去除會耗用a `system-node-critical` 優先級類別是Kubernetes中最高的優先級類別、可確保Astra Trident在正常節點關機期間識別並清理磁碟區、並允許Trident的取消安裝Pod在資源壓力較高的叢集中預先配置優先級較低的工作負載。

為了達成此目標、Astra Trident採用 `ResourceQuota` 確保在Trident取消程式集上達到「系統節點關鍵」優先順序類別的物件。在部署和建立實體化設定之前、Astra Trident會先尋找 `ResourceQuota` 物件、如果未探索到、則套用它。

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

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、安裝會在之後失敗 `ResourceQuota` 物件已建立、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 `ResourceQuota` 物件：

移除 ResourceQuota

如果您偏好控制自己的資源配置、可以移除Astra Trident `ResourceQuota` 使用命令的物件：

```
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、Astra Trident可在安裝期間啟用特定權限。以下各節詳細說明Astra Trident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、並使用主機網路與iSCSI精靈進行通訊。
主機iPC	NFS使用程序間通訊（IPC）與nfsd通訊。
主機 PID	必須啟動 `rpc-statd` NFS：Astra Trident會查詢主機程序以判斷是否 `rpc-statd` 在掛載NFS磁碟區之前執行。

功能	。 `SYS_ADMIN` 此功能是專為特殊權限容器提供的預設功能之一。例如、Docker會針對特殊權限容器設定下列功能：
	`CapPrm: 0000003fffffffff`
	`CapEff: 0000003fffffffff`

Seccomp	Seccomp設定檔在特殊權限容器中一律為「未限制」、因此無法在Astra Trident中啟用。
SELinux	在OpenShift上、有權限的容器會在中執行 `spc_t` （「超級權限容器」）網域和無權限容器會在中執行 `container_t` 網域：開啟 `containerd`、搭配 `container-selinux` 安裝後、所有的容器都會在中執行 `spc_t` 網域、有效停用SELinux。因此、Astra Trident並未新增 `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、請檢查命名空間標籤是否適用於 `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
`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
`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項目 `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`	否 `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`

欄位	說明	預設
hostPID	需要主機PID才能檢查是否有 rpc-statd 正在節點上執行。	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	未設定Trident seLinuxOptions 因為目前容器執行時間與Kubernetes發行版本處理SELinux的方式有差異。	空白
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才能與進行通訊 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才能檢查是否有 rpc-statd 正在節點上執行。	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項目 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

標籤	說明	預設
`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`	未設定Trident `seLinuxOptions` 因為目前容器執行時間與Kubernetes發行版本處理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
`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`