最佳實務做法與建議
Trident

NetApp
March 05, 2026

目錄

最佳實務做法與建議

部署

部署 Trident 時、請使用此處列出的建議。

部署至專屬命名空間

"命名空間"在不同的應用程式之間提供管理上的分離、是資源共用的障礙。例如、某個命名空間的某個永久虛電路無法從另一個命名空間使用。Trident 為 Kubernetes 叢集中的所有命名空間提供 PV 資源、因此會運用 Privileges 提升的服務帳戶。

此外、存取Trident Pod可能會讓使用者存取儲存系統認證和其他敏感資訊。請務必確保應用程式使用者和管理應用程式無法存取Trident物件定義或Pod本身。

使用配額和範圍限制來控制儲存使用量

Kubernetes有兩項功能、一旦結合、就能提供強大的機制來限制應用程式的資源使用量。 "儲存配額機制"可讓系統管理員以每個命名空間為基礎，實作全域及特定儲存類別、容量及物件數使用量限制。此外、使用 "範圍限制" 可確保在將要求轉送至資源配置程式之前、 PVC 要求的最小值和最大值都在內。

這些值是以每個命名空間為基礎來定義、這表示每個命名空間都應該定義符合其資源需求的值。如需相關資訊、請參閱此處 "如何運用配額"。

儲存組態

NetApp產品組合中的每個儲存平台都有獨特的功能、無論應用程式是否為容器化的應用程式帶來好處。

平台總覽

Trident可搭配ONTAP 使用沒有一個平台比其他平台更適合所有應用程式和案例、不過在選擇平台時、應考慮應用程式和管理裝置團隊的需求。

您應該遵循主機作業系統的基礎最佳實務做法、以及您所使用的傳輸協定。或者、您可能想要考慮在可用的情況下、將應用程式最佳實務做法與後端、儲存類別和永久虛擬基礎架構設定整合、以最佳化特定應用程式的儲存。

最佳實務做法**ONTAP Cloud Volumes ONTAP**

瞭解設定ONTAP 適用於Cloud Volumes ONTAP Trident的功能性及功能性的最佳實務做法。

以下建議是設定ONTAP 以容器化工作負載為基礎的功能指南、這些功能會消耗Trident動態配置的磁碟區。每個項目都應考量及評估是否適合您的環境。

使用**Trident**專用的**SVM**

儲存虛擬機器（SVM）可隔離ONTAP 及管理各個客戶在一個系統上的區隔。將SVM專用於應用程式可委派權限、並可套用最佳實務做法來限制資源使用量。

SVM管理有多種選項可供選擇：

- 在後端組態中提供叢集管理介面、以及適當的認證、然後指定SVM名稱。
- 使用ONTAP 支援功能的支援中心或CLI、為SVM建立專屬的管理介面。
- 與NFS資料介面共用管理角色。

在每種情況下、介面都應該位於DNS中、而且在設定Trident時、應該使用DNS名稱。這有助於推動一些DR案例、例如不使用網路身分保留功能的SVM-DR。

不過、您並不偏好為SVM設定專屬或共享的管理LIF、不過您應該確保網路安全性原則符合您選擇的方法。無論如何、管理 LIF 應可透過 DNS 存取、以提供最大的靈活度 "SVM-DR"、並搭配 Trident 一起使用。

限制最大**Volume**數

根據軟體版本和硬體平台、系統可提供最大的Volume數。ONTAP請參閱 "NetApp Hardware Universe"以瞭解您的特定平台和 ONTAP 版本、以確定確切的限制。當磁碟區數用盡時、資源配置作業不僅會針對Trident、也會針對所有儲存要求失敗。

Trident `ontap-nas` 和 `ontap-san`驅動程式會為每個建立的 Kubernetes 持續 Volume （ PV ）提供 FlexVolume 。 `ontap-nas-economy`驅動程式每 200 部 PV 建立一個 FlexVolume （可設定在 50 到 300 之間）。 `ontap-san-economy`驅動程式每 100 部 PV 建立一個 FlexVolume （可設定在 50 到 200 之間）。若要避免Trident佔用儲存系統上的所有可用磁碟區、您應該在SVM上設定限制。您可以從命令列執行此動作：

```
vserver modify -vserver <svm_name> -max-volumes <num_of_volumes>
```

的值 `max-volumes`會因環境特定的幾個條件而異：

- 在叢集中現有的Volume數量ONTAP
- 您預期在Trident外部配置其他應用程式的Volume數量
- Kubernetes應用程式預期會使用的持續磁碟區數量

此 `max-volumes`值是在 ONTAP 叢集中所有節點上、而非在個別 ONTAP 節點上、配置的總磁碟區。因此ONTAP 、您可能會遇到一些情況、例如、某個叢集節點的資源配置量可能遠高於或低於其他節點。

例如，雙節點 ONTAP 叢集最多可裝載 2000 個 FlexVol 磁碟區。將最大Volume數設為1250似乎非常合理。不過、如果只有 "集合體"一個節點指派給 SVM 、或是無法針對一個節點指派的集合體進行資源配置（例如、由於容量）、則另一個節點會成為所有 Trident 資源配置 Volume 的目標。這表示在達到該值之前、可能會達到該節點的磁碟區限制 `max-volumes`、進而影響使用該節點的 Trident 和其他磁碟區作業。您可以確保叢集中每個節點的集合體都指派給**Trident**使用的**SVM**、數量相等、藉此避免這種情況。

限制**Trident**所建立的**Volume**大小上限

若要設定可由 Trident 建立的磁碟區大小上限、請使用 `limitVolumeSize`**定義中的參數** `backend.json`。

除了控制儲存陣列的磁碟區大小、您也應該善用Kubernetes功能。

限制 **Trident** 所建立的 **FlexVols** 大小上限

若要將 FlexVols 的最大大小設定為用於 ONTAP SAN 經濟型和 ONTAP NAS 經濟型驅動程式的集區、請使用 `limitVolumePoolSize` `backend.json`定義中的參數。

設定**Trident**使用雙向**CHAP**

您可以在後端定義中指定CHAP啟動器和目標使用者名稱和密碼、並在SVM上啟用Trident啟用CHAP。使用 `useCHAP`後端組態中的參數、 Trident 會使用 CHAP 驗證 ONTAP 後端的 iSCSI 連線。

建立並使用**SVM QoS**原則

運用ONTAP 套用至SVM的SVM的SQoS原則、限制Trident佈建磁碟區所耗用的IOPS數量。這有助於 "預防欺凌" 或無法控制容器、避免影響 Trident SVM 以外的工作負載。

您可以在幾個步驟中建立SVM的QoS原則。如ONTAP 需最準確的資訊、請參閱您的版次更新文件。以下範例建立QoS原則、將SVM可用的總IOPS限制為5000。

```
# create the policy group for the SVM
qos policy-group create -policy-group <policy_name> -vserver <svm_name>
-max-throughput 5000iops

# assign the policy group to the SVM, note this will not work
# if volumes or files in the SVM have existing QoS policies
vserver modify -vserver <svm_name> -qos-policy-group <policy_name>
```

此外、如果ONTAP 您的版本支援此功能、您可以考慮使用QoS下限來保證容器化工作負載的處理量。調適性QoS與SVM層級原則不相容。

容器化工作負載專用的IOPS數量取決於許多層面。其中包括：

- 使用儲存陣列的其他工作負載。如果有其他工作負載與Kubernetes部署無關、請善用儲存資源、確保這些工作負載不會意外受到不良影響。
- 預期的工作負載會在容器中執行。如果將在容器中執行高IOPS需求的工作負載、低QoS原則會導致不良體驗。

請務必記住、在SVM層級指派的QoS原則會導致所有已配置給SVM的磁碟區共用相同的IOPS集區。如果其中一種或少數幾種容器化應用程式的IOPS需求較高、可能會成為其他容器化工作負載的一大功臣。如果是這種情況、您可能需要考慮使用外部自動化來指派每個Volume QoS原則。

如果您的版本早於ONTAP 9.8、您應該將QoS原則群組指派給SVM * Only *。

為**Trident**建立**QoS**原則群組

服務品質（QoS）可確保關鍵工作負載的效能不會因競爭工作負載而降級。支援QoS原則群組的QoS選項可用於磁碟區、並可讓使用者定義一或多個工作負載的處理量上限。ONTAP如需 QoS 的詳細資訊、請 "保證QoS的處理量"參閱。您可以在後端或儲存資源池中指定QoS原則群組、並將其套用至該資源池或後端中建立的每個磁碟區。

包含兩種QoS原則群組：傳統和可調適。ONTAP傳統原則群組可在IOPS中提供最大（或最小）的單位處理量（在較新版本中）。調適性QoS會自動將處理量調整至工作負載大小、並隨著工作負載大小變更、維持IOPS與TBs的比率。當您在大型部署中管理數百個或數千個工作負載時、這項優勢就相當顯著。

建立QoS原則群組時、請考量下列事項：

- 您應該在後端組態區塊中設定 `qosPolicy`**金鑰** `defaults`。請參閱下列後端組態範例：

```
---
version: 1
storageDriverName: ontap-nas
managementLIF: 0.0.0.0
dataLIF: 0.0.0.0
svm: svm0
username: user
password: pass
defaults:
  qosPolicy: standard-pg
storage:
  - labels:
      performance: extreme
    defaults:
      adaptiveQosPolicy: extremely-adaptive-pg
  - labels:
      performance: premium
    defaults:
      qosPolicy: premium-pg
```

- 您應該為每個Volume套用原則群組、以便每個Volume都能獲得原則群組指定的整個處理量。不支援共用原則群組。

如需 QoS 原則群組的詳細資訊，請 "ONTAP 命令參照"參閱。

限制**Kubernetes**叢集成員存取儲存資源

限制對 Trident 所建立的 NFS 磁碟區， iSCSI LUN 和 FC LUN 的存取，是 Kubernetes 部署安全性態勢的重要元件。這樣做可防止非Kubernetes叢集一部分的主機存取磁碟區、並可能意外修改資料。

請務必瞭解命名空間是Kubernetes中資源的邏輯邊界。假設相同命名空間中的資源可以共用、但重要的是、沒有跨命名空間功能。這表示即使PV是全域物件、但只有在同一個命名空間中的Pod才能存取它們。確保命名空間在適當時用於提供分隔是非常重要的。

大多數組織對於Kubernetes內容中的資料安全性、主要關注的是、容器中的程序可以存取掛載到主機的儲存設備、但不適用於容器。 "命名空間"旨在防止此類入侵。不過、有一個例外：特殊權限容器。

與正常情況相比、特權容器的執行主機層級權限大幅增加。默認情況下不會拒絕這些功能，因此請確保使用禁用此功能 "Pod安全性原則"。

對於需要從Kubernetes和外部主機存取的磁碟區、儲存設備應以傳統方式進行管理、由系統管理員引進PV、而

非由Trident管理。這可確保只有在Kubernetes和外部主機中斷連線且不再使用磁碟區時、才會銷毀儲存磁碟區。此外、也可以套用自訂匯出原則、以便從Kubernetes叢集節點和Kubernetes叢集以外的目標伺服器存取。

對於具有專用基礎架構節點（例如OpenShift）或其他節點無法排程使用者應用程式的部署、應使用個別的匯出原則、進一步限制對儲存資源的存取。這包括為部署至這些基礎架構節點的服務（例如OpenShift Metrics和記錄服務）、以及部署至非基礎架構節點的標準應用程式建立匯出原則。

使用專屬的匯出原則

您應該確保每個後端都有一個匯出原則、只允許存取Kubernetes叢集中的節點。Trident 可以自動建立及管理匯出原則。如此一來、Trident就能限制對Kubernetes叢集中節點所配置之磁碟區的存取、並簡化節點的新增/刪除作業。

或者、您也可以手動建立匯出原則、並以一或多個匯出規則填入、以處理每個節點存取要求：

- 使用 `vserver export-policy create` ONTAP CLI 命令建立匯出原則。
- 使用 ONTAP CLI 命令將規則新增至匯出原則 `vserver export-policy rule create`。

執行這些命令可讓您限制哪些Kubernetes節點可以存取資料。

停用 **`showmount`**應用程式 **SVM**

此 `showmount`功能可讓 NFS 用戶端查詢 SVM 以取得可用 NFS 匯出清單。部署至 Kubernetes 叢集的 Pod 可針對發出 `showmount -e`命令，並接收可用掛載清單，包括無法存取的掛載。雖然這本身並不是安全威脅、但它確實提供不必要的資訊、可能有助於未獲授權的使用者連線至NFS匯出。

您應該使用 SVM 層級的 ONTAP CLI 命令來停用 `showmount`：

```
vserver nfs modify -vserver <svm_name> -showmount disabled
```

最佳實務做法**SolidFire**

瞭解設定SolidFire Trident之用的功能完善的功能。

建立**SolidFire** 支援帳戶

每SolidFire 個驗證帳戶都代表唯一的磁碟區擁有者、並會收到自己的挑戰握手驗證傳輸協定（CHAP）認證資料。您可以使用帳戶名稱和相對CHAP認證、或是透過Volume存取群組、來存取指派給帳戶的磁碟區。帳戶最多可指派2、000個磁碟區、但一個磁碟區只能屬於一個帳戶。

建立**QoS**原則

如果您想建立並儲存可套用至許多Volume的標準化服務品質設定、請使用SolidFire 「服務品質（QoS）」原則。

您可以設定每個Volume的QoS參數。設定三個可設定的參數來定義QoS、以確保每個Volume的效能：最小IOPS、最大IOPS和爆發IOPS。

以下是4KB區塊大小的可能最小、最大和尖峰IOPS值。

IOPS 參數	定義	最小值	預設值	最大值（4KB）
最小IOPS	保證磁碟區效能等級。	50	50	15000
最大IOPS	效能不會超過此限制。	50	15000	20 萬
暴增IOPS	在短時間暴增案例中允許的最大IOPS。	50	15000	20 萬

雖然最大IOPS和爆發IOPS可設定為高達20、000、但實際的Volume最大效能卻受到叢集使用量和每節點效能的限制。

區塊大小和頻寬會直接影響IOPS的數量。隨著區塊大小增加、系統會將頻寬增加至處理較大區塊大小所需的層級。隨著頻寬增加、系統能夠達到的IOPS數量也隨之減少。如需 QoS 和效能的詳細資訊、請參閱 "服務品質SolidFire"。

驗證SolidFire

Element支援兩種驗證方法：CHAP和Volume Access Groups（VAG）。CHAP使用CHAP傳輸協定驗證主機到後端的驗證。Volume存取群組可控制對其所配置之Volume的存取。NetApp建議使用CHAP進行驗證、因為它更簡單、而且沒有擴充限制。

Trident搭配增強的csi佈置程式、可支援使用CHAP驗證。VAG只能在傳統的非csi操作模式下使用。

CHAP驗證（驗證啟動器是否為預定的Volume使用者）僅支援帳戶型存取控制。如果您使用CHAP進行驗證、則有兩個選項可供使用：單向CHAP和雙向CHAP。單向CHAP使用SolidFire 驗證帳戶名稱和啟動器密碼來驗證Volume存取。雙向CHAP選項提供最安全的驗證磁碟區方法、因為磁碟區會透過帳戶名稱和啟動器密碼來驗證主機、然後主機會透過帳戶名稱和目標密碼來驗證磁碟區。

但是、如果無法啟用CHAP且需要VAG、請建立存取群組、然後將主機啟動器和磁碟區新增至存取群組。您新增至存取群組的每個IQN都可以使用或不使用CHAP驗證來存取群組中的每個磁碟區。如果iSCSI啟動器設定為使用CHAP驗證、則會使用帳戶型存取控制。如果iSCSI啟動器未設定為使用CHAP驗證、則會使用Volume Access Group存取控制。

哪裡可以找到更多資訊？

以下列出部分最佳實務做法文件。搜尋 "NetApp資料庫"最新版本。

《》ONTAP

- "NFS最佳實務與實作指南"
- "SAN 管理"（適用於 iSCSI ）
- "適用於RHEL的iSCSI Express組態"

元件軟體

- "設定SolidFire 適用於Linux的功能"
- NetApp HCI *
- "部署先決條件NetApp HCI"
- "存取NetApp部署引擎"

應用程式最佳實務做法資訊

- "MySQL ONTAP 的最佳實務做法"
- "MySQL SolidFire 的最佳實務做法"
- "NetApp SolidFire 的功能與Cassandra"
- "Oracle SolidFire 的最佳實務做法"
- "PostgreSQL SolidFire 的最佳實務做法"

並非所有應用程式都有特定準則、請務必與 NetApp 團隊合作、並使用 "NetApp資料庫"來尋找最新的文件。

整合 Trident

若要整合 Trident 、下列設計和架構元素需要整合：驅動程式選擇和部署、儲存類別設計、虛擬集區設計、持續 Volume Claim （永久 Volume Claim ）對使用 Trident 的儲存資源配置、 Volume 作業和 OpenShift 服務部署的影響。

驅動程式選擇與部署

為您的儲存系統選取並部署後端驅動程式。

背後驅動程式**ONTAP**

以使用的傳輸協定和儲存系統上的磁碟區配置方式來區分後端驅動程式ONTAP 。因此、在決定要部署的驅動程式時、請謹慎考量。

較高層級的應用程式若有需要共用儲存設備的元件（多個Pod存取相同的PVc）、則以NAS為基礎的驅動程式將是預設選擇、而區塊型iSCSI驅動程式則可滿足非共用儲存設備的需求。根據應用程式的需求、以及儲存設備和基礎架構團隊的舒適度來選擇傳輸協定。一般而言、大多數應用程式的差異不大、因此通常是根據是否需要共用儲存設備（如果有多個Pod需要同時存取）來決定。

可用ONTAP 的支援功能包括：

- `ontap-nas`：配置的每個 PV 都是完整的 ONTAP FlexVolume 。
- `ontap-nas-economy`：配置的每個 PV 都是 qtree ，每個 FlexVolume 都有可設定的 qtree 數量（預設值為 200 ）。
- `ontap-nas-flexgroup`：每個 PV 配置爲完整 ONTAP FlexGroup ，並使用分配給 SVM 的所有聚合。
- `ontap-san`：配置的每個 PV 都是其自身 FlexVolume 內的 LUN 。
- `ontap-san-economy`：每個配置的 PV 都是一個 LUN ，每個 FlexVolume 具有可配置的 LUN 數量（默認值爲 100 ）。

在這三種NAS驅動程式之間選擇、會對應用程式可用的功能產生一些影響。

請注意、在下表中、並非所有功能都是透過 Trident 公開的。如果需要這些功能、儲存管理員必須在資源配置後套用部分功能。上標註可區分每項功能和驅動程式的功能。

ONTAP NAS 驅動程式	快照	複製	動態匯出原則	多重附加	**QoS**	調整大小	複寫
`ontap-nas`	是的	是的	是註腳：5[]	是的	是註腳：1[]	是的	是註腳：1[]
`ontap-nas-economy`	NO腳 註：3[]	NO腳 註：3[]	是註腳：5[]	是的	NO腳 註：3[]	是的	NO腳 註：3[]
`ontap-nas-flexgroup`	是註腳：1[]	否	是註腳：5[]	是的	是註腳：1[]	是的	是註腳：1[]

Trident 提供 2 個適用於 ONTAP 的 SAN 驅動程式、其功能如下所示。

ONTAP SAN 驅動程式	快照	複製	多重附加	雙向**CHAP**	**QoS**	調整大小	複寫
`ontap-san`	是的	是的	是註腳：4[]	是的	是註腳：1[]	是的	是註腳：1[]
`ontap-san-economy`	是的	是的	是註腳：4[]	是的	NO腳 註：3[]	是的	NO腳 註：3[]

> 上表的註腳： Yesporte:1[] ：非由 Trident 管理 Yesporte:2[] ：由 Trident 管理，但非 PV 精細 NO腳 註 :3[] ：非由 Trident 管理，非 PV 精細腳註： 4[] ：支援原始區塊磁碟區 Yesport:5[] ：由 Trident 支援

非PV精細的功能會套用至整個FlexVolume、而所有PV（即共享FlexVols中的qtree或LUN）都會共用一個共同排程。

如上表所示、和 `ontap-nas-economy`之間的大部分功能都是 `ontap-nas`相同的。不過、由於驅動程式限制了以每 `PV` 精細度控制排程的能力、因此 `ontap-nas-economy`這特別會影響您的災難恢復和備份規劃。對於想要在 `ONTAP` 儲存設備上使用 `PVC` 複製功能的開發團隊、這只有在使用、 `ontap-san`或 `ontap-san-economy`驅動程式時才可行 `ontap-nas`。

> 此 `solidfire-san`驅動程式也能複製 PVCS 。

背後驅動程式**Cloud Volumes ONTAP**

支援資料控管功能、並提供企業級的儲存功能、適用於各種使用案例、包括檔案共用、區塊層級儲存設備（NFS、SMB / CIFS及iSCSI）Cloud Volumes ONTAP 。Cloud Volume ONTAP 的相容驅動程式包括 `ontap-nas`、 `ontap-nas-economy` `ontap-san`和 `ontap-san-economy`。適用於ONTAP Azure的Cloud Volume供應、適用於ONTAP GCP的Cloud Volume供應。

Amazon FSXfor ONTAP Sendbackend驅動程式

Amazon FSX for NetApp ONTAP 可讓您運用熟悉的 NetApp 功能、效能和管理功能、同時充分利用在 AWS 上儲存資料的簡易性、敏捷度、安全性和擴充性。適用於 ONTAP 的 FSX 支援許多 ONTAP 檔案系統功能和管理

API 。Cloud Volume ONTAP 的相容驅動程式有 `ontap-nas`、 `ontap-nas-economy`、 `ontap-nas-flexgroup` `ontap-san`和 `ontap-san-economy`。

NetApp HCI / SolidFire後端驅動程式

`solidfire-san`搭配 NetApp HCI / SolidFire 平台使用的驅動程式可協助管理員根據 QoS 限制、為 Trident 設定元素後端。如果您想設計後端來設定 Trident 所佈建之磁碟區的特定 QoS 限制、請使用 `type`後端檔案中的參數。管理員也可以使用參數來限制可在儲存設備上建立的磁碟區大小 `limitVolumeSize`。目前、磁碟區大小調整和磁碟區複寫等元素儲存功能不支援透過 `solidfire-san`驅動程式。這些作業應透過Element Software Web UI手動完成。

驅動程式**SolidFire**	快照	複製	多重附加	**CHAP**	**QoS**	調整大小	複寫
solidfire-san	是的	是的	是註腳：2[]	是的	是的	是的	是註腳：1[]

> 註腳：是註腳： 1[] ：非由 Trident 管理是註腳： 2[] ：支援原始區塊磁碟區

背後驅動程式**Azure NetApp Files**

Trident 使用 `azure-netapp-files`驅動程式來管理"Azure NetApp Files"服務。

有關此驅動程式及其設定方式"適用於 Azure NetApp Files 的 Trident 後端組態"的詳細資訊，請參閱。

驅動程式**Azure NetApp Files**	快照	複製	多重附加	**QoS**	展開	複寫
azure-netapp-files	是的	是的	是的	是的	是的	是註腳： 1[]

> 註腳： Yes腳 註： 1[] ：非由 Trident 管理

在**Google Cloud**後端驅動程式上執行**Cloud Volumes Service**

Trident 使用 `gcp-cvs`驅動程式連結 Google Cloud 上的 Cloud Volumes Service 。

驅動程式會 `gcp-cvs`使用虛擬集區來抽象化後端、並允許 Trident 判斷磁碟區的放置位置。系統管理員會定義檔案中的虛擬集區 `backend.json`。儲存類別會使用選取器來依標籤識別虛擬資源池。

- 如果在後端定義虛擬集區、 Trident 將嘗試在 Google Cloud 儲存池中建立一個磁碟區、而這些虛擬集區則受到限制。
- 如果未在後端定義虛擬集區、 Trident 會從該區域的可用儲存集區中選取 Google Cloud 儲存集區。

若要在 Trident 上設定 Google Cloud 後端、您必須在後端檔案中指定 `projectNumber`、 `apiRegion`和 `apiKey`。您可以在Google Cloud主控台找到專案編號。API金鑰取自您在Google Cloud Volumes Service Cloud上設定API存取功能時所建立的服務帳戶私密金鑰檔案。

如需 Cloud Volumes Service on Google Cloud 服務類型和服務層級的詳細資訊"瞭解 Trident 對 CVS for GCP

的支援"、請參閱。

適用於Google Cloud驅動程式Cloud Volumes Service	快照	複製	多重附加	QoS	展開	複寫
gcp-cvs	是的	是的	是的	是的	是的	僅適用於CVS效能服務類型。

複寫附註

- 複寫並非由 Trident 管理。
- 該實體複本會建立在與來源Volume相同的儲存資源池中。

儲存層級設計

需要設定並套用個別的儲存類別、才能建立Kubernetes儲存類別物件。本節將討論如何為應用程式設計儲存類別。

特定後端使用率

篩選功能可在特定的儲存類別物件內使用、以決定要搭配該特定儲存類別使用的儲存資源池或集區集區集區。可以在 Storage Class （存儲類）中設置三組篩選器： `storagePools`、 `additionalStoragePools`和（或 `excludeStoragePools`）。

此 `storagePools`參數有助於將儲存限制為符合任何指定屬性的集區集。此 `additionalStoragePools`參數用於擴充 Trident 用於資源配置的集區集區集、以及由屬性和參數所選取的集區集 `storagePools`。您可以單獨使用參數或同時使用兩者、以確保已選取適當的儲存資源池集區集區。

此 `excludeStoragePools`參數用於明確排除列出的一組符合屬性的集區。

模擬QoS原則

如果您想設計儲存類別來模擬服務品質原則、請建立屬性為 `hdd`或 `ssd`的儲存類別 `media`。根據 `media`儲存類別中提及的屬性、 Trident 會選取適當的後端來提供 `hdd`或 `ssd`集合以符合媒體屬性、然後將磁碟區的資源配置導向特定的集合體。因此、我們可以建立儲存類別 Premium 、將 `media`屬性設定為 `ssd`可歸類為 Premium QoS 原則。我們可以建立另一個儲存類別標準、將媒體屬性設為「HDD」、並將其歸類為標準QoS原則。我們也可以使用儲存類別中的「IOPS」屬性、將資源配置重新導向至可定義為QoS原則的元素應用裝置。

根據特定功能使用後端

儲存類別可設計用於將Volume資源配置導向特定後端、啟用精簡與完整資源配置、快照、複製及加密等功能。若要指定要使用的儲存設備、請建立儲存設備類別、以指定啟用所需功能的適當後端。

虛擬資源池

所有 Trident 後端均可使用虛擬集區。您可以使用 Trident 提供的任何驅動程式、為任何後端定義虛擬集區。

虛擬集區可讓系統管理員在後端建立抽象層級、以便透過「儲存類別」加以參考、以提高磁碟區在後端的靈活度與效率。不同的後端可以使用相同的服務類別來定義。此外、您也可以在相同的後端上建立多個儲存資源池、但

其特性不同。當儲存類別設定為具有特定標籤的選取器時、 Trident 會選擇符合所有選取器標籤的後端來放置磁碟區。如果儲存類別選取器標籤符合多個儲存集區、 Trident 將會選擇其中一個標籤來配置磁碟區。

虛擬資源池設計

建立後端時、您通常可以指定一組參數。系統管理員無法以相同的儲存認證和一組不同的參數來建立另一個後端。隨著虛擬資源池的推出、這個問題已經減輕。虛擬集區是後端與Kubernetes儲存類別之間的層級抽象、可讓系統管理員定義參數及標籤、並以不受後端限制的方式透過Kubernetes儲存類別做為選取元來參考。您可以使用 Trident 為所有支援的 NetApp 後端定義虛擬集區。這份清單包括SolidFire/NetApp HCI、ONTAP 《關於Cloud Volumes Service GCP的功能、功能、功能、功能Azure NetApp Files 、功能、以及

> 定義虛擬資源池時、建議您不要嘗試重新排列後端定義中現有虛擬資源池的順序。此外、建議您不要編輯/修改現有虛擬資源池的屬性、改為定義新的虛擬資源池。

模擬不同的服務層級/QoS

您可以設計虛擬集區來模擬服務類別。使用適用於Azure NetApp Files 支援功能的Cloud Volume Service for效益的虛擬資源池實作、讓我們來看看如何設定不同的服務類別。使用代表不同效能層級的多個標籤來設定 Azure NetApp Files 後端。將 Aspect 設 `servicelevel`為適當的效能層級、並在每個標籤下新增其他必要的層面。現在請建立不同的Kubernetes儲存類別、以便對應至不同的虛擬資源池。使用此 `parameters.selector`欄位、每個 StorageClass 都會呼叫哪些虛擬集區可用於主控磁碟區。

指派特定的層面組合

可從單一儲存後端設計多個具有特定層面的虛擬集區。若要這麼做、請使用多個標籤來設定後端、並在每個標籤下設定所需的層面。現在請使用對應至不同虛擬集區的欄位、建立不同的 Kubernetes 儲存類別 `parameters.selector`。在後端上進行資源配置的磁碟區、將會在所選的虛擬資源池中定義各個層面。

會影響儲存資源配置的永久儲存設備特性

建立 PVC 時、超出所要求儲存類別的部分參數可能會影響 Trident 資源配置決策程序。

存取模式

透過永久虛擬網路申請儲存時、其中一個必填欄位是存取模式。所需的模式可能會影響所選的後端、以裝載儲存要求。

Trident 將嘗試將使用的儲存傳輸協定與根據下列對照表所指定的存取方法配對。這與基礎儲存平台無關。

	ReadWriteOnce	ReadOnlyMany	ReadWriteMany
iSCSI	是的	是的	是（原始區塊）
NFS	是的	是的	是的

如果要求將ReadWriteMany永久虛擬磁碟提交至Trident部署、但未設定NFS後端、則不會配置任何磁碟區。因此、申請者應使用適合其應用程式的存取模式。

Volume作業

修改持續磁碟區

持續磁碟區除了兩個例外、都是Kubernetes中不可變的物件。建立後、即可修改回收原則和大小。不過、這並不會妨礙磁碟區的某些層面在 Kubernetes 之外進行修改。這可能是理想的做法、以便針對特定應用程式自訂磁碟區、確保容量不會意外耗用、或是單純地將磁碟區移至不同的儲存控制器。

> Kubernetes 樹內置備程式目前不支援 NFS ， iSCSI 或 FC PV 的 Volume resize 作業。Trident 支援擴充 NFS ， iSCSI 和 FC 磁碟區。

PV的連線詳細資料無法在建立後修改。

建立隨需磁碟區快照

Trident 支援隨需建立磁碟區快照、以及使用 CSI 架構從快照建立 PVC 。Snapshot提供便利的方法來維護資料的時間點複本、並使Kubernetes中的來源PV在生命週期上獨立不受影響。這些快照可用於複製PVCS。

從快照建立磁碟區

Trident 也支援從磁碟區快照建立 PersistentVolumes 。若要達成此目標、只要建立 PersistentVolume Claim 、並將提及作為建立磁碟區所需的快照即可 `datasource`。Trident 會建立一個含有快照資料的磁碟區來處理此 PVC 。有了這項功能、您可以跨區域複製資料、建立測試環境、完整取代毀損或毀損的正式作業磁碟區、或擷取特定檔案和目錄、然後將它們傳輸到其他附加磁碟區。

在叢集中移動磁碟區

儲存管理員能夠在ONTAP 整個叢集中的集合體和控制器之間、不中斷營運地將磁碟區移至儲存使用者。只要目的地 Aggregate 是 Trident 使用的 SVM 具有存取權、此作業就不會影響 Trident 或 Kubernetes 叢集。重要的是、如果新增 Aggregate 至 SVM 、則需要重新將後端新增至 Trident 以重新整理。這會觸發 Trident 重新清查 SVM 、以便辨識新的 Aggregate 。

不過、 Trident 並不自動支援在後端之間移動磁碟區。這包括在同一個叢集中的 SVM 之間、叢集之間或不同的儲存平台上（即使該儲存系統是連線至 Trident 的儲存系統）。

如果將磁碟區複製到其他位置、則可使用 Volume 匯入功能將目前的磁碟區匯入 Trident 。

展開**Volume**

Trident 支援調整 NFS ， iSCSI 和 FC PV 的大小。這可讓使用者透過Kubernetes層直接調整磁碟區大小。所有主要的NetApp儲存平台皆可進行Volume擴充、包括ONTAP ：NetApp、SolidFire/NetApp HCI及Cloud Volumes Service 背後端點。若要稍後允許擴充、請在與該磁碟區相關的 StorageClass 中設定 `allowVolumeExpansion`為 `true`。每當需要調整「持續 Volume 」的大小時、請將「持續 Volume 」宣告中的註釋編輯 `spec.resources.requests.storage`為所需的 Volume 大小。Trident會自動調整儲存叢集上的磁碟區大小。

將現有磁碟區匯入**Kubernetes**

Volume匯入功能可將現有的儲存磁碟區匯入Kubernetes環境。目前、、 `ontap-nas-flexgroup` `solidfire-san`、 `azure-netapp-files`和 `gcp-cvs`驅動程式都支援這項 `ontap-nas`功能。當將現有應用程式移轉至Kubernetes或發生災難恢復時、此功能非常實用。

使用 ONTAP 和 `solidfire-san`驅動程式時、請使用命令 `tridentctl import volume <backend-name> <volume-name> -f /path/pvc.yaml`將現有的磁碟區匯入 Kubernetes 、以便由 Trident 管理。匯入 Volume 命令中使用的

PVC YAML 或 JSON 檔案會指向將 Trident 識別為資源配置程式的儲存類別。使用NetApp HCI / SolidFire後端時、請確定磁碟區名稱是唯一的。如果磁碟區名稱重複、請將磁碟區複製成唯一名稱、以便磁碟區匯入功能能夠區分它們。

如果 `azure-netapp-files`使用或 `gcp-cvs`驅動程式、請使用命令 `tridentctl import volume <backend-name> <volume path> -f /path/pvc.yaml`將磁碟區匯入 Kubernetes 、以便由 Trident 管理。如此可確保唯一的Volume參考。

執行上述命令時、 Trident 會在後端找到該 Volume 並讀取其大小。它會自動新增（並在必要時覆寫）已設定的 PVC Volume Size 。然後 Trident 建立新的 PV 、 Kubernetes 會將 PVC 與 PV 連結起來。

如果部署的容器需要特定匯入的PVc、則會保持擱置狀態、直到PVC/PV配對透過Volume匯入程序繫結為止。在PVC/PV配對繫結之後、如果沒有其他問題、則應啟動容器。

登錄服務

有關登錄的儲存設備部署與管理"部落格"，請參閱"NetApp.IO"。

記錄服務

如同其他 OpenShift 服務、記錄服務是使用 Ansible 搭配庫存檔案（即主機）所提供的組態參數來部署、這些主機是提供給教戰手冊的。其中包括兩種安裝方法：在初始OpenShift安裝期間部署記錄、以及在安裝OpenShift之後部署記錄。

> 從Red Hat OpenShift版本3.9起、官方文件建議您不要使用NFS來執行記錄服務、因為您擔心資料毀損。這是以Red Hat測試其產品為基礎。ONTAP NFS 伺服器沒有這些問題、而且可以輕鬆地備份記錄部署。最後、記錄服務的通訊協定選擇取決於您、只要知道兩者在使用NetApp平台時都能順利運作、而且如果您偏好NFS、就沒有理由不使用NFS。

如果您選擇搭配記錄服務使用 NFS 、則必須設定 Ansible 變數 `openshift_enable_unsupported_configurations`、以 `true`防止安裝程式發生故障。

開始使用

記錄服務可選擇性地同時部署給應用程式、以及OpenShift叢集本身的核心作業。如果您選擇部署作業記錄、請將變數指定 `openshift_logging_use_ops`為 `true`、將會建立兩個服務執行個體。控制作業記錄執行個體的變數包含「ops」、而應用程式執行個體則不包含。

根據部署方法設定 Ansible 變數非常重要、如此才能確保基礎服務使用正確的儲存設備。讓我們來看看每種部署方法的選項。

> 下表僅包含與記錄服務相關的儲存組態變數。您可以找到其他選項、這些選項"Red Hat OpenShift 記錄文件"應根據您的部署進行檢閱、設定及使用。

下表中的變數會使用提供的詳細資料、產生Ansible教戰手冊、為記錄服務建立PV和PVc。這種方法的彈性遠低於OpenShift安裝後使用元件安裝方針、不過如果您有現有的磁碟區可用、這是一個選項。

變動	詳細資料
`openshift_logging_storage_kind`	設定為 `nfs`讓安裝程式為記錄服務建立 NFS PV 。

變動	詳細資料
`openshift_logging_storage_host`	NFS主機的主機名稱或IP位址。這應該設定為虛擬機器的 dataLIF 。
`openshift_logging_storage_nfs_directory`	NFS匯出的掛載路徑。例如、如果該 Volume 是與 `/openshift_logging`一同連接的、您就會為此變數使用該路徑。
`openshift_logging_storage_volume_name`	要建立的 PV 名稱、例如 `pv_ose_logs`。
`openshift_logging_storage_volume_size`	NFS 匯出的大小 `100Gi`、例如。

如果您的OpenShift叢集已在執行中、因此已部署及設定Trident、則安裝程式可以使用動態資源配置來建立磁碟區。需要設定下列變數。

變動	詳細資料
`openshift_logging_es_pvc_dynamic`	設為true可使用動態資源配置的磁碟區。
`openshift_logging_es_pvc_storage_class_name`	將在PVc中使用的儲存類別名稱。
`openshift_logging_es_pvc_size`	在永久虛擬磁碟中要求的磁碟區大小。
`openshift_logging_es_pvc_prefix`	記錄服務使用的PVCS前置詞。
`openshift_logging_es_ops_pvc_dynamic`	設為 `true`、可將動態佈建的磁碟區用於營運記錄執行個體。
`openshift_logging_es_ops_pvc_storage_class_name`	作業記錄執行個體的儲存類別名稱。
`openshift_logging_es_ops_pvc_size`	作業執行個體的Volume要求大小。
`openshift_logging_es_ops_pvc_prefix`	ops執行個體PVCS的前置詞。

部署記錄堆疊

如果您將記錄部署為初始OpenShift安裝程序的一部分、則只需遵循標準部署程序即可。Ansible會設定及部署所需的服務和OpenShift物件、以便在可執行的完成後立即提供服務。

不過、如果您在初始安裝之後進行部署、Ansible將需要使用元件方針。此程序可能會隨著 OpenShift 的不同版本而稍有變更、因此請務必閱讀並遵循"Red Hat OpenShift Container Platform 3.11 文件"您的版本。

度量服務

度量服務可針對OpenShift叢集的狀態、資源使用率及可用度、提供寶貴的資訊給系統管理員。此外、也需要Pod自動擴充功能、許多組織會使用指標服務的資料來支付費用和/或顯示應用程式。

如同記錄服務和OpenShift整體、Ansible可用於部署度量服務。此外、與記錄服務一樣、度量服務也可以在叢集初始設定期間或使用元件安裝方法在其運作後進行部署。下表包含在設定度量服務的持續儲存時、重要的變數。

下表僅包含與度量服務相關的儲存組態相關變數。文件中還有許多其他選項、您應該根據部署情況來檢閱、設定及使用。

變動	詳細資料
`openshift_metrics_storage_kind`	設定為 `nfs`讓安裝程式為記錄服務建立 NFS PV 。
`openshift_metrics_storage_host`	NFS主機的主機名稱或IP位址。這應該設定為 SVM 的 dataLIF 。
`openshift_metrics_storage_nfs_directory`	NFS匯出的掛載路徑。例如、如果該 Volume 是與 `/openshift_metrics`一同連接的、您就會為此變數使用該路徑。
`openshift_metrics_storage_volume_name`	要建立的 PV 名稱、例如 `pv_ose_metrics`。
`openshift_metrics_storage_volume_size`	NFS 匯出的大小 `100Gi`、例如。

如果您的OpenShift叢集已在執行中、因此已部署及設定Trident、則安裝程式可以使用動態資源配置來建立磁碟區。需要設定下列變數。

變動	詳細資料
`openshift_metrics_cassandra_pvc_prefix`	用於度量PVCS的前置詞。
`openshift_metrics_cassandra_pvc_size`	要要求的磁碟區大小。
`openshift_metrics_cassandra_storage_type`	用於度量的儲存類型、必須設定為動態、Ansible才能建立具有適當儲存類別的PVCS。
`openshift_metrics_cassanda_pvc_storage_class_name`	要使用的儲存類別名稱。

部署度量服務

在您的主機/庫存檔案中定義適當的可Ansible變數後、使用Ansible部署服務。如果您是在OpenShift安裝時間進行部署、則會自動建立及使用PV。如果您是使用元件教戰手冊進行部署、則在安裝 OpenShift 之後、 Ansible 會建立所需的任何 PVCS 、並在 Trident 為其提供儲存設備之後、部署服務。

上述變數及部署程序可能會隨OpenShift的每個版本而變更。請務必檢閱並遵循"Red Hat 的 OpenShift 部署指南"您的版本、以便針對您的環境進行設定。

資料保護與災難恢復

瞭解使用 Trident 建立的 Trident 和磁碟區的保護與還原選項。對於每個應用程式、您都應該有持續性需求的資料保護與還原策略。

Trident 複寫與還原

您可以建立備份、以便在發生災難時還原 Trident 。

Trident 複寫

Trident 使用 Kubernetes CRD 來儲存及管理其本身的狀態、並使用 Kubernetes 叢集 etcd 來儲存其中繼資料。

步驟

1. 使用備份 Kubernetes 叢集 etcd "Kubernetes ：備份 etcd 叢集"。

2. 將備份產出工件放在 FlexVol volume 上

> NetApp 建議您保護 FlexVol 所在的 SVM ，並與另一個 SVM 建立 SnapMirror 關係。

Trident 恢復

您可以使用 Kubernetes CRD 和 Kubernetes 叢集 etcd 快照來復原 Trident 。

步驟

1. 從目的地 SVM 、將包含 Kubernetes etcd 資料檔案和憑證的磁碟區掛載到將設定為主要節點的主機上。
2. 複製下 Kubernetes 叢集的所有必要憑證 `/etc/kubernetes/pki`、以及下的 etcd 成員檔案 `/var/lib/etcd`。
3. 使用從 etcd 備份還原 Kubernetes 叢集"Kubernetes ：還原 etcd 叢集"。
4. 執行 `kubectl get crd`以驗證所有 Trident 自訂資源都已出現、並擷取 Trident 物件以驗證所有資料是否可用。

SVM 複寫與還原

Trident 無法設定複寫關係、不過儲存管理員可以使用 "ONTAP SnapMirror"複寫 SVM 。

發生災難時、您可以啟動SnapMirror目的地SVM、開始提供資料服務。系統還原時、您可以切換回主要系統。

關於這項工作

使用 SnapMirror SVM 複寫功能時、請考量下列事項：

- 您應該為每個啟用 SVM-DR 的 SVM 建立不同的後端。
- 設定儲存類別、僅在需要時才選取複寫的後端、以避免將不需要複寫的磁碟區佈建到支援 SVM-DR 的後端。
- 應用程式管理員應瞭解複寫的額外成本與複雜度、並在開始此程序之前仔細考慮其還原計畫。

SVM 複寫

您可以使用"ONTAP ： SnapMirror SVM 複寫"建立 SVM 複寫關係。

SnapMirror 可讓您設定選項、以控制要複寫的內容。您需要知道您在進行預先設定時所選擇使用 Trident 進行 SVM 恢復的選項。

- "-identity 保留為真"複寫整個 SVM 組態。
- "-discard 配置網路"不包括生命和相關的網路設定。
- "-identity 保留錯誤"僅複寫磁碟區和安全組態。

使用 Trident 進行 SVM 恢復

Trident 不會自動偵測 SVM 故障。發生災難時、管理員可以手動啟動 Trident 容錯移轉至新的 SVM 。

步驟

1. 取消已排程和持續的 SnapMirror 傳輸、中斷複寫關係、停止來源 SVM 、然後啟動 SnapMirror 目的地 SVM 。
2. 如果您指定 `-identity-preserve false`或 `-discard-config network`設定 SVM 複寫、請在 Trident 後端定義檔中更新 `managementLIF`和 `dataLIF`。
3. Trident 後端定義檔案中存在確認 `storagePrefix`。此參數無法變更。省略 `storagePrefix`會導致後端更新失敗。
4. 更新所有必要的後端、以反映新的目的地 SVM 名稱、使用：

```
./tridentctl update backend <backend-name> -f <backend-json-file> -n
<namespace>
```

5. 如果您指定 `-identity-preserve false`或 `discard-config network`、則必須退回所有應用程式 Pod 。

> 如果您指定 `-identity-preserve true`、則當目的地 SVM 啟動時、 Trident 所佈建的所有磁碟區都會開始提供資料。

Volume 複寫與還原

Trident 無法設定 SnapMirror 複寫關係、不過儲存管理員可以使用"ONTAP SnapMirror 複寫與還原"複寫 Trident 建立的磁碟區。

然後，您可以使用將恢復的卷導入 Trident "tridentctl Volume 匯入"。

> `ontap-san-economy`、或 `ontap-flexgroup-economy`驅動程式不支援匯入 `ontap-nas-economy`。

Snapshot 資料保護

您可以使用下列項目來保護及還原資料：

- 外部快照控制器和 CRD 、用於建立持續磁碟區（ PV ）的 Kubernetes Volume 快照。

  "Volume快照"

- ONTAP 快照可還原磁碟區的全部內容、或是還原個別檔案或 LUN 。

  "ONTAP 快照"

安全性

安全性

請使用此處列出的建議、確保 Trident 安裝安全無虞。

在自己的命名空間中執行 Trident

請務必防止應用程式、應用程式管理員、使用者和管理應用程式存取 Trident 物件定義或 Pod 、以確保可靠的儲存設備、並封鎖潛在的惡意活動。

要將其他應用程序和用戶與 Trident 分開，請始終在其自己的 Kubernetes 命名空間中安裝 Trident (`trident`) 。將 Trident 置於自己的命名空間中、可確保只有 Kubernetes 管理人員能夠存取 Trident Pod 、以及儲存在命名 CRD 物件中的成品（例如後端和 CHAP 機密、如果適用）。您應確保只允許系統管理員存取 Trident 命名空間、進而存取 `tridentctl`應用程式。

使用CHAP驗證搭配ONTAP 使用支援SAN的功能

Trident 支援 ONTAP SAN 工作負載的 CHAP 型驗證（使用 `ontap-san`和 `ontap-san-economy`驅動程式）。NetApp 建議在主機和儲存後端之間使用 Trident 的雙向 CHAP 進行驗證。

對於使用 SAN 儲存驅動程式的 ONTAP 後端、 Trident 可以設定雙向 CHAP 、並透過管理 CHAP 使用者名稱和機密 `tridentctl`。請參閱"準備使用ONTAP 不完善的SAN驅動程式來設定後端"以瞭解 Trident 如何在 ONTAP 後端上設定 CHAP 。

使用CHAP驗證NetApp HCI 搭配不景和SolidFire 不景的後端

NetApp建議部署雙向CHAP、以確保主機與NetApp HCI 支援功能及SolidFire 支援功能之間的驗證。Trident 使用的是每個租戶包含兩個 CHAP 密碼的秘密物件。安裝 Trident 時、它會管理 CHAP 機密、並將其儲存在相關 PV 的 CR 物件中 `tridentvolume`。建立 PV 時、 Trident 會使用 CHAP 機密來啟動 iSCSI 工作階段、並透過 CHAP 與 NetApp HCI 和 SolidFire 系統通訊。

由 Trident 建立的磁碟區不會與任何 Volume 存取群組相關聯。

搭配 NVE 和 NAE 使用 Trident

NetApp ONTAP 支援閒置資料加密、可在磁碟遭竊、退回或重新使用時、保護敏感資料。如需詳細資訊、請 "設定NetApp Volume Encryption總覽"參閱。

- 如果在後端上啟用 NAE 、則 Trident 中配置的任何 Volume 都將啟用 NAE 。
  - 。您可以將 NVE 加密旗標設定為 `""`建立啟用 NAE 的磁碟區。
- 如果後端未啟用 NAE ，則除非在後端組態中將 NVE 加密旗標設為（預設值），否則在 Trident 中配置的任何 Volume 都將啟用 NVE `false` 。

在啟用 NAE 的後端 Trident 中建立的磁碟區必須加密 NVE 或 NAE 。

- 您可以在 Trident 後端組態中將 NVE 加密旗標設為 `true`、以覆寫 NAE 加密、並以每個磁碟區為基礎使用特定的加密金鑰。
- 在啟用 NAE 的後端上、將 NVE 加密旗標設定為 `false`**會建立啟用 NAE 的 Volume 。您無法透過將 NVE 加密旗標設定為來停用 NAE 加密 `false`。**

- 您可以在 Trident 中手動建立 NVE Volume 、方法是將 NVE 加密旗標明確設定為 `true`。

如需後端組態選項的詳細資訊、請參閱：

- "ONTAP SAN 組態選項"

- "ONTAP NAS 組態選項"

Linux統一化金鑰設定（LUKS）

您可以啟用 Linux 統一金鑰設定（ LUKS ）來加密 Trident 上的 ONTAP SAN 和 ONTAP SAN 經濟磁碟區。Trident 支援使用複雜密碼的旋轉和磁碟區擴充、適用於使用 LUKS 加密的磁碟區。

在 Trident 中， LUKS 加密的磁碟區使用 AES-XTS-plain64 cypher 和模式"NIST"，如所建議。

開始之前

- 工作者節點必須安裝密碼設定2.1或更高版本（但低於3.0）。如需更多資訊"Gitlab：密碼設定"、請造訪。
- 基於效能理由， NetApp 建議工作節點支援進階加密標準新指令（ AES-NI ）。若要驗證AES-NI支援、請執行下列命令：

```
grep "aes" /proc/cpuinfo
```

如果沒有歸還任何內容、您的處理器就不支援AES-NI。如需 AES-NI 的詳細資訊、請參閱"Intel：進階加密標準指令（AES-NI）"：。

啟用LUKS加密

您可以使用Linux Unified Key Setup（LUKS）來啟用每個Volume、主機端的加密功能、以利ONTAP 執行SAN和ONTAP 支援SAN經濟效益的磁碟區。

步驟

1. 在後端組態中定義LUKS加密屬性。有關 ONTAP SAN 後端組態選項的詳細資訊，請參閱"ONTAP SAN 組態選項"。

```
{
  "storage": [
    {
      "labels": {
        "luks": "true"
      },
      "zone": "us_east_1a",
      "defaults": {
        "luksEncryption": "true"
      }
    },
    {
      "labels": {
        "luks": "false"
      },
      "zone": "us_east_1a",
      "defaults": {
        "luksEncryption": "false"
      }
    }
  ]
}
```

2. 使用 `parameters.selector`以 LUKS 加密定義儲存池。例如：

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: luks
provisioner: csi.trident.netapp.io
parameters:
  selector: "luks=true"
  csi.storage.k8s.io/node-stage-secret-name: luks-${pvc.name}
  csi.storage.k8s.io/node-stage-secret-namespace: ${pvc.namespace}
```

3. 建立包含LUKS通關密碼的秘密。例如：

```
kubectl -n trident create -f luks-pvc1.yaml
apiVersion: v1
kind: Secret
metadata:
  name: luks-pvc1
stringData:
  luks-passphrase-name: A
  luks-passphrase: secretA
```

限制

LUKS加密磁碟區無法利用ONTAP 重複資料刪除技術與壓縮技術。

用於匯入 **LUKS Volume** 的後端組態

若要匯入 LUKS Volume 、您必須在後端將設 `luksEncryption`為(`true。 `luksEncryption`選項告訴 `Trident` 卷是否符合 `LUKS （`false）(`true`或不符合 LUKS ），如下例所示。

```
version: 1
storageDriverName: ontap-san
managementLIF: 10.0.0.1
dataLIF: 10.0.0.2
svm: trident_svm
username: admin
password: password
defaults:
  luksEncryption: 'true'
  spaceAllocation: 'false'
  snapshotPolicy: default
  snapshotReserve: '10'
```

用於匯入 **LUKS Volume** 的 **PVC** 組態

若要動態匯入 LUKS Volume 、請將註釋設 `trident.netapp.io/luksEncryption`為 `true、並在 PVC 中包含啟用 LUKS 的儲存類別、如本範例所示。

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: luks-pvc
  namespace: trident
  annotations:
    trident.netapp.io/luksEncryption: "true"
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: luks-sc
```

旋轉LUKS複雜密碼

您可以旋轉LUKS複雜密碼並確認輪調。

> 請勿忘記密碼、除非您已驗證它不再被任何磁碟區、快照或機密所引用。如果參考的通關密碼遺失、您可能無法掛載磁碟區、而且資料將保持加密且無法存取。

關於這項工作

如果在指定新的LUKS通關密碼之後建立裝載磁碟區的Pod、則會發生LUKS通關密碼循環。建立新的 Pod 時、Trident 會將磁碟區上的 LUKS 複雜密碼與機密中的作用中複雜密碼進行比較。

- 如果磁碟區上的通關密碼與機密中的作用中通關密碼不相符、就會發生輪調。
- 如果磁碟區上的複雜密碼與機密中的作用中複雜密碼相符、則會忽略此 `previous-luks-passphrase`參數。

步驟

1. 新增 `node-publish-secret-name`和 `node-publish-secret-namespace` StorageClass 參數。例如：

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: csi-san
provisioner: csi.trident.netapp.io
parameters:
  trident.netapp.io/backendType: "ontap-san"
  csi.storage.k8s.io/node-stage-secret-name: luks
  csi.storage.k8s.io/node-stage-secret-namespace: ${pvc.namespace}
  csi.storage.k8s.io/node-publish-secret-name: luks
  csi.storage.k8s.io/node-publish-secret-namespace: ${pvc.namespace}
```

2. 識別磁碟區或快照上的現有密碼。

Volume

```
tridentctl -d get volume luks-pvc1
GET http://127.0.0.1:8000/trident/v1/volume/<volumeID>

...luksPassphraseNames:["A"]
```

Snapshot

```
tridentctl -d get snapshot luks-pvc1
GET http://127.0.0.1:8000/trident/v1/volume/<volumeID>/<snapshotID>

...luksPassphraseNames:["A"]
```

3. 更新磁碟區的LUKS機密、以指定新的和先前的密碼。確保 `previous-luke-passphrase-name`與 `previous-luks-passphrase`先前的密碼相符。

```
apiVersion: v1
kind: Secret
metadata:
  name: luks-pvc1
stringData:
  luks-passphrase-name: B
  luks-passphrase: secretB
  previous-luks-passphrase-name: A
  previous-luks-passphrase: secretA
```

4. 建立新的Pod以掛載Volume。這是啟動旋轉所需的。
5. 確認複雜密碼已旋轉。

Volume

```
tridentctl -d get volume luks-pvc1
GET http://127.0.0.1:8000/trident/v1/volume/<volumeID>

...luksPassphraseNames:["B"]
```

Snapshot

```
tridentctl -d get snapshot luks-pvc1
GET http://127.0.0.1:8000/trident/v1/volume/<volumeID>/<snapshotID>

...luksPassphraseNames:["B"]
```

結果

只有在磁碟區和快照上傳回新的通關密碼時、才會旋轉通關密碼。

> 如果傳回兩個密碼短語、例如 `luksPassphraseNames: ["B", "A"]`、旋轉不完整。您可以觸發新的Pod以嘗試完成旋轉。

啟用**Volume**擴充

您可以在LUKS加密的Volume上啟用Volume擴充。

步驟

1. 啟用 `CSINodeExpandSecret`功能安全門（ beta 1.25+ ）。如 "Kubernetes 1.25：使用Secrets進行節點導向的SCSI Volume擴充" 需詳細資訊、請參閱。
2. 新增 `node-expand-secret-name`和 `node-expand-secret-namespace` StorageClass 參數。例如：

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: luks
provisioner: csi.trident.netapp.io
parameters:
  selector: "luks=true"
  csi.storage.k8s.io/node-stage-secret-name: luks-${pvc.name}
  csi.storage.k8s.io/node-stage-secret-namespace: ${pvc.namespace}
  csi.storage.k8s.io/node-expand-secret-name: luks-${pvc.name}
  csi.storage.k8s.io/node-expand-secret-namespace: ${pvc.namespace}
allowVolumeExpansion: true
```

結果

當您啟動線上儲存擴充時、kubelet會將適當的認證資料傳遞給驅動程式。

Kerberos 執行中加密

使用 Kerberos 在線上加密，您可以針對託管叢集與儲存後端之間的流量啟用加密，藉此改善資料存取安全性。

Trident 支援 ONTAP 的 Kerberos 加密作為儲存後端：

- * 內部部署 ONTAP * ： Trident 支援透過 NFSv3 和 NFSv4 連線進行 Kerberos 加密，從 Red Hat OpenShift 和上游 Kubernetes 叢集到內部部署 ONTAP 磁碟區。

您可以建立、刪除、調整大小、快照、複製、 唯讀複製及匯入使用 NFS 加密的磁碟區。

使用內部部署的 **ONTAP** 磁碟區來設定在線上 **Kerberos** 加密

您可以在託管叢集與內部部署 ONTAP 儲存後端之間的儲存流量上啟用 Kerberos 加密。

內部部署 ONTAP 儲存後端的 NFS 流量 Kerberos 加密僅支援使用 `ontap-nas`儲存驅動程式。

開始之前

- 請確定您可以存取 `tridentctl` 公用程式。
- 確保您具有 ONTAP 儲存後端的管理員存取權。
- 確保您知道將從 ONTAP 儲存後端共用的磁碟區名稱。
- 請確定您已準備好 ONTAP 儲存 VM 、以支援 NFS 磁碟區的 Kerberos 加密。請參閱 "在 dataLIF 上啟用 Kerberos" 以取得指示。
- 請確定您使用 Kerberos 加密的任何 NFSv4 磁碟區都已正確設定。請參閱的 NetApp NFSv4 網域組態一節（第 13 頁） "NetApp NFSv4 增強與最佳實務指南"。

新增或修改 **ONTAP** 匯出原則

您需要將規則新增至現有的 ONTAP 匯出原則、或建立新的匯出原則、以支援 ONTAP 儲存 VM 根磁碟區的 Kerberos 加密、以及與上游 Kubernetes 叢集共用的任何 ONTAP 磁碟區。您新增的匯出原則規則或您建立的新匯出原則需要支援下列存取通訊協定和存取權限：

存取傳輸協定

使用 NFS 、 NFSv3 和 NFSv4 存取通訊協定來設定匯出原則。

存取詳細資料

您可以根據對磁碟區的需求、設定 Kerberos 加密的三個不同版本之一：

- * Kerberos 5* - （驗證與加密）
- * Kerberos 5i* - （身分識別保護的驗證與加密）
- * Kerberos 5p* - （身分識別與隱私保護的驗證與加密）

使用適當的存取權限來設定 ONTAP 匯出原則規則。例如、如果叢集將使用 Kerberos 5i 和 Kerberos 5p 加密混合安裝 NFS 磁碟區、請使用下列存取設定：

類型	唯讀存取	讀取 / 寫入存取權	超級使用者存取權
UNIX	已啟用	已啟用	已啟用
Kerberos 5i	已啟用	已啟用	已啟用
Kerberos 5p	已啟用	已啟用	已啟用

請參閱下列文件、瞭解如何建立 ONTAP 匯出原則和匯出原則規則：

- "建立匯出原則"
- "新增規則至匯出原則"

建立儲存後端

您可以建立內含 Kerberos 加密功能的 Trident 儲存後端組態。

關於這項工作

當您建立設定 Kerberos 加密的儲存後端組態檔時、可以使用參數指定 Kerberos 加密的三個不同版本之一 `spec.nfsMountOptions` ：

- `spec.nfsMountOptions: sec=krb5` （驗證與加密）
- `spec.nfsMountOptions: sec=krb5i` （身分識別保護的驗證與加密）
- `spec.nfsMountOptions: sec=krb5p` （身分識別與隱私保護的驗證與加密）

只指定一個 Kerberos 層級。如果您在參數清單中指定多個 Kerberos 加密層級、則只會使用第一個選項。

步驟

1. 在託管叢集上、使用下列範例建立儲存後端組態檔案。以您環境的資訊取代括弧 <> 中的值：

```
apiVersion: v1
kind: Secret
metadata:
  name: backend-ontap-nas-secret
type: Opaque
stringData:
  clientID: <CLIENT_ID>
  clientSecret: <CLIENT_SECRET>
---
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-ontap-nas
spec:
  version: 1
  storageDriverName: "ontap-nas"
  managementLIF: <STORAGE_VM_MGMT_LIF_IP_ADDRESS>
  dataLIF: <PROTOCOL_LIF_FQDN_OR_IP_ADDRESS>
  svm: <STORAGE_VM_NAME>
  username: <STORAGE_VM_USERNAME_CREDENTIAL>
  password: <STORAGE_VM_PASSWORD_CREDENTIAL>
  nasType: nfs
  nfsMountOptions: ["sec=krb5i"] #can be krb5, krb5i, or krb5p
  qtreesPerFlexvol:
  credentials:
    name: backend-ontap-nas-secret
```

2. 使用您在上一個步驟中建立的組態檔來建立後端：

```
tridentctl create backend -f <backend-configuration-file>
```

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
```

識別並修正組態檔的問題之後、您可以再次執行create命令。

建立儲存類別

您可以建立儲存類別、以使用 Kerberos 加密來配置磁碟區。

關於這項工作

當您建立儲存類別物件時、可以使用下列參數、指定 Kerberos 加密的三個不同版本之一 `mountOptions` ：

- `mountOptions: sec=krb5` （驗證與加密）
- `mountOptions: sec=krb5i` （身分識別保護的驗證與加密）
- `mountOptions: sec=krb5p` （身分識別與隱私保護的驗證與加密）

只指定一個 Kerberos 層級。如果您在參數清單中指定多個 Kerberos 加密層級、則只會使用第一個選項。如果您在儲存後端組態中指定的加密層級與您在儲存類別物件中指定的層級不同、則儲存類別物件會優先。

步驟

1. 使用以下範例建立 StorageClass Kubernetes 物件：

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontap-nas-sc
provisioner: csi.trident.netapp.io
mountOptions:
  - sec=krb5i #can be krb5, krb5i, or krb5p
parameters:
  backendType: ontap-nas
  storagePools: ontapnas_pool
  trident.netapp.io/nasType: nfs
allowVolumeExpansion: true
```

2. 建立儲存類別：

```
kubectl create -f sample-input/storage-class-ontap-nas-sc.yaml
```

3. 確定已建立儲存類別：

```
kubectl get sc ontap-nas-sc
```

您應該會看到類似下列的輸出：

```
NAME          PROVISIONER             AGE
ontap-nas-sc    csi.trident.netapp.io   15h
```

配置 Volume

建立儲存後端和儲存類別之後、您現在可以配置 Volume 。有關說明，請參閱 "配置 Volume"。

使用 **Azure NetApp Files** 磁碟區設定在線上 **Kerberos** 加密

您可以在託管叢集與單一 Azure NetApp Files 儲存後端或 Azure NetApp Files 儲存後端的虛擬集區之間的儲存流量上啟用 Kerberos 加密。

開始之前

- 確保您已在託管的 Red Hat OpenShift 叢集上啟用 Trident 。
- 請確定您可以存取 `tridentctl` 公用程式。
- 請注意中的要求並遵循中的指示、以確保您已準備好 Azure NetApp Files 儲存後端進行 Kerberos 加密 "本文檔 Azure NetApp Files"。
- 請確定您使用 Kerberos 加密的任何 NFSv4 磁碟區都已正確設定。請參閱的 NetApp NFSv4 網域組態一節（第 13 頁） "NetApp NFSv4 增強與最佳實務指南"。

建立儲存後端

您可以建立包含 Kerberos 加密功能的 Azure NetApp Files 儲存後端組態。

關於這項工作

當您建立儲存後端組態檔案來設定 Kerberos 加密時、您可以加以定義、以便將其套用至下列兩種可能的層級之一：

- 使用欄位的 * 儲存後端層級 * `spec.kerberos`
- 使用欄位的 * 虛擬集區層級 * `spec.storage.kerberos`

當您在虛擬集區層級定義組態時、會使用儲存類別中的標籤來選取集區。

在任一層級、您都可以指定 Kerberos 加密的三個不同版本之一：

- `kerberos: sec=krb5` （驗證與加密）
- `kerberos: sec=krb5i` （身分識別保護的驗證與加密）
- `kerberos: sec=krb5p` （身分識別與隱私保護的驗證與加密）

步驟

1. 在託管叢集上、根據您需要定義儲存後端（儲存後端層級或虛擬集區層級）的位置、使用下列其中一個範例建立儲存後端組態檔案。以您環境的資訊取代括弧 <> 中的值：

儲存後端層級範例

```
apiVersion: v1
kind: Secret
metadata:
  name: backend-tbc-secret
type: Opaque
stringData:
  clientID: <CLIENT_ID>
  clientSecret: <CLIENT_SECRET>

---
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc
spec:
  version: 1
  storageDriverName: azure-netapp-files
  subscriptionID: <SUBSCRIPTION_ID>
  tenantID: <TENANT_ID>
  location: <AZURE_REGION_LOCATION>
  serviceLevel: Standard
  networkFeatures: Standard
  capacityPools: <CAPACITY_POOL>
  resourceGroups: <RESOURCE_GROUP>
  netappAccounts: <NETAPP_ACCOUNT>
  virtualNetwork: <VIRTUAL_NETWORK>
  subnet: <SUBNET>
  nasType: nfs
  kerberos: sec=krb5i #can be krb5, krb5i, or krb5p
  credentials:
    name: backend-tbc-secret
```

虛擬集區層級範例

```
---
apiVersion: v1
kind: Secret
metadata:
  name: backend-tbc-secret
type: Opaque
stringData:
  clientID: <CLIENT_ID>
  clientSecret: <CLIENT_SECRET>

---
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc
spec:
  version: 1
  storageDriverName: azure-netapp-files
  subscriptionID: <SUBSCRIPTION_ID>
  tenantID: <TENANT_ID>
  location: <AZURE_REGION_LOCATION>
  serviceLevel: Standard
  networkFeatures: Standard
  capacityPools: <CAPACITY_POOL>
  resourceGroups: <RESOURCE_GROUP>
  netappAccounts: <NETAPP_ACCOUNT>
  virtualNetwork: <VIRTUAL_NETWORK>
  subnet: <SUBNET>
  nasType: nfs
  storage:
    - labels:
        type: encryption
      kerberos: sec=krb5i #can be krb5, krb5i, or krb5p
  credentials:
    name: backend-tbc-secret
```

2. 使用您在上一個步驟中建立的組態檔來建立後端：

```
tridentctl create backend -f <backend-configuration-file>
```

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
```

識別並修正組態檔的問題之後、您可以再次執行create命令。

建立儲存類別

您可以建立儲存類別、以使用 Kerberos 加密來配置磁碟區。

步驟

1. 使用以下範例建立 StorageClass Kubernetes 物件：

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: sc-nfs
provisioner: csi.trident.netapp.io
parameters:
  backendType: azure-netapp-files
  trident.netapp.io/nasType: nfs
  selector: type=encryption
```

2. 建立儲存類別：

```
kubectl create -f sample-input/storage-class-sc-nfs.yaml
```

3. 確定已建立儲存類別：

```
kubectl get sc -sc-nfs
```

您應該會看到類似下列的輸出：

```
NAME          PROVISIONER             AGE
sc-nfs        csi.trident.netapp.io   15h
```

配置 **Volume**

建立儲存後端和儲存類別之後、您現在可以配置 Volume 。有關說明，請參閱 "配置 Volume"。