最佳實務做法與建議
Trident

NetApp
October 30, 2026

This PDF was generated from https://docs.netapp.com/zh-tw/trident-2606/trident-reco/deploy-reco.html on October 30, 2026. Always check docs.netapp.com for the latest.

目錄

最佳實務做法與建議 . . . 1
Trident 部署建議 . . . 1
部署到專用命名空間 . . . 1
使用配額和範圍限制來控制儲存使用量 . . . 1
Trident 儲存組態最佳實務 . . . 1
平台概覽 . . . 1
ONTAP 和 Cloud Volumes ONTAP 最佳實務做法 . . . 1
SolidFire 最佳實踐 . . . 5
哪裡可以找到更多資訊？ . . . 7
將 Trident 與 Kubernetes 整合 . . . 7
驅動程式選擇與部署 . . . 7
儲存類別設計 . . . 9
虛擬資源池設計 . . . 10
磁碟區作業 . . . 11
指標服務 . . . 14
Trident 資料保護與災難恢復 . . . 15
Trident 複寫與還原 . . . 15
SVM 複製與復原 . . . 16
Volume 複寫與還原 . . . 17
快照資料保護 . . . 17
使用 Trident 自動化具狀態應用程式的容錯移轉 . . . 17
強制分離的詳細資訊 . . . 17
有關自動容錯移轉的詳細資訊 . . . 18
安全性 . . . 22
Trident 安全性建議 . . . 22
Trident Linux 統一金鑰設定 （LUKS） . . . 23
Trident Kerberos 傳輸中加密 . . . 28

最佳實務做法與建議

Trident 部署建議

部署 Trident 時、請使用此處列出的建議。

部署到專用命名空間

"命名空間" 提供不同應用程式之間的管理隔離，並構成資源共享的障礙。例如，一個命名空間中的 PVC 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，因此利用具有提升權限的服務帳戶。

此外，存取 Trident pod 可能使用戶能夠存取儲存系統憑證和其他敏感資訊。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 具有兩項特性，二者結合使用可提供強大的機制來限制應用程式的資源消耗。 "儲存配額機制"可讓管理員以每個命名空間為基礎，實施全域以及儲存類別特定的容量和物件數量消耗限制。此外，使用 "範圍限制" 可確保在將請求轉送給配置器之前，PVC 請求的值均在最小值和最大值範圍內。

這些值是基於命名空間定義的，這意味著每個命名空間都應該定義與其資源需求相符的值。請參閱此處以取得相關資訊 "如何利用配額"。

Trident 儲存組態最佳實務

NetApp 產品組合中的每個儲存平台都具有獨特的功能，無論應用程式是否容器化，都能從中受益。

平台概覽

Trident 可與 ONTAP 和 Element 搭配使用。雖然沒有哪個平台比其他平台更適合所有應用程式和場景，但在選擇平台時，應考慮應用程式的需求以及管理設備的團隊的需求。

您應該遵循所用協定對應的主機作業系統的最佳實務。此外，您還可以考慮在後端、儲存類別和 PVC 設定中融入應用程式最佳實務（如有），以優化特定應用程式的儲存。

ONTAP 和 Cloud Volumes ONTAP 最佳實務做法

了解為 Trident 配置 ONTAP 和 Cloud Volumes ONTAP 的最佳實務做法。

以下建議是為容器化工作負載配置 ONTAP 的指導原則，這些工作負載會使用由 Trident 動態配置的磁碟區。每項建議都應根據您的環境進行考慮和評估，以確定其適用性。

使用專用於 **Trident** 的 **SVM**

儲存虛擬機器（SVM）為 ONTAP 系統上的租用戶提供隔離和管理分離。將 SVM 專用於應用程式可實現權限委派，並支援應用限制資源消耗的最佳實務做法。

SVM 的管理有多種選擇：

- 在後端組態中提供叢集管理介面、適當的認證資料、並指定 SVM 名稱。
- 使用 ONTAP System Manager 或 CLI 為 SVM 建立專用管理介面。
- 與 NFS 資料介面共用管理角色。

無論哪種情況，介面都應在 DNS 中配置，並且在配置 Trident 時應使用 DNS 名稱。這有助於簡化某些災難復原場景，例如無需網路身分保留的 SVM-DR。

對於 SVM 而言，採用專用管理 LIF 或共享管理 LIF 並無優劣之分，但您應確保網路安全策略與所選方案相符。無論如何，管理 LIF 都應可透過 DNS 訪問，以便在 "SVM-DR"與 Trident 搭配使用時實現最大的靈活性。

限制磁碟區數量上限

ONTAP 儲存系統存在最大磁碟區數限制，此限制因軟體版本和硬體平台而異。請參閱 "NetApp Hardware Universe"以了解您特定平台和 ONTAP 版本的確切限制。當磁碟區數達到上限時，不僅 Trident 的資源配置作業會失敗，所有儲存請求也會失敗。

Trident 的 `ontap-nas`和 `ontap-san`驅動程式會為每個建立的 Kubernetes 持久性磁碟區（PV）配置一個 FlexVolume。 `ontap-nas-economy`驅動程式大約每 200 個 PV 建立一個 FlexVolume（可在 50 到 300 之間配置）。 `ontap-san-economy`驅動程式大約每 100 個 PV 建立一個 FlexVolume（可在 50 到 200 之間配置）。為防止 Trident 佔用儲存系統上所有可用磁碟區，您應該設定 SVM 的限制。您可以透過命令列進行此操作：

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

`max-volumes` 的值會根據您環境的幾個特定條件而有所不同：

- ONTAP 叢集中現有磁碟區的數量
- 您預計在 Trident 之外為其他應用程式配置的磁碟區數量
- Kubernetes 應用程式預計使用的持續磁碟區數量

該 `max-volumes`值是 ONTAP 叢集中所有節點上配置的磁碟區總數、而非單一 ONTAP 節點上的磁碟區數。因此、您可能會遇到某些情況、其中 ONTAP 叢集節點的 Trident 配置磁碟區數可能遠多於或遠少於其他節點。

例如，一個雙節點 ONTAP 叢集最多可以託管 2000 個 FlexVol 磁碟區。將最大磁碟區數設為 1250 看起來非常合理。但是，如果僅 "Aggregate"將來自一個節點的 Aggregate 分配給 SVM，或者來自一個節點的 Aggregate 無法進行配置（例如，由於容量不足），則另一個節點將成為所有 Trident 配置磁碟區的目標。這意味著該節點的磁碟區限制可能在達到 `max-volumes`值之前就已經達到，從而影響 Trident 以及使用該節點的其他磁碟區操作。您可以透過確保將叢集中每個節點的 **Aggregate** 以相等的數量分配給 **Trident** 使用的 **SVM** 來避免這種情況。

複製磁碟區

NetApp Trident 在使用 `ontap-nas`、 `ontap-san`和 `solidfire-san`儲存驅動程式時支援複製磁碟區。使用 `ontap-nas-flexgroup`或 `ontap-nas-economy`驅動程式時，不支援複製。從現有磁碟區建立新磁碟區將建立新的快照。

避免克隆與不同 StorageClass 關聯的 PVC。在同一 StorageClass 內執行克隆操作，以確保相容性並防止意外行為。

限制 **Trident** 建立的磁碟區大小上限

若要設定 Trident 可建立的磁碟區最大大小，請在您的 `limitVolumeSize` 定義中使用 `backend.json` 參數。

除了控制儲存陣列的磁碟區大小之外，還應該利用 Kubernetes 功能。

限制 **Trident** 建立的 **FlexVols** 大小上限

若要為作為 ontap-san-economy 和 ontap-nas-economy 驅動程式資源池所使用的 FlexVols 設定最大大小，請在您的 `limitVolumePoolSize` 定義中使用 `backend.json` 參數。

設定 **Trident** 使用雙向 **CHAP**

您可以在後端定義中指定 CHAP 發起方和目標的使用者名稱和密碼，並讓 Trident 在 SVM 上啟用 CHAP。透過後端設定中的 `useCHAP` 參數，Trident 可使用 CHAP 對 ONTAP 後端的 iSCSI 連線進行驗證。

建立並使用 **SVM QoS** 原則

透過對 SVM 應用 ONTAP QoS 原則，可限制 Trident 已配置磁碟區可使用的 IOPS 數量。這有助於 "阻止霸凌行為"防止失控容器影響 Trident SVM 以外的工作負載。

您只需幾個步驟即可為 SVM 建立 QoS 策略。如需有關 ONTAP 的最準確資訊、請參閱您所用 ONTAP 版本的文件。以下範例建立了一個 QoS 策略、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，如果您的 ONTAP 版本支援，您可以考慮使用 QoS 最低要求來確保容器化工作負載的吞吐量。自適應 QoS 與 SVM 層級的原則不相容。

分配給容器化工作負載的 IOPS 數量取決於諸多因素。其中包括：

- 其他使用儲存陣列的工作負載。如果存在與 Kubernetes 部署無關的其他工作負載正在使用儲存資源，則應注意確保這些工作負載不會受到意外的不利影響。
- 預期工作負載將在容器中運作。如果具有高 IOPS 要求的工作負載將在容器中運作，則低 QoS 原則會導致不佳的體驗。

需要注意的是，在 SVM 層級指派的 QoS 策略會導致佈建給該 SVM 的所有磁碟區共用同一個 IOPS 資源池。如果一個或少數幾個容器化應用程式的 IOPS 需求很高，則可能會對其他容器化工作負載造成嚴重影響。在這種情況下，您可能需要考慮使用外部自動化工具來為每個磁碟區指派 QoS 策略。

只有當您的 ONTAP 版本低於 9.8 時，才應將 QoS 原則群組指派給 SVM。

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低。ONTAP QoS 策略群組為磁碟區提供 QoS 選項，並允許使用者為一個或多個工作負載定義吞吐量上限。有關 QoS 的更多資訊，請參閱 "透過 QoS 保證處理量"。您可以在後端或儲存池中指定 QoS 策略群組，這些策略群組將套用於在該儲存池或後端中建立的每個磁碟區。

ONTAP 有兩種 QoS 原則群組：傳統和自適應。傳統原則群組提供固定的最大（或在較新版本中為最小）IOPS 處理量。自適應 QoS 會根據工作負載大小自動調整處理量，在工作負載大小變化時維持 IOPS 與 TB|GB 的比率。當您在大型部署中管理數百或數千個工作負載時，這提供了顯著的優勢。

建立 QoS 策略群組時、請考慮以下事項：

- 您應該在後端設定的 `defaults`區塊中設定 `qosPolicy`鍵。請參閱以下後端設定範例：

```
---
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
```

- 您應該按磁碟區套用原則群組，以便每個磁碟區都能獲得原則群組指定的全部處理量。不支援共享原則群組。

如需 QoS 策略群組的詳細資訊，請參閱 "ONTAP 指令參考"。

限制 **Kubernetes** 叢集成員的儲存資源存取

限制對 Trident 所建立的 NFS 磁碟區、iSCSI LUN 和 FC LUN 的存取，是 Kubernetes 部署安全態勢的關鍵組成部分。這樣做可以防止非 Kubernetes 叢集成員的主機存取這些磁碟區，從而避免資料被意外修改。

理解命名空間是 Kubernetes 中資源的邏輯邊界至關重要。雖然同一命名空間內的資源可以共享，但需要注意的

是，它們之間不支援跨命名空間存取。這意味著，即使 PV 是全域對象，當綁定到 PVC 時，也只有位於相同命名空間的 Pod 才能存取它們。務必確保在適當情況下使用命名空間來實現資源隔離。

在 Kubernetes 環境中，大多數組織對資料安全的主要擔憂是容器中的進程可以存取掛載到主機上的儲存，但這些儲存並非容器所期望的。 "命名空間" 旨在防止此類安全漏洞。然而，特權容器是個例外。

特權容器是指擁有比普通容器高得多的主機級權限的容器。這些權限預設不會被拒絕，因此請務必使用 "Pod 安全性原則"停用此功能。

對於需要同時從 Kubernetes 和外部主機存取的磁碟區，應採用傳統方式管理儲存設備，即由管理員建立 PV，而不是由 Trident 管理。這樣可以確保僅在 Kubernetes 和外部主機都已中斷連線且不再使用該磁碟區時才銷毀儲存磁碟區。此外，還可以套用自訂匯出原則，從而允許從 Kubernetes 叢集節點和 Kubernetes 叢集外部的目標伺服器存取。

對於具有專用基礎架構節點（例如 OpenShift）或其他無法調度使用者應用程式的節點的部署，應使用單獨的匯出策略來進一步限制對儲存資源的存取。這包括為部署到這些基礎架構節點的服務（例如 OpenShift Metrics 和 Logging 服務）以及部署到非基礎架構節點的標準應用程式建立匯出策略。

使用專用的匯出原則

您應確保每個後端都存在匯出策略，該策略僅允許存取 Kubernetes 叢集中的節點。Trident 可以自動建立和管理匯出策略。這樣，Trident 將對其配置的磁碟區的存取限制在 Kubernetes 叢集中的節點上，並簡化節點的新增/刪除操作。

或者、您也可以手動建立匯出原則、並在其中填入一或多個匯出規則、以處理每個節點存取要求：

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

執行這些命令可以限制哪些 Kubernetes 節點可以存取資料。

停用應用程式 **SVM** 的 **`showmount`**

`showmount` 功能可讓 `NFS` 用戶端向 `SVM` 查詢可用 `NFS` 匯出的清單。部署至 `Kubernetes` 叢集的 `Pod` 可以針對 `SVM` 發出 `showmount -e` 命令、並接收可用掛載的清單、包括其無權存取的掛載。雖然這本身並不構成安全性危害、但它確實提供了不必要的資訊、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 ONTAP CLI 命令來停用 `showmount`：

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

SolidFire 最佳實踐

了解為 Trident 配置 SolidFire 儲存設備的最佳實務做法。

建立 **SolidFire** 帳戶

每個 SolidFire 帳戶代表一個唯一的磁碟區擁有者，並擁有自己的一組 Challenge-Handshake Authentication Protocol (CHAP) 憑證。您可以透過帳戶名稱和對應的 CHAP 憑證，或透過磁碟區存取群組來存取指派給某個帳戶的磁碟區。一個帳戶最多可以分配兩千個磁碟區，但一個磁碟區只能屬於一個帳戶。

建立 **QoS** 原則

如果您想要建立並儲存可套用於多個磁碟區的標準化服務品質設定，請使用 SolidFire 服務品質（QoS）原則。

您可以按磁碟區設定 QoS 參數。透過設定定義 QoS 的三個可設定參數（Min IOPS、Max IOPS 和 Burst IOPS），可以確保每個磁碟區的效能。

以下是 4Kb 區塊大小的可能最小、最大和突發 IOPS 值。

IOPS 參數	定義	最小值	預設值	最大值 **(4Kb)**
最小 IOPS	磁碟區的保證效能層級。	50	50	15000
最大 IOPS	效能不會超過此限制。	50	15000	200,000
突發 IOPS	短時突發場景下允許的最大 IOPS 。	50	15000	200,000

儘管 Max IOPS 和 Burst IOPS 可以設定為高達 200,000，但磁碟區的實際最大效能受叢集使用情況和每個節點效能的限制。

資料區塊大小和頻寬直接影響 IOPS 值。隨著資料區塊大小的增加，系統會提升頻寬以處理更大的資料區塊。頻寬增加的同時，系統能夠達到的 IOPS 值反而會下降。如需 QoS 和效能的詳細資訊，請參閱 "SolidFire 服務品質"。

SolidFire 驗證

Element 支援兩種驗證方法：CHAP 和磁碟區存取群組（VAG）。CHAP 使用 CHAP 協定對主機進行身份驗證，以連接到後端伺服器。磁碟區存取群組控制對其配置的磁碟區的存取。NetApp 建議使用 CHAP 進行身份驗證，因為它更簡單且沒有擴展限制。

Trident 配備增強型 CSI 配置程式的 Trident 支援使用 CHAP 驗證。VAG 只能在傳統的非 CSI 作業模式下使用。

CHAP 驗證（驗證啟動器是否為預期的磁碟區使用者）僅支援以帳戶為基礎的存取控制。如果您使用 CHAP 進行驗證，有兩個選項可用：單向 CHAP 和雙向 CHAP。單向 CHAP 使用 SolidFire 帳戶名稱和啟動器密碼來驗證磁碟區存取。雙向 CHAP 選項提供最安全的磁碟區驗證方式，因為磁碟區透過帳戶名稱和啟動器密碼來驗證主機，然後主機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，請建立存取群組並將主機啟動器和磁碟區新增至該存取群組。新增至存取群組的每個 IQN 都可以存取群組中的每個磁碟區，無論是否使用 CHAP 驗證。如果 iSCSI 啟動器設定為使用 CHAP 驗證，則使用基於帳戶的存取控制。如果 iSCSI 啟動器未設定為使用 CHAP 驗證，則使用 Volume Access Group 存取控制。

哪裡可以找到更多資訊？

以下列出了一些最佳實踐文件。搜尋 "NetApp 資料庫"以取得最新版本。

ONTAP

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

Element 軟體

- "配置 SolidFire for Linux"

NetApp HCI

- "NetApp HCI 部署先決條件"
- "存取 NetApp Deployment Engine"

應用程式最佳實務資訊

- "ONTAP 上 MySQL 的最佳實務做法"
- "MySQL 在 SolidFire 上的最佳實踐"
- "NetApp SolidFire 和 Cassandra"
- "Oracle 最佳實務做法 SolidFire"
- "PostgreSQL 在 SolidFire 上的最佳實踐"

並非所有應用程式都有具體指南，重要的是與您的 NetApp 團隊合作，並使用 "NetApp 資料庫"尋找最新文件。

將 Trident 與 Kubernetes 整合

要整合 Trident，需要整合以下設計和架構元素：驅動程式選擇和部署、儲存類別設計、虛擬池設計、持久性磁碟區聲明 (PVC) 對儲存配置的影響、磁碟區操作以及使用 Trident 部署 OpenShift 服務。

驅動程式選擇與部署

為您的儲存系統選擇並部署後端驅動程式。

ONTAP 後端驅動程式

ONTAP 後端驅動程式之間的差異在於所使用的傳輸協定以及磁碟區在儲存系統上的配置方式。因此，在決定部署哪個驅動程式時，請務必仔細考慮。

從更高層次來看，如果您的應用程式包含需要共用儲存的元件（多個 Pod 存取同一個 PVC），則基於 NAS 的驅動程式是預設選擇；而基於區塊的 iSCSI 驅動程式則符合非共用儲存的需求。協議的選擇應基於應用程式的需求以及儲存和基礎架構團隊的熟悉程度。一般來說，對於大多數應用程式而言，兩者之間的差異很小，因此通

常取決於是否需要共用儲存（即多個 Pod 需要同時存取）。

可用的 ONTAP 後端驅動程式包括：

- `ontap-nas`：每個已配置的 PV 都是一個完整的 ONTAP FlexVolume。
- `ontap-nas-economy`：每個已配置的 PV 都是一個 qtree，每個 FlexVolume 可配置的 qtree 數量（預設值為 200）。
- `ontap-nas-flexgroup`：每個 PV 都配置為完整的 ONTAP FlexGroup，並且分配給 SVM 的所有聚合都將被使用。
- `ontap-san`：每個已配置的 PV 都是其自身 FlexVolume 內部的一個 LUN。
- `ontap-san-economy`：每個已佈建的 PV 都是一個 LUN，每個 FlexVolume 可配置的 LUN 數量（預設值為 100）。

在三個 NAS 驅動程式之間進行選擇會對應用程式可用的功能產生一些影響。

請注意，下表中的並非所有功能都透過 Trident 公開。如果需要該功能，則必須由儲存管理員在配置後套用其中一些功能。上標腳註區分了每個功能和驅動程式的功能。

ONTAP NAS 驅動程式	快照	複本	動態匯出原則	多重附加	**QoS**	調整大小	複寫
`ontap-nas`	是的	是的	是	是的	是	是的	是
`ontap-nas-economy`	NO[3]	NO[3]	是	是的	NO[3]	是的	NO[3]
`ontap-nas-flexgroup`	是	否	是	是的	是	是的	是

Trident 為 ONTAP 提供 2 個 SAN 驅動程式，其功能如下所示。

ONTAP SAN 驅動程式	快照	複本	多重附加	雙向 **CHAP**	**QoS**	調整大小	複寫
`ontap-san`	是的	是的	是	是的	是	是的	是
`ontap-san-economy`	是的	是的	是	是的	NO[3]	是的	NO[3]

> 以上表格的註腳：Yes[1]：非 Trident 管理 Yes[2]：由 Trident 管理，但不支援 PV 粒度 NO[3]：非 Trident 管理且不支援 PV 粒度 Yes[4]：支援原始區塊磁碟區 Yes[5]：由 Trident 支援

非 PV 粒度的功能會套用至整個 FlexVolume 和所有 PV（即共享 FlexVols 中的 qtree 或 LUN），並將共用一個共同的排程。

如上表所示、 `ontap-nas`和 `ontap-nas-economy`之間的許多功能都是相同的。但是、由於 `ontap-nas-economy`驅動程式限制了以每個 PV 精細度控制排程的能力、這可能會特別影響您的災難恢復和備份規劃。對於希望在 ONTAP 儲存設備上利用 PVC 複製功能的開發團隊、只有在使用 `ontap-nas`、`ontap-san`或 `ontap-san-economy`驅動程式時才有可能。

> 該 `solidfire-san` 驅動程式還能夠克隆 PVC 。

Cloud Volumes ONTAP 後端驅動程式

Cloud Volumes ONTAP 提供資料控制以及企業級儲存功能，適用於各種使用案例，包括檔案共用和區塊層級儲存，並支援 NAS 和 SAN 協定（NFS、SMB / CIFS 和 iSCSI）。Cloud Volumes ONTAP 的相容驅動程式為 `ontap-nas`、`ontap-nas-economy`、`ontap-san`和 `ontap-san-economy`。這些適用於 Cloud Volumes ONTAP for Azure、Cloud Volumes ONTAP for GCP。

Amazon FSx for ONTAP 後端驅動程式

Amazon FSx for NetApp ONTAP 讓您能夠利用您熟悉的 NetApp 特性、效能和管理功能，同時享受在 AWS 上儲存資料的簡易性、敏捷性、安全性和可擴充性。FSx for ONTAP 支援許多 ONTAP 檔案系統特性和管理 API。Cloud Volume ONTAP 的相容驅動程式包括 `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`與 `NetApp HCI/SolidFire` 平台配合使用的驅動程式可協助管理員根據 `QoS` 限制為 `Trident` 配置 `Element` 後端。如果您希望設計後端以對 `Trident` 配置的磁碟區設定特定的 `QoS` 限制，請在後端檔案中使用 `type`參數。管理員也可以使用 `limitVolumeSize`參數限制儲存上可建立的磁碟區大小。目前， `solidfire-san`驅動程式不支援 `Element` 儲存功能，例如磁碟區調整大小和磁碟區複製。這些操作需要透過 `Element Software Web UI` 手動完成。

SolidFire 驅動程式	快照	複本	多重附加	CHAP	QoS	調整大小	複寫
`solidfire-san`	是的	是的	是 [2]	是的	是的	是的	是

> 註腳：Yes[1]：不由 Trident 管理 Yes[2]：支援原始區塊磁碟區

Azure NetApp Files 後端驅動程式

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配置方法的更多資訊，請參見 "Azure NetApp Files 的 Trident 後端組態"。

Azure NetApp Files 驅動程式	快照	複本	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：是註腳：1[]：非由 Trident 管理

儲存類別設計

需要配置並套用個別儲存類別，才能建立 Kubernetes Storage Class 物件。本節將討論如何為您的應用程式設計儲存類別。

特定後端使用率

在特定的儲存類別物件中可以使用篩選功能來確定要與該特定儲存類別一起使用的儲存池或儲存池集合。可以在儲存類別中設定三組過濾器： `storagePools`、 `additionalStoragePools` 和/或 `excludeStoragePools`。

`storagePools` 參數有助於將儲存空間限制在符合任何指定屬性的儲存池集合中。 `additionalStoragePools` 參數用於擴展 Trident 用於資源配置的儲存池集合，以及透過屬性和 `storagePools` 參數選擇的儲存池集合。您可以單獨使用任一參數，也可以同時使用兩個參數，以確保選擇合適的儲存池集合。

`excludeStoragePools`參數用於專門排除符合屬性的已列出資源池集合。

模擬 **QoS** 原則

如果您希望設計儲存類別來模擬服務品質政策，請建立一個儲存類別，並將 `media`屬性設為 `hdd`或 `ssd`。根據儲存類別中提及的 `media`屬性，`Trident` 將選擇適當的後端來提供 `hdd`或 `ssd`集合體以符合媒體屬性，然後將磁碟區的配置導向至特定的集合體。因此，我們可以建立一個名為 `PREMIUM` 的儲存類別，其 `media`屬性設為 `ssd`，這可以歸類為 PREMIUM QoS 政策。我們可以建立另一個名為 STANDARD 的儲存類別，其媒體屬性設為 `hdd`，這可以歸類為 STANDARD QoS 政策。我們也可以使用儲存類別中的 `IOPS` 屬性將配置重新導向至 Element 設備，這可以定義為 QoS 政策。

根據特定功能使用後端

儲存類別可以設計用於指導在特定後端上進行磁碟區配置，這些後端啟用了精簡配置、厚配置、快照、複製和加密等功能。若要指定要使用的儲存，請建立儲存類別，並指定啟用了所需功能的相應後端。

虛擬資源池

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

虛擬池允許管理員在後端之上建立一層抽象層，並透過 Storage Classes 來引用這些後端，從而提高磁碟區在後端上的部署靈活性和效率。不同的後端可以使用相同的服務類別。此外，可以在同一個後端上建立多個具有不同特性的儲存池。當使用具有特定標籤的選擇器配置 Storage Class 時，Trident 會選擇一個與所有選擇器標籤都相符的後端來部署磁碟區。如果 Storage Class 選擇器標籤符合多個儲存池，Trident 將從中選擇其中一個來配置磁碟區。

虛擬資源池設計

在建立後端時，您通常可以指定一組參數。管理員無法使用相同的儲存憑證並搭配不同的參數組來建立另一個後端。隨著虛擬資源池的引入，這個問題已經得到緩解。虛擬資源池是在後端與 Kubernetes Storage Class 之間引入的一層抽象，讓管理員可以定義參數及標籤，這些標籤可以透過 Kubernetes Storage Classes 作為選擇器來引用，並且與後端無關。虛擬資源池可以為所有支援的 NetApp 後端與 Trident 一起定義。該清單包括 SolidFire/NetApp HCI、ONTAP，以及 Azure NetApp Files。

定義虛擬資源池時，建議不要嘗試在後端定義中重新排列現有虛擬資源池的順序。此外，也不建

議編輯 / 修改現有虛擬資源池的屬性，而是定義一個新的虛擬資源池。

模擬不同的服務等級 / **QoS**

可以設計虛擬池來模擬服務類別。使用 Cloud Volume Service for Azure NetApp Files 的虛擬池實作，讓我們來探討如何設定不同的服務類別。使用多個標籤設定 Azure NetApp Files 後端，代表不同的效能等級。將 `servicelevel`方面設定為適當的效能等級，並在每個標籤下新增其他所需的方面。現在建立不同的 Kubernetes Storage Classes，對應到不同的虛擬池。使用 `parameters.selector`欄位，每個 StorageClass 會指定哪些虛擬池可用於託管磁碟區。

指定特定的層面集

可以從單一儲存後端設計多個具有特定屬性的虛擬儲存池。為此、請為後端配置多個標籤、並在每個標籤下設定所需的屬性。現在使用 `parameters.selector` 欄位建立不同的 Kubernetes Storage Classes、以對應到不同的虛擬儲存池。在後端配置的磁碟區將具有所選虛擬儲存池中定義的屬性。

影響儲存資源配置的 **PVC** 特性

在建立 PVC 時，要求的儲存類別以外的某些參數可能會影響 Trident 配置決策過程。

存取模式

透過 PVC 請求儲存時，必填欄位之一是存取模式。所需的模式可能會影響選擇用於託管儲存請求的後端。

Trident 將嘗試根據下列矩陣將所使用的儲存協定與指定的存取方法進行比對。這與底層儲存平台無關。

	ReadWriteOnce	**ReadOnlyMany**	**ReadWriteMany**
iSCSI	是的	是的	是（Raw block）
NFS	是的	是的	是的

如果向未配置 NFS 後端的 Trident 部署提交 ReadWriteMany PVC 請求，則不會建立任何磁碟區。因此，請求者應使用適合其應用程式的存取模式。

磁碟區作業

修改持續磁碟區

持久性磁碟區在 Kubernetes 中除兩種例外情況外，都是不可變物件。創建後，可以修改其回收策略和大小。但是，這並不妨礙在 Kubernetes 外部修改磁碟區的某些方面。這樣做可能有助於為特定應用程式定製磁碟區，確保容量不會被意外耗盡，或者只是為了將磁碟區遷移到不同的儲存控制器。

目前，Kubernetes 內建的配置器不支援對 NFS、iSCSI 或 FC PV 進行磁碟區大小調整操作。Trident 支援擴充 NFS、iSCSI 和 FC 磁碟區。

PV 的連線詳細資料在建立後無法修改。

建立隨需磁碟區快照

Trident 支援隨需建立 Volume 快照，並可使用 CSI 架構從快照建立 PVC。快照提供了一種便捷的方法來維護資

料的時間點複本，並且其生命週期獨立於 Kubernetes 中的來源 PV。這些快照可用於複製 PVC。

從快照建立磁碟區

Trident 也支援從磁碟區快照建立PersistentVolumes。為此、只需建立PersistentVolumeClaim並提及 `datasource`作為需要從中建立磁碟區的所需快照。Trident 將使用快照中存在的資料建立磁碟區來處理此 PVC。透過此功能、可以跨區域複製資料、建立測試環境、完整替換損壞或毀損的正式作業磁碟區、或擷取特定檔案和目錄並將其傳輸到另一個附加的磁碟區。

在叢集中移動磁碟區

儲存管理員可以在 ONTAP 叢集中無中斷地將磁碟區在聚合和控制器之間移動，而不會對儲存使用者造成任何影響。只要目標聚合是 Trident 使用的 SVM 可以存取的聚合，此操作就不會影響 Trident 或 Kubernetes 叢集。需要注意的是，如果聚合是新添加到 SVM 的，則需要透過將其重新新增至 Trident 來重新整理後端。這將觸發 Trident 重新清點 SVM，以便識別新聚合。

然而，Trident 不支援在後端之間自動移動磁碟區。這包括在同一叢集中的 SVM 之間、叢集之間，或移動到不同的儲存平台（即使該儲存系統已連接到 Trident）。

如果將磁碟區複製到另一個位置，則可以使用磁碟區匯入功能將目前磁碟區匯入到 Trident 。

擴充磁碟區

Trident 支援調整 NFS、iSCSI 和 FC PV 的大小。這使用戶能夠直接透過 Kubernetes 層調整磁碟區的大小。所有主流 NetApp 儲存平台（包括 ONTAP 和 SolidFire/NetApp HCI 後端）均支援磁碟區擴充。為了允許日後擴展，請在與磁碟區關聯的 StorageClass 中將 `allowVolumeExpansion` 設定為 `true`。每當需要調整 Persistent Volume 的大小時，請將 Persistent Volume Claim 中的 `spec.resources.requests.storage` 註解編輯為所需的磁碟區大小。Trident 將自動處理儲存叢集上的磁碟區大小調整。

將現有磁碟區匯入 **Kubernetes**

磁碟區匯入功能可將現有儲存磁碟區匯入 Kubernetes 環境。目前 `ontap-nas`、 `ontap-nas-flexgroup`、 `solidfire-san`和 `azure-netapp-files`驅動程式支援此功能。此功能在將現有應用程式移植到 Kubernetes 或災難恢復情境中非常實用。

使用 ONTAP 和 `solidfire-san`驅動程式時，請使用指令 `tridentctl import volume <backend-name> <volume-name> -f /path/pvc.yaml`將現有磁碟區匯入 Kubernetes 以便由 Trident 管理。匯入磁碟區指令中使用的 PVC YAML 或 JSON 檔案指向一個儲存類別，該儲存類別會將 Trident 識別為佈建程式。使用 NetApp HCI/SolidFire 後端時，請確保磁碟區名稱是唯一的。如果磁碟區名稱重複，請將磁碟區複製為唯一名稱，以便磁碟區匯入功能能夠區分它們。

如果使用 `azure-netapp-files`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。這樣可以確保磁碟區參照的唯一性。

執行上述指令後、 Trident 將在後端尋找磁碟區並讀取其大小。它會自動新增（必要時會覆蓋）已配置 PVC 的磁碟區大小。然後、 Trident 會建立新的 PV 、 Kubernetes 會將 PVC 綁定到該 PV 。

如果容器的部署需要特定的匯入 PVC，則該容器將保持待處理狀態，直到透過磁碟區匯入流程綁定 PVC/PV 對。PVC/PV 對綁定後，如果沒有其他問題，容器即可啟動。

登錄服務

註冊表的部署和管理儲存已在 "netapp.io" 的 "部落格" 中記錄。

日誌服務

與其他 OpenShift 服務一樣，日誌服務也使用 Ansible 進行部署，設定參數由提供給 playbook 的清單檔案（也稱為 hosts）提供。將介紹兩種安裝方法：在初始 OpenShift 安裝期間部署日誌服務，以及在 OpenShift 安裝完成後部署日誌服務。

從 Red Hat OpenShift 3.9 版本開始，官方文件建議不要使用 NFS 作為日誌服務，因為有資料損壞的風險。這是基於 Red Hat 對其產品的測試結果。ONTAP NFS 伺服器不存在這些問題，可以輕鬆支援日誌部署。最終，日誌服務的協定選擇取決於您，但需要注意的是，這兩種協定在使用 NetApp 平台時都能很好地工作，如果您偏好 NFS，則沒有理由避免使用 NFS。

如果選擇將 NFS 與日誌服務一起使用，則需要設定 Ansible 變數 `openshift_enable_unsupported_configurations`為 `true`以防止安裝程序失敗。

開始使用

日誌服務可以選擇性地部署在應用程式和 OpenShift 叢集的核心操作上。如果您選擇部署操作日誌服務，透過指定變數 `openshift_logging_use_ops` 為 `true`，系統將建立兩個服務實例。控制操作日誌實例的變數包含「ops」參數，而控制應用程式日誌實例的變數則不包含。

根據部署方法配置 Ansible 變數至關重要，以確保底層服務使用正確的儲存。讓我們來看看每種部署方法的選項。

下表僅包含與日誌服務相關的儲存配置變數。您可以在 "Red Hat OpenShift 日誌文檔" 中找到其他選項，這些選項應根據您的部署情況進行檢視、配置和使用。

下表中的變數將使 Ansible playbook 使用提供的詳細資訊為日誌服務建立 PV 和 PVC 。與 OpenShift 安裝後使用元件安裝 playbook 相比，此方法彈性要差得多，但如果您已有可用的磁碟區，則此方法也是一種選擇。

變數	詳細資料
`openshift_logging_storage_kind`	設定為 `nfs` 讓安裝程式為日誌服務建立 NFS PV。
`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 值。
`openshift_logging_storage_nfs_directory`	NFS 匯出的掛載路徑。例如，如果磁碟區已連接為 `/openshift_logging`，則此變數應使用該路徑。
`openshift_logging_storage_volume_name`	要建立的 PV 的名稱，例如 `pv_ose_logs`。
`openshift_logging_storage_volume_size`	NFS 匯出的大小，例如 `100Gi`。

如果您的 OpenShift 叢集已在運行，並且 Trident 已部署和配置，則安裝程式可以使用動態配置來建立磁碟區。需要配置以下變數。

變數	詳細資料
`openshift_logging_es_pvc_dynamic`	設定為 true 以使用動態配置磁碟區。

變數	詳細資料
`openshift_logging_es_pvc_storage_class_name`	PVC 中將使用的儲存類別名稱。
`openshift_logging_es_pvc_size`	PVC 中請求的磁碟區大小。
`openshift_logging_es_pvc_prefix`	日誌記錄服務使用的 PVC 前置碼。
`openshift_logging_es_ops_pvc_dynamic`	設定為 `true` 以使用動態配置的磁碟區來執行運維日誌實例。
`openshift_logging_es_ops_pvc_storage_class_name`	運維日誌執行個體的儲存類別名稱。
`openshift_logging_es_ops_pvc_size`	ops 實例的磁碟區請求大小。
`openshift_logging_es_ops_pvc_prefix`	ops 執行個體 PVC 的前置詞。

部署日誌堆疊

如果您在初始 OpenShift 安裝過程中部署日誌記錄，則只需遵循標準部署流程即可。Ansible 將配置並部署所需的服務和 OpenShift 對象，以便在 Ansible 完成後服務即可使用。

但是，如果您在初始安裝後進行部署，則需要使用 Ansible 元件 playbook。此過程可能會因 OpenShift 版本而略有不同，因此請務必閱讀並遵循"Red Hat OpenShift Container Platform 3.11 文件"適用於您版本的相關說明。

指標服務

指標服務為管理員提供有關 OpenShift 叢集狀態、資源利用率和可用性的重要資訊。它對於 Pod 自動擴縮容功能也至關重要，許多組織也利用指標服務中的資料來實施成本分攤和/或成本展示應用。

與日誌服務和整個 OpenShift 類似，指標服務也使用 Ansible 進行部署。此外，與日誌服務類似，指標服務既可以在叢集初始設定期間部署，也可以在叢集運行後使用元件安裝方法進行部署。下表包含配置指標服務持久性儲存時的重要變數。

下表僅包含與指標服務相關的儲存配置變數。文件中還提供了許多其他選項，您應該根據部署情況進行查閱、配置和使用。

變數	詳細資料
`openshift_metrics_storage_kind`	設定為 `nfs` 讓安裝程式為日誌服務建立 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，如果磁碟區已連接為 `/openshift_metrics`，則此變數應使用該路徑。
`openshift_metrics_storage_volume_name`	要建立的 PV 的名稱，例如 `pv_ose_metrics`。
`openshift_metrics_storage_volume_size`	NFS 匯出的大小，例如 `100Gi`。

如果您的 OpenShift 叢集已在運行，並且 Trident 已部署和配置，則安裝程式可以使用動態配置來建立磁碟區。需要配置以下變數。

變數	詳細資料
`openshift_metrics_cassandra_pvc_prefix`	用於指標 PVC 的前綴。
`openshift_metrics_cassandra_pvc_size`	要求的磁碟區大小。
`openshift_metrics_cassandra_storage_type`	用於指標的儲存類型，必須設定為 dynamic，以便 Ansible 建立具有適當儲存類別的 PVC。
`openshift_metrics_cassanda_pvc_storage_class_name`	要使用的儲存類別名稱。

部署指標服務

在 hosts/inventory 檔案中定義好對應的 Ansible 變數後，即可使用 Ansible 部署服務。如果在 OpenShift 安裝時部署，PV 將自動建立並使用。如果使用元件 playbook 進行部署，在 OpenShift 安裝完成後，Ansible 會建立所需的 PVC，並在 Trident 為其配置儲存後部署服務。

上述變數和部署流程可能會隨 OpenShift 的每個版本而變化。請務必查看並遵循"Red Hat OpenShift 部署指南"您所用版本的相關說明，以便根據您的環境進行配置。

Trident 資料保護與災難恢復

了解使用 Trident 建立的 Trident 和磁碟區的保護和復原選項。您應該為每個具有持久性要求的應用程式製定資料保護和復原策略。

Trident 複寫與還原

您可以建立備份，以便在災難發生時還原 Trident。

Trident 複寫

Trident 使用 Kubernetes CRD 來儲存和管理自己的狀態，並使用 Kubernetes 叢集 etcd 來儲存其中繼資料。

步驟

1. 使用 "Kubernetes：備份 etcd 叢集" 備份 Kubernetes 叢集 etcd。
2. 將備份檔案放置在 FlexVol 磁碟區上

NetApp 建議您使用 SnapMirror 關係，將 FlexVol 所在的 SVM 保護到另一個 SVM。

Trident 恢復

使用 Kubernetes CRD 和 Kubernetes 叢集 etcd 快照、您可以還原 Trident。

步驟

1. 從目的地 SVM 掛載包含 Kubernetes etcd 資料檔案和憑證的磁碟區到將設定為主節點的主機。
2. 複製 Kubernetes 叢集下的所有必要憑證 `/etc/kubernetes/pki` 以及 etcd 成員檔案 `/var/lib/etcd`。
3. 使用 "Kubernetes：還原 etcd 叢集" 從 etcd 備份還原 Kubernetes 叢集。

4. 執行 `kubectl get crd`以驗證所有 Trident 自訂資源是否已啟動，並擷取 Trident 物件以驗證所有資料是否可用。

SVM 複製與復原

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

發生災難時，您可以啟動 SnapMirror 目標 SVM 開始提供資料服務。系統復原後，您可以切換回主 SVM。

關於此任務

使用 SnapMirror SVM 複製功能時，請考慮以下事項：

- 您應該為每個啟用 SVM-DR 的 SVM 建立一個獨立的後端。
- 配置儲存類別，僅在需要時選擇複寫後端，以避免將不需要複寫的磁碟區配置到支援 SVM-DR 的後端上。
- 應用程式管理員應了解與複寫相關的額外成本和複雜性，並在開始此程序之前仔細考慮其恢復計畫。

SVM 複製

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

SnapMirror 允許您設定選項來控制要複寫的內容。執行 使用 Trident 進行 SVM 恢復 時，您需要知道選擇了哪些選項。

- "-identity-preserve true" 複製整個 SVM 組態。
- "-discard-configs network" 不包括 LIF 和相關網路設定。
- "-identity-preserve false" 僅複製磁碟區和安全性組態。

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. 取消已排程和正在進行的 SnapMirror 傳輸、中斷複寫關係、停止來源 SVM、然後啟動 SnapMirror 目的地 SVM。
2. 如果您在設定 SVM 複寫時指定了 `-identity-preserve false`或 `-discard-config network`，請更新 Trident 後端定義檔中的 `managementLIF`和 `dataLIF`。
3. 確認 `storagePrefix`已存在於 Trident 後端定義檔中。此參數不可更改。省略 `storagePrefix`將導致後端更新失敗。
4. 使用下列命令更新所有必要的後端，以反映新的目的地 SVM 名稱：

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

5. 如果您指定了 `-identity-preserve false`或 `discard-config network`，則必須重新啟動所有應用程式 pod。

如果您指定了 `-identity-preserve true`，則當目標 SVM 啟動時，Trident 配置的所有磁碟區都會開始提供資料服務。

Volume 複寫與還原

Trident 無法設定 SnapMirror 複製關係，但儲存管理員可以使用 "ONTAP SnapMirror 複製和恢復" 來複製由 Trident 建立的磁碟區。

然後，您可以使用 "tridentctl volume import" 將復原的磁碟區匯入到 Trident 中。

此 `ontap-san-economy` 驅動程式支援磁碟區匯入。使用此 `ontap-nas-economy` 驅動程式匯入 Qtree 是一項技術預覽功能，不支援在生產環境中使用。此 `ontap-flexgroup-economy` 驅動程式不支援匯入。如需詳細資訊，請參閱 "匯入 Trident Volume"。

快照資料保護

您可以使用以下方法保護和還原資料：

- 外部快照控制器和 CRD 用於建立持久磁碟區（PV）的 Kubernetes 磁碟區快照。

  "Volume 快照"

- ONTAP 快照可還原磁碟區的全部內容，或還原個別檔案或 LUN。

  "ONTAP Snapshot"

使用 Trident 自動化具狀態應用程式的容錯移轉

Trident 的強制分離功能可讓您自動將磁碟區從 Kubernetes 叢集中不健康的節點分離，從而防止資料損壞並確保應用程式的可用性。此功能在節點無回應或因維護而離線的情況下尤其有用。

強制分離的詳細資訊

強制分離僅適用於 `ontap-san`、 `ontap-san-economy`、 `ontap-nas` 和 `ontap-nas-economy`。啟用強制分離前，必須在 Kubernetes 叢集上啟用非優雅節點關閉（NGNS）。Kubernetes 1.28 及更高版本預設啟用 NGNS。如需詳細資訊，請參閱 "Kubernetes：非正常節點關機"。

使用 `ontap-nas`或 `ontap-nas-economy`驅動程式時，需要在後端配置中將 `autoExportPolicy`參數設定為 `true`，以便 Trident 可以使用託管匯出原則限制對應用了污點的 Kubernetes 節點的存取。

由於 Trident 依賴 Kubernetes NGNS，因此在所有不可接受的工作負載重新調度之前，請勿從不健康節點上移除 `out-of-service`污點。貿然應用或移除污點可能會危及後端資料保護。

當 Kubernetes 叢集管理員將 `node.kubernetes.io/out-of-service=nodeshutdown:NoExecute`污點套用到節點並 `enableForceDetach`設定為 `true`時、Trident 將確定節點狀態並：

1. 停止對掛載到該節點的磁碟區的後端 I/O 存取。
2. 將 Trident 節點物件標記為 `dirty`（不適合新發佈）。

> Trident 控制器將拒絕新的發布磁碟區請求，直到 Trident 節點 pod 重新驗證節點是否符合條件（標記為 `dirty`之後）為止。即使叢集節點運作正常且已準備就緒，任何使用已掛載 PVC 調度的工作負載也不會被接受，直到 `Trident` 驗證節點 `clean（可以安全地進行新發布）為止。

當節點健全狀況恢復且污點被移除後、Trident 將：

1. 識別並清理節點上過期的已發布路徑。
2. 如果節點處於 `cleanable`狀態（已移除服務中斷污點，節點處於 `Ready`狀態），並且所有過期的已發布路徑都已清理，`Trident` 將重新接納該節點為 `clean，並允許向該節點發布新的磁碟區。

有關自動容錯移轉的詳細資訊

您可以透過與"節點健全狀況檢查 (NHC) operator"整合來自動執行強制分離程序。當節點發生故障時、NHC 會透過在 Trident 的命名空間中建立定義故障節點的 TridentNodeRemediation CR 來觸發 Trident 節點補救（TNR）並自動執行強制分離。TNR 僅在節點發生故障時建立、並在節點恢復上線或節點被刪除後由 NHC 移除。

故障節點 **Pod** 移除程序

自動故障轉移會選擇要從故障節點移除的工作負載。建立 TNR 時、TNR 控制器會將該節點標記為髒節點、阻止任何新的磁碟區發佈、並開始移除支援強制分離的 Pod 及其磁碟區附件。

所有受強制分離支援的磁碟區 / PVC 均受自動故障轉移支援：

- NAS 和使用自動匯出原則的 NAS 經濟型磁碟區（尚未支援 SMB）。
- SAN 和 SAN-economy 磁碟區。

請參閱 [強制分離的詳細資訊]。

預設行為：

- 使用強制分離（force-detach）支援的磁碟區的 Pod 將從故障節點移除。Kubernetes 會將這些 Pod 重新調度到健康的節點上。
- 使用不支援強制分離的磁碟區（包括非 Trident 磁碟區）的 Pod 不會從故障節點中移除。
- 無狀態 Pod （非 PVC）不會從故障節點中移除，除非設定了 pod 註解 `trident.netapp.io/podRemediationPolicy: delete`。

覆寫 **pod** 移除行為：

可使用 Pod 註解自訂 Pod 移除行為：`trident.netapp.io/podRemediationPolicy[retain, delete]`。發生故障轉移時，系統會檢查並使用這些註解。將註解應用於 Kubernetes 部署 / 複本集 Pod 規格，以防止註解在故障轉移後消失：

- `retain` - 在自動故障轉移期間，Pod 不會從故障節點中移除。
- `delete` - 在自動故障轉移期間，Pod 將從故障節點中移除。

這些註解可以應用於任何 pod 。

- 只有在支援強制分離的磁碟區上，發生故障的節點才會阻塞 I/O 操作。
- 對於不支援強制分離的磁碟區，存在資料損毀和多重附加問題的風險。

TridentNodeRemediation CR

TridentNodeRemediation（TNR）CR 定義了一個故障節點。TNR 的名稱就是故障節點的名稱。

TNR 範例：

```
apiVersion: trident.netapp.io/v1
kind: TridentNodeRemediation
metadata:
  name: <K8s-node-name>
spec: {}
```

TNR 狀態：使用以下命令檢視 TNR 的狀態：
`kubectl get tnr <name> -n <trident-namespace>`

TNR 可能處於下列幾種狀態之一：

- 修復中：
  - 停止對強制分離掛載到該節點所支援磁碟區的後端 I/O 存取。
  - Trident 節點物件被標記為髒（不適合新發佈）。
  - 從節點移除 Pod 和磁碟區附件
- *NodeRecoveryPending*：
  - 控制器正在等待節點重新上線。
  - 一旦節點上線，publish-enforcement 將確保節點乾淨且已準備好發布新磁碟區。
- 如果節點從 K8s 中刪除，TNR 控制器將移除 TNR 並停止協調。
- 成功：
  - 所有修復和節點恢復步驟均已成功完成。節點已清理乾淨，可以發布新的磁碟區。
- *Failed*：
  - 無法恢復的錯誤。錯誤原因在 CR 的 status.message 欄位中設定。

啟用自動容錯移轉

先決條件 ：

- 請確保在啟用自動故障轉移之前已啟用強制分離。如需詳細資訊，請參閱 [強制分離的詳細資訊]。
- 在 Kubernetes 叢集中安裝節點健康檢查（NHC）。
  - "安裝 operator-sdk"。

- 如果叢集中尚未安裝 Operator Lifecycle Manager（OLM），請安裝它：`operator-sdk olm install`
- 安裝 Node Health check Operator：`kubectl create -f https://operatorhub.io/install/node-healthcheck-operator.yaml`。

> 您也可以使用以下[整合自訂節點健全狀況檢查解決方案]章節中指定的其他方法來偵測節點故障。

如需詳細資訊，請參閱 "節點健康檢查 Operator"。

步驟

1. 在 Trident 命名空間中建立一個 NodeHealthCheck (NHC) CR，用於監控叢集中的工作節點。範例：

```
apiVersion: remediation.medik8s.io/v1alpha1
kind: NodeHealthCheck
metadata:
  name: <CR name>
spec:
  selector:
    matchExpressions:
      - key: node-role.kubernetes.io/control-plane
        operator: DoesNotExist
      - key: node-role.kubernetes.io/master
        operator: DoesNotExist
  remediationTemplate:
    apiVersion: trident.netapp.io/v1
    kind: TridentNodeRemediationTemplate
    namespace: <Trident installation namespace>
    name: trident-node-remediation-template
  minHealthy: 0 # Trigger force-detach upon one or more node failures
  unhealthyConditions:
    - type: Ready
      status: "False"
      duration: 0s
    - type: Ready
      status: Unknown
      duration: 0s
```

2. 在 `trident` 命名空間中應用節點健康檢查 CR。

```
kubectl apply -f <nhc-cr-file>.yaml -n <trident-namespace>
```

上述 CR 設定用於監控 K8s 工作節點的節點狀況 Ready: false 和 Unknown。當節點進入 Ready: false 或 Ready: Unknown 狀態時，將觸發 Automated-Failover。

> `unhealthyConditions` 在 CR 中使用 0 秒寬限期。這會導致自動容錯移轉在 K8s

設定節點條件 `Ready: false` 時立即觸發，該條件會在 `K8s` 失去節點的活動訊號後設定。`K8s` 預設在最後一次活動訊號後等待 `40` 秒，然後才設定 `Ready: false`。此寬限期可在 `K8s` 部署選項中自訂。

如需其他組態選項，請參閱 "Node-Healthcheck-Operator 說明文件"。

其他設定資訊

當 Trident 安裝時啟用了強制分離功能，Trident 命名空間中會自動建立兩個額外的資源，以方便與 NHC 整合：TridentNodeRemediationTemplate（TNRT）和 ClusterRole。

TridentNodeRemediationTemplate (TNRT)：

TNRT 可作為 NHC 控制器的範本，NHC 控制器可依需求使用 TNRT 產生 TNR 資源。

```
apiVersion: trident.netapp.io/v1
kind: TridentNodeRemediationTemplate
metadata:
  name: trident-node-remediation-template
  namespace: trident
spec:
  template:
    spec: {}
```

ClusterRole：

當啟用 force-detach 時，安裝過程中也會新增一個叢集角色。這會賦予 NHC 在 Trident 命名空間中對 TNRs 的權限。

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  labels:
    rbac.ext-remediation/aggregate-to-ext-remediation: "true"
  name: tridentnoderemediation-access
rules:
- apiGroups:
  - trident.netapp.io
  resources:
  - tridentnoderemediationtemplates
  - tridentnoderemediations
  verbs:
  - get
  - list
  - watch
  - create
```

```
    - update
    - patch
    - delete
```

K8s 叢集升級與維護

為防止任何故障轉移，請在 K8s 維護或升級期間暫停自動故障轉移，因為預計節點會當機或重新啟動。您可以透過修補 NHC CR（如上所述）來暫停其 CR：

```
kubectl patch NodeHealthCheck <cr-name> --patch
'{"spec":{"pauseRequests":["<description-for-reason-of-pause>"]}}' --type=merge
```

這將暫停自動故障轉移。若要重新啟用自動故障轉移、請在維護完成後從規格中移除 pauseRequests。

限制

- 僅對受強制分離支援的磁碟區，在發生故障的節點上才會阻止 I/O 操作。只有使用受強制分離支援的磁碟區 / PVC 的 Pod 才會被自動移除。
- 自動故障轉移和強制分離作業在 trident-controller pod 內部運作。如果託管 trident-controller 的節點發生故障，自動故障轉移將會延遲，直到 K8s 將 pod 遷移到健康的節點。

整合自訂節點健全狀況檢查解決方案

您可以使用其他節點故障偵測工具取代 Node Healthcheck Operator，以觸發自動故障轉移。為確保與自動故障轉移機制相容，您的自訂解決方案應：

- 當偵測到節點故障時，建立 TNR，使用故障節點的名稱作為 TNR CR 名稱。
- 當節點恢復且 TNR 處於 Succeeded 狀態時，刪除 TNR。

安全性

Trident 安全性建議

請按照此處列出的建議、確保您的 Trident 安裝安全可靠。

在其自己的命名空間中執行 **Trident**

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 非常重要，以確保可靠的儲存並封鎖潛在的惡意活動。

為了將其他應用程式和使用者與 Trident 分開，請務必將 Trident 安裝在其專屬的 Kubernetes 命名空間中 (`trident`。將 Trident 置於其專屬的命名空間可確保只有 Kubernetes 管理人員才能存取 Trident Pod 以及儲存在命名空間 CRD 物件中的成品（例如後端和 CHAP 密碼，如果適用）。您應該確保僅允許管理員存取 Trident 命名空間，從而存取 `tridentctl` 應用程式。

使用 **CHAP** 驗證與 **ONTAP SAN** 後端

Trident 支援基於 CHAP 的 ONTAP SAN 工作負載驗證（使用 `ontap-san`和 `ontap-san-economy`驅動程式

）。NetApp 建議 Trident 在主機和儲存後端之間使用雙向 CHAP 進行身份驗證。

對於使用 SAN 儲存驅動程式的 ONTAP 後端， Trident 可以透過 `tridentctl`設定雙向 CHAP 並管理 CHAP 使用者名稱和金鑰。請參閱"準備使用 ONTAP SAN 驅動程式配置後端"以了解 Trident 在 ONTAP 後端上的 CHAP 設定方式。

在 **NetApp HCI** 和 **SolidFire** 後端使用 **CHAP** 驗證

NetApp 建議部署雙向 CHAP，以確保主機與 NetApp HCI 及 SolidFire 後端之間的驗證。Trident 會使用一個包含每個租戶兩組 CHAP 密碼的 secret 物件。安裝 Trident 時，它會管理 CHAP 機密並將其儲存在對應 PV 的 `tridentvolume` CR 物件中。當你建立 PV 時，Trident 會使用 CHAP 機密來啟動 iSCSI 連線，並透過 CHAP 與 NetApp HCI 及 SolidFire 系統進行通訊。

由 Trident 建立的磁碟區不會與任何磁碟區存取群組關聯。

將 **Trident** 與 **NVE** 和 **NAE** 結合使用

NetApp ONTAP 提供靜態資料加密，以保護磁碟被盜、退回或重新利用時的敏感資料。如需詳細資料，請參閱 "設定 NetApp Volume Encryption 總覽"。

- 如果後端啟用了 NAE，則在 Trident 中配置的任何磁碟區都會啟用 NAE。
  - 您可以設定 NVE 加密標誌以 "" 建立啟用 NAE 的磁碟區。
- 如果後端未啟用 NAE，則在 Trident 中配置的任何磁碟區都會啟用 NVE，除非在後端組態中將 NVE 加密旗標設為 `false`（預設值）。

在啟用 NAE 的後端上使用 Trident 建立的磁碟區必須使用 NVE 或 NAE 加密。

- 您可以在 Trident 後端設定中將 NVE 加密標誌設定為 `true`，以覆蓋 NAE 加密並按磁碟區使用特定的加密金鑰。
- 在啟用 NAE 的後端上將 NVE 加密標誌設為 `false`會建立一個啟用 NAE 的磁碟區。您無法將 NVE 加密標誌設為 `false`來停用 NAE 加密。

- 您可以透過明確地將 NVE 加密標誌設為 `true` 來在 Trident 中手動建立 NVE 磁碟區。

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"

Trident Linux 統一金鑰設定 （LUKS）

您可以啟用 Linux Unified Key Setup （LUKS） 來加密 Trident 上的 ONTAP SAN 和 ONTAP SAN ECONOMY 磁碟區。Trident 支援對 LUKS 加密磁碟區進行密碼短語輪換和磁碟區擴展。

在 Trident 中，LUKS 加密磁碟區使用 aes-xts-plain64 密碼和模式，如 "NIST" 所建議。

ASA r2 系統不支援 LUKS 加密。如需 ASA r2 系統的相關資訊、請參閱"了解 ASA r2 儲存系統

"。

開始之前

- 工作節點必須安裝 cryptsetup 2.1 或更高版本（但低於 3.0）。如需詳細資訊，請造訪 "Gitlab：cryptsetup"。
- 出於效能考慮，NetApp 建議工作節點支援高級加密標準新指令集（AES-NI）。若要驗證是否支援 AES-NI，請執行下列命令：

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

如果沒有回傳任何內容，則表示您的處理器不支援 AES-NI。有關 AES-NI 的更多資訊，請造訪："Intel：Advanced Encryption Standard Instructions（AES-NI）"。

啟用 **LUKS** 加密

您可以使用 Linux Unified Key Setup (LUKS) 為 ONTAP SAN 和 ONTAP SAN ECONOMY 磁碟區啟用按磁碟區主機端加密。

步驟

1. 在後端配置中定義 LUKS 加密屬性。有關 ONTAP SAN 後端配置選項的更多資訊,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` 來定義使用 LUKS 加密的儲存池。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. 建立一個包含 LUKS 密碼的密鑰。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 重複資料刪除和壓縮功能。

匯入 **LUKS** 磁碟區的後端組態

若要匯入 LUKS 磁碟區、您必須在後端將 `luksEncryption`設為(`true。 luksEncryption`選項會告知 Trident 磁碟區是否符合 LUKS 規範(`true`或不符合 LUKS 規範(`false、如下列範例所示。

```
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** 磁碟區的 **PVC** 組態

若要動態匯入 LUKS 卷，請將註解 `trident.netapp.io/luksEncryption`設為 `true`並在 PVC 中包含啟用 LUKS 的儲存類，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
```

輪換 **LUKS** 密碼

您可以輪換 LUKS 密碼並確認輪換。

> 請務必在確認所有磁碟區、快照或金鑰均不再引用該密碼短語後再忘記它。如果引用的密碼短語遺失，您可能無法掛載該磁碟區，且資料將保持加密狀態且無法存取。

關於此任務

當指定新的 LUKS 密碼後建立掛載磁碟區的 Pod 時，就會發生 LUKS 密碼輪替。建立新 Pod 時，Trident 會將磁碟區上的 LUKS 密碼與金鑰中的活動密碼進行比較。

- 如果磁碟區上的密碼與密碼中的作用中密碼不符、就會發生輪替。
- 如果磁碟區上的密碼與密碼中的作用中密碼相符、則會忽略 `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現有的複雜密碼。

   磁碟區

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

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

   快照

```
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 金鑰，以指定新的和先前的密碼短語。確保 `previous-luke-passphrase-name`和 `previous-luks-passphrase`與先前的密碼短語一致。

```
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 並掛載磁碟區。這是啟動輪替所必需的。
5. 確認密碼已輪換。

   磁碟區

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

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

快照

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

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

結果

當磁碟區和快照上僅傳回新複雜密碼時，複雜密碼就會輪替。

> 如果傳回兩個密碼短語，例如 `luksPassphraseNames: ["B", "A"]`，則輪換尚未完成。您可以觸發一個新的 pod 來嘗試完成輪換。

啟用磁碟區擴充

您可以對 LUKS 加密磁碟區啟用磁碟區擴充。

步驟

1. 啟用 `CSINodeExpandSecret` 功能閘控（beta 1.25+）。如需詳細資訊，請參閱 "Kubernetes 1.25：使用 Secrets 實作 CSI 磁碟區的節點驅動擴展"。
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

啟動線上儲存擴充功能時，kubelet 會將對應的憑證傳遞給驅動程式。

Trident Kerberos 傳輸中加密

使用 Kerberos 傳輸中加密，您可以對託管叢集和儲存後端之間的流量啟用加密，從而提高資料存取安全性。

Trident 支援以 ONTAP 作為儲存後端的 Kerberos 加密：

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

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

使用內部部署 **ONTAP** 磁碟區設定傳輸中 **Kerberos** 加密

您可以為託管叢集和本機 ONTAP 儲存後端之間的儲存流量啟用 Kerberos 加密。

> 對於使用本機 ONTAP 儲存後端的 NFS 流量，僅支援使用 `ontap-nas` 儲存驅動程式進行 Kerberos 加密。

開始之前

- 請確保您可以使用 `tridentctl` 公用程式。
- 請確保您擁有 ONTAP 儲存後端的管理員存取權限。
- 請確保您知道要從 ONTAP 儲存後端共用的磁碟區名稱。
- 請確保您已設定 ONTAP 儲存虛擬機器以支援 NFS 磁碟區的 Kerberos 加密。請參閱 "在資料 LIF 上啟用 Kerberos"以取得指示。
- 請確保所有與 Kerberos 加密一起使用的 NFSv4 磁碟區都已正確設定。請參閱 NetApp NFSv4 域配置部分（第 13 頁） "NetApp NFSv4 增強功能與最佳實務指南"。

新增或修改 **ONTAP** 匯出原則

您需要為現有的 ONTAP 匯出策略新增規則，或建立新的匯出策略，以支援對 ONTAP 儲存 VM 根磁碟區以及與上游 Kubernetes 叢集共用的任何 ONTAP 磁碟區進行 Kerberos 加密。您新增的匯出策略規則或建立的新匯出策略需要支援以下存取協定和存取權限：

存取通訊協定

使用 NFS、NFSv3 和 NFSv4 存取協定設定匯出原則。

存取詳細資料

您可以根據磁碟區的需求、配置三種不同版本的 Kerberos 加密之一：

- **Kerberos 5** -（驗證和加密）
- **Kerberos 5i** -（驗證和加密，具有身分保護功能）
- **Kerberos 5p** -（驗證和加密，提供身分和隱私保護）

使用適當的存取權限設定 ONTAP 匯出原則規則。例如，如果叢集將使用 Kerberos 5i 和 Kerberos 5p 加密混合掛載 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
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

有關如何建立 ONTAP 匯出原則和匯出原則規則的資訊，請參閱下列文件：

- "建立匯出原則"
- "將規則新增至匯出原則"

建立儲存後端

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

關於此任務

建立配置 Kerberos 加密的儲存後端設定檔時，可以使用 `spec.nfsMountOptions` 參數指定三種不同版本的 Kerberos 加密之一：

- `spec.nfsMountOptions: sec=krb5` （驗證與加密）
- `spec.nfsMountOptions: sec=krb5i`（驗證與加密及身分保護）
- `spec.nfsMountOptions: sec=krb5p`（驗證與加密，具備身分與隱私保護）

只能指定一個 Kerberos 加密等級。如果在參數清單中指定了多個 Kerberos 加密等級，則僅使用第一個選項。

步驟

1. 在託管叢集上，使用以下範例建立儲存後端組態檔。將方括號 <> 中的值替換為您環境中的資訊：

```
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. 使用上一步建立的組態檔建立後端：

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

如果後端建立失敗，則表示後端配置存在問題。您可以執行以下命令查看日誌以確定原因：

```
tridentctl logs
```

在您識別並修正組態檔的問題後、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來配置具有 Kerberos 加密的磁碟區。

關於此任務

建立儲存類別物件時，可以使用 `mountOptions` 參數指定三種不同版本的 Kerberos 加密之一：

- `mountOptions: sec=krb5` （驗證與加密）
- `mountOptions: sec=krb5i`（驗證與加密及身分保護）
- `mountOptions: sec=krb5p`（驗證與加密，具備身分與隱私保護）

只能指定一個 Kerberos 加密等級。如果在參數清單中指定了多個 Kerberos 加密等級，則僅使用第一個選項。如果在儲存後端組態中指定的加密等級與在儲存類別物件中指定的加密等級不同，則以儲存類別物件為準。

步驟

1. 建立一個 StorageClass Kubernetes 物件，請參考以下範例：

```
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. 請確保已建立儲存類別：

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

您應該會看到類似以下內容的輸出：

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

配置磁碟區

建立儲存後端和儲存類別後，即可配置磁碟區。如需相關指示，請參閱 "配置磁碟區"。

使用 **Azure NetApp Files** 磁碟區設定傳輸中 **Kerberos** 加密

您可以為託管叢集與單一 Azure NetApp Files 儲存後端或 Azure NetApp Files 儲存後端虛擬池之間的儲存流量啟用 Kerberos 加密。

開始之前

- 請確保已在受管理的 Red Hat OpenShift 叢集上啟用 Trident。
- 請確保您可以使用 `tridentctl` 公用程式。
- 請確保您已為 Kerberos 加密準備好 Azure NetApp Files 儲存後端，請注意需求並遵循 "Azure NetApp Files 文件"中的說明。
- 請確保所有與 Kerberos 加密一起使用的 NFSv4 磁碟區都已正確設定。請參閱 NetApp NFSv4 域配置部分（第 13 頁） "NetApp NFSv4 增強功能與最佳實務指南"。

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。

關於此任務

建立配置 Kerberos 加密的儲存後端組態檔時，您可以將其定義為套用於下列兩個層級之一：

- 使用 `spec.kerberos`欄位的 **storage backend level**
- 使用 `spec.storage.kerberos`欄位的*虛擬資源池層級*

在虛擬資源池層級定義組態時，會使用儲存類別中的標籤來選取資源池。

無論在哪個級別、您都可以指定三種不同版本的 Kerberos 加密之一：

- `kerberos: sec=krb5` （驗證與加密）
- `kerberos: sec=krb5i`（驗證與加密及身分保護）

- `kerberos: sec=krb5p`（驗證與加密，具備身分與隱私保護）

步驟

1. 在託管叢集上，根據您需要定義儲存後端的位置（儲存後端等級或虛擬資源池等級），使用下列其中一個範例建立儲存後端組態檔。將方括號 <> 中的值替換為您環境中的資訊：

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
```

虛擬資源池層級範例

```
---
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. 使用上一步建立的組態檔建立後端：

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

如果後端建立失敗，則表示後端配置存在問題。您可以執行以下命令查看日誌以確定原因：

```
tridentctl logs
```

在您識別並修正組態檔的問題後、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來配置具有 Kerberos 加密的磁碟區。

步驟

1. 建立一個 StorageClass Kubernetes 物件，請參考以下範例：

```
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. 請確保已建立儲存類別：

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

您應該會看到類似以下內容的輸出：

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

配置磁碟區

建立儲存後端和儲存類別後，即可配置磁碟區。如需相關指示，請參閱 "配置磁碟區"。

版權資訊

Copyright © 2026 NetApp, Inc. 版權所有。台灣印製。非經版權所有人事先書面同意，不得將本受版權保護文件的任何部分以任何形式或任何方法（圖形、電子或機械）重製，包括影印、錄影、錄音或儲存至電子檢索系統中。

由 NetApp 版權資料衍伸之軟體必須遵守下列授權和免責聲明：

此軟體以 NETAPP「原樣」提供，不含任何明示或暗示的擔保，包括但不限於有關適售性或特定目的適用性之擔保，特此聲明。於任何情況下，就任何已造成或基於任何理論上責任之直接性、間接性、附隨性、特殊性、懲罰性或衍生性損害（包括但不限於替代商品或服務之採購；使用、資料或利潤上的損失；或企業營運中斷），無論是在使用此軟體時以任何方式所產生的契約、嚴格責任或侵權行為（包括疏忽或其他）等方面，NetApp 概不負責，即使已被告知有前述損害存在之可能性亦然。

NetApp 保留隨時變更本文所述之任何產品的權利，恕不另行通知。NetApp 不承擔因使用本文所述之產品而產生的責任或義務，除非明確經過 NetApp 書面同意。使用或購買此產品並不會在依據任何專利權、商標權或任何其他 NetApp 智慧財產權的情況下轉讓授權。

本手冊所述之產品受到一項（含）以上的美國專利、國外專利或申請中專利所保障。

有限權利說明：政府機關的使用、複製或公開揭露須受 DFARS 252.227-7013（2014 年 2 月）和 FAR 52.227-19（2007 年 12 月）中的「技術資料權利 - 非商業項目」條款 (b)(3) 小段所述之限制。

此處所含屬於商業產品和 / 或商業服務（如 FAR 2.101 所定義）的資料均為 NetApp, Inc. 所有。根據本協議提供的所有 NetApp 技術資料和電腦軟體皆屬於商業性質，並且完全由私人出資開發。 美國政府對於該資料具有非專屬、非轉讓、非轉授權、全球性、有限且不可撤銷的使用權限，僅限於美國政府為傳輸此資料所訂合約所允許之範圍，並基於履行該合約之目的方可使用。除非本文另有規定，否則未經 NetApp Inc. 事前書面許可，不得逕行使用、揭露、重製、修改、履行或展示該資料。美國政府授予國防部之許可權利，僅適用於 DFARS 條款 252.227-7015(b)（2014 年 2 月）所述權利。

商標資訊

NETAPP、NETAPP 標誌及 http://www.netapp.com/TM 所列之標章均為 NetApp, Inc. 的商標。文中所涉及的所有其他公司或產品名稱，均為其各自所有者的商標，不得侵犯。