最佳實務做法與建議
Astra Trident

NetApp
April 03, 2024

目錄

最佳實務做法與建議

部署

部署Astra 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。Astra Trident為Kubernetes叢集中的所有命名空間提供PV資源、並因此運用具有較高權限的服務帳戶。

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" 可讓系統管理員針對每個命名空間來實作全域和儲存類別的容量和物件數使用限制。此外、請使用 "範圍限制" 確保在將要求轉送至資源配置程式之前、永久虛擬機器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都應透過DNS存取、以達到最大的靈活度 "SVM-DR" 與Trident搭配使用。

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" 針對您的特定平台和ONTAP 版本、決定確切的限制。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的 `ontap-nas` 和 `ontap-san` 驅動程式會針對所建立的每個Kubernetes持續Volume（PV）來配置FlexVolume。。 `ontap-nas-economy` 每200個PV約需建立一個FlexVolume（可設定為50至300）。。 `ontap-san-economy` 每100個PV約需建立一個FlexVolume（可設定為50到200）。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>
```

的價值 `max-volumes` 視您環境的幾項特定條件而異：

- 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

。 `max-volumes` 值是ONTAP 指在整個叢集中所有節點上配置的總Volume、而非在個別ONTAP 的節點上配置的總Volume。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。

舉例ONTAP 來說、雙節點的「叢集」最多可裝載2000個FlexVolumes。將最大Volume數設為1250似乎非常合理。 不過、如果只有 "集合體" 從某個節點指派給SVM、或是從某個節點指派的集合體無法根據（例如、由於容量）進行資源配置、則另一個節點會成為所有Trident資源配置磁碟區的目標。這表示該節點可能會在之前達到磁碟區限制 `max-volumes` 達到該值、會影響Trident和其他使用該節點的Volume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Trident可建立之磁碟區的最大大小、請使用 `limitVolumeSize` 中的參數 `backend.json` 定義。

除了控制儲存陣列的磁碟區大小、您也應該善用Kubernetes功能。

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會驗證iSCSI連線ONTAP 、以CHAP作為後端。

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。 這對我們有幫助 "預防欺凌" 或失控的容器、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、請參閱 "保證QoS的處理量"。
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：

- 您應該設定 `qosPolicy` 鍵入 `defaults` 後端組態區塊。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、請參閱 "Sof 9.8 QoS命令ONTAP"。

限制**Kubernetes**叢集成員存取儲存資源

限制存取Trident所建立的NFS磁碟區和iSCSI LUN、是Kubernetes部署安全態勢的重要元件。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" 旨在防止這類入侵。 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。依預設不會拒絕這些功能、因此請務必使用停用該功能 "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` 建立匯出原則的CLI命令。ONTAP
- 使用新增規則至匯出原則 `vserver export-policy rule create` CLI命令。ONTAP

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` 適用於應用程式**SVM**

。 `showmount` 功能可讓NFS用戶端查詢SVM、取得可用NFS匯出的清單。部署至Kubernetes叢集的Pod可能會發出問題 `showmount -e` 針對資料LIF執行命令、並接收可用掛載的清單、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。

您應該停用 `showmount` 使用SVM層級ONTAP 的SVM CLI命令：

```
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	最小價值	預設值	最大價值（**4KB**）
最小IOPS	保證磁碟區效能等級。	50	50	15000

IOPS 參數	定義	最小價值	預設值	最大價值（4KB）
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。請參閱 "服務品質SolidFire" 如需QoS和效能的詳細資訊、請參閱。

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" 適用於最新版本。

《》ONTAP

- "NFS最佳實務與實作指南"
- "SAN管理指南"（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的準則、請務必與您的NetApp團隊合作並使用 "NetApp資料庫" 以尋找最新的文件。

整合Astra Trident

若要整合Astra Trident、下列設計與架構元素需要整合：驅動程式選擇與部署、儲存類別設計、虛擬集區設計、持續磁碟區宣告（PVc）、使用Astra Trident對儲存資源配置、磁碟區作業及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ONTAP 完整的FlexVolume。
- `ontap-nas-economy`：每個配置的PV都是qtree、每個FlexVolume可設定的qtree數量（預設值為200）。
- `ontap-nas-flexgroup`：每個PV均配置為完整ONTAP FlexGroup 的功能、並使用指派給SVM的所有集合體。
- `ontap-san`：每個已配置的PV都是其FlexVolume中的LUN。
- `ontap-san-economy`：配置的每個PV都是一個LUN、每個FlexVolume有可設定的LUN數量（預設值為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、並非所有功能都透過Astra Trident公開。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。

ASNAS驅動程式ONTAP	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	是註腳：3[]	是註腳：3[]	是註腳：5[]	是的	是註腳：3[]	是的	是註腳：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[]

Astra Trident提供2個ONTAP SAN驅動程式來支援功能如下所示。

支援SAN驅動程式ONTAP	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	是註腳：3[]	是的	是註腳：3[]

> 上述表格的註腳：
>
> 是註腳： 1[] ：不由 Astra Trident 管理
>
> 是註腳： 2[] ：由 Astra Trident 管理、但非 PV Granular 管理
>
> 是註腳： 3[] ：不由 Astra Trident 管理、而非 PV Granular 管理
>
> 是註腳： 4[] ：支援原始區塊磁碟區
>
> 是註腳： 5[] ： Astra 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、兩者之間的大部分功能 `ontap-nas` 和 `ontap-nas-economy` 相同。不過、因為 `ontap-nas-economy` 驅動程式會限制以每個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與備份規劃。如果開發團隊想要在ONTAP 不支援的儲存設備上使用永久虛擬複製功能、只有在使用時才有可能做到這一點 `ontap-nas`、 `ontap-san` 或 `ontap-san-economy` 驅動程式：

。 `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 的相容驅動程式就是 `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 的相容驅動程式就是 `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平台搭配使用的驅動程式、可協助管理員根據QoS限制、設定Trident的元素後端。如果您想要設計後端、以便針對Trident提供的磁碟區設定特定的QoS限制、請使用 `type` 後端檔案中的參數。管理員也可以使用來限制儲存設備上可建立的磁碟區大小 `limitVolumeSize` 參數。目前、不支援磁碟區大小調整和磁碟區複寫等元素儲存功能 `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[] ：不由 Astra 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**

Astra Trident使用 `azure-netapp-files` 管理的驅動程式 "Azure NetApp Files" 服務：

如需此驅動程式及其設定方式的詳細資訊、請參閱 "Astra Trident的Azure NetApp Files 後端組態、適用於"。

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：
>
> 是註腳： 1[] ：不由 Astra Trident 管理

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

Astra Trident使用 `gcp-cvs` 在Cloud Volumes Service Google Cloud上連結至該解決方案的驅動程式。

。 `gcp-cvs` 驅動程式使用虛擬資源池來抽象化後端、並允許Astra Trident決定磁碟區放置。系統管理員會在中

定義虛擬資源池 `backend.json` 檔案：儲存類別會使用選取器來依標籤識別虛擬資源池。

- 如果在後端定義了虛擬資源池、Astra Trident會嘗試在Google Cloud儲存資源池中建立該虛擬資源池所限制的磁碟區。
- 如果後端未定義虛擬資源池、Astra Trident會從該地區可用的儲存資源池中選取Google Cloud儲存資源池。

若要在Astra Trident上設定Google Cloud後端、您必須指定 `projectNumber`、 `apiRegion`和 `apiKey` 在後端檔案中。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 有關Google Cloud服務類型與服務層級的詳細資訊、請參閱 "瞭解Astra Trident對CVS for GCP的支援"。

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

- 複寫不由Astra 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。可在儲存類別中設定三組篩選器： `storagePools`、 `additionalStoragePools`和/或 `excludeStoragePools`。

。 `storagePools` 參數有助於將儲存區限制在符合任何指定屬性的集區集合。。 `additionalStoragePools` 參數可用來擴充Astra Trident將用於資源配置的集區集區集區集區、以及由屬性和所選的集區集區集區集區集區集區集區集區集區 `storagePools` 參數。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。

。 `excludeStoragePools` 參數用於明確排除所列的符合屬性的集區集區集區集區。

模擬**QoS**原則

如果您想要設計儲存類別來模擬服務品質原則、請使用建立儲存類別 `media` 屬性為 `hdd` 或 `ssd`。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服務 `hdd` 或 `ssd` 集合體以符合媒體屬性、然後將磁碟區的資源配置導向特定的集合體。因此、我們可以建立一個儲存等級Premium `media` 屬性設為 `ssd` 可歸類為優質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Astra Trident後端均可使用虛擬資源池。您可以使用Astra Trident提供的任何驅動程式、為任何後端定義虛擬資源池。

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、Astra 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資源池、Astra Trident會選擇其中一個來配置磁碟區。

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。可使用Astra Trident為所有支援的NetApp後端定義虛擬資源池。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。設定 `servicelevel` 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 `parameters.selector` 欄位中、每個StorageClass會呼叫哪些虛擬資源池可用於裝載Volume。

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、請使用建立不同的Kubernetes儲存類別 `parameters.selector` 對應至不同虛擬資源池的欄位。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

在建立永久虛擬儲存設備時、超出所要求儲存類別的部分參數可能會影響Astra Trident的資源配置決策程序。

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。

Astra Trident會嘗試將所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樹狀目錄內建資源配置程式目前不支援NFS或iSCSI PV的磁碟區大小調整作業。Astra Trident支援同時擴充NFS和iSCSI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

Astra Trident支援隨需磁碟區快照建立、並使用csi架構從快照建立PVCS。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

Astra Trident也支援從Volume快照建立PersistentVolumes。為達成此目的、只要建立一個PeristentVolume Claim並提及即可 `datasource` 所需的快照、以便建立磁碟區。Astra Trident會利用快照上的資料建立磁碟區、以處理此永久虛擬磁碟。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。此作業不會影響Astra Trident或Kubernetes叢集、只要目的地Aggregate是Astra Trident所使用的SVM能夠存取的集合體。重要的是、如果新將Aggregate新增至SVM、則需要重新將其新增至Astra Trident來重新整理後端。這會觸發Astra Trident重新清查SVM、以便辨識新的Aggregate。

然而、Astra Trident並不支援跨後端移動磁碟區。這包括在同一個叢集內的SVM之間、叢集之間或不同的儲存平台（即使該儲存系統是連接至Astra Trident的儲存系統）。

如果將磁碟區複製到其他位置、則磁碟區匯入功能可用於將目前的磁碟區匯入Astra Trident。

展開**Volume**

Astra Trident支援調整NFS和iSCSI 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。若要允許稍後擴充、請設定 `allowVolumeExpansion` 至 `true` 在與磁碟區相關的StorageClass中。每當需要調整持續Volume的大小時、請編輯 `spec.resources.requests.storage` 持

續Volume中的註釋會宣告為所需的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。目前支援此功能 `ontap-nas`、 `ontap-nas-flexgroup`、 `solidfire-san`、 `azure-netapp-files`和 `gcp-cvs` 驅動程式：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 the功能時 `solidfire-san` 驅動程式、請使用命令 `tridentctl import volume <backend-name> <volume-name> -f /path/pvc.yaml` 將現有磁碟區匯入Kubernetes、由Astra Trident管理。匯入Volume命令中使用的PVc Yaml或Json檔案會指向儲存類別、以將Astra Trident識別為資源配置程式。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。

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

執行上述命令時、Astra Trident會在後端找到磁碟區並讀取其大小。它會自動新增（並在必要時覆寫）已設定的PVC Volume Size 。Astra Trident接著會建立新的PV、Kubernetes則會將PVc繫結至PV。

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。

部署**OpenShift**服務

OpenShift加值叢集服務可為叢集管理員和託管的應用程式提供重要功能。這些服務所使用的儲存設備可以使用節點本機資源進行資源配置、但這通常會限制服務的容量、效能、可恢復性及永續性。運用企業儲存陣列來提供這些服務的容量、可大幅改善服務品質、不過OpenShift和儲存管理員應該密切合作、以決定每個服務的最佳選項。Red Hat文件應充分運用、以判斷需求、並確保符合規模調整與效能需求。

登錄服務

登錄的儲存設備部署與管理已記錄在中 "NetApp.IO" 在中 "部落格"。

記錄服務

如同其他OpenShift服務、記錄服務是使用Ansible搭配庫存檔案所提供的組態參數（即k.a.）來部署主機、提供給教戰手冊。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 OpenShift 之後部署記錄已安裝。

> 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。

如果您選擇使用NFS搭配記錄服務、則必須設定Ansible變數 `openshift_enable_unsupported_configurations` 至 `true` 以避免安裝程式失敗。

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

指定變數 `openshift_logging_use_ops` 做為 `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。您可以在中找到其他選項 "RedHat OpenShift記錄文件" 應根據您的部署情況來審查、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` 若要讓安裝程式為記錄服務建立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資料LIF。
`openshift_logging_storage_nfs_directory`	NFS匯出的掛載路徑。例如、如果該磁碟區的輔助狀態為 `/openshift_logging`您可以將該路徑用於此變數。
`openshift_logging_storage_volume_name`	名稱、例如 `pv_ose_logs`的。
`openshift_logging_storage_volume_size`	例如、NFS匯出的大小 `100Gi`。

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`	設定為 `true` 使用動態資源配置的磁碟區來執行作業記錄執行個體。
`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。不同版本的OpenShift可能會稍微改變此程序、因此請務必閱讀並遵循 "RedHat OpenShift Container Platform 3.11文件" 適用於您的版本。

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` 若要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的資料LIF。
`openshift_metrics_storage_nfs_directory`	NFS匯出的掛載路徑。例如、如果該磁碟區的輔助狀態為 `/openshift_metrics`您可以將該路徑用於此變數。
`openshift_metrics_storage_volume_name`	名稱、 例如 `pv_ose_metrics`的。
`openshift_metrics_storage_volume_size`	例如、NFS匯出的大小 `100Gi`。

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。如果您使用元件教戰手冊進行部署、則在安裝 OpenShift 之後、 Ansible 會建立任何需要的 PVCS 、並在 Astra Trident 為其配置儲存設備之後、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 "RedHat的OpenShift部署指南" 以供您的環境使用。

資料保護與災難恢復

瞭解 Astra Trident 的保護與恢復選項、以及使用 Astra Trident 建立的 Volume 。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。

Astra Trident 複寫與還原

您可以建立備份、在發生災難時還原 Astra Trident 。

Astra Trident 複寫

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

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 上。

> 我們建議您保護 FlexVol 所在的 SVM 、並與另一個 SVM 建立 SnapMirror 關係。

Astra Trident 恢復

使用 Kubernetes CRD 和 Kubernetes 叢集 etcd 快照、您可以復原 Astra 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` 若要驗證所有 Trident 自訂資源都已出現、請擷取 Trident 物件、以驗證所有資料是否可用。

SVM 複寫與還原

Astra 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。您必須知道自己在進行預先設定時所選擇的選項 使用 Astra Trident 進行 SVM 恢復。

- "-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 **Astra Trident** 進行 **SVM** 恢復

Astra 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 複寫時、請更新 `managementLIF` 和 `dataLIF` 在 Trident 後端定義檔案中。
3. 確認 `storagePrefix` 存在於 Trident 後端定義檔案中。此參數無法變更。省略 `storagePrefix` 將導致後端更新失敗。
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`、您必須退回所有應用程式 Pod 。

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

Volume 複寫與還原

Astra 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 Astra Trident 所建立的 Volume 。

然後、您可以使用將復原的磁碟區匯入 Astra Trident "tridentctl Volume 匯入"。

> 匯入不受支援 `ontap-nas-economy`、 `ontap-san-economy`或 `ontap-flexgroup-economy` 驅動程式：

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"

Astra Control Center 應用程式複寫

使用 Astra Control 、您可以使用 SnapMirror 的非同步複寫功能、將資料和應用程式變更從一個叢集複寫到另一個叢集。

"Astra Control ：使用 SnapMirror 技術將應用程式複寫到遠端系統"

安全性

安全性

請使用此處列出的建議、確保您的Astra Trident安裝安全無虞。

在自己的命名空間中執行**Astra Trident**

防止應用程式、應用程式管理員、使用者及管理應用程式存取Astra Trident物件定義或Pod、以確保可靠的儲存並封鎖潛在的惡意活動、這一點非常重要。

若要將其他應用程式和使用者與Astra Trident區隔開、請務必在自己的Kubernetes命名空間中安裝Astra Trident (`trident`）。將Astra Trident放在自己的命名空間中、可確保只有Kubernetes管理人員能夠存取Astra Trident Pod、以及儲存在命名式CRD物件中的成品（例如後端和CHAP機密）。
您應確保只允許系統管理員存取Astra 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

Astra Trident支援ONTAP 以CHAP為基礎的驗證功能、適用於各種不實的SAN工作負載（使用 `ontap-san` 和 `ontap-san-economy` 驅動程式）。NetApp建議使用雙向CHAP搭配Astra Trident進行主機與儲存後端之間的驗證。

針對使用SAN儲存驅動程式的幕後作業、Astra Trident可設定雙向CHAP、並透過管理CHAP使用者名稱和機密ONTAP `tridentctl`。
請參閱 "請按這裡" 瞭解Astra Trident如何在ONTAP 不景點上設定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。Astra Trident使用每個租戶包含兩個CHAP密碼的秘密物件。安裝 Astra Trident 時、它會管理 CHAP 機密、並將其儲存在中 `tridentvolume` 對應PV的CR物件。建立 PV 時、 Astra 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。

> Astra Trident 所建立的磁碟區不會與任何 Volume Access Group 相關聯。

使用**Astra Trident**搭配**NVE**和**NAE**

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、請參閱 "設定NetApp Volume Encryption總覽"。

- 如果在後端啟用NAE、則Astra Trident中配置的任何磁碟區都將啟用NAE。
- 如果後端未啟用NAE、則在Astra Trident中配置的任何Volume都會啟用NVE、除非您將NVE加密旗標設為 `false` 在後端組態中。

> 在啟用NAE的後端Astra Trident中建立的磁碟區、必須加密NVE或NAE。
>
> - 您可以將NVE加密旗標設為 `true` 在Trident後端組態中、覆寫NAE加密、並以每個磁碟區為基礎使用特定的加密金鑰。
> - 將NVE加密旗標設定為 `false` 在啟用NAE的後端上、將會建立啟用NAE的Volume。您無法將NVE加密旗標設為、以停用NAE加密 `false`。

- 您可以在Astra Trident中手動建立NVE磁碟區、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：

- "支援SAN組態選項ONTAP"
- "ASNAS組態選項ONTAP"

Linux統一化金鑰設定（LUKS）

您可以在ONTAP Astra Trident上啟用Linux Unified Key Setup（LUKS）來加密支援的SAN和ONTAP 支援的SAN經濟版磁碟區。Astra Trident支援使用密碼的輪替和磁碟區擴充、適用於使用LUKS加密的磁碟區。

在Astra Trident中、LUKS加密的磁碟區會依照所建議的方式使用AES-XTS-plain64 cypher和模式 "NIST"。

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）。如需詳細資訊、請造訪 "Gitlab：密碼設定"。
- 基於效能考量、我們建議工作節點支援進階加密標準新增指令（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、請造訪： "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。如需ONTAP 有關支援不支援SAN的後端組態選項的詳細資訊、請參閱 "支援SAN組態選項ONTAP"。

```
"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` 使用LUKS加密定義儲存資源池。例如：

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: luks
provisioner: netapp.io/trident
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 、您必須設定 `luksEncryption` 至(`true` 在後端。。 `luksEncryption` 選項會告訴 Astra Trident 、磁碟區是否符合 LUKS 標準 (`true`）或不符合 LUKS 規範 (`false`）如以下範例所示。

```
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'
```

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、Astra Trident會比較磁碟區上的LUKS通關密碼與機密中的作用中通關密碼。

- 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。
- 如果磁碟區上的通關密碼與機密中的作用中通關密碼相符 `previous-luks-passphrase` 參數被忽略。

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` 和 `previous-luks-passphrase` 請與先前的通關密碼相符。

```
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。

例如、如果傳回兩個複雜密碼 `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` 功能閘道（beta 1.25 +）。請參閱 "Kubernetes 1.25：使用Secrets進行節點導向的SCSI Volume擴充" 以取得詳細資料。
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: netapp.io/trident
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。

版權資訊

商標資訊