



使用Astra Trident

Astra Trident

NetApp
April 16, 2024

目錄

使用Astra Trident	1
設定後端	1
使用kubectl建立後端	64
以KECBECVL執行後端管理	69
使用tridentctl執行後端管理	71
在後端管理選項之間切換	72
管理儲存類別	79
執行Volume作業	81
準備工作節點	106
自動準備工作節點	109
監控Astra Trident	109

使用Astra Trident

設定後端

後端定義了Astra Trident與儲存系統之間的關係。它告訴Astra Trident如何與該儲存系統通訊、以及Astra Trident如何從該儲存系統配置磁碟區。Astra Trident會自動從後端提供符合儲存類別所定義需求的儲存資源池。深入瞭解如何根據您擁有的儲存系統類型來設定後端。

- "[設定Azure NetApp Files 一個靜態後端](#)"
- "[設定Cloud Volumes Service 適用於Google Cloud Platform後端的功能](#)"
- "[設定NetApp HCI 一個不只是功能的SolidFire 後端](#)"
- "[使用ONTAP 功能不一的Cloud Volumes ONTAP NAS驅動程式來設定後端](#)"
- "[使用ONTAP 不支援的Cloud Volumes ONTAP SAN驅動程式來設定後端](#)"
- "[使用Astra Trident搭配Amazon FSX for NetApp ONTAP 解決方案](#)"

設定Azure NetApp Files 一個靜態後端

瞭解Azure NetApp Files 解如何使用提供的範例組態、將靜態（anf）設定為Astra Trident安裝的後端。



該支援服務不支援低於100 GB的磁碟區。Azure NetApp Files如果要求較小的磁碟區、Astra Trident會自動建立100-GB磁碟區。

您需要的產品

若要設定及使用 "[Azure NetApp Files](#)" 後端、您需要下列項目：

- Azure訂閱提供的「SubscriptionID」 Azure NetApp Files（含功能不支援的功能）。
- 來自於的「TenantId」、「clientID」和「clientSecret」 "[應用程式註冊](#)" 在Azure Active Directory中、具備Azure NetApp Files 充分的權限執行此功能。應用程式登錄應使用Azure預先定義的「擁有者」或「貢獻者」角色。



若要深入瞭解Azure內建角色、請參閱 "[Azure文件](#)"。

- Azure的「位置」至少包含一個 "[委派的子網路](#)"。從Trident 22.01起、「位置」參數是後端組態檔最上層的必填欄位。會忽略虛擬資源池中指定的位置值。
- 如果Azure NetApp Files 您是第一次使用或是在新的位置使用、則需要進行一些初始組態。請參閱 "[快速入門指南](#)"。

關於這項工作

Trident會根據後端組態（子網路、虛擬網路、服務層級和位置）、在所要求位置可用的容量集區上建立ANF磁碟區、並符合所要求的服務層級和子網路。



附註：Astra Trident不支援手動QoS容量資源池。

後端組態選項

如需後端組態選項、請參閱下表：

參數	說明	預設
「分度」		永遠為1
「storageDriverName」	儲存驅動程式名稱	「Azure - NetApp-Files」
「後端名稱」	自訂名稱或儲存後端	驅動程式名稱+「_」+隨機字元
《訂閱ID》	Azure訂閱的訂閱ID	
「TenantId」	應用程式註冊的租戶ID	
"clientId"	應用程式註冊的用戶端ID	
「客戶機密」	應用程式註冊的用戶端機密	
《服務層級》	其中一種是「標準」、「高級」或「超高」	"" (隨機)
位置	要建立新磁碟區的Azure位置名稱	
"來源群組"	用於篩選已探索資源的資源群組清單	「[]」 (無篩選器)
《netappAccounts》	篩選探索資源的NetApp帳戶清單	「[]」 (無篩選器)
《容量Pools》	用於篩選已探索資源的容量集區清單	「[]」 (無篩選器、隨機)
「虛擬化網路」	具有委派子網路的虛擬網路名稱	"
《Subnet》	委派給「microsoft.Netapp/volumes」的子網路名稱	"
「nfsMountOptions」	精細控制NFS掛載選項。	"nfsves=3"
《限制Volume大小》	如果要求的磁碟區大小高於此值、則資源配置失敗	"" (預設不強制執行)
「DebugTraceFlags」	疑難排解時要使用的偵錯旗標。範例：「{"API":假、「方法」:真、「探索」:true}。除非您正在進行疑難排解並需要詳細的記錄傾印、否則請勿使用此功能。	null



如果嘗試建立永久虛擬基礎架構時發生「找不到容量資源池」錯誤、您的應用程式登錄可能沒有相關的必要權限和資源（子網路、虛擬網路、容量資源池）。Astra Trident會在啟用偵錯功能時、記錄在建立後端時所探索到的Azure資源。請務必檢查是否使用適當的角色。



如果您想使用NFS 4.1版掛載磁碟區、可以在以逗號分隔的掛載選項清單中加入「nfsves=4」、以選擇NFS v4.1。儲存類別中設定的任何掛載選項、都會覆寫在後端組態檔中設定的掛載選項。

可以使用短名稱或全限定名稱來指定"sourceGroups"、"netappAccounts"、"capacityPools"、"virtualNetwork"（虛擬化網路）和"shubnet"的值。簡短名稱可能會比對同名的多個資源、因此在大多數情況下、建議使用完整名稱。「資源群組」、「netappAccounts」和「capacityPools」值是篩選器、可將探索到的資源集限制在此儲存

後端可用的資源集、並可任意組合指定。完整名稱格式如下：

類型	格式
資源群組	<資源群組>
NetApp帳戶	資源群組//<NetApp帳戶>
容量資源池	資源群組//<NetApp帳戶>/<容量資源池>
虛擬網路	資源群組//<虛擬網路>
子網路	資源群組//<虛擬網路>/<子網路>

您可以在組態檔的特殊區段中指定下列選項、以控制預設的每個Volume佈建方式。請參閱下列組態範例。

參數	說明	預設
「匯出規則」	新磁碟區的匯出規則	「0.00.0.0/0」
「napshotDir	控制.snapshot目錄的可見度	"假"
《大小》	新磁碟區的預設大小	100公克
「unixPermissions」	新磁碟區的UNIX權限（4個八進位數字）	""（預覽功能、訂閱時需要白名單）

「匯出規則」值必須是以逗號分隔的清單、以CIDR表示法列出所有的IPv4位址或IPv4子網路組合。



對於在ANF後端上建立的所有磁碟區、Astra Trident會將儲存資源池上的所有標籤複製到資源配置時的儲存磁碟區。儲存管理員可以定義每個儲存資源池的標籤、並將儲存資源池中建立的所有磁碟區分組。這是根據後端組態中提供的一組可自訂標籤、方便區分磁碟區的方法。

範例1：最低組態

這是絕對最低的后端組態。使用此組態、Astra Trident會在設定的位置探索所有NetApp帳戶、容量集區和委派給ANF的子網路、並隨機將新磁碟區放在其中一個集區和子網路上。

當您剛開始使用ANF並嘗試各種功能時、這種組態是理想的選擇、但實際上您想要為您所配置的磁碟區提供額外的範圍。

```
{
  "version": 1,
  "storageDriverName": "azure-netapp-files",
  "subscriptionID": "9f87c765-4774-fake-ae98-a721add45451",
  "tenantID": "68e4f836-edc1-fake-bff9-b2d865ee56cf",
  "clientID": "dd043f63-bf8e-fake-8076-8de91e5713aa",
  "clientSecret": "SECRET",
  "location": "eastus"
}
```

範例2：使用容量集區篩選器的特定服務層級組態

此後端組態可將磁碟區置於Azure的「東向」位置、位於「超」容量資源池中。Astra Trident會自動探索該位置委派給ANF的所有子網路、並隨機在其中一個磁碟區上放置新磁碟區。

```
{
  "version": 1,
  "storageDriverName": "azure-netapp-files",
  "subscriptionID": "9f87c765-4774-fake-ae98-a721add45451",
  "tenantID": "68e4f836-edc1-fake-bff9-b2d865ee56cf",
  "clientID": "dd043f63-bf8e-fake-8076-8de91e5713aa",
  "clientSecret": "SECRET",
  "location": "eastus",
  "serviceLevel": "Ultra",
  "capacityPools": [
    "application-group-1/account-1/ultra-1",
    "application-group-1/account-1/ultra-2"
  ],
}
```

範例3：進階組態

此後端組態可進一步將磁碟區放置範圍縮小至單一子網路、並修改部分Volume資源配置預設值。

```

{
  "version": 1,
  "storageDriverName": "azure-netapp-files",
  "subscriptionID": "9f87c765-4774-fake-ae98-a721add45451",
  "tenantID": "68e4f836-edc1-fake-bff9-b2d865ee56cf",
  "clientID": "dd043f63-bf8e-fake-8076-8de91e5713aa",
  "clientSecret": "SECRET",
  "location": "eastus",
  "serviceLevel": "Ultra",
  "capacityPools": [
    "application-group-1/account-1/ultra-1",
    "application-group-1/account-1/ultra-2"
  ],
  "virtualNetwork": "my-virtual-network",
  "subnet": "my-subnet",
  "nfsMountOptions": "vers=3,proto=tcp,timeo=600",
  "limitVolumeSize": "500Gi",
  "defaults": {
    "exportRule": "10.0.0.0/24,10.0.1.0/24,10.0.2.100",
    "snapshotDir": "true",
    "size": "200Gi",
    "unixPermissions": "0777"
  }
}
=====
}
}

```

範例4：虛擬儲存池組態

此後端組態可在單一檔案中定義多個儲存集區。當您有多個容量集區支援不同的服務層級、而且想要在Kubernetes中建立代表這些層級的儲存類別時、這很有用。

```

{
  "version": 1,
  "storageDriverName": "azure-netapp-files",
  "subscriptionID": "9f87c765-4774-fake-ae98-a721add45451",
  "tenantID": "68e4f836-edc1-fake-bff9-b2d865ee56cf",
  "clientID": "dd043f63-bf8e-fake-8076-8de91e5713aa",
  "clientSecret": "SECRET",
  "location": "eastus",
  "resourceGroups": ["application-group-1"],
  "nfsMountOptions": "vers=3,proto=tcp,timeo=600",
  "labels": {
    "cloud": "azure"
  },
  "location": "eastus",

  "storage": [
    {
      "labels": {
        "performance": "gold"
      },
      "serviceLevel": "Ultra",
      "capacityPools": ["ultra-1", "ultra-2"]
    },
    {
      "labels": {
        "performance": "silver"
      },
      "serviceLevel": "Premium",
      "capacityPools": ["premium-1"]
    },
    {
      "labels": {
        "performance": "bronze"
      },
      "serviceLevel": "Standard",
      "capacityPools": ["standard-1", "standard-2"]
    }
  ]
}

```

下列「儲存類別」定義是指上述的儲存資源池。您可以使用「parameters.selector」欄位、為每個「儲存類別」指定用於裝載磁碟區的虛擬資源池。該磁碟區會在所選的資源池中定義各個層面。


```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: gold
provisioner: csi.trident.netapp.io
parameters:
  selector: "performance=gold"
allowVolumeExpansion: true
---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: silver
provisioner: csi.trident.netapp.io
parameters:
  selector: "performance=silver"
allowVolumeExpansion: true
---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: bronze
provisioner: csi.trident.netapp.io
parameters:
  selector: "performance=bronze"
allowVolumeExpansion: true
```

接下來呢？

建立後端組態檔之後、請執行下列命令：

```
tridentctl create backend -f <backend-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。

針對GCP後端設定CVS

瞭解Cloud Volumes Service 解如何使用所提供的範例組態、將NetApp for Google Cloud Platform (GCP) 設定為Astra Trident安裝的後端。



適用於 Google Cloud 的 NetApp Cloud Volumes Service 支援的 CVS 效能磁碟區大小不得低於 100 GiB、或 CVS 磁碟區大小不得低於 300 GiB。如果所要求的磁碟區小於最小大小、Astra Trident 會自動建立最小大小的磁碟區。

您需要的產品

以設定及使用 "適用於 Google Cloud Cloud Volumes Service" 後端、您需要下列項目：

- 使用 NetApp CVS 設定的 Google Cloud 帳戶
- Google Cloud 帳戶的專案編號
- Google Cloud 服務帳戶的角色為「netappcloudvolumes.admin」
- CVS 服務帳戶的 API 金鑰檔

Astra Trident 現在支援預設的較小磁碟區 "GCP 上的 CVS 服務類型"。用於建立的後端 storageClass=software、現在、磁碟區的資源配置大小最小可達 300 GiB。CVS 目前在「管制可用度」下提供此功能、並不提供技術支援。使用者必須註冊才能存取低於 1 TiB 的磁碟區 "請按這裡"。NetApp 建議客戶使用低於 1 TiB 的磁碟區來處理*非正式作業*的工作負載。



使用預設 CVS 服務類型（「儲存類別=軟體」）部署後端時、使用者必須取得 GCP 上的子 1 TiB 磁碟區功能、以取得相關專案編號和專案 ID 的存取權。這是 Astra Trident 配置子 1 TiB 磁碟區所需的。如果沒有、則低於 600 GiB 的 PVCS 將無法建立 Volume。使用取得對低於 1 TiB 磁碟區的存取權 "這份表格"。

由 Astra Trident 針對預設 CVS 服務層級所建立的磁碟區、將會配置如下：

- 小於 300 GiB 的 PVCS 會導致 Astra Trident 建立 300 GiB CVS Volume。
- 介於 300 GiB 到 600 GiB 之間的 PVCS 會導致 Astra Trident 建立一個所需大小的 CVS Volume。
- 介於 600 GiB 和 1 TiB 之間的 PVCS 會導致 Astra Trident 建立 1 TiB CVS Volume。
- 大於 1 TiB 的 PVCS 會導致 Astra Trident 建立所需大小的 CVS Volume。

後端組態選項

如需後端組態選項、請參閱下表：

參數	說明	預設
「分度」		永遠為 1
「storageDriverName」	儲存驅動程式名稱	「GCP-CVS」
「後端名稱」	自訂名稱或儲存後端	驅動程式名稱+「_」+ API 金鑰的一部分
「storageClass」	儲存類型：可選擇「硬體」（效能最佳化）或「軟體」（CVS 服務類型）	
「ProjectNumber」	Google Cloud 帳戶專案編號。此值可在 Google Cloud 入口網站的首頁找到。	

參數	說明	預設
《apiRegion》	CVS帳戶區域。這是後端配置磁碟區的區域。	
「apiKey」	Google Cloud服務帳戶的API金鑰、其角色為「netappcloudVolumes.admin」。其中包含Google Cloud服務帳戶私密金鑰檔案（逐字複製到後端組態檔）的JSON-格式內容。	
"proxyurl"	Proxy URL（如果需要代理伺服器才能連線至CVS帳戶）。Proxy伺服器可以是HTTP Proxy或HTTPS Proxy。對於HTTPS Proxy、會跳過憑證驗證、以允許在Proxy伺服器中使用自我簽署的憑證。不支援已啟用驗證的Proxy伺服器。	
「nfsMountOptions」	精細控制NFS掛載選項。	"nfsves=3"
《限制Volume大小》	如果要求的磁碟區大小高於此值、則資源配置失敗	""（預設不強制執行）
《服務層級》	新磁碟區的CVS服務層級。這些值包括「標準」、「高級」和「極端」。	"標準"
網路	用於CVS磁碟區的GCP網路	「預設」
「DebugTraceFlags」	疑難排解時要使用的偵錯旗標。例如：「 <code>\{"API":fals,"method":true}</code> 」。除非您正在進行疑難排解並需要詳細的記錄傾印、否則請勿使用此功能。	null

如果使用共享VPC網路、則必須同時指定「ProjectNumber」和「hostProjectNumber」。在這種情況下、「專案編號」是服務專案、而「主機專案編號」則是主機專案。

「apiRegion」代表Astra Trident建立CVS磁碟區的GCP區域。在建立跨區域Kubernetes叢集時、在「apiRegion」中建立的CVS磁碟區可用於多個GCP區域的節點上排程的工作負載。請注意、跨區域流量會產生額外成本。

- 若要啟用跨區域存取、您的StorageClass定義「allegtopologies」必須包含所有地區。例如：



```
- key: topology.kubernetes.io/region
  values:
  - us-east1
  - europe-west1
```

- 「storageClass」是選用參數、可用來選取所需的參數 "CVS服務類型"。您可以從基礎CVS服務類型（「儲存類別=軟體」）或CVS效能服務類型（「儲存類別=硬體」）中進行選擇、這是Trident預設使用的類型。請務必指定「apiRegion」、在後端定義中提供各自的CVS「儲存類別」。



Astra Trident與Google Cloud上的基礎CVS服務類型整合、是一項*測試版功能、不適用於正式作業工作負載。Trident 完全支援 CVS效能服務類型、並依預設使用。

每個後端都會在單一Google Cloud區域中配置磁碟區。若要在其他區域建立磁碟區、您可以定義其他後端。

您可以在組態檔的特殊區段中指定下列選項、以控制預設的每個Volume佈建方式。請參閱下列組態範例。

參數	說明	預設
「匯出規則」	新磁碟區的匯出規則	「0.00.0.0/0」
「napshotDir	存取「.snapshot」目錄	"假"
「快照保留區」	保留給快照的磁碟區百分比	""（接受CVS預設值為0）
《大小》	新磁碟區的大小	「100Gi」

「匯出規則」值必須是以逗號分隔的清單、以CIDR表示法列出所有的IPv4位址或IPv4子網路組合。



針對在CVS Google Cloud後端上建立的所有磁碟區、Trident會在儲存資源池上的所有標籤配置時複製到儲存磁碟區。儲存管理員可以定義每個儲存資源池的標籤、並將儲存資源池中建立的所有磁碟區分組。這是根據後端組態中提供的一組可自訂標籤、方便區分磁碟區的方法。

範例1：最低組態

這是絕對最低的后端組態。

```
{
  "version": 1,
  "storageDriverName": "gcp-cvs",
  "projectNumber": "012345678901",
  "apiRegion": "us-west2",
  "apiKey": {
    "type": "service_account",
    "project_id": "my-gcp-project",
    "private_key_id": "12345678901234567890123456789012345678901234567890",
  }
}
```

```

    "private_key": "-----BEGIN PRIVATE KEY-----
\nznHczZsrtrHisIsAbOguSaPIKeyAZNchRAGzllZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZ
srtrHisIsAbOguSaPIKeyAZNchRAGzllZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrtrHisI
sAbOguSaPIKeyAZNchRAGzllZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrtrHisIsAbOguSa
PIKeyAZNchRAGzllZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrtrHisIsAbOguSaPIKeyAZN
chRAGzllZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrtrHisIsAbOguSaPIKeyAZNchRAGzll
ZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrtrHisIsAbOguSaPIKeyAZNchRAGzllZE4jK3bl
/qp8B4Kws8zX5ojY9m\nznHczZsrtrHisIsAbOguSaPIKeyAZNchRAGzllZE4jK3bl/qp8B4K
ws8zX5ojY9m\nznHczZsrtrHisIsAbOguSaPIKeyAZNchRAGzllZE4jK3bl/qp8B4Kws8zX5oj
Y9m\nznHczZsrtrHisIsAbOguSaPIKeyAZNchRAGzllZE4jK3bl/qp8B4Kws8zX5ojY9m\nznH
czZsrtrHisIsAbOguSaPIKeyAZNchRAGzllZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrtr
HisIsAbOguSaPIKeyAZNchRAGzllZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrtrHisIsAbO
guSaPIKeyAZNchRAGzllZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrtrHisIsAbOguSaPIKe
yAZNchRAGzllZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrtrHisIsAbOguSaPIKeyAZNchR
AGzllZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrtrHisIsAbOguSaPIKeyAZNchRAGzllZE4
jK3bl/qp8B4Kws8zX5ojY9m\nXsYg6gyxy4zq70lwWgLwGa==\n-----END PRIVATE
KEY-----\n",
    "client_email": "cloudvolumes-admin-sa@my-gcp-
project.iam.gserviceaccount.com",
    "client_id": "123456789012345678901",
    "auth_uri": "https://accounts.google.com/o/oauth2/auth",
    "token_uri": "https://oauth2.googleapis.com/token",
    "auth_provider_x509_cert_url":
"https://www.googleapis.com/oauth2/v1/certs",
    "client_x509_cert_url":
"https://www.googleapis.com/robot/v1/metadata/x509/cloudvolumes-admin-
sa%40my-gcp-project.iam.gserviceaccount.com"
  }
}

```

範例2：基礎CVS服務類型組態

此範例顯示使用基本CVS服務類型的後端定義、此服務類型適用於一般用途的工作負載、提供輕度/中度效能、以及高分區可用度。

```

{
  "version": 1,
  "storageDriverName": "gcp-cvs",

```



```

    "client_x509_cert_url":
      "https://www.googleapis.com/robot/v1/metadata/x509/cloudvolumes-admin-
      sa%40my-gcp-project.iam.gserviceaccount.com"
  },
  "proxyURL": "http://proxy-server-hostname/",
  "nfsMountOptions": "vers=3,proto=tcp,timeo=600",
  "limitVolumeSize": "10Ti",
  "serviceLevel": "premium",
  "defaults": {
    "snapshotDir": "true",
    "snapshotReserve": "5",
    "exportRule": "10.0.0.0/24,10.0.1.0/24,10.0.2.100",
    "size": "5Ti"
  }
}

```

範例4：虛擬儲存池組態

此範例顯示使用虛擬儲存資源池設定的後端定義檔、以及回溯到這些資源池的「儲存類別」。

在下圖所示的範例後端定義檔中、會針對所有儲存資源池設定特定的預設值、將「快照保留」設為5%、將「匯出規則」設為0.00.0/0。虛擬儲存資源池是在「儲存區」區段中定義的。在此範例中、每個個別的儲存資源池都會設定自己的「服務層級」、有些資源池則會覆寫預設值。

```

{
  "version": 1,
  "storageDriverName": "gcp-cvs",
  "projectNumber": "012345678901",
  "apiRegion": "us-west2",
  "apiKey": {
    "type": "service_account",
    "project_id": "my-gcp-project",
    "private_key_id": "1234567890123456789012345678901234567890",
    "private_key": "-----BEGIN PRIVATE KEY-----
\nznHczZsrrtHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZ
srrtHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrrtHisI
sAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrrtHisIsAbOguSa
PIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrrtHisIsAbOguSaPIKeyAZN
chRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrrtHisIsAbOguSaPIKeyAZNchRAGz1z
ZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrrtHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl
/qp8B4Kws8zX5ojY9m\nznHczZsrrtHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kw
s8zX5ojY9m\nznHczZsrrtHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY
9m\nznHczZsrrtHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHc
zZsrrtHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrrtHi
sIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrrtHisIsAbOgu
SaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrrtHisIsAbOguSaPIKeyA

```



```

ZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrtrHisIsAbOguSaPIKeyAZNchRAGz
1zZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrtrHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3
bl/qp8B4Kws8zX5ojY9m\nznHczZsrtrHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4
Kws8zX5ojY9m\nznHczZsrtrHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5o
jY9m\nznHczZsrtrHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m\nzn
HczZsrtrHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrtr
HisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrtrHisIsAbO
guSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrtrHisIsAbOguSaPIKe
yAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrtrHisIsAbOguSaPIKeyAZNchRA
Gz1zZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrtrHisIsAbOguSaPIKeyAZNchRAGz1zZE4j
K3bl/qp8B4Kws8zX5ojY9m\nXsYg6gyxy4zq70lwWgLwGa==\n-----END PRIVATE
KEY-----\n",
    "client_email": "cloudvolumes-admin-sa@my-gcp-
project.iam.gserviceaccount.com",
    "client_id": "123456789012345678901",
    "auth_uri": "https://accounts.google.com/o/oauth2/auth",
    "token_uri": "https://oauth2.googleapis.com/token",
    "auth_provider_x509_cert_url":
"https://www.googleapis.com/oauth2/v1/certs",
    "client_x509_cert_url":
"https://www.googleapis.com/robot/v1/metadata/x509/cloudvolumes-admin-
sa%40my-gcp-project.iam.gserviceaccount.com"
  },
  "nfsMountOptions": "vers=3,proto=tcp,timeo=600",

  "defaults": {
    "snapshotReserve": "5",
    "exportRule": "0.0.0.0/0"
  },

  "labels": {
    "cloud": "gcp"
  },
  "region": "us-west2",

  "storage": [
    {
      "labels": {
        "performance": "extreme",
        "protection": "extra"
      },
      "serviceLevel": "extreme",
      "defaults": {
        "snapshotDir": "true",
        "snapshotReserve": "10",
        "exportRule": "10.0.0.0/24"
      }
    }
  ]
}

```

```

    }
  },
  {
    "labels": {
      "performance": "extreme",
      "protection": "standard"
    },
    "serviceLevel": "extreme"
  },
  {
    "labels": {
      "performance": "premium",
      "protection": "extra"
    },
    "serviceLevel": "premium",
    "defaults": {
      "snapshotDir": "true",
      "snapshotReserve": "10"
    }
  },
  {
    "labels": {
      "performance": "premium",
      "protection": "standard"
    },
    "serviceLevel": "premium"
  },
  {
    "labels": {
      "performance": "standard"
    },
    "serviceLevel": "standard"
  }
]
}

```

下列StorageClass定義係指上述儲存資源池。您可以使用「parameters.selector」欄位、為每個StorageClass指定用於裝載磁碟區的虛擬集區。該磁碟區會在所選的資源池中定義各個層面。

第一個StorageClass（「CVS極致額外保護」）會對應至第一個虛擬儲存資源池。這是唯一提供極致效能、快照保留率為10%的資源池。最後一個StorageClass（「CVS額外保護」）會呼叫提供快照保留10%的任何儲存資源池。Astra Trident決定選取哪個虛擬儲存池、並確保符合快照保留需求。

```
apiVersion: storage.k8s.io/v1
```

```

kind: StorageClass
metadata:
  name: cvs-extreme-extra-protection
provisioner: netapp.io/trident
parameters:
  selector: "performance=extreme; protection=extra"
allowVolumeExpansion: true
---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: cvs-extreme-standard-protection
provisioner: netapp.io/trident
parameters:
  selector: "performance=premium; protection=standard"
allowVolumeExpansion: true
---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: cvs-premium-extra-protection
provisioner: netapp.io/trident
parameters:
  selector: "performance=premium; protection=extra"
allowVolumeExpansion: true
---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: cvs-premium
provisioner: netapp.io/trident
parameters:
  selector: "performance=premium; protection=standard"
allowVolumeExpansion: true
---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: cvs-standard
provisioner: netapp.io/trident
parameters:
  selector: "performance=standard"
allowVolumeExpansion: true
---
apiVersion: storage.k8s.io/v1
kind: StorageClass

```

```
metadata:
  name: cvs-extra-protection
  provisioner: netapp.io/trident
  parameters:
    selector: "protection=extra"
  allowVolumeExpansion: true
```

接下來呢？

建立後端組態檔之後、請執行下列命令：

```
tridentctl create backend -f <backend-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。

設定NetApp HCI 一個不只是功能的SolidFire 後端

瞭解如何在安裝Astra Trident時建立及使用元素後端。

您需要的產品

- 支援的儲存系統、可執行Element軟體。
- 提供給NetApp HCI / SolidFire叢集管理員或租戶使用者的認證、以管理磁碟區。
- 您所有的Kubernetes工作節點都應該安裝適當的iSCSI工具。請參閱 "[工作節點準備資訊](#)"。

您需要知道的資訊

「Poolidfire - san」儲存驅動程式支援兩種Volume模式：檔案和區塊。對於「檔案系統」磁碟區模式、Astra Trident會建立一個磁碟區並建立檔案系統。檔案系統類型由StorageClass指定。

驅動程式	傳輸協定	Volume模式	支援的存取模式	支援的檔案系統
「olidfire - san」	iSCSI	區塊	rwo、ROX、rwx	無檔案系統。原始區塊裝置。
「olidfire - san」	iSCSI	區塊	rwo、ROX、rwx	無檔案系統。原始區塊裝置。
「olidfire - san」	iSCSI	檔案系統	Rwo、ROX	《xfs》、《ext3》、《ext4》

驅動程式	傳輸協定	Volume模式	支援的存取模式	支援的檔案系統
「olidfire - san」	iSCSI	檔案系統	Rwo、ROX	《xfs》、《ext3》、 《ext4》



Astra Trident在做為增強型的csi資源配置程式時使用CHAP。如果您使用的是CHAP（這是「csi」的預設值）、則不需要進一步準備。建議將「UseCHAP」選項明確設定為搭配非「csi Trident」使用CHAP。否則請參閱 ["請按這裡"](#)。



Volume存取群組僅受Astra Trident的傳統非csi架構支援。當Astra Trident設定為以「csi」模式運作時、會使用CHAP。

如果未設定「存取群組」或「UseCHAP」、則適用下列其中一項規則：

- 如果偵測到預設的「Trident」存取群組、就會使用存取群組。
- 如果未偵測到存取群組、且Kubernetes版本為1.7或更新版本、則會使用CHAP。

後端組態選項

如需後端組態選項、請參閱下表：

參數	說明	預設
「分度」		永遠為1
「storageDriverName」	儲存驅動程式名稱	永遠是「solidfire-san」
「後端名稱」	自訂名稱或儲存後端	「S指_」+儲存設備（iSCSI）IP位址SolidFire
端點	MVIP、適用於SolidFire 採用租戶認證的不含用戶身分證明的叢集	
《VIP》	儲存設備（iSCSI）IP位址和連接埠	
《標籤》	套用到磁碟區的任意JSON-格式化標籤集。	「」
《天王名稱》	要使用的租戶名稱（如果找不到、請建立）	
《初始器IFACE》	將iSCSI流量限制在特定的主機介面	「預設」
《UseCHAP》	使用CHAP驗證iSCSI	是的
《存取群組》	要使用的存取群組ID清單	尋找名為「Trident」的存取群組ID
《類型》	QoS規格	
《限制Volume大小》	如果要求的磁碟區大小高於此值、則資源配置失敗	「」（預設不強制執行）
「DebugTraceFlags」	疑難排解時要使用的偵錯旗標。範例：{"API":假、「方法」:true}	null



除非您正在進行疑難排解並需要詳細的記錄傾印、否則請勿使用「debugTraceFlags」。



對於所有建立的磁碟區、Astra Trident會在儲存資源池上的所有標籤配置時、複製到備用儲存LUN。儲存管理員可以定義每個儲存資源池的標籤、並將儲存資源池中建立的所有磁碟區分組。這是根據後端組態中提供的一組可自訂標籤、方便區分磁碟區的方法。

範例1：的後端組態 solidfire-san 三種磁碟區類型的驅動程式

此範例顯示使用CHAP驗證的後端檔案、並建立具有特定QoS保證的三種Volume類型模型。您很可能會使用「IOPS」儲存類別參數來定義儲存類別、以使用每個類別。

```
{
  "version": 1,
  "storageDriverName": "solidfire-san",
  "Endpoint": "https://<user>:<password>@<mvip>/json-rpc/8.0",
  "SVIP": "<svip>:3260",
  "TenantName": "<tenant>",
  "labels": {"k8scluster": "dev1", "backend": "dev1-element-cluster"},
  "UseCHAP": true,
  "Types": [{"Type": "Bronze", "Qos": {"minIOPS": 1000, "maxIOPS": 2000,
"burstIOPS": 4000}},
            {"Type": "Silver", "Qos": {"minIOPS": 4000, "maxIOPS": 6000,
"burstIOPS": 8000}},
            {"Type": "Gold", "Qos": {"minIOPS": 6000, "maxIOPS": 8000,
"burstIOPS": 10000}}]
}
```

範例2：的後端與儲存類別組態 solidfire-san 驅動程式搭配虛擬儲存資源池

此範例顯示使用虛擬儲存資源池設定的後端定義檔案、以及參照回溯的StorageClass。

在下圖所示的範例後端定義檔中、會針對所有儲存資源池設定特定的預設值、將「type」設為Silver。虛擬儲存資源池是在「儲存區」區段中定義的。在此範例中、有些儲存資源池會設定自己的類型、有些資源池則會覆寫上述設定的預設值。

```

{
  "version": 1,
  "storageDriverName": "solidfire-san",
  "Endpoint": "https://<user>:<password>@<mvip>/json-rpc/8.0",
  "SVIP": "<svip>:3260",
  "TenantName": "<tenant>",
  "UseCHAP": true,
  "Types": [{"Type": "Bronze", "Qos": {"minIOPS": 1000, "maxIOPS": 2000,
"burstIOPS": 4000}},
            {"Type": "Silver", "Qos": {"minIOPS": 4000, "maxIOPS": 6000,
"burstIOPS": 8000}},
            {"Type": "Gold", "Qos": {"minIOPS": 6000, "maxIOPS": 8000,
"burstIOPS": 10000}}],

  "type": "Silver",
  "labels":{"store":"solidfire", "k8scluster": "dev-1-cluster"},
  "region": "us-east-1",

  "storage": [
    {
      "labels":{"performance":"gold", "cost":"4"},
      "zone":"us-east-1a",
      "type":"Gold"
    },
    {
      "labels":{"performance":"silver", "cost":"3"},
      "zone":"us-east-1b",
      "type":"Silver"
    },
    {
      "labels":{"performance":"bronze", "cost":"2"},
      "zone":"us-east-1c",
      "type":"Bronze"
    },
    {
      "labels":{"performance":"silver", "cost":"1"},
      "zone":"us-east-1d"
    }
  ]
}

```

下列StorageClass定義係指上述虛擬儲存資源池。使用「parameters.selector」欄位、每個StorageClass都會呼叫哪些虛擬資源池可用於裝載磁碟區。磁碟區將會在所選的虛擬資源池中定義各個層面。

第一個StorageClass（「Poolidfire gold-s4」）會對應至第一個虛擬儲存集區。這是唯一提供黃金級效能的「Volume Type QoS」（Volume Type QoS）。最後一個StorageClass（「Poolidfire Silver」）會呼叫任何提

供銀級效能的儲存資源池。Astra Trident將決定選取哪個虛擬儲存資源池、並確保符合儲存需求。


```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: solidfire-gold-four
provisioner: csi.trident.netapp.io
parameters:
  selector: "performance=gold; cost=4"
  fsType: "ext4"
---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: solidfire-silver-three
provisioner: csi.trident.netapp.io
parameters:
  selector: "performance=silver; cost=3"
  fsType: "ext4"
---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: solidfire-bronze-two
provisioner: csi.trident.netapp.io
parameters:
  selector: "performance=bronze; cost=2"
  fsType: "ext4"
---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: solidfire-silver-one
provisioner: csi.trident.netapp.io
parameters:
  selector: "performance=silver; cost=1"
  fsType: "ext4"
---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: solidfire-silver
provisioner: csi.trident.netapp.io
parameters:
  selector: "performance=silver"
  fsType: "ext4"
```

如需詳細資訊、請參閱

- "[Volume存取群組](#)"

使用ONTAP 不支援的Cloud Volumes ONTAP SAN驅動程式來設定後端

深入瞭解如何使用ONTAP 支援功能的功能和功能性SAN驅動程式來設定功能性的後端。ONTAP Cloud Volumes ONTAP

- "[準備](#)"
- "[組態與範例](#)"

使用者權限

Astra Trident希望以ONTAP 支援或SVM管理員的身分執行、通常是使用「admin」叢集使用者或「vsadmin」SVM使用者、或使用具有相同角色的不同名稱使用者。對於Amazon FSx for NetApp ONTAP 等部署、Astra Trident預期會使用ONTAP 叢集「fsxadmin」使用者或「vsadmin」SVM使用者、或是具有相同角色之不同名稱的使用者、來執行成一個或多個SVM管理員。「fsxadmin」使用者是叢集管理使用者的有限替代。



如果您使用「limitAggregateusage」參數、則需要叢集管理權限。使用Amazon FSx for NetApp ONTAP 支援Astra Trident時、「limitAggregateusage」參數無法搭配「vsadmin」和「fsxadmin」使用者帳戶使用。如果您指定此參數、組態作業將會失敗。

雖然可以在ONTAP 功能區內建立更嚴格的角色、讓Trident驅動程式能夠使用、但我們不建議您這麼做。Trident的大多數新版本都會呼叫額外的API、而這些API必須納入考量、使升級變得困難且容易出錯。

準備

瞭解如何準備使用ONTAP 支援不支援的SAN驅動程式來設定支援功能的後端。ONTAP對於所有ONTAP 的不支援端點、Astra Trident至少需要指派一個集合體給SVM。

請記住、您也可以執行多個驅動程式、並建立指向一個或多個驅動程式的儲存類別。例如、您可以設定使用「ONTAP-SAN」驅動程式的「SAN開發」類別、以及使用「ONTAP-SAN經濟」類別的「SAN預設」類別。

您所有的Kubernetes工作節點都必須安裝適當的iSCSI工具。請參閱 "[請按這裡](#)" 以取得更多詳細資料。

驗證

Astra Trident提供兩種驗證ONTAP 證功能來驗證支援的後端。

- 認證型：ONTAP 對具備所需權限的使用者名稱和密碼。建議使用預先定義的安全登入角色、例如「admin」或「vsadmin」、以確保與ONTAP 各種版本的最大相容性。
- 憑證型：Astra Trident也能ONTAP 使用安裝在後端的憑證與某個叢集進行通訊。在此處、後端定義必須包含用戶端憑證、金鑰及信任的CA憑證（建議使用）的Base64編碼值。

使用者也可以選擇更新現有的後端、選擇從認證移至憑證型、反之亦然。如果*同時提供認證資料和憑證*、Astra Trident將預設使用憑證、同時發出警告、從後端定義中移除認證資料。

啟用認證型驗證

Astra Trident需要SVM範圍/叢集範圍管理員的認證資料、才能與ONTAP 該後端進行通訊。建議使用預先定義的

標準角色、例如「admin」或「vsadmin」。這可確保與未來ONTAP 的支援版本保持前瞻相容、因為未來的Astra Trident版本可能會使用功能API。您可以建立自訂的安全登入角色、並與Astra Trident搭配使用、但不建議使用。

後端定義範例如下所示：

```
{
  "version": 1,
  "backendName": "ExampleBackend",
  "storageDriverName": "ontap-san",
  "managementLIF": "10.0.0.1",
  "dataLIF": "10.0.0.2",
  "svm": "svm_nfs",
  "username": "vsadmin",
  "password": "secret",
}
```

請記住、後端定義是唯一以純文字儲存認證的位置。建立後端之後、使用者名稱/密碼會以Base64編碼、並儲存為Kubernetes機密。建立/更新後端是唯一需要知道認證資料的步驟。因此、這是一項純管理員操作、由Kubernetes /儲存管理員執行。

啟用憑證型驗證

新的和現有的後端可以使用憑證、並與ONTAP 該後端通訊。後端定義需要三個參數。

- 用戶端憑證：用戶端憑證的Base64編碼值。
- 用戶端私密金鑰：關聯私密金鑰的Base64編碼值。
- 信任的CACertificate：受信任CA憑證的Base64編碼值。如果使用信任的CA、則必須提供此參數。如果未使用信任的CA、則可忽略此問題。

典型的工作流程包括下列步驟。

步驟

1. 產生用戶端憑證和金鑰。產生時、請將Common Name (CN) (一般名稱 (CN)) 設定為ONTAP 驗證身分。

```
openssl req -x509 -nodes -days 1095 -newkey rsa:2048 -keyout k8senv.key
-out k8senv.pem -subj "/C=US/ST=NC/L=RTP/O=NetApp/CN=admin"
```

2. 將信任的CA憑證新增ONTAP 至整個叢集。這可能已由儲存管理員處理。如果未使用信任的CA、請忽略。

```
security certificate install -type server -cert-name <trusted-ca-cert-name> -vserver <vserver-name>
ssl modify -vserver <vserver-name> -server-enabled true -client-enabled true -common-name <common-name> -serial <SN-from-trusted-CA-cert> -ca <cert-authority>
```

3. 在ONTAP 支援叢集上安裝用戶端憑證和金鑰（步驟1）。

```
security certificate install -type client-ca -cert-name <certificate-name> -vserver <vserver-name>
security ssl modify -vserver <vserver-name> -client-enabled true
```

4. 確認ONTAP 支援「cert」驗證方法的支援功能。

```
security login create -user-or-group-name admin -application ontapi -authentication-method cert
security login create -user-or-group-name admin -application http -authentication-method cert
```

5. 使用產生的憑證測試驗證。以ONTAP Management LIF IP和SVM名稱取代<SfManagement LIF>和<vserver name>。

```
curl -X POST -Lk https://<ONTAP-Management-LIF>/servlets/netapp.servlets.admin.XMLrequest_filer --key k8senv.key --cert ~/k8senv.pem -d '<?xml version="1.0" encoding="UTF-8"?><netapp xmlns="http://www.netapp.com/filer/admin" version="1.21" vfiler="<vserver-name>"><vserver-get></vserver-get></netapp>'
```

6. 使用Base64編碼憑證、金鑰和信任的CA憑證。

```
base64 -w 0 k8senv.pem >> cert_base64
base64 -w 0 k8senv.key >> key_base64
base64 -w 0 trustedca.pem >> trustedca_base64
```

7. 使用從上一步取得的值建立後端。

```

$ cat cert-backend.json
{
  "version": 1,
  "storageDriverName": "ontap-san",
  "backendName": "SanBackend",
  "managementLIF": "1.2.3.4",
  "dataLIF": "1.2.3.8",
  "svm": "vserver_test",
  "clientCertificate": "Faaaakkkkeeee...Vaaalllluuueeeee",
  "clientPrivateKey": "LS0tFaKE...0VaLuES0tLS0K",
  "trustedCACertificate": "QNFinfO...SiqOyN",
  "storagePrefix": "myPrefix_"
}

$ tridentctl create backend -f cert-backend.json -n trident
+-----+-----+-----+
+-----+-----+
|   NAME   | STORAGE DRIVER |          UUID          |
STATE | VOLUMES |
+-----+-----+-----+
+-----+-----+
| SanBackend | ontap-san      | 586b1cd5-8cf8-428d-a76c-2872713612c1 |
online |         0 |
+-----+-----+-----+
+-----+-----+

```

更新驗證方法或旋轉認證資料

您可以更新現有的後端、以使用不同的驗證方法、或是旋轉其認證資料。這兩種方法都可行：使用使用者名稱/密碼的後端可更新以使用憑證；使用憑證的後端可更新為使用者名稱/密碼。若要這麼做、請使用更新的「backend.json」檔案、其中包含執行「tridentctl後端更新」所需的參數。

```

$ cat cert-backend-updated.json
{
"version": 1,
"storageDriverName": "ontap-san",
"backendName": "SanBackend",
"managementLIF": "1.2.3.4",
"dataLIF": "1.2.3.8",
"svm": "vserver_test",
"username": "vsadmin",
"password": "secret",
"storagePrefix": "myPrefix_"
}

#Update backend with tridentctl
$ tridentctl update backend SanBackend -f cert-backend-updated.json -n
trident
+-----+-----+-----+
+-----+-----+
|   NAME   | STORAGE DRIVER |          UUID          |
STATE | VOLUMES |
+-----+-----+-----+
+-----+-----+
| SanBackend | ontap-san      | 586b1cd5-8cf8-428d-a76c-2872713612c1 |
online |         9 |
+-----+-----+-----+
+-----+-----+

```



當您旋轉密碼時、儲存管理員必須先更新ONTAP 使用者的密碼（位於BIOS）。接著是後端更新。在循環憑證時、可將多個憑證新增至使用者。然後更新後端以使用新的憑證、之後可從ONTAP 該叢集刪除舊的憑證。

更新後端不會中斷對已建立之磁碟區的存取、也不會影響之後建立的磁碟區連線。成功的後端更新顯示Astra Trident可以與ONTAP 該後端通訊、並處理未來的Volume作業。

指定igroup

Astra Trident使用igroup來控制其所配置的磁碟區（LUN）存取。系統管理員在指定後端的igroup時有兩種選擇：

- Astra Trident可自動建立及管理每個後端的igroup。如果後端定義中未包含「groupName」、Astra Trident會在SVM上建立名為「trident -」的igroup。如此可確保每個後端都有專屬的igroup、並處理Kubernetes節點IQN的自動新增/刪除作業。
- 或者、也可以在後端定義中提供預先建立的igroup。這可以使用「groupName」組態參數來完成。Astra Trident會將Kubernetes節點IQN新增/刪除至預先存在的igroup。

對於定義了「groupName」的後端、可以使用「tridentctl後端更新」刪除「groupName」、以便使用Astra Trident自動處理igroup。這不會中斷對已附加至工作負載之磁碟區的存取。未來的連線將使用建立的igroup

Astra Trident來處理。



針對每個獨特的Astra Trident執行個體指定igroup是最適合Kubernetes管理員和儲存管理員的最佳實務做法。「csi Trident」可自動新增及移除igroup的叢集節點IQN、大幅簡化其管理。在Kubernetes環境中使用相同的SVM（以及Astra Trident安裝）時、使用專屬的igroup可確保對Kubernetes叢集所做的變更不會影響與其他叢集相關的igroup。此外、也必須確保Kubernetes叢集中的每個節點都有唯一的IQN。如上所述、Astra Trident會自動處理IQN的新增與移除。重複使用主機間的IQN可能會導致主機彼此誤用、並拒絕存取LUN的不良情況。

如果將Astra Trident設定為使用「csi資源配置程式」、則Kubernetes節點IQN會自動新增至igroup或從其中移除。當節點新增至Kubernetes叢集時、「Trident - csi」示範Set會在新增的節點上部署pod（「trident - csi-xxxxX」）、並登錄可附加磁碟區的新節點。節點IQN也會新增至後端的igroup。當節點封鎖、排放及從Kubernetes刪除時、類似的一組步驟可處理刪除IQN。

如果Astra Trident並未以csi資源配置程式的形式執行、則必須手動更新igroup、以包含Kubernetes叢集中每個工作節點的iSCSI IQN。加入Kubernetes叢集的節點IQN必須新增至igroup。同樣地、從Kubernetes叢集移除的節點IQN也必須從igroup移除。

使用雙向CHAP驗證連線

Astra Trident可以使用雙向CHAP驗證iSCSI工作階段的「ONTAP-SAN」和「ONTAP-san經濟」驅動程式。這需要在後端定義中啟用「useCHAP」選項。當設為「true」時、Astra Trident會將SVM的預設啟動器安全性設定為雙向CHAP、並從後端檔案設定使用者名稱和機密。NetApp建議使用雙向CHAP來驗證連線。請參閱下列組態範例：

```
{
  "version": 1,
  "storageDriverName": "ontap-san",
  "backendName": "ontap_san_chap",
  "managementLIF": "192.168.0.135",
  "svm": "ontap_iscsi_svm",
  "useCHAP": true,
  "username": "vsadmin",
  "password": "FaKePaSsWoRd",
  "igroupName": "trident",
  "chapInitiatorSecret": "cl9qxIm36DKyawxy",
  "chapTargetInitiatorSecret": "rqxigXgkesIpwxyz",
  "chapTargetUsername": "iJF4heBRT0TCwxyz",
  "chapUsername": "uh2aNCLSD6cNwxyz",
}
```



「useCHAP」參數是布林選項、只能設定一次。預設值設為假。將其設為true之後、您就無法將其設為假。

除了"useCHAP=true"之外、"chapInitiator Secret (chapInitiator機密)"、"chaptarketatorSecret (chaptarketusername)"、"chaptarketusername (chaptargetuseamuse)"和"chapusername (chamus在建立後端後端之後、可以執行「tridentctl update」來變更機密。

運作方式

儲存管理員將「useCHAP」設定為true、指示Astra Trident在儲存後端上設定CHAP。這包括下列項目：

- 在SVM上設定CHAP：
 - 如果SVM的預設啟動器安全性類型為「無」（預設設定）和、則磁碟區中已沒有預先存在的LUN、Astra Trident會將預設安全性類型設為「CHAP」、並繼續設定CHAP啟動器和目標使用者名稱和機密。
 - 如果SVM包含LUN、Astra Trident將不會在SVM上啟用CHAP。如此可確保不限制存取SVM上已存在的LUN。
- 設定CHAP啟動器和目標使用者名稱和機密；這些選項必須在後端組態中指定（如上所示）。
- 管理在後端的「igroupName」中新增的初始化。如果未指定、則預設為「Trident」。

建立後端之後、Astra Trident會建立對應的「tridentbackend」CRD、並將CHAP機密和使用者名稱儲存為Kubernetes機密。由Astra Trident在此後端上建立的所有PV、都會掛載並附加於CHAP上。

旋轉認證資料並更新後端

您可以更新「backend.json」檔案中的CHAP參數、以更新CHAP認證。這需要更新CHAP機密、並使用「tridentctl update」命令來反映這些變更。



更新後端的CHAP機密時、您必須使用「tridentctl」來更新後端。請勿透過CLI/ONTAP UI更新儲存叢集上的認證資料、因為Astra Trident無法接受這些變更。


```

$ cat backend-san.json
{
  "version": 1,
  "storageDriverName": "ontap-san",
  "backendName": "ontap_san_chap",
  "managementLIF": "192.168.0.135",
  "svm": "ontap_iscsi_svm",
  "useCHAP": true,
  "username": "vsadmin",
  "password": "FaKePaSsWoRd",
  "igroupName": "trident",
  "chapInitiatorSecret": "c19qxUpDaTeD",
  "chapTargetInitiatorSecret": "rqxigXgkeUpDaTeD",
  "chapTargetUsername": "iJF4heBRT0TCwxyz",
  "chapUsername": "uh2aNCLSD6cNwxyz",
}

$ ./tridentctl update backend ontap_san_chap -f backend-san.json -n
trident
+-----+-----+-----+
+-----+-----+
|  NAME          | STORAGE DRIVER |          UUID          |
STATE | VOLUMES |
+-----+-----+-----+
+-----+-----+
| ontap_san_chap | ontap-san      | aa458f3b-ad2d-4378-8a33-1a472ffbeb5c |
online |       7 |
+-----+-----+-----+
+-----+-----+

```

現有的連線不會受到影響；如果SVM上的Astra Trident更新認證、它們將繼續保持作用中狀態。新連線將使用更新的認證資料、而現有連線仍保持作用中狀態。中斷舊PV的連線並重新連線、將會使用更新的認證資料。

組態選項與範例

瞭解如何透過ONTAP Astra Trident安裝來建立及使用支援NetApp的SAN驅動程式。本節提供後端組態範例、以及如何將後端對應至StorageClass的詳細資料。

後端組態選項

如需後端組態選項、請參閱下表：

參數	說明	預設
「分度」		永遠為1

參數	說明	預設
「storageDriverName」	儲存驅動程式名稱	「ONTAP-NAS」、「ONTAP-NAS-節約型」、「ONTAP-NAS-flexgroup」、「ONTAP-SAN」、「ONTAP-san經濟型」
「後端名稱」	自訂名稱或儲存後端	驅動程式名稱+「_」+dataLIF
《馬納格門達利》	叢集或SVM管理LIF的IP位址	「10.0.0.1」、「[2001:1234:abcd::fefo]」
「DataLIF」	傳輸協定LIF的IP位址。IPv6使用方括弧。設定後無法更新	除非另有說明、否則由SVM衍生
《使用CHAP》	使用CHAP驗證iSCSI以供ONTAP支援不支援的SAN驅動程式使用[布林值]	錯
《chapInitiator機密》	CHAP啟動器密碼。如果是"useCHAP=true"、則為必要項目	「」
《標籤》	套用到磁碟區的任意JSON-格式化標籤集	「」
《chapTargetInitiator機密》	CHAP目標啟動器機密。如果是"useCHAP=true"、則為必要項目	「」
「chapUsername」	傳入使用者名稱。如果是"useCHAP=true"、則為必要項目	「」
《chapTargetUsername》	目標使用者名稱。如果是"useCHAP=true"、則為必要項目	「」
「用戶端憑證」	用戶端憑證的Base64編碼值。用於憑證型驗證	「」
「clientPrivate Key」	用戶端私密金鑰的Base64編碼值。用於憑證型驗證	「」
「可信賴的CACertificate」	受信任CA憑證的Base64編碼值。選用。用於憑證型驗證	「」
《使用者名稱》	連線至叢集/ SVM的使用者名稱。用於認證型驗證	「」
密碼	連線至叢集/ SVM的密碼。用於認證型驗證	「」
《虛擬機器》	要使用的儲存虛擬機器	如果指定SVM "managementLIF"則衍生
「igroupName」	要使用之SAN磁碟區的igroup名稱	「Trident -<後端-UUID>」
「storagePrefix」	在SVM中配置新磁碟區時所使用的前置碼。設定後無法更新	「Trident」
「限制Aggregateusage」	如果使用率高於此百分比、則無法進行資源配置。*不適用於Amazon FSX for ONTAP Sfor Sfor *	「」（預設不強制執行）

參數	說明	預設
《限制Volume大小》	如果要求的磁碟區大小高於經濟驅動程式的此值、則無法進行資源配置。	「」（預設不強制執行）
《lunsPerFlexvol》	每FlexVol 個LUN的最大LUN數量、範圍必須在[50、200]	「100」
「DebugTraceFlags」	疑難排解時要使用的偵錯旗標。範例： { "API" : 假、 「方法」 : true }	null
《useREST》	使用ONTAP Isrest API的布林參數。技術預覽	錯



「useREST」是一種建議用於測試環境、而非用於正式作業工作負載的技術預覽。當設為「true」時、Astra Trident會使用ONTAP「Isrest API」與後端通訊。此功能需要ONTAP 使用更新版本的版本。此外ONTAP、所使用的登入角色必須能夠存取「ONTAP 發揮作用」應用程式。這對預先定義的「vsadmin」和「叢集管理」角色感到滿意。

若要與ONTAP 此叢集通訊、您應該提供驗證參數。這可能是安全登入或安裝憑證的使用者名稱/密碼。



如果您使用Amazon FSX for NetApp ONTAP Sendbackend、請勿指定「limitAggregateusage」參數。Amazon FSx for NetApp ONTAP 的「fsxadmin」和「vsadmin」角色不包含擷取Aggregate使用量並透過Astra Trident加以限制的必要存取權限。



除非您正在進行疑難排解並需要詳細的記錄傾印、否則請勿使用「debugTraceFlags」。

對於「ONTAP-san」驅動程式、預設為使用SVM的所有資料LIF IP、並使用iSCSI多重路徑。為「ONTAP-san」驅動程式的dataLIF指定IP位址、會強制它們停用多重路徑、並只使用指定的位址。



建立後端時、請記住、建立後無法修改「dataLIF」和「儲存前置字元」。若要更新這些參數、您需要建立新的後端。

「igroupName」可設定為ONTAP 已在該叢集上建立的igroup。如果未指定、Astra Trident會自動建立名為Trident的igroup -<後端UUID >。如果提供預先定義的igroupName、則如果要在環境之間共用SVM、NetApp 建議使用每個Kubernetes叢集的igroup。這是Astra Trident自動維護IQN新增/刪除的必要條件。

後端也可以在建立後更新igroup：

- 可以更新igroupName、以指向在Astra Trident以外的SVM上建立及管理的新igroup。
- 可以省略igroupName。在此案例中、Astra Trident會自動建立及管理Trident -<backend-UUUUID> igroup。

在這兩種情況下、仍可繼續存取Volume附件。未來的Volume附件將使用更新的igroup。此更新不會中斷對後端磁碟區的存取。

可以為「managementLIF」選項指定完整網域名稱（FQDN）。

所有ONTAP 的不完整驅動程式也可設定為IPv6位址。請務必使用「-use IPv6」旗標來安裝Trident。必須謹慎定義方括弧內的「managementLIF」IPv6位址。



使用IPv6位址時、請務必在方括弧內定義「managementLIF」和「dataLIF」（若包含在後端定義中）、例如[28e8:d9fb:a825:b7bf:69a8:d02f:9e7b:3555]。如果未提供「data LIF」、Astra Trident會從SVM擷取IPv6資料LIF。

若要讓ONTAP-SAN驅動程式使用CHAP、請在後端定義中將「useCHAP」參數設為「true」。然後Astra Trident會設定並使用雙向CHAP做為後端所指定SVM的預設驗證。請參閱 ["請按這裡"](#) 以瞭解其運作方式。

對於「ONTAP-san經濟」驅動程式、「limitVolume Sizes」選項也會限制其管理的qtree和LUN磁碟區大小上限。



Astra Trident在所有使用「ONTAP-san」驅動程式建立的磁碟區的「Comments」（註解）欄位中設定資源配置標籤。針對所建立的每個Volume、FlexVol 將會在顯示於其儲存資源池中的「Comments」（註解）欄位中填入所有標籤。儲存管理員可以定義每個儲存資源池的標籤、並將儲存資源池中建立的所有磁碟區分組。這是根據後端組態中提供的一組可自訂標籤、方便區分磁碟區的方法。

用於資源配置磁碟區的後端組態選項

您可以在組態的特定區段中、使用這些選項來控制預設配置每個Volume的方式。如需範例、請參閱下列組態範例。

參數	說明	預設
"paceAllocate (配置) "	LUN的空間分配	「真的」
《保護區》	空間保留模式；「無」（精簡）或「Volume」（完整）	「無」
「快照原則」	要使用的Snapshot原則	「無」
「qosPolicy」	要指派給所建立磁碟區的QoS原則群組。選擇每個儲存集區/後端的其中一個qosPolicy或adaptiveQosPolicy	「」
《adaptiveQosPolicy》	要指派給所建立磁碟區的調適性QoS原則群組。選擇每個儲存集區/後端的其中一個qosPolicy或adaptiveQosPolicy	「」
「快照保留區」	保留給快照「0」的磁碟區百分比	如果「快照原則」為「無」、則其他為「」
「PlitOnClone」	建立複本時、從其父複本分割複本	「假」
「PlitOnClone」	建立複本時、從其父複本分割複本	「假」
加密	啟用NetApp Volume加密	「假」
《生態樣式》	新磁碟區的安全樣式	「UNIX」
「分層政策」	分層原則以使用「無」	ONTAP 9.5之前的SVM-DR組態為「純快照」



搭配Astra Trident使用QoS原則群組需要ONTAP 使用更新版本的版本。建議使用非共用的QoS原則群組、並確保原則群組會個別套用至每個組成群組。共享的QoS原則群組將強制所有工作負載的總處理量上限。

以下是已定義預設值的範例：

```
{
  "version": 1,
  "storageDriverName": "ontap-san",
  "managementLIF": "10.0.0.1",
  "dataLIF": "10.0.0.2",
  "svm": "trident_svm",
  "username": "admin",
  "password": "password",
  "labels": {"k8scluster": "dev2", "backend": "dev2-sanbackend"},
  "storagePrefix": "alternate-trident",
  "igroupName": "custom",
  "debugTraceFlags": {"api":false, "method":true},
  "defaults": {
    "spaceReserve": "volume",
    "qosPolicy": "standard",
    "spaceAllocation": "false",
    "snapshotPolicy": "default",
    "snapshotReserve": "10"
  }
}
```



對於使用「ONTAP-san」驅動程式建立的所有磁碟區、Astra Trident會在FlexVol the支援LUN中繼資料的情況下、為該系統的功能增加10%的容量。LUN的配置大小與使用者在PVC中要求的大小完全相同。Astra Trident在FlexVol 整個過程中增加10%的速度（顯示ONTAP 在畫面上可用的尺寸）。使用者現在可以取得所要求的可用容量。此變更也可防止LUN成為唯讀、除非可用空間已充分利用。這不適用於ONTAP-san經濟型。

對於定義「快照保留區」的後端、Astra Trident會計算磁碟區大小、如下所示：

```
Total volume size = [(PVC requested size) / (1 - (snapshotReserve
percentage) / 100)] * 1.1
```

1.1是額外10%的Astra Trident加入FlexVol 到the支援LUN中繼資料的功能。對於「快照保留區」= 5%、而PVC要求= 5GiB、磁碟區總大小為5.79GiB、可用大小為5.5GiB。「volume show (Volume show) 命令應顯示類似以下範例的結果：

Vserver	Volume	Aggregate	State	Type	Size	Available	Used%
		_pvc_89f1c156_3801_4de4_9f9d_034d54c395f4	online	RW	10GB	5.00GB	0%
		_pvc_e42ec6fe_3baa_4af6_996d_134adbbb8e6d	online	RW	5.79GB	5.50GB	0%
		_pvc_e8372153_9ad9_474a_951a_08ae15e1c0ba	online	RW	1GB	511.8MB	0%

3 entries were displayed.

目前、只有調整大小、才能將新計算用於現有的Volume。

最低組態範例

下列範例顯示基本組態、讓大部分參數保留預設值。這是定義後端最簡單的方法。



如果您在NetApp ONTAP 支援Astra Trident的NetApp上使用Amazon FSX、建議您指定lifs的DNS名稱、而非IP位址。

ontap-san 具有憑證型驗證的驅動程式

這是最小的後端組態範例。「clientCertificate」、「clientPrivate Key」和「Trusted CACertificate」（選用、如果使用信任的CA）會分別填入「backend.json」、並採用用戶端憑證、私密金鑰和信任的CA憑證的64編碼值。

```
{
  "version": 1,
  "storageDriverName": "ontap-san",
  "backendName": "DefaultSANBackend",
  "managementLIF": "10.0.0.1",
  "dataLIF": "10.0.0.3",
  "svm": "svm_iscsi",
  "useCHAP": true,
  "chapInitiatorSecret": "cl9qxIm36DKyawxy",
  "chapTargetInitiatorSecret": "rqxigXgkesIpwxyz",
  "chapTargetUsername": "iJF4heBRT0TCwxyz",
  "chapUsername": "uh2aNCLSD6cNwxyz",
  "igroupName": "trident",
  "clientCertificate": "ZXR0ZXJwYXB...ICMgJ3BhcGVyc2",
  "clientPrivateKey": "vciwKIyAgZG...0cnksIGRlc2NyaX",
  "trustedCACertificate": "zcyBbaG...b3Igb3duIGNsYXNz"
}
```

ontap-san 使用雙向CHAP的驅動程式

這是最小的後端組態範例。此基本組態會建立一個「ONTAP-SAN」後端、並將「useCHAP」設為「true」。

```

{
  "version": 1,
  "storageDriverName": "ontap-san",
  "managementLIF": "10.0.0.1",
  "dataLIF": "10.0.0.3",
  "svm": "svm_iscsi",
  "labels": {"k8scluster": "test-cluster-1", "backend": "testcluster1-
sanbackend"},
  "useCHAP": true,
  "chapInitiatorSecret": "cl9qxIm36DKyawxy",
  "chapTargetInitiatorSecret": "rqxigXgkesIpwxyz",
  "chapTargetUsername": "iJF4heBRT0TCwxyz",
  "chapUsername": "uh2aNCLSD6cNwxyz",
  "igroupName": "trident",
  "username": "vsadmin",
  "password": "secret"
}

```

ontap-san-economy 驅動程式

```

{
  "version": 1,
  "storageDriverName": "ontap-san-economy",
  "managementLIF": "10.0.0.1",
  "svm": "svm_iscsi_eco",
  "useCHAP": true,
  "chapInitiatorSecret": "cl9qxIm36DKyawxy",
  "chapTargetInitiatorSecret": "rqxigXgkesIpwxyz",
  "chapTargetUsername": "iJF4heBRT0TCwxyz",
  "chapUsername": "uh2aNCLSD6cNwxyz",
  "igroupName": "trident",
  "username": "vsadmin",
  "password": "secret"
}

```

虛擬儲存資源池的後端範例

在下圖所示的範例後端定義檔中、會針對所有儲存資源池設定特定的預設值、例如「無的儲存資源保留」、「假的資源配置」、以及「假的加密」。虛擬儲存資源池是在儲存區段中定義。

在此範例中、有些儲存資源池會設定自己的「資源保留」、「資源配置」和「加密」值、有些資源池會覆寫上述設定的預設值。

```

{

```

```

"version": 1,
"storageDriverName": "ontap-san",
"managementLIF": "10.0.0.1",
"dataLIF": "10.0.0.3",
"svm": "svm_iscsi",
"useCHAP": true,
"chapInitiatorSecret": "cl9qxIm36DKyawxy",
"chapTargetInitiatorSecret": "rqxigXgkesIpwxyz",
"chapTargetUsername": "iJF4heBRT0TCwxyz",
"chapUsername": "uh2aNCLSD6cNwxyz",
"igroupName": "trident",
"username": "vsadmin",
"password": "secret",

"defaults": {
  "spaceAllocation": "false",
  "encryption": "false",
  "qosPolicy": "standard"
},
"labels":{"store": "san_store", "kubernetes-cluster": "prod-cluster-
1"},
"region": "us_east_1",
"storage": [
  {
    "labels":{"protection":"gold", "creditpoints":"40000"},
    "zone":"us_east_1a",
    "defaults": {
      "spaceAllocation": "true",
      "encryption": "true",
      "adaptiveQosPolicy": "adaptive-extreme"
    }
  },
  {
    "labels":{"protection":"silver", "creditpoints":"20000"},
    "zone":"us_east_1b",
    "defaults": {
      "spaceAllocation": "false",
      "encryption": "true",
      "qosPolicy": "premium"
    }
  },
  {
    "labels":{"protection":"bronze", "creditpoints":"5000"},
    "zone":"us_east_1c",
    "defaults": {
      "spaceAllocation": "true",

```



```

        "encryption": "false"
    }
}
]
}

```

以下是「ONTAP-san經濟」驅動程式的iSCSI範例：

```

{
  "version": 1,
  "storageDriverName": "ontap-san-economy",
  "managementLIF": "10.0.0.1",
  "svm": "svm_iscsi_eco",
  "useCHAP": true,
  "chapInitiatorSecret": "cl9qxIm36DKyawxy",
  "chapTargetInitiatorSecret": "rqxigXgkesIpwxyz",
  "chapTargetUsername": "iJF4heBRT0TCwxyz",
  "chapUsername": "uh2aNCLSD6cNwxyz",
  "igroupName": "trident",
  "username": "vsadmin",
  "password": "secret",

  "defaults": {
    "spaceAllocation": "false",
    "encryption": "false"
  },
  "labels": {"store": "san_economy_store"},
  "region": "us_east_1",
  "storage": [
    {
      "labels": {"app": "oracledb", "cost": "30"},
      "zone": "us_east_1a",
      "defaults": {
        "spaceAllocation": "true",
        "encryption": "true"
      }
    },
    {
      "labels": {"app": "postgresdb", "cost": "20"},
      "zone": "us_east_1b",
      "defaults": {
        "spaceAllocation": "false",
        "encryption": "true"
      }
    }
  ],
}

```

```

    {
      "labels":{"app":"mysqldb", "cost":"10"},
      "zone":"us_east_1c",
      "defaults": {
        "spaceAllocation": "true",
        "encryption": "false"
      }
    }
  ]
}

```

將後端對應至StorageClass

下列StorageClass定義係指上述虛擬儲存資源池。使用「parameters.selector」欄位、每個StorageClass都會呼叫哪些虛擬資源池可用於裝載磁碟區。磁碟區將會在所選的虛擬資源池中定義各個層面。

- 第一部StorageClass（「protection黃金級」）將對應至「ONTAP-NAS Flexgroup」後端的第一個、第二個虛擬儲存集區、以及「ONTAP-SAN」後端的第一個虛擬儲存集區。這是唯一提供金級保護的資源池。
- 第二個StorageClass（「protection非gold」）將對應至第三個、第四個虛擬儲存集區、位於「ONTAP-NAS」後端、第二個是第三個虛擬儲存集區、位於「ONTAP-SAN」後端。這是唯一提供金級以外保護層級的資源池。
- 第三個StorageClass（「app-mysqldb」）將對應至「ontap-nas」後端的第四個虛擬儲存資源池、以及「ontap-san經濟」後端的第三個虛擬儲存資源池。這些是唯一提供mysqldb類型應用程式儲存池組態的集區。
- 第四個StorageClass（「protection—silver—creditpoints—20k」）將對應至第三個虛擬儲存資源池（位於「ONTAP—NAS—flexgroup」後端）、第二個虛擬儲存資源池（位於「ONTAP—san」後端）。這些資源池是唯一能以20000個信用點數提供金級保護的資源池。
- 第五個StorageClass（'creditpoint-5k'）將對應到第二個位於「ONTAP-NAS經濟」後端的虛擬儲存資源池、以及位於「ONTAP-SAN」後端的第三個虛擬儲存資源池。這些是唯一提供5000個信用點數的資源池產品。

Astra Trident將決定選取哪個虛擬儲存資源池、並確保符合儲存需求。

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: protection-gold
provisioner: netapp.io/trident
parameters:
  selector: "protection=gold"
  fsType: "ext4"
---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: protection-not-gold
provisioner: netapp.io/trident
parameters:
  selector: "protection!=gold"
  fsType: "ext4"
---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: app-mysqldb
provisioner: netapp.io/trident
parameters:
  selector: "app=mysqldb"
  fsType: "ext4"
---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: protection-silver-creditpoints-20k
provisioner: netapp.io/trident
parameters:
  selector: "protection=silver; creditpoints=20000"
  fsType: "ext4"
---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: creditpoints-5k
provisioner: netapp.io/trident
parameters:
  selector: "creditpoints=5000"
  fsType: "ext4"
```

使用ONTAP NetApp NAS驅動程式設定後端

深入瞭解如何使用ONTAP 功能性和功能性NAS驅動程式來設定功能性的後端。ONTAP Cloud Volumes ONTAP

- ["準備"](#)
- ["組態與範例"](#)

使用者權限

Astra Trident希望以ONTAP 支援或SVM管理員的身分執行、通常是使用「admin」叢集使用者或「vsadmin」SVM使用者、或使用具有相同角色的不同名稱使用者。對於Amazon FSx for NetApp ONTAP 等部署、Astra Trident預期會使用ONTAP 叢集「fsxadmin」使用者或「vsadmin」SVM使用者、或是具有相同角色之不同名稱的使用者、來執行成一個或多個SVM管理員。「fsxadmin」使用者是叢集管理使用者的有限替代。



如果您使用「limitAggregateusage」參數、則需要叢集管理權限。使用Amazon FSx for NetApp ONTAP 支援Astra Trident時、「limitAggregateusage」參數無法搭配「vsadmin」和「fsxadmin」使用者帳戶使用。如果您指定此參數、組態作業將會失敗。

雖然可以在ONTAP 功能區內建立更嚴格的角色、讓Trident驅動程式能夠使用、但我們不建議您這麼做。Trident的大多數新版本都會呼叫額外的API、而這些API必須納入考量、使升級變得困難且容易出錯。

準備

瞭解如何準備使用ONTAP 不含NetApp功能的NAS驅動程式來設定功能完善的後端。ONTAP對於所有ONTAP 的不支援端點、Astra Trident至少需要指派一個集合體給SVM。

對於所有ONTAP 的不支援端點、Astra Trident至少需要指派一個集合體給SVM。

請記住、您也可以執行多個驅動程式、並建立指向一個或多個驅動程式的儲存類別。例如、您可以設定金級、使用「ONTAP-NAS」驅動程式、以及使用「ONTAP-NAS-經濟」的銅級。

您所有的Kubernetes工作節點都必須安裝適當的NFS工具。請參閱 ["請按這裡"](#) 以取得更多詳細資料。

驗證

Astra Trident提供兩種驗證ONTAP 證功能來驗證支援的後端。

- 認證型：ONTAP 對具備所需權限的使用者名稱和密碼。建議使用預先定義的安全登入角色、例如「admin」或「vsadmin」、以確保與ONTAP 各種版本的最大相容性。
- 憑證型：Astra Trident也能ONTAP 使用安裝在後端的憑證與某個叢集進行通訊。在此處、後端定義必須包含用戶端憑證、金鑰及信任的CA憑證（建議使用）的Base64編碼值。

使用者也可以選擇更新現有的後端、選擇從認證移至憑證型、反之亦然。如果*同時提供認證資料和憑證*、Astra Trident將預設使用憑證、同時發出警告、從後端定義中移除認證資料。

啟用認證型驗證

Astra Trident需要SVM範圍/叢集範圍管理員的認證資料、才能與ONTAP 該後端進行通訊。建議使用預先定義的標準角色、例如「admin」或「vsadmin」。這可確保與未來ONTAP 的支援版本保持前瞻相容、因為未來的Astra Trident版本可能會使用功能API。您可以建立自訂的安全登入角色、並與Astra Trident搭配使用、但不建議使用。

後端定義範例如下所示：

```
{
  "version": 1,
  "backendName": "ExampleBackend",
  "storageDriverName": "ontap-nas",
  "managementLIF": "10.0.0.1",
  "dataLIF": "10.0.0.2",
  "svm": "svm_nfs",
  "username": "vsadmin",
  "password": "secret"
}
```

請記住、後端定義是唯一以純文字儲存認證的位置。建立後端之後、使用者名稱/密碼會以Base64編碼、並儲存為Kubernetes機密。建立/更新後端是唯一需要知道認證資料的步驟。因此、這是一項純管理員操作、由Kubernetes /儲存管理員執行。

啟用憑證型驗證

新的和現有的後端可以使用憑證、並與ONTAP 該後端通訊。後端定義需要三個參數。

- 用戶端憑證：用戶端憑證的Base64編碼值。
- 用戶端私密金鑰：關聯私密金鑰的Base64編碼值。
- 信任的CACertificate：受信任CA憑證的Base64編碼值。如果使用信任的CA、則必須提供此參數。如果未使用信任的CA、則可忽略此問題。

典型的工作流程包括下列步驟。

步驟

1. 產生用戶端憑證和金鑰。產生時、請將Common Name (CN) (一般名稱 (CN)) 設定為ONTAP 驗證身分。

```
openssl req -x509 -nodes -days 1095 -newkey rsa:2048 -keyout k8senv.key
-out k8senv.pem -subj "/C=US/ST=NC/L=RTP/O=NetApp/CN=vsadmin"
```

2. 將信任的CA憑證新增ONTAP 至整個叢集。這可能已由儲存管理員處理。如果未使用信任的CA、請忽略。

```
security certificate install -type server -cert-name <trusted-ca-cert-
name> -vserver <vserver-name>
ssl modify -vserver <vserver-name> -server-enabled true -client-enabled
true -common-name <common-name> -serial <SN-from-trusted-CA-cert> -ca
<cert-authority>
```

3. 在ONTAP 支援叢集上安裝用戶端憑證和金鑰 (步驟1)。

```
security certificate install -type client-ca -cert-name <certificate-  
name> -vserver <vserver-name>  
security ssl modify -vserver <vserver-name> -client-enabled true
```

4. 確認ONTAP 支援「cert」 驗證方法的支援功能。

```
security login create -user-or-group-name vsadmin -application ontapi  
-authentication-method cert -vserver <vserver-name>  
security login create -user-or-group-name vsadmin -application http  
-authentication-method cert -vserver <vserver-name>
```

5. 使用產生的憑證測試驗證。以ONTAP Management LIF IP和SVM名稱取代<SfManagement LIF>和<vserver name>。您必須確保LIF的服務原則設定為「預設資料管理」。

```
curl -X POST -Lk https://<ONTAP-Management-  
LIF>/servlets/netapp.servlets.admin.XMLrequest_filer --key k8senv.key  
--cert ~/k8senv.pem -d '<?xml version="1.0" encoding="UTF-8"?><netapp  
xmlns="http://www.netapp.com/filer/admin" version="1.21"  
vfiler="<vserver-name>"><vserver-get></vserver-get></netapp>'
```

6. 使用Base64編碼憑證、金鑰和信任的CA憑證。

```
base64 -w 0 k8senv.pem >> cert_base64  
base64 -w 0 k8senv.key >> key_base64  
base64 -w 0 trustedca.pem >> trustedca_base64
```

7. 使用從上一步取得的值建立後端。

```

$ cat cert-backend-updated.json
{
  "version": 1,
  "storageDriverName": "ontap-nas",
  "backendName": "NasBackend",
  "managementLIF": "1.2.3.4",
  "dataLIF": "1.2.3.8",
  "svm": "vserver_test",
  "clientCertificate": "Faaaakkkkeeee...Vaaalllluuueeeee",
  "clientPrivateKey": "LS0tFaKE...0VaLuES0tLS0K",
  "storagePrefix": "myPrefix_"
}

#Update backend with tridentctl
$ tridentctl update backend NasBackend -f cert-backend-updated.json -n
trident
+-----+-----+-----+-----+
+-----+-----+
|   NAME   | STORAGE DRIVER |           UUID           |
STATE | VOLUMES |
+-----+-----+-----+-----+
+-----+-----+
| NasBackend | ontap-nas      | 98e19b74-aec7-4a3d-8dcf-128e5033b214 |
online |          9 |
+-----+-----+-----+-----+
+-----+-----+

```

更新驗證方法或旋轉認證資料

您可以更新現有的後端、以使用不同的驗證方法、或是旋轉其認證資料。這兩種方法都可行：使用使用者名稱/密碼的後端可更新以使用憑證；使用憑證的後端可更新為使用者名稱/密碼。若要這麼做、請使用更新的「backend.json」檔案、其中包含執行「tridentctl後端更新」所需的參數。

```

$ cat cert-backend-updated.json
{
"version": 1,
"storageDriverName": "ontap-nas",
"backendName": "NasBackend",
"managementLIF": "1.2.3.4",
"dataLIF": "1.2.3.8",
"svm": "vserver_test",
"username": "vsadmin",
"password": "secret",
"storagePrefix": "myPrefix_"
}

#Update backend with tridentctl
$ tridentctl update backend NasBackend -f cert-backend-updated.json -n
trident
+-----+-----+-----+
+-----+-----+
|   NAME   | STORAGE DRIVER |           UUID           |
STATE | VOLUMES |
+-----+-----+-----+
+-----+-----+
| NasBackend | ontap-nas      | 98e19b74-aec7-4a3d-8dcf-128e5033b214 |
online |         9 |
+-----+-----+-----+
+-----+-----+

```



當您旋轉密碼時、儲存管理員必須先更新ONTAP 使用者的密碼（位於BIOS）。接著是後端更新。在循環憑證時、可將多個憑證新增至使用者。然後更新後端以使用新的憑證、之後可從ONTAP 該叢集刪除舊的憑證。

更新後端不會中斷對已建立之磁碟區的存取、也不會影響之後建立的磁碟區連線。成功的後端更新顯示Astra Trident可以與ONTAP 該後端通訊、並處理未來的Volume作業。

管理NFS匯出原則

Astra Trident使用NFS匯出原則來控制其所配置之磁碟區的存取。

使用匯出原則時、Astra Trident提供兩種選項：

- Astra Trident可動態管理匯出原則本身；在此作業模式中、儲存管理員會指定代表可接受IP位址的CIDR區塊清單。Astra Trident會自動將這些範圍內的節點IP新增至匯出原則。或者、如果未指定CIDR、則會將節點上找到的任何全域範圍單點傳送IP新增至匯出原則。
- 儲存管理員可以建立匯出原則、並手動新增規則。除非在組態中指定不同的匯出原則名稱、否則Astra Trident會使用預設的匯出原則。

動態管理匯出原則

「csi Trident」的20.04版提供動態管理輸出原則的能力ONTAP、以利實現幕後。這可讓儲存管理員為工作節點IP指定允許的位址空間、而非手動定義明確的規則。它可大幅簡化匯出原則管理；修改匯出原則不再需要在儲存叢集上進行手動介入。此外、這有助於將儲存叢集的存取限制在指定範圍內有IP的工作者節點、以支援精簡且自動化的管理。



只有「csi Trident」才能動態管理匯出原則。請務必確保工作節點未被NATed。

範例

必須使用兩種組態選項。以下是後端定義範例：

```
{
  "version": 1,
  "storageDriverName": "ontap-nas",
  "backendName": "ontap_nas_auto_export",
  "managementLIF": "192.168.0.135",
  "svm": "svm1",
  "username": "vsadmin",
  "password": "FaKePaSsWoRd",
  "autoExportCIDRs": ["192.168.0.0/24"],
  "autoExportPolicy": true
}
```



使用此功能時、您必須確保SVM中的根連接點具有預先設定的匯出原則、並具有允許節點CIDR區塊（例如預設匯出原則）的匯出規則。請務必遵循NetApp建議的最佳實務做法、為Astra Trident指定SVM。

以下是使用上述範例說明此功能的運作方式：

- 將「AutoExportPolicy」設為「true」。這表示Astra Trident將為「svm1」SVM建立匯出原則、並使用「AutoExportCIDR」位址區塊來處理新增和刪除規則的作業。例如、在SVM上、UUID為403b5326-8482-40dB/96d0-d83fb3f4daec和「autoExportPolicy」設為「true」的後端會建立名為「trident 403b5326-8482-40dB/96d0-d83fb3f4daec」的匯出原則。
- 「AutoExportCIDR」包含位址區塊清單。此欄位為選用欄位、預設為「0.00.0.0/0」、「:/0」。如果未定義、Astra Trident會新增在工作者節點上找到的所有全域範圍單點傳送位址。

在此範例中、會提供「192.168.0.0/24」位址空間。這表示、屬於此位址範圍的Kubernetes節點IP將新增至Astra Trident所建立的匯出原則。當Astra Trident登錄其執行的節點時、會擷取節點的IP位址、並對照「AutoExportCIDR」中提供的位址區塊進行檢查。篩選IP之後、Astra Trident會針對所探索的用戶端IP建立匯出原則規則、並針對所識別的每個節點建立一個規則。

您可以在建立後端後、更新「AutoExportPolicy」和「AutoExportCTR」。您可以為自動管理或刪除現有CIDR的後端附加新的CIDR。刪除CIDR時請務必謹慎、以確保不會中斷現有的連線。您也可以選擇停用後端的「autoExportPolicy」、然後回到手動建立的匯出原則。這需要在後端組態中設定「exportPolicy」參數。

Astra Trident建立或更新後端之後、您可以使用「tridentctl」或對應的「tridentbackend」CRD來檢查後端：

```

$ ./tridentctl get backends ontap_nas_auto_export -n trident -o yaml
items:
- backendUUID: 403b5326-8482-40db-96d0-d83fb3f4daec
  config:
    aggregate: ""
    autoExportCIDRs:
    - 192.168.0.0/24
    autoExportPolicy: true
    backendName: ontap_nas_auto_export
    chapInitiatorSecret: ""
    chapTargetInitiatorSecret: ""
    chapTargetUsername: ""
    chapUsername: ""
    dataLIF: 192.168.0.135
    debug: false
    debugTraceFlags: null
    defaults:
      encryption: "false"
      exportPolicy: <automatic>
      fileType: ext4

```

當節點新增至Kubernetes叢集並向Astra Trident控制器註冊時、現有後端的匯出原則會更新（前提是它們位於後端的「AutoExportCIDR」中指定的位址範圍內）。

移除節點時、Astra Trident會檢查所有線上的後端、以移除節點的存取規則。Astra Trident將此節點IP從託管後端的匯出原則中移除、可防止惡意掛載、除非叢集中的新節點重複使用此IP。

對於先前存在的後端、使用「tridentctl update backend」更新後端、可確保Astra Trident自動管理匯出原則。這會建立以後端UUID命名的新匯出原則、而後端上的磁碟區會在重新掛載時使用新建立的匯出原則。



刪除具有自動管理匯出原則的後端、將會刪除動態建立的匯出原則。如果重新建立後端、則會將其視為新的後端、並導致建立新的匯出原則。

如果即時節點的IP位址已更新、您必須重新啟動節點上的Astra Trident Pod。Astra Trident接著會更新其管理的後端匯出原則、以反映此IP變更。

組態選項與範例

瞭解如何透過ONTAP Astra Trident安裝來建立及使用NetApp NAS驅動程式。本節提供後端組態範例、以及如何將後端對應至StorageClass的詳細資料。

後端組態選項

如需後端組態選項、請參閱下表：

參數	說明	預設
「分度」		永遠為1

參數	說明	預設
「storageDriverName」	儲存驅動程式名稱	「ONTAP-NAS」、「ONTAP-NAS-節約型」、「ONTAP-NAS-flexgroup」、「ONTAP-SAN」、「ONTAP-san經濟型」
「後端名稱」	自訂名稱或儲存後端	驅動程式名稱+「_」+dataLIF
《馬納格門達利》	叢集或SVM管理LIF的IP位址	「10.0.0.1」、「[2001:1234:abcd:::fefo]」
「DataLIF」	傳輸協定LIF的IP位址。IPv6使用方括弧。設定後無法更新	除非另有說明、否則由SVM衍生
「AutoExpportPolicy」	啟用自動匯出原則建立與更新[布林值]	錯
《AutoExpportCIDR》（自動匯出CTR）	啟用「AutoExpportPolicy」時、用來篩選Kubernetes節點IP的CIDR清單	[「0.00.0/0」、「:/0」]
《標籤》	套用到磁碟區的任意JSON-格式化標籤集	「」
「用戶端憑證」	用戶端憑證的Base64編碼值。用於憑證型驗證	「」
「clientPrivate Key」	用戶端私密金鑰的Base64編碼值。用於憑證型驗證	「」
「可信賴的CACertificate」	受信任CA憑證的Base64編碼值。選用。用於憑證型驗證	「」
《使用者名稱》	連線至叢集/ SVM的使用者名稱。用於認證型驗證	
密碼	連線至叢集/ SVM的密碼。用於認證型驗證	
《虛擬機器》	要使用的儲存虛擬機器	如果指定SVM "managementLIF"則衍生
「igroupName」	要使用之SAN磁碟區的igroup名稱	「Trident -<後端-UUID>」
「storagePrefix」	在SVM中配置新磁碟區時所使用的前置碼。設定後無法更新	「Trident」
「限制Aggregateusage」	如果使用率高於此百分比、則無法進行資源配置。*不適用於Amazon FSX for ONTAP Sfor Sfor *	「」（預設不強制執行）
《限制Volume大小》	如果要求的磁碟區大小高於經濟驅動程式的此值、則無法進行資源配置。	「」（預設不強制執行）
《lunsPerFlexvol》	每FlexVol 個LUN的最大LUN數量、範圍必須在[50、200]	「100」
「DebugTraceFlags」	疑難排解時要使用的偵錯旗標。範例：{"API":假、「方法」:true}	null

參數	說明	預設
「nfsMountOptions」	以逗號分隔的NFS掛載選項清單	「」
"qtreesPerFlexvol"	每FlexVol 個邊的最大qtree數、必須在範圍內[50、300]	「200」
《useREST》	使用ONTAP Isrest API的布林參數。技術預覽	錯



「useREST」是一種建議用於測試環境、而非用於正式作業工作負載的技術預覽。當設為「true」時、Astra Trident會使用ONTAP「Isrest API」與後端通訊。此功能需要ONTAP 使用更新版本的版本。此外ONTAP、所使用的登入角色必須能夠存取「ONTAP 發揮作用」應用程式。這對預先定義的「vsadmin」和「叢集管理」角色感到滿意。

若要與ONTAP 此叢集通訊、您應該提供驗證參數。這可能是安全登入或安裝憑證的使用者名稱/密碼。



如果您使用Amazon FSX for NetApp ONTAP Sendbackend、請勿指定「limitAggregateusage」參數。Amazon FSx for NetApp ONTAP 的「fsxadmin」和「vsadmin」角色不包含擷取Aggregate使用量並透過Astra Trident加以限制的必要存取權限。



除非您正在進行疑難排解並需要詳細的記錄傾印、否則請勿使用「debugTraceFlags」。



建立後端時、請記住、建立後無法修改「dataLIF」和「儲存前置字元」。若要更新這些參數、您需要建立新的後端。

可以為「managementLIF」選項指定完整網域名稱（FQDN）。也可以為「dataLIF」選項指定FQDN、在這種情況下、FQDN將用於NFS裝載作業。如此一來、您就能建立循環DNS、在多個資料生命期之間實現負載平衡。

所有ONTAP 的不完整驅動程式也可設定為IPv6位址。請務必使用「-use IPv6」旗標來安裝Astra Trident。必須謹慎定義方括弧內的「managementLIF」IPv6位址。



使用IPv6位址時、請務必在方括弧內定義「managementLIF」和「dataLIF」（若包含在後端定義中）、例如[28e8:d9fb:a825:b7bf:69a8:d02f:9e7b:3555]。如果未提供「data LIF」、Astra Trident會從SVM擷取IPv6資料LIF。

使用「AutoExport Policy」（自動匯出原則）和「AutoExport CIDR」（自動匯出CTR）選項、「csi Trident」（SCSI Trident）可以自動管理匯出原則。所有的ONTAP-NAS-*驅動程式均支援此功能。

對於「ONTAP-NAS-EAS-經濟」驅動程式、「limitVolume Sizes」選項也會限制其管理的qtree和LUN磁碟區大小上限、而「qtreesPerFlexvol」選項則可自訂每FlexVol 個經濟體系的最高配額樹數。

可以使用「nfsMountOptions」參數來指定掛載選項。Kubernetes持續磁碟區的掛載選項通常會在儲存類別中指定、但如果儲存類別中未指定掛載選項、則Astra Trident會改回使用儲存後端組態檔中指定的掛載選項。如果儲存類別或組態檔中未指定掛載選項、則Astra Trident不會在相關的持續磁碟區上設定任何掛載選項。



Astra Trident在所有使用「ONTAP-NAS」和「ONTAP-NAS-flexgroup」建立的磁碟區的「Comments」（註解）欄位中設定資源配置標籤。根據所使用的驅動程式、意見會設定在FlexVol以下項目上：（「ONTAP-NAS」）或FlexGroup（「ONTAP-NAS-flexgroup」）。Astra Trident會在儲存資源池配置時、將儲存資源池上的所有標籤複製到儲存磁碟區。儲存管理員可以定義每個儲存資源池的標籤、並將儲存資源池中建立的所有磁碟區分組。這是根據後端組態中提供的一組可自訂標籤、方便區分磁碟區的方法。

用於資源配置磁碟區的后端組態選項

您可以在組態的特定區段中、使用這些選項來控制預設配置每個Volume的方式。如需範例、請參閱下列組態範例。

參數	說明	預設
"paceAllocate (配置) "	LUN的空間分配	「真的」
《保護區》	空間保留模式；「無」（精簡）或「Volume」（完整）	「無」
「快照原則」	要使用的Snapshot原則	「無」
「qosPolicy」	要指派給所建立磁碟區的QoS原則群組。選擇每個儲存集區/後端的其中一個qosPolicy或adaptiveQosPolicy	「」
《adaptiveQosPolicy》	要指派給所建立磁碟區的調適性QoS原則群組。選擇每個儲存集區/後端的其中一個qosPolicy或adaptiveQosPolicy。不受ONTAP-NAS-經濟支援。	「」
「快照保留區」	保留給快照「0」的磁碟區百分比	如果「快照原則」為「無」、則其他為「」
「PlitOnClone」	建立複本時、從其父複本分割複本	「假」
加密	啟用NetApp Volume加密	「假」
《生態樣式》	新磁碟區的安全樣式	「UNIX」
「分層政策」	分層原則以使用「無」	ONTAP 9.5之前的SVM-DR組態為「純快照」
unix權限	新磁碟區的模式	「777」
snapshotDir	控制「.snapshot」目錄的可見度	「假」
匯出原則	要使用的匯出原則	「預設」
安全性樣式	新磁碟區的安全樣式	「UNIX」



搭配Astra Trident使用QoS原則群組需要ONTAP 使用更新版本的版本。建議使用非共用的QoS原則群組、並確保原則群組會個別套用至每個組成群組。共享的QoS原則群組將強制所有工作負載的總處理量上限。

以下是已定義預設值的範例：

```

{
  "version": 1,
  "storageDriverName": "ontap-nas",
  "backendName": "customBackendName",
  "managementLIF": "10.0.0.1",
  "dataLIF": "10.0.0.2",
  "labels": {"k8scluster": "dev1", "backend": "dev1-nasbackend"},
  "svm": "trident_svm",
  "username": "cluster-admin",
  "password": "password",
  "limitAggregateUsage": "80%",
  "limitVolumeSize": "50Gi",
  "nfsMountOptions": "nfsvers=4",
  "debugTraceFlags": {"api":false, "method":true},
  "defaults": {
    "spaceReserve": "volume",
    "qosPolicy": "premium",
    "exportPolicy": "myk8scluster",
    "snapshotPolicy": "default",
    "snapshotReserve": "10"
  }
}

```

對於「ONTAP-NAS」和「ONTAP-NAS-flexGroups」、Astra Trident現在採用新的計算方法、確保FlexVol 利用snapshotReserve百分比和PVC正確調整其規模。當使用者要求使用PVCs時、Astra Trident會FlexVol 使用新的計算方式、建立原始的包含更多空間的候選區。此計算可確保使用者在永久虛擬磁碟中獲得所要求的可寫入空間、且空間不得小於所要求的空間。在v21.07之前、當使用者要求使用PVC（例如5GiB）、快照保留區達到50%時、他們只能獲得2.5GiB的可寫入空間。這是因為使用者要求的是整個磁碟區、而「快照保留區」則是其中的一個百分比。使用Trident 21.07時、使用者要求的是可寫入空間、而Astra Trident則將「快照保留區」編號定義為整個Volume的百分比。這不適用於「ONTAP-NAS-經濟」。請參閱下列範例以瞭解此功能的運作方式：

計算方式如下：

$$\text{Total volume size} = (\text{PVC requested size}) / (1 - (\text{snapshotReserve percentage}) / 100)$$

對於snapshotReserve = 50%、而PVC要求= 5GiB、磁碟區總大小為2/0.5 = 10GiB、可用大小為5GiB、這是使用者在PVC要求中要求的大小。「volume show (Volume show) 命令應顯示類似以下範例的結果：

Vserver	Volume	Aggregate	State	Type	Size	Available	Used%
	_pvc_89f1c156_3801_4de4_9f9d_034d54c395f4		online	RW	10GB	5.00GB	0%
	_pvc_e8372153_9ad9_474a_951a_08ae15e1c0ba		online	RW	1GB	511.8MB	0%

2 entries were displayed.

在升級Astra Trident時、先前安裝的現有後端會按照上述說明來配置磁碟區。對於在升級之前建立的磁碟區、您應該調整其磁碟區大小、以便觀察變更。例如、較早前使用「快照保留區=50」的2GiB室早導致磁碟區提供1GiB的可寫入空間。例如、將磁碟區大小調整為3GiB、可讓應用程式在6 GiB磁碟區上擁有3GiB的可寫入空間。

最低組態範例

下列範例顯示基本組態、讓大部分參數保留預設值。這是定義後端最簡單的方法。



如果您在NetApp ONTAP 支援Trident的NetApp支援上使用Amazon FSX、建議您指定lifs的DNS名稱、而非IP位址。

ontap-nas 具有憑證型驗證的驅動程式

這是最小的後端組態範例。「clientCertificate」、「clientPrivate Key」和「Trusted CACertificate」（選用、如果使用信任的CA）會分別填入「backend.json」、並採用用戶端憑證、私密金鑰和信任的CA憑證的64編碼值。

```
{
  "version": 1,
  "backendName": "DefaultNASBackend",
  "storageDriverName": "ontap-nas",
  "managementLIF": "10.0.0.1",
  "dataLIF": "10.0.0.15",
  "svm": "nfs_svm",
  "clientCertificate": "ZXR0ZXJwYXB...ICMgJ3BhcGVyc2",
  "clientPrivateKey": "vcIwKIyAgZG...0cnksIGRlc2NyaX",
  "trustedCACertificate": "zcyBbaG...b3Igb3duIGNsYXNz",
  "storagePrefix": "myPrefix_"
}
```

ontap-nas 具有自動匯出原則的驅動程式

本範例說明如何指示Astra Trident使用動態匯出原則來自動建立及管理匯出原則。這對「ONTAP-NAS-節約」和「ONTAP-NAS-Flexgroup」驅動程式的運作方式相同。

```

{
  "version": 1,
  "storageDriverName": "ontap-nas",
  "managementLIF": "10.0.0.1",
  "dataLIF": "10.0.0.2",
  "svm": "svm_nfs",
  "labels": {"k8scluster": "test-cluster-east-1a", "backend": "test1-
nasbackend"},
  "autoExportPolicy": true,
  "autoExportCIDRs": ["10.0.0.0/24"],
  "username": "admin",
  "password": "secret",
  "nfsMountOptions": "nfsvers=4",
}

```

ontap-nas-flexgroup 驅動程式

```

{
  "version": 1,
  "storageDriverName": "ontap-nas-flexgroup",
  "managementLIF": "10.0.0.1",
  "dataLIF": "10.0.0.2",
  "labels": {"k8scluster": "test-cluster-east-1b", "backend": "test1-
ontap-cluster"},
  "svm": "svm_nfs",
  "username": "vsadmin",
  "password": "secret",
}

```

ontap-nas 使用IPv6的驅動程式

```

{
  "version": 1,
  "storageDriverName": "ontap-nas",
  "backendName": "nas_ipv6_backend",
  "managementLIF": "[5c5d:5edf:8f:7657:bef8:109b:1b41:d491]",
  "labels": {"k8scluster": "test-cluster-east-1a", "backend": "test1-ontap-
ipv6"},
  "svm": "nas_ipv6_svm",
  "username": "vsadmin",
  "password": "netapp123"
}

```


ontap-nas-economy 驅動程式

```
{
  "version": 1,
  "storageDriverName": "ontap-nas-economy",
  "managementLIF": "10.0.0.1",
  "dataLIF": "10.0.0.2",
  "svm": "svm_nfs",
  "username": "vsadmin",
  "password": "secret"
}
```

虛擬儲存資源池的後端範例

在下圖所示的範例後端定義檔中、會針對所有儲存資源池設定特定的預設值、例如「無的儲存資源保留」、「假的資源配置」、以及「假的加密」。虛擬儲存資源池是在儲存區段中定義。

在此範例中、有些儲存資源池會設定自己的「資源保留」、「資源配置」和「加密」值、有些資源池會覆寫上述設定的預設值。

ontap-nas 驅動程式

```
{
  {
    "version": 1,
    "storageDriverName": "ontap-nas",
    "managementLIF": "10.0.0.1",
    "dataLIF": "10.0.0.2",
    "svm": "svm_nfs",
    "username": "admin",
    "password": "secret",
    "nfsMountOptions": "nfsvers=4",

    "defaults": {
      "spaceReserve": "none",
      "encryption": "false",
      "qosPolicy": "standard"
    },
    "labels": {"store": "nas_store", "k8scluster": "prod-cluster-1"},
    "region": "us_east_1",
    "storage": [
      {
        "labels": {"app": "msoffice", "cost": "100"},
        "zone": "us_east_1a",
        "defaults": {
          "spaceReserve": "volume",

```

```

        "encryption": "true",
        "unixPermissions": "0755",
        "adaptiveQosPolicy": "adaptive-premium"
    }
},
{
    "labels":{"app":"slack", "cost":"75"},
    "zone":"us_east_1b",
    "defaults": {
        "spaceReserve": "none",
        "encryption": "true",
        "unixPermissions": "0755"
    }
},
{
    "labels":{"app":"wordpress", "cost":"50"},
    "zone":"us_east_1c",
    "defaults": {
        "spaceReserve": "none",
        "encryption": "true",
        "unixPermissions": "0775"
    }
},
{
    "labels":{"app":"mysqldb", "cost":"25"},
    "zone":"us_east_1d",
    "defaults": {
        "spaceReserve": "volume",
        "encryption": "false",
        "unixPermissions": "0775"
    }
}
]
}

```

ontap-nas-flexgroup 驅動程式

```

{
    "version": 1,
    "storageDriverName": "ontap-nas-flexgroup",
    "managementLIF": "10.0.0.1",
    "dataLIF": "10.0.0.2",
    "svm": "svm_nfs",
    "username": "vsadmin",
    "password": "secret",
}

```

```

"defaults": {
  "spaceReserve": "none",
  "encryption": "false"
},
"labels":{"store":"flexgroup_store", "k8scluster": "prod-cluster-1"},
"region": "us_east_1",
"storage": [
  {
    "labels":{"protection":"gold", "creditpoints":"50000"},
    "zone":"us_east_1a",
    "defaults": {
      "spaceReserve": "volume",
      "encryption": "true",
      "unixPermissions": "0755"
    }
  },
  {
    "labels":{"protection":"gold", "creditpoints":"30000"},
    "zone":"us_east_1b",
    "defaults": {
      "spaceReserve": "none",
      "encryption": "true",
      "unixPermissions": "0755"
    }
  },
  {
    "labels":{"protection":"silver", "creditpoints":"20000"},
    "zone":"us_east_1c",
    "defaults": {
      "spaceReserve": "none",
      "encryption": "true",
      "unixPermissions": "0775"
    }
  },
  {
    "labels":{"protection":"bronze", "creditpoints":"10000"},
    "zone":"us_east_1d",
    "defaults": {
      "spaceReserve": "volume",
      "encryption": "false",
      "unixPermissions": "0775"
    }
  }
]
}

```

```
{
  "version": 1,
  "storageDriverName": "ontap-nas-economy",
  "managementLIF": "10.0.0.1",
  "dataLIF": "10.0.0.2",
  "svm": "svm_nfs",
  "username": "vsadmin",
  "password": "secret",

  "defaults": {
    "spaceReserve": "none",
    "encryption": "false"
  },
  "labels":{"store":"nas_economy_store"},
  "region": "us_east_1",
  "storage": [
    {
      "labels":{"department":"finance", "creditpoints":"6000"},
      "zone":"us_east_1a",
      "defaults": {
        "spaceReserve": "volume",
        "encryption": "true",
        "unixPermissions": "0755"
      }
    },
    {
      "labels":{"department":"legal", "creditpoints":"5000"},
      "zone":"us_east_1b",
      "defaults": {
        "spaceReserve": "none",
        "encryption": "true",
        "unixPermissions": "0755"
      }
    },
    {
      "labels":{"department":"engineering", "creditpoints":"3000"},
      "zone":"us_east_1c",
      "defaults": {
        "spaceReserve": "none",
        "encryption": "true",
        "unixPermissions": "0775"
      }
    },
    {
```

```

    "labels":{"department":"humanresource",
"creditpoints":"2000"},
    "zone":"us_east_1d",
    "defaults": {
        "spaceReserve": "volume",
        "encryption": "false",
        "unixPermissions": "0775"
    }
}
]
}

```

將後端對應至StorageClass

下列StorageClass定義係指上述虛擬儲存資源池。使用「parameters.selector」欄位、每個StorageClass都會呼叫哪些虛擬資源池可用於裝載磁碟區。磁碟區將會在所選的虛擬資源池中定義各個層面。

- 第一部StorageClass（「protection黃金級」）將對應至「ONTAP-NAS Flexgroup」後端的第一個、第二個虛擬儲存集區、以及「ONTAP-SAN」後端的第一個虛擬儲存集區。這是唯一提供金級保護的資源池。
- 第二個StorageClass（「protection非gold」）將對應至第三個、第四個虛擬儲存集區、位於「ONTAP-NAS」後端、第二個是第三個虛擬儲存集區、位於「ONTAP-SAN」後端。這是唯一提供金級以外保護層級的資源池。
- 第三個StorageClass（「app-mysqldb」）將對應至「ontap-nas」後端的第四個虛擬儲存資源池、以及「ontap-san經濟」後端的第三個虛擬儲存資源池。這些是唯一提供mysqldb類型應用程式儲存池組態的集區。
- 第四個StorageClass（「protection-silver-creditpoints-20k」）將對應至第三個虛擬儲存資源池（位於「ONTAP-NAS-flexgroup」後端）、第二個虛擬儲存資源池（位於「ONTAP-san」後端）。這些資源池是唯一能以20000個信用點數提供金級保護的資源池。
- 第五個StorageClass（'creditpoint-5k'）將對應到第二個位於「ONTAP-NAS經濟」後端的虛擬儲存資源池、以及位於「ONTAP-SAN」後端的第三個虛擬儲存資源池。這些是唯一提供5000個信用點數的資源池產品。

Astra Trident將決定選取哪個虛擬儲存資源池、並確保符合儲存需求。

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: protection-gold
provisioner: netapp.io/trident
parameters:
  selector: "protection=gold"
  fsType: "ext4"
---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: protection-not-gold
provisioner: netapp.io/trident
parameters:
  selector: "protection!=gold"
  fsType: "ext4"
---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: app-mysqldb
provisioner: netapp.io/trident
parameters:
  selector: "app=mysqldb"
  fsType: "ext4"
---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: protection-silver-creditpoints-20k
provisioner: netapp.io/trident
parameters:
  selector: "protection=silver; creditpoints=20000"
  fsType: "ext4"
---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: creditpoints-5k
provisioner: netapp.io/trident
parameters:
  selector: "creditpoints=5000"
  fsType: "ext4"
```

使用Astra Trident搭配Amazon FSX for NetApp ONTAP 解決方案

"Amazon FSX for NetApp ONTAP 產品"是一項完全託管的AWS服務、可讓客戶啟動及執行採用NetApp ONTAP的一套儲存作業系統的檔案系統。Amazon FSX for NetApp ONTAP 功能可讓您運用熟悉的NetApp功能、效能和管理功能、同時充分發揮儲存AWS資料的簡易性、敏捷度、安全性和擴充性。FSX支援ONTAP的許多檔案系統功能和管理API。

檔案系統是Amazon FSX的主要資源、類似ONTAP 於內部部署的一個叢集。在每個SVM中、您可以建立一個或多個磁碟區、這些磁碟區是儲存檔案系統中檔案和資料夾的資料容器。有了Amazon FSX for NetApp ONTAP 的功能、Data ONTAP 即可在雲端以託管檔案系統的形式提供支援。新的檔案系統類型稱為* NetApp ONTAP Sing*。

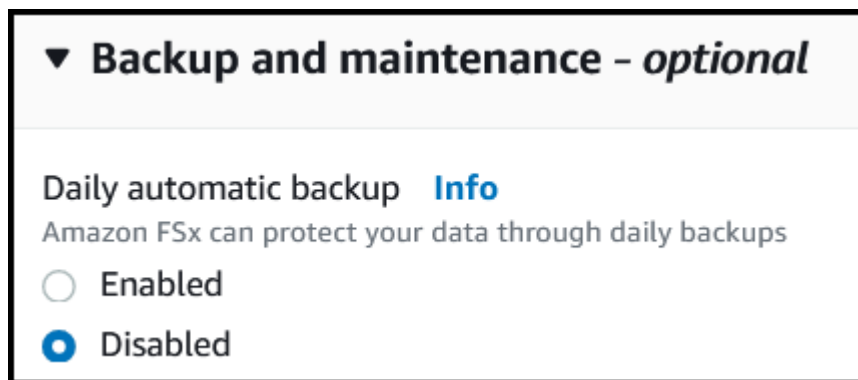
使用Astra Trident搭配Amazon FSX for NetApp ONTAP 供應NetApp時、您可以確保在Amazon Elastic Kubernetes Service (EKS) 中執行的Kubernetes叢集、能夠配置區塊和檔案以ONTAP 支援的持續磁碟區。

建立Amazon FSX for ONTAP Sif供 檔案系統使用

在啟用自動備份的Amazon FSX檔案系統上建立的磁碟區、無法由Trident刪除。若要刪除PVCs、您需要手動刪除PV和FSXfor ONTAP the Sesfvolume。

若要避免此問題：

- 請勿使用「快速建立」來建立FSX for ONTAP the Suse檔案系統。快速建立工作流程可自動備份、但不提供退出選項。
- 使用「標準建立」時、請停用自動備份。停用自動備份可讓Trident成功刪除磁碟區、而無需進一步手動介入。



瞭解Astra Trident

如果您是Astra Trident的新手、請使用下列連結來熟悉您的操作：

- ["常見問題集"](#)
- ["使用Astra Trident的要求"](#)
- ["部署Astra Trident"](#)
- ["最佳實務做法、可設定ONTAP 適用於Cloud Volumes ONTAP NetApp ONTAP 的功能、包括功能、功能及Amazon FSX"](#)
- ["整合Astra Trident"](#)

- ["支援SAN後端組態ONTAP"](#)
- ["ASNAS後端組態ONTAP"](#)

深入瞭解驅動程式功能 ["請按這裡"](#)。

適用於NetApp ONTAP 的Amazon FSX ["FabricPool"](#) 管理儲存層。它可讓您根據資料是否經常存取、將資料儲存在一個層級中。

Astra Trident預期以「vsadmin」SVM使用者或具有相同角色之不同名稱的使用者身分執行。Amazon FSx for NetApp ONTAP 的使用者是「fsxadmin」使用者、只能有限地取代ONTAP「admin」叢集使用者。不建議將「fsxadmin」使用者搭配Trident使用、因為「vsadmin」SVM使用者可以存取更多Astra Trident功能。

驅動程式

您ONTAP 可以使用下列驅動程式、將Astra Trident與Amazon FSX for NetApp整合：

- 「ONTAP-san」：每個提供的PV都是其專屬Amazon FSX for NetApp ONTAP 的LUN。
- 「ONTAP-san經濟型」：每個配置的PV都是LUN、每個Amazon FSX for NetApp ONTAP 的LUN數量可設定。
- 「ONTAP-NAS」：每個提供的PV都是適用於NetApp ONTAP 的完整Amazon FSX。
- 「ONTAP-NAS-EAS」：每個提供的PV都是qtree、每個Amazon FSX的NetApp ONTAP 功能是可設定的配額樹數。
- 「ONTAP-NAS-Flexgroup」：每個提供的PV都是適用於NetApp ONTAP FlexGroup 的完整Amazon FSX。

驗證

Astra Trident提供兩種驗證模式：

- 憑證型：Astra Trident會使用SVM上安裝的憑證、與FSX檔案系統上的SVM進行通訊。
- 認證型：您可以將「fsxadmin」使用者用於您的檔案系統、或是為SVM設定的「vsadmin」使用者。



我們強烈建議您使用「vsadmin」使用者、而非「fssadmin」來設定後端。Astra Trident將使用此使用者名稱和密碼與FSX檔案系統通訊。

若要深入瞭解驗證、請參閱下列連結：

- ["NAS ONTAP"](#)
- ["SAN ONTAP"](#)

在EKS上部署及設定Astra Trident搭配Amazon FSX for NetApp ONTAP

您需要的產品

- 現有的Amazon EKS叢集或自行管理的Kubernetes叢集、已安裝「kubectll」。
- 現有的Amazon FSX-適用於NetApp ONTAP 的支援資料系統和儲存虛擬機器（SVM）、可從叢集的工作節點存取。
- 已準備好的工作節點 ["NFS和/或iSCSI"](#)。



請務必遵循Amazon Linux和Ubuntu所需的節點準備步驟 "[Amazon機器映像](#)" (AMIs) 、視您的EKS AMI類型而定。

如需其他Astra Trident需求、請參閱 "[請按這裡](#)"。

步驟

1. 使用以下其中一種方式部署Astra Trident：./ Trident入門/ Kubernetes-Deploy .html[部署方法]。
2. 設定Astra Trident如下：
 - a. 收集SVM的管理LIF DNS名稱。例如、使用AWS CLI執行下列命令後、在「Endings」→「管理」下找到「DNSName」項目：

```
aws fsx describe-storage-virtual-machines --region <file system region>
```

3. 建立及安裝驗證憑證。如果您使用的是「ONTAP-SAN」後端、請參閱 "[請按這裡](#)"。如果您使用的是「ONTAP-NAS」後端、請參閱 "[請按這裡](#)"。



您可以使用SSH從任何位置登入檔案系統（例如安裝憑證）、而該SSH可連至檔案系統。使用「fsxadmin」使用者、您在建立檔案系統時設定的密碼、以及「AWS FSx file-systems」中的管理DNS名稱。

4. 使用您的憑證和管理LIF的DNS名稱建立後端檔案、如下例所示：

```
{
  "version": 1,
  "storageDriverName": "ontap-san",
  "backendName": "customBackendName",
  "managementLIF": "svm-XXXXXXXXXXXXXXXXXX.fs-XXXXXXXXXXXXXXXXXX.fsx.us-east-2.aws.internal",
  "svm": "svm01",
  "clientCertificate": "ZXR0ZXJwYXB...ICMgJ3BhcGVyc2",
  "clientPrivateKey": "vcIwKIyAgZG...0cnksIGRlc2NyaX",
  "trustedCACertificate": "zcyBbaG...b3Igb3duIGNsYXNz",
}
```

如需建立後端的相關資訊、請參閱下列連結：

- "[使用ONTAP NetApp NAS驅動程式設定後端](#)"
- "[使用ONTAP SAN驅動程式設定後端](#)"



不要指定"ontap-san"和"ONTAP-san經濟"驅動程序的"data LIF"允許Astra Trident使用多重路徑。



「limitAggregateusage」參數無法搭配「vsadmin」和「fsxadmin」使用者帳戶使用。如果您指定此參數、組態作業將會失敗。

部署之後、請執行建立的步驟 "[儲存類別、配置磁碟區、然後將磁碟區掛載到Pod中](#)"。

如需詳細資訊、請參閱

- "[Amazon FSX for NetApp ONTAP 的支援文件](#)"
- "[Amazon FSX for NetApp ONTAP 的部落格文章](#)"

使用kubectll建立後端

後端定義了Astra Trident與儲存系統之間的關係。它告訴Astra Trident如何與該儲存系統通訊、以及Astra Trident如何從該儲存系統配置磁碟區。安裝Astra Trident之後、下一步是建立後端。「TridentBackendConfig」自訂資源定義（CRD）可讓您直接透過Kubernetes介面建立及管理Trident後端。您可以使用「kubectll」或Kubernetes發佈的等效CLI工具來執行此作業。

TridentBackendConfig

「TridentBackendConfig」（「tbc」、「tbconfig」、「tbackendconfig」）是前端、名稱為CRD、可讓您使用「kubectll」來管理Astra Trident後端。Kubernetes與儲存管理員現在可以直接透過Kubernetes CLI建立及管理後端、而不需要使用專屬的命令列公用程式（「tridentctl」）。

建立「TridentBackendConfig」物件之後、會發生下列情況：

- Astra Trident會根據您提供的組態自動建立後端。內部代表的是「TridentBackend」（「tbc」、「tridentbackend」）、CR。
- 《TridentBackendConfig》與由Astra Trident所建立的《TridentBackend》有獨特的關聯。

每個「TridentBackendConfig」都有一對一的對應、並有「TridentBackend」。前者是提供給使用者設計及設定後端的介面、後者是Trident代表實際後端物件的方式。



「TridentBackend」CRS是由Astra Trident自動建立的。您*不應該*修改這些項目。如果您想要更新後端、請修改「TridentBackendConfig」物件來執行此動作。

請參閱下列範例、以瞭解「TridentBackendConfig」CR的格式：

```

apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-ontap-san
spec:
  version: 1
  backendName: ontap-san-backend
  storageDriverName: ontap-san
  managementLIF: 10.0.0.1
  dataLIF: 10.0.0.2
  svm: trident_svm
  credentials:
    name: backend-tbc-ontap-san-secret

```

您也可以查看中的範例 "[Trident安裝程式](#)" 所需儲存平台/服務的範例組態目錄。

「show」會採用後端特定的組態參數。在此範例中、後端使用「ONTAP-san」儲存驅動程式、並使用此處表格列出的組態參數。如需所需儲存驅動程式的組態選項清單、請參閱 "[儲存驅動程式的後端組態資訊](#)"。

在《TridentBackendConfig》（CRR）中新推出的「sPEC」一節也包含「認證」和「刪除原則」欄位：

- 「認證資料」：此參數為必填欄位、包含用於驗證儲存系統/服務的認證資料。此設定為使用者建立的Kubernetes Secret。認證資料無法以純文字格式傳遞、因此會產生錯誤。
- 「刪除原則」：此欄位可定義刪除「TridentBackendConfig」時應發生的情況。可能需要兩種可能的值之一：
 - 「刪除」：這會同時刪除「TridentBackendConfig」和相關後端。這是預設值。
 - 「保留」：刪除「TridentBackendConfig」（TridentBackendConfig）CR時、後端定義仍會存在、並可使用「tridentctl」進行管理。將刪除原則設為「保留」可讓使用者降級至較早版本（21.04之前）、並保留建立的後端。此欄位的值可在建立「TridentBackendConfig」之後更新。



後端名稱是使用「sPEC.backendName」來設定。如果未指定、則會將後端名稱設為「TridentBackendConfig」物件（metadata.name）的名稱。建議使用「sPEC.backendName」明確設定後端名稱。



使用「tridentctl」建立的後端沒有關聯的「TridentBackendConfig」物件。您可以建立「TridentBackendConfig」（TridentBackendConfig）的CR、選擇以「kubectll」管理這類後端。必須謹慎指定相同的組態參數（例如「s.pec.backendName」、「sec.storagePrefix」、「sPEec.storageDriverName」等）。Astra Trident會自動將新建立的「TridentBackendConfig」連結至預先存在的後端。

步驟總覽

若要使用「kubectll」建立新的後端、您應該執行下列動作：

1. 建立 "[Kubernetes機密](#)"。此機密包含Astra Trident與儲存叢集/服務通訊所需的認證資料。
2. 建立「TridentBackendConfig」物件。其中包含有關儲存叢集/服務的詳細資訊、並參考上一步建立的機密。

建立後端之後、您可以使用「`kubectl Get tbc <tbc-name>-n <trident命名空間>`」來觀察其狀態、並收集其他詳細資料。

步驟1：建立Kubernetes機密

建立包含後端存取認證的秘密。這是每個儲存服務/平台所獨有的功能。以下是範例：

```
$ kubectl -n trident create -f backend-tbc-ontap-san-secret.yaml
apiVersion: v1
kind: Secret
metadata:
  name: backend-tbc-ontap-san-secret
type: Opaque
stringData:
  username: cluster-admin
  password: t@Ax@7q(>
```

下表摘要說明每個儲存平台的機密必須包含的欄位：

儲存平台機密欄位說明	秘密	欄位說明
Azure NetApp Files	ClientID	應用程式註冊的用戶端ID
適用於 GCP Cloud Volumes Service	Private金鑰ID	私密金鑰的ID。GCP服務帳戶API金鑰的一部分、具有CVS管理員角色
適用於 GCP Cloud Volumes Service	Private金鑰	私密金鑰：GCP服務帳戶API金鑰的一部分、具有CVS管理員角色
元素 (NetApp HCI / SolidFire)	端點	MVIP、適用於SolidFire 採用租戶認證的不含用戶身分證明的叢集
ONTAP	使用者名稱	連線至叢集/ SVM的使用者名稱。用於認證型驗證
ONTAP	密碼	連線至叢集/ SVM的密碼。用於認證型驗證
ONTAP	用戶端權限金鑰	用戶端私密金鑰的Base64編碼值。用於憑證型驗證
ONTAP	chap使用者名稱	傳入使用者名稱。如果useCHAP=true則需要。適用於「ONTAP-SAN」和「ONTAP-san經濟」

儲存平台機密欄位說明	秘密	欄位說明
ONTAP	chapInitiator機密	CHAP啟動器密碼。如果useCHAP=true則需要。適用於「ONTAP-SAN」和「ONTAP-san經濟」
ONTAP	chapTargetUsername	目標使用者名稱。如果useCHAP=true則需要。適用於「ONTAP-SAN」和「ONTAP-san經濟」
ONTAP	chapTargetInitiator機密	CHAP目標啟動器機密。如果useCHAP=true則需要。適用於「ONTAP-SAN」和「ONTAP-san經濟」

在此步驟中建立的機密會參照下一步所建立之「TridentBackendConfig」物件的「sapec.ecent」欄位。

步驟2：建立 TridentBackendConfig CR

您現在可以建立「TridentBackendConfig」的CR了。在此範例中、使用「ONTAP-SAN」驅動程式的後端是使用「TridentBackendConfig」物件建立、如下所示：

```
$ kubectl -n trident create -f backend-tbc-ontap-san.yaml
```

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-ontap-san
spec:
  version: 1
  backendName: ontap-san-backend
  storageDriverName: ontap-san
  managementLIF: 10.0.0.1
  dataLIF: 10.0.0.2
  svm: trident_svm
  credentials:
    name: backend-tbc-ontap-san-secret
```

步驟3：確認的狀態 TridentBackendConfig CR

現在您已經建立了「TridentBackendConfig」（TridentBackendConfig）CR、您就可以驗證其狀態。請參閱下列範例：

```
$ kubectl -n trident get tbc backend-tbc-ontap-san
NAME                                BACKEND NAME                BACKEND UUID
PHASE    STATUS
backend-tbc-ontap-san    ontap-san-backend          8d24fce7-6f60-4d4a-8ef6-
bab2699e6ab8    Bound    Success
```

已成功建立後端、並連結至「TridentBackendConfig」CR。

階段可以採用下列其中一個值：

- 「綁定」：「TridentBackendConfig」CR與後端相關聯、後端包含「configRef」設定為「TridentBackendConfig」的CR uid。
- 《Unbound》：使用「」表示。「TridentBackendConfig」物件不會繫結至後端。根據預設、所有新建立的「TridentBackendConfig」CRS均處於此階段。階段變更之後、就無法再恢復為Unbound（未綁定）。
- 「刪除」：「TridentBackendConfig」的「刪除原則」已設定為刪除。刪除「TridentBackendConfig」CR時、它會轉換為「刪除」狀態。
 - 如果後端不存在持續磁碟區宣告（PVCS）、刪除「TridentBackendConfig」（TridentBackendConfig）會導致Astra Trident刪除後端、以及刪除「TridentBackendConfig」（TridentBackendConfig）。
 - 如果後端上有一個或多個PVCS、則會進入刪除狀態。隨後、「TridentBackendConfig」CR也會進入刪除階段。只有刪除所有的PVCS之後、才會刪除後端和「TridentBackendConfig」。
- 「遺失」：與「TridentBackendConfig」CR相關的後端意外或刻意刪除、而「TridentBackendConfig」CR仍有刪除後端的參考資料。無論「刪除原則」值為何、「TridentBackendConfig」CR仍可刪除。
- 「未知」：Astra Trident無法判斷與「TridentBackendConfig」CR相關的後端狀態或存在。例如、如果API伺服器沒有回應、或是缺少「tridentbackends.trident.netapp.io`CRD」。這可能需要使用者介入。

在此階段、成功建立後端！還有多種作業可以額外處理、例如 ["後端更新和後端刪除"](#)。

（選用）步驟4：取得更多詳細資料

您可以執行下列命令來取得有關後端的詳細資訊：

```
kubectl -n trident get tbc backend-tbc-ontap-san -o wide
```

```
NAME                                BACKEND NAME                BACKEND UUID
PHASE    STATUS    STORAGE DRIVER    DELETION POLICY
backend-tbc-ontap-san    ontap-san-backend          8d24fce7-6f60-4d4a-8ef6-
bab2699e6ab8    Bound    Success    ontap-san          delete
```

此外、您也可以取得「TridentBackendConfig」的YAML/Json傾印。

```
$ kubectl -n trident get tbc backend-tbc-ontap-san -o yaml
```

```

apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  creationTimestamp: "2021-04-21T20:45:11Z"
  finalizers:
  - trident.netapp.io
  generation: 1
  name: backend-tbc-ontap-san
  namespace: trident
  resourceVersion: "947143"
  uid: 35b9d777-109f-43d5-8077-c74a4559d09c
spec:
  backendName: ontap-san-backend
  credentials:
    name: backend-tbc-ontap-san-secret
  managementLIF: 10.0.0.1
  dataLIF: 10.0.0.2
  storageDriverName: ontap-san
  svm: trident_svm
  version: 1
status:
  backendInfo:
    backendName: ontap-san-backend
    backendUUID: 8d24fce7-6f60-4d4a-8ef6-bab2699e6ab8
  deletionPolicy: delete
  lastOperationStatus: Success
  message: Backend 'ontap-san-backend' created
  phase: Bound

```

《backendInfo》 (backendInfo) 包含了針對「TridentBackendConfig」 (TridentBackendConfig) 的CR而建立的後端「backendName」 (背端名稱) 和「backendUUID」 (背端UUID)。 「lastoperationStatus」欄位代表上次執行的「TridentBackendConfig」 (TridentBackendConfig) CR狀態、可由使用者觸發 (例如、使用者在「show」中變更內容)、或由Astra Trident觸發 (例如、Astra Trident重新啟動期間)。可能是「成功」或「失敗」。「階段」代表「TridentBackendConfig」與後端之間的關係狀態。在上述範例中、「階段」具有界限值、這表示「TridentBackendConfig」CR與後端相關聯。

您可以執行「`kubectl -n triident描述tbc <tbc-cr-name>`」命令、以取得事件記錄的詳細資料。



您無法使用「tridentctl」來更新或刪除包含相關「TridentBackendConfig」物件的後端。若要瞭解在「tridentctl」和「TridentBackendConfig」之間切換的步驟、["請參閱此處"](#)。

以KECBE CVL執行後端管理

瞭解如何使用「kubectl」來執行後端管理作業。

刪除後端

刪除「TridentBackendConfig」（TridentBackendConfig）之後、即指示Astra Trident刪除/保留後端（根據「刪除原則」）。若要刪除後端、請確定「刪除原則」已設定為刪除。如果只要刪除「TridentBackendConfig」、請確定「刪除原則」已設定為保留。這可確保後端仍存在、並可使用「tridentctl」進行管理。

執行下列命令：

```
$ kubectl delete tbc <tbc-name> -n trident
```

Astra Trident並不會刪除「TridentBackendConfig」所使用的Kubernetes Secrets。Kubernetes使用者負責清除機密。刪除機密時必須小心。只有在後端未使用機密時、才應刪除這些機密。

檢視現有的後端

執行下列命令：

```
$ kubectl get tbc -n trident
```

您也可以執行「tridentctl Get backend -n trident」或「tridentctl Get backend -o yaml -n trident」、以取得所有後端的清單。這份清單也會包含以「tridentctl」建立的後端。

更新後端

更新後端可能有多種原因：

- 儲存系統的認證資料已變更。若要更新認證資料、必須更新「TridentBackendConfig」物件中使用的Kubernetes Secret。Astra Trident會自動以提供的最新認證資料更新後端。執行下列命令以更新Kubernetes Secret：

```
$ kubectl apply -f <updated-secret-file.yaml> -n trident
```

- 需要ONTAP 更新參數（例如使用的SVM名稱）。在這種情況下、「TridentBackendConfig」物件可直接透過Kubernetes更新。

```
$ kubectl apply -f <updated-backend-file.yaml>
```

或者、您也可以執行下列命令來變更現有的「TridentBackendConfig」CR：

```
$ kubectl edit tbc <tbc-name> -n trident
```

如果後端更新失敗、後端仍會繼續維持其最後已知的組態。您可以執行「kubectl Get tbc <tbc-name>-o yaml -n trident」或「kubectl描述tbc <tbc-name>-n trident」來檢視記錄以判斷原因。

識別並修正組態檔的問題之後、即可重新執行update命令。

使用tridentctl執行後端管理

瞭解如何使用「tridentctl」來執行後端管理作業。

建立後端

建立之後 "後端組態檔"，執行下列命令：

```
$ tridentctl create backend -f <backend-file> -n trident
```

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 -n trident
```

識別並修正組態檔的問題之後、您只需再次執行「create」命令即可。

刪除後端

若要從Astra Trident刪除後端、請執行下列步驟：

1. 擷取後端名稱：

```
$ tridentctl get backend -n trident
```

2. 刪除後端：

```
$ tridentctl delete backend <backend-name> -n trident
```



如果Astra Trident已從這個後端配置磁碟區和快照、但該後端仍存在、則刪除後端會使新的磁碟區無法由其進行資源配置。後端將繼續處於「刪除」狀態、而Trident將繼續管理這些磁碟區和快照、直到它們被刪除為止。

檢視現有的後端

若要檢視Trident知道的後端、請執行下列步驟：

- 若要取得摘要、請執行下列命令：

```
$ tridentctl get backend -n trident
```

- 若要取得所有詳細資料、請執行下列命令：

```
$ tridentctl get backend -o json -n trident
```

更新後端

建立新的後端組態檔之後、請執行下列命令：

```
$ tridentctl update backend <backend-name> -f <backend-file> -n trident
```

如果後端更新失敗、表示後端組態有問題、或是您嘗試了無效的更新。您可以執行下列命令來檢視記錄、以判斷原因：

```
$ tridentctl logs -n trident
```

識別並修正組態檔的問題之後、您只需再次執行「update」命令即可。

識別使用後端的儲存類別

這是您可以用Json回答的問題類型範例、其中的「tridentctl」會輸出後端物件。這會使用您需要安裝的「jq」公用程式。

```
$ tridentctl get backend -o json | jq '[.items[] | {backend: .name, storageClasses: [.storage[].storageClasses]|unique}]'
```

這也適用於使用「TridentBackendConfig」建立的後端。

在後端管理選項之間切換

瞭解Astra Trident管理後端的不同方法。隨著「TridentBackendConfig」的推出、系統管理員現在有兩種獨特的後端管理方法。這會提出下列問題：

- 使用「tridentctl」建立的後端、是否能以「TridentBackendConfig」來管理？
- 使用「TridentBackendConfig」建立的後端、是否可以使用「tridentctl」來管理？

管理 tridentctl 後端使用 TridentBackendConfig

本節說明透過Kubernetes介面建立「TridentBackendConfig」物件、直接透過「tridentctl」建立的後端管理所需的步驟。

這將適用於下列案例：

- 由於使用「tridentctl」建立的後端沒有「TridentBackendConfig」、因此它們是預先存在的後端。

- 使用「tridentctl」建立的新後端、而其他「TridentBackendConfig」物件則存在。

在這兩種情況下、後端仍會繼續存在、Astra Trident排程磁碟區會繼續運作。系統管理員有兩種選擇之一：

- 繼續使用「tridentctl」來管理使用它建立的後端。
- 將使用「tridentctl」建立的後端連結至新的「TridentBackendConfig」物件。這樣做將意味着後端將使用“kubedl”而不是“tridentctl”來管理。

若要使用「kubectl」管理預先存在的後端、您需要建立連結至現有後端的「TridentBackendConfig」。以下是如何運作的總覽：

1. 建立Kubernetes機密。此機密包含Astra Trident與儲存叢集/服務通訊所需的認證資料。
2. 建立「TridentBackendConfig」物件。其中包含有關儲存叢集/服務的詳細資訊、並參考上一步建立的機密。必須謹慎指定相同的組態參數（例如「s.pec.backendName」、「sec.storagePrefix」、「sPEec.storageDriverName」等）。必須將「Pec.backendName」設定為現有後端的名稱。

步驟0：識別後端

若要建立連結至現有後端的「TridentBackendConfig」、您必須取得後端的組態。在此範例中、假設使用下列Json定義建立後端：

```
$ tridentctl get backend ontap-nas-backend -n trident
+-----+-----+-----+
+-----+-----+-----+
|          NAME          | STORAGE DRIVER |          UUID          |
| STATE  | VOLUMES |          |
+-----+-----+-----+
+-----+-----+-----+
| ontap-nas-backend    | ontap-nas      | 52f2eb10-e4c6-4160-99fc-
96b3be5ab5d7 | online |          25 |
+-----+-----+-----+
+-----+-----+-----+
```

```
$ cat ontap-nas-backend.json

{
  "version": 1,
  "storageDriverName": "ontap-nas",
  "managementLIF": "10.10.10.1",
  "dataLIF": "10.10.10.2",
  "backendName": "ontap-nas-backend",
  "svm": "trident_svm",
  "username": "cluster-admin",
  "password": "admin-password",

  "defaults": {
    "spaceReserve": "none",
```

```

    "encryption": "false"
  },
  "labels":{"store":"nas_store"},
  "region": "us_east_1",
  "storage": [
    {
      "labels":{"app":"msoffice", "cost":"100"},
      "zone":"us_east_1a",
      "defaults": {
        "spaceReserve": "volume",
        "encryption": "true",
        "unixPermissions": "0755"
      }
    },
    {
      "labels":{"app":"mysqldb", "cost":"25"},
      "zone":"us_east_1d",
      "defaults": {
        "spaceReserve": "volume",
        "encryption": "false",
        "unixPermissions": "0775"
      }
    }
  ]
}

```

步驟1：建立Kubernetes機密

建立包含後端認證的秘密、如以下範例所示：

```

$ cat tbc-ontap-nas-backend-secret.yaml

apiVersion: v1
kind: Secret
metadata:
  name: ontap-nas-backend-secret
type: Opaque
stringData:
  username: cluster-admin
  passWord: admin-password

$ kubectl create -f tbc-ontap-nas-backend-secret.yaml -n trident
secret/backend-tbc-ontap-san-secret created

```

步驟2：建立 TridentBackendConfig CR

下一步是建立一個「TridentBackendConfig」（TridentBackendConfig）CR、它會自動連結至現有的「ONTAP-NAS-backend」（如本範例所示）。確保符合下列要求：

- 相同的後端名稱是在「s.pec.backendName」中定義。
- 組態參數與原始後端相同。
- 虛擬儲存資源池（若有）必須維持與原始後端相同的順序。
- 認證資料是透過Kubernetes Secret提供、而非以純文字提供。

在這種情況下、「TridentBackendConfig」將會如下所示：

```

$ cat backend-tbc-ontap-nas.yaml
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: tbc-ontap-nas-backend
spec:
  version: 1
  storageDriverName: ontap-nas
  managementLIF: 10.10.10.1
  dataLIF: 10.10.10.2
  backendName: ontap-nas-backend
  svm: trident_svm
  credentials:
    name: mysecret
  defaults:
    spaceReserve: none
    encryption: 'false'
  labels:
    store: nas_store
  region: us_east_1
  storage:
  - labels:
    app: msoffice
    cost: '100'
    zone: us_east_1a
    defaults:
      spaceReserve: volume
      encryption: 'true'
      unixPermissions: '0755'
  - labels:
    app: mysqldb
    cost: '25'
    zone: us_east_1d
    defaults:
      spaceReserve: volume
      encryption: 'false'
      unixPermissions: '0775'

$ kubectl create -f backend-tbc-ontap-nas.yaml -n trident
tridentbackendconfig.trident.netapp.io/tbc-ontap-nas-backend created

```

步驟3：確認的狀態 TridentBackendConfig **CR**

在建立「TridentBackendConfig」之後、其階段必須是「綁定」。它也應反映與現有後端相同的後端名稱和UUID。

```

$ kubectl -n trident get tbc tbc-ontap-nas-backend -n trident
NAME                                BACKEND NAME                BACKEND UUID
PHASE    STATUS
tbc-ontap-nas-backend  ontap-nas-backend          52f2eb10-e4c6-4160-99fc-
96b3be5ab5d7    Bound    Success

#confirm that no new backends were created (i.e., TridentBackendConfig did
not end up creating a new backend)
$ tridentctl get backend -n trident
+-----+-----+
+-----+-----+-----+-----+
|          NAME          | STORAGE DRIVER |          UUID          |
| STATE  | VOLUMES |          |
+-----+-----+-----+-----+
| ontap-nas-backend     | ontap-nas      | 52f2eb10-e4c6-4160-99fc-
96b3be5ab5d7 | online |          25 |
+-----+-----+-----+-----+
+-----+-----+-----+-----+

```

現在可以使用「tbc-ontap-nas-backend」 「TridentBackendConfig」物件來完全管理後端。

管理 TridentBackendConfig 後端使用 tridentctl

可以使用「tridentctl」來列出使用「TridentBackendConfig」建立的後端。此外、系統管理員也可以刪除「TridentBackendConfig」、並確定「pec.deletionPolicy」設為「效能」、藉此選擇透過「tridentctl」來完全管理此類後端。

步驟0：識別後端

例如、假設使用「TridentBackendConfig」建立下列後端：

```

$ kubectl get tbc backend-tbc-ontap-san -n trident -o wide
NAME                                BACKEND NAME          BACKEND UUID
PHASE  STATUS  STORAGE DRIVER  DELETION POLICY
backend-tbc-ontap-san  ontap-san-backend  81abcb27-ea63-49bb-b606-
0a5315ac5f82  Bound  Success  ontap-san  delete

$ tridentctl get backend ontap-san-backend -n trident
+-----+-----+
+-----+-----+-----+-----+
|          NAME          | STORAGE DRIVER |                               UUID
| STATE  | VOLUMES |
+-----+-----+-----+-----+
| ontap-san-backend | ontap-san      | 81abcb27-ea63-49bb-b606-
0a5315ac5f82 | online |          33 |
+-----+-----+-----+-----+
+-----+-----+-----+-----+

```

從輸出中可以看到、「TridentBackendConfig」已成功建立、並繫結至後端（觀察後端的UUID）。

步驟1：確認 `deletionPolicy` 設為 `retain`

讓我們來看看「改革政策」的價值。這需要設定為「維護」。這將確保刪除「TridentBackendConfig」（TridentBackendConfig）的CR時、後端定義仍會存在、而且可以使用「tridentctl」進行管理。

```

$ kubectl get tbc backend-tbc-ontap-san -n trident -o wide
NAME                                BACKEND NAME          BACKEND UUID
PHASE  STATUS  STORAGE DRIVER  DELETION POLICY
backend-tbc-ontap-san  ontap-san-backend  81abcb27-ea63-49bb-b606-
0a5315ac5f82  Bound  Success  ontap-san  delete

# Patch value of deletionPolicy to retain
$ kubectl patch tbc backend-tbc-ontap-san --type=merge -p
'{"spec":{"deletionPolicy":"retain"}}' -n trident
tridentbackendconfig.trident.netapp.io/backend-tbc-ontap-san patched

#Confirm the value of deletionPolicy
$ kubectl get tbc backend-tbc-ontap-san -n trident -o wide
NAME                                BACKEND NAME          BACKEND UUID
PHASE  STATUS  STORAGE DRIVER  DELETION POLICY
backend-tbc-ontap-san  ontap-san-backend  81abcb27-ea63-49bb-b606-
0a5315ac5f82  Bound  Success  ontap-san  retain

```



除非將「刪除原則」設定為「需要」、否則請勿繼續下一步。

步驟2：刪除 TridentBackendConfig CR

最後一個步驟是刪除「TridentBackendConfig」（TridentBackendConfig）。確認「刪除原則」設為「保留」之後、您可以繼續刪除：

```
$ kubectl delete tbc backend-tbc-ontap-san -n trident
tridentbackendconfig.trident.netapp.io "backend-tbc-ontap-san" deleted

$ tridentctl get backend ontap-san-backend -n trident
+-----+-----+
+-----+-----+-----+-----+
|          NAME          | STORAGE DRIVER |          UUID          |
| STATE  | VOLUMES |
+-----+-----+-----+-----+
+-----+-----+-----+-----+
| ontap-san-backend | ontap-san      | 81abcb27-ea63-49bb-b606-
0a5315ac5f82 | online |          33 |
+-----+-----+-----+-----+
+-----+-----+-----+-----+
```

刪除「TridentBackendConfig」物件之後、Astra Trident便會移除該物件、而不會實際刪除後端本身。

管理儲存類別

尋找建立儲存類別、刪除儲存類別及檢視現有儲存類別的相關資訊。

設計儲存類別

請參閱 ["請按這裡"](#) 以取得有關儲存類別及其設定方式的詳細資訊。

建立儲存類別

取得儲存類別檔案之後、請執行下列命令：

```
kubectl create -f <storage-class-file>
```

「<storage-class file」（儲存設備類別檔案）應以您的儲存設備類別檔案名稱取代。

刪除儲存類別

若要從Kubernetes刪除儲存類別、請執行下列命令：

```
kubectl delete storageclass <storage-class>
```

「<storage-class>」應改用您的儲存類別。

透過此儲存類別所建立的任何持續磁碟區都將維持不變、Astra Trident將繼續管理這些磁碟區。



Astra Trident會對其建立的磁碟區執行空白的「FSType」。對於iSCSI後端、建議在StorageClass中強制執行「參數.FSType」。您應該刪除「existing StorageClass」、然後使用指定的「參數.FSType」重新建立。

檢視現有的儲存類別

- 若要檢視現有的Kubernetes儲存類別、請執行下列命令：

```
kubectl get storageclass
```

- 若要檢視Kubernetes儲存類別詳細資料、請執行下列命令：

```
kubectl get storageclass <storage-class> -o json
```

- 若要檢視Astra Trident的同步儲存類別、請執行下列命令：

```
tridentctl get storageclass
```

- 若要檢視Astra Trident的同步儲存類別詳細資料、請執行下列命令：

```
tridentctl get storageclass <storage-class> -o json
```

設定預設儲存類別

Kubernetes 1.6新增了設定預設儲存類別的功能。如果使用者未在「持續磁碟區宣告」（PVC）中指定一個、則此儲存類別將用於配置「持續磁碟區」。

- 在儲存類別定義中、將「storageclass.Kubernetes.IO/as-default-Class」註釋設為true、以定義預設儲存類別。根據規格、任何其他值或不存在附註都會解譯為假。
- 您可以使用下列命令、將現有的儲存類別設定為預設的儲存類別：

```
kubectl patch storageclass <storage-class-name> -p '{"metadata": {"annotations":{"storageclass.kubernetes.io/is-default-class":"true"}}}'
```

- 同樣地、您也可以使用下列命令移除預設儲存類別註釋：

```
kubectl patch storageclass <storage-class-name> -p '{"metadata": {"annotations":{"storageclass.kubernetes.io/is-default-class":"false"}}}'
```

Trident安裝程式套件中也有包含此附註的範例。



在任何指定時間、您的叢集中都只應有一個預設儲存類別。Kubernetes在技術上並不妨礙您擁有多個儲存類別、但它的行為方式就如同完全沒有預設的儲存類別一樣。

識別儲存類別的後端

這是您可以用Json回答的問題類型範例、其中的「tridentctl」輸出用於Astra Trident後端物件。這會使用「jq」公用程式、您可能需要先安裝。

```
tridentctl get storageclass -o json | jq '[.items[] | {storageClass: .Config.name, backends: [.storage]|unique}]'
```

執行Volume作業

深入瞭解Astra Trident提供的功能、協助您管理磁碟區。

- "使用「csi拓撲」"
- "使用快照"
- "展開Volume"
- "匯入磁碟區"

使用「csi拓撲」

Astra Trident可以利用、選擇性地建立磁碟區、並將磁碟區附加至Kubernetes叢集中的節點 "「csi拓撲」功能"。使用「csi拓撲」功能、可根據區域和可用性區域、限制對磁碟區的存取、只能存取一部分節點。如今、雲端供應商可讓Kubernetes管理員建立以區域為基礎的節點。節點可位於某個區域內的不同可用度區域、或位於不同區域之間。為了協助在多區域架構中配置工作負載的磁碟區、Astra Trident使用了csi拓撲。



深入瞭解「csi拓撲」功能 ["請按這裡"](#)。

Kubernetes提供兩種獨特的Volume繫結模式：

- Astra Trident將「Volume BindingMode」設為「Immediate（立即）」、可在沒有任何拓撲感知的情況下建立磁碟區。建立永久虛擬磁碟時、即會處理磁碟區繫結和動態資源配置。這是預設的「Volume BindingMode」、適用於未強制拓撲限制的叢集。建立永續性磁碟區時、不會對要求的Pod排程需求有任何相依性。
- 將「Volume BindingMode」設為「WaitForFirst消費者」時、會延遲建立和繫結永久磁碟區、直到排程並建立使用永久磁碟的Pod為止。如此一來、就能建立磁碟區、以符合拓撲需求所強制執行的排程限制。



「等待使用者」繫結模式不需要拓撲標籤。這可獨立於「csi拓撲」功能使用。

您需要的產品

若需要使用「csi拓撲」、您需要下列項目：

- 執行1.17或更新版本的Kubernetes叢集。

```
$ kubectl version
Client Version: version.Info{Major:"1", Minor:"19",
GitVersion:"v1.19.3",
GitCommit:"1e11e4a2108024935ecfcb2912226cedeadfd99df",
GitTreeState:"clean", BuildDate:"2020-10-14T12:50:19Z",
GoVersion:"go1.15.2", Compiler:"gc", Platform:"linux/amd64"}
Server Version: version.Info{Major:"1", Minor:"19",
GitVersion:"v1.19.3",
GitCommit:"1e11e4a2108024935ecfcb2912226cedeadfd99df",
GitTreeState:"clean", BuildDate:"2020-10-14T12:41:49Z",
GoVersion:"go1.15.2", Compiler:"gc", Platform:"linux/amd64"}
```

- 叢集中的節點應該有標籤來介紹拓撲認知（「topology.kubernetes.io/region」和「topology.kubernetes.io/Zone」）。在安裝Astra Trident以識別拓撲之前、這些標籤*應該會出現在叢集*的節點上。

```
$ kubectl get nodes -o=jsonpath='{range .items[*]}[.metadata.name],
[.metadata.labels]]{"\n"}{end}' | grep --color "topology.kubernetes.io"
[nod1,
{"beta.kubernetes.io/arch":"amd64","beta.kubernetes.io/os":"linux","kuber-
netes.io/arch":"amd64","kubernetes.io/hostname":"nod1","kubernetes.io/
os":"linux","node-
role.kubernetes.io/master":"","topology.kubernetes.io/region":"us-
east1","topology.kubernetes.io/zone":"us-east1-a"}]
[node2,
{"beta.kubernetes.io/arch":"amd64","beta.kubernetes.io/os":"linux","kuber-
netes.io/arch":"amd64","kubernetes.io/hostname":"node2","kubernetes.io/
os":"linux","node-
role.kubernetes.io/worker":"","topology.kubernetes.io/region":"us-
east1","topology.kubernetes.io/zone":"us-east1-b"}]
[node3,
{"beta.kubernetes.io/arch":"amd64","beta.kubernetes.io/os":"linux","kuber-
netes.io/arch":"amd64","kubernetes.io/hostname":"node3","kubernetes.io/
os":"linux","node-
role.kubernetes.io/worker":"","topology.kubernetes.io/region":"us-
east1","topology.kubernetes.io/zone":"us-east1-c"}]
```

步驟1：建立可感知拓撲的後端

Astra Trident儲存後端可根據可用性區域、選擇性地配置磁碟區。每個後端都可以攜帶一個可選的「支援拓撲」

區塊、代表必須支援的區域和區域清單。對於使用此類後端的StorageClass、只有在受支援地區/區域中排程的應用程式要求時、才會建立Volume。

以下是後端定義的範例：

```
{
  "version": 1,
  "storageDriverName": "ontap-san",
  "backendName": "san-backend-us-east1",
  "managementLIF": "192.168.27.5",
  "svm": "iscsi_svm",
  "username": "admin",
  "password": "xxxxxxxxxxxx",
  "supportedTopologies": [
    {"topology.kubernetes.io/region": "us-east1",
     "topology.kubernetes.io/zone": "us-east1-a"},
    {"topology.kubernetes.io/region": "us-east1",
     "topology.kubernetes.io/zone": "us-east1-b"}
  ]
}
```



「支援拓撲」是用來提供每個後端的區域和區域清單。這些區域和區域代表StorageClass中可提供的允許值清單。對於包含後端所提供之區域和區域子集的StorageClass、Astra Trident會在後端建立磁碟區。

您也可以定義每個儲存資源池的「支援拓撲」。請參閱下列範例：

```

{"version": 1,
"storageDriverName": "ontap-nas",
"backendName": "nas-backend-us-central1",
"managementLIF": "172.16.238.5",
"svm": "nfs_svm",
"username": "admin",
"password": "Netapp123",
"supportedTopologies": [
  {"topology.kubernetes.io/region": "us-central1",
"topology.kubernetes.io/zone": "us-central1-a"},
  {"topology.kubernetes.io/region": "us-central1",
"topology.kubernetes.io/zone": "us-central1-b"}
]
"storage": [
  {
    "labels": {"workload":"production"},
    "region": "Iowa-DC",
    "zone": "Iowa-DC-A",
    "supportedTopologies": [
      {"topology.kubernetes.io/region": "us-central1",
"topology.kubernetes.io/zone": "us-central1-a"}
    ]
  },
  {
    "labels": {"workload":"dev"},
    "region": "Iowa-DC",
    "zone": "Iowa-DC-B",
    "supportedTopologies": [
      {"topology.kubernetes.io/region": "us-central1",
"topology.kubernetes.io/zone": "us-central1-b"}
    ]
  }
]
}

```

在此範例中、「REGion」和「Zone」標籤代表儲存資源池的位置。「topology、Kubernetes.io/region」和「topology、Kubernetes.io/Zone」決定儲存資源池的使用來源。

步驟2：定義可感知拓撲的StorageClass

根據提供給叢集中節點的拓撲標籤、可以定義StorageClass以包含拓撲資訊。這將決定做為所提出之永久虛擬磁碟要求候選的儲存資源池、以及可以使用Trident所提供之磁碟區的節點子集。

請參閱下列範例：

```

apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: netapp-san-us-east1
provisioner: csi.trident.netapp.io
volumeBindingMode: WaitForFirstConsumer
allowedTopologies:
- matchLabelExpressions:
- key: topology.kubernetes.io/zone
  values:
  - us-east1-a
  - us-east1-b
- key: topology.kubernetes.io/region
  values:
  - us-east1
parameters:
  fsType: "ext4"

```

在上述StorageClass定義中、「volumeBindingMode」設定為「WaitForFirst消費者」。在Pod中引用此StorageClass所要求的PVCS之前、系統不會對其採取行動。此外、「allowedTopologies」也提供要使用的區域和區域。《NetApp-san—use-east1》StorageClass將在上述定義的「SAN後端-us-east1」後端上建立PVCS。

步驟3：建立並使用PVC

建立StorageClass並對應至後端後端後端之後、您現在就可以建立PVCS。

請參閱以下「SPEC」範例：

```

---
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: pvc-san
spec:
  accessModes:
  - ReadWriteOnce
  resources:
    requests:
      storage: 300Mi
  storageClassName: netapp-san-us-east1

```

使用此資訊清單建立永久虛擬環境可能會產生下列結果：

```

$ kubectl create -f pvc.yaml
persistentvolumeclaim/pvc-san created
$ kubectl get pvc
NAME          STATUS      VOLUME      CAPACITY   ACCESS MODES   STORAGECLASS
AGE
pvc-san      Pending
2s
$ kubectl describe pvc
Name:          pvc-san
Namespace:    default
StorageClass: netapp-san-us-east1
Status:       Pending
Volume:
Labels:       <none>
Annotations:  <none>
Finalizers:   [kubernetes.io/pvc-protection]
Capacity:
Access Modes:
VolumeMode:   Filesystem
Mounted By:   <none>
Events:
  Type    Reason              Age   From
  ----    -
  Normal  WaitForFirstConsumer 6s    persistentvolume-controller
waiting for first consumer to be created before binding

```

若要Trident建立磁碟區並將其連結至PVC、請在Pod中使用PVC。請參閱下列範例：


```

apiVersion: v1
kind: Pod
metadata:
  name: app-pod-1
spec:
  affinity:
    nodeAffinity:
      requiredDuringSchedulingIgnoredDuringExecution:
        nodeSelectorTerms:
          - matchExpressions:
              - key: topology.kubernetes.io/region
                operator: In
                values:
                  - us-east1
            preferredDuringSchedulingIgnoredDuringExecution:
              - weight: 1
                preference:
                  matchExpressions:
                    - key: topology.kubernetes.io/zone
                      operator: In
                      values:
                        - us-east1-a
                        - us-east1-b
      securityContext:
        runAsUser: 1000
        runAsGroup: 3000
        fsGroup: 2000
    volumes:
      - name: voll
        persistentVolumeClaim:
          claimName: pvc-san
    containers:
      - name: sec-ctx-demo
        image: busybox
        command: [ "sh", "-c", "sleep 1h" ]
        volumeMounts:
          - name: voll
            mountPath: /data/demo
        securityContext:
          allowPrivilegeEscalation: false

```

此pod化 規範會指示Kubernetes在「us-east1」區域的節點上排程pod、並從「us-east1-a」或「us-east1-b」區域中的任何節點中進行選擇。

請參閱下列輸出：

```

$ kubectl get pods -o wide
NAME          READY   STATUS    RESTARTS   AGE   IP              NODE
NOMINATED NODE READINESS GATES
app-pod-1    1/1     Running   0          19s   192.168.25.131  node2
<none>      <none>
$ kubectl get pvc -o wide
NAME          STATUS   VOLUME                                     CAPACITY
ACCESS MODES  STORAGECLASS          AGE   VOLUMEMODE
pvc-san      Bound   pvc-ecb1e1a0-840c-463b-8b65-b3d033e2e62b  300Mi
RWO          netapp-san-us-east1  48s   Filesystem

```

更新後端以納入 supportedTopologies

您可以使用「tridentctl後端更新」來更新現有的後端、以納入「最上層拓撲」清單。這不會影響已配置的磁碟區、而且只會用於後續的PVCS。

如需詳細資訊、請參閱

- ["管理容器的資源"](#)
- ["節點選取器"](#)
- ["關聯性與反關聯性"](#)
- ["污染與容許"](#)

使用快照

從2001版Astra Trident開始、您可以在Kubernetes層建立PV快照。您可以使用這些快照來維護由Astra Trident所建立之磁碟區的時間點複本、並排程建立其他磁碟區（複本）。Volume Snapshot可支援「ONTAP-NAS」、「ONTAP-SAN」、「ONTAP-SAN經濟」、「Poolidfire - SAN」、「GCP - CVS」、以及「azure-NetApp-file-fil」驅動程式。



此功能可從Kubernetes 1.17（試用版）取得、GA版本為1.20。若要瞭解從試用版移轉至GA所涉及的變更、請參閱 ["版本部落格"](#)。隨着GA的問世、推出了「v1」API版本、並向下相容於「v1Beta1」快照。

您需要的產品

- 建立Volume快照需要建立外部快照控制器和自訂資源定義（CRD）。這是正在使用的Kubernetes Orchestrator的責任（例如：Kubeadm、GKE、OpenShift）。

如果您的Kubernetes發佈版本未包含快照控制器和客戶需求日、您可以依照下列方式進行部署。

1. 建立Volume Snapshot客戶需求日。

對於Kubernetes 1.20及更新版本、請使用v1 Snapshot CRD搭配v5.0或更新版本的Snapshot元件。對於Kubernetes 1.18和1.19版、請使用v1beta1搭配v3.0.3快照元件。

V5.0元件

```
$ cat snapshot-setup.sh
#!/bin/bash
# Create volume snapshot CRDs
kubectl apply -f https://raw.githubusercontent.com/kubernetes-
csi/external-snapshotter/release-
5.0/client/config/crd/snapshot.storage.k8s.io_volumesnapshotclasses.
yaml
kubectl apply -f https://raw.githubusercontent.com/kubernetes-
csi/external-snapshotter/release-
5.0/client/config/crd/snapshot.storage.k8s.io_volumesnapshotcontents
.yaml
kubectl apply -f https://raw.githubusercontent.com/kubernetes-
csi/external-snapshotter/release-
5.0/client/config/crd/snapshot.storage.k8s.io_volumesnapshots.yaml
```

v3.0.3元件

```
$ cat snapshot-setup.sh
#!/bin/bash
# Create volume snapshot CRDs
kubectl apply -f https://raw.githubusercontent.com/kubernetes-
csi/external-
snapshotter/v3.0.3/client/config/crd/snapshot.storage.k8s.io_volumes
napshotclasses.yaml
kubectl apply -f https://raw.githubusercontent.com/kubernetes-
csi/external-
snapshotter/v3.0.3/client/config/crd/snapshot.storage.k8s.io_volumes
napshotcontents.yaml
kubectl apply -f https://raw.githubusercontent.com/kubernetes-
csi/external-
snapshotter/v3.0.3/client/config/crd/snapshot.storage.k8s.io_volumes
napshots.yaml
```

2. 在所需的命名空間中建立Snapshot控制器。編輯下方的Yami清單以修改命名空間。

對於Kubernetes 1.20及更新版本、請使用5.0版或更新版本。對於Kubernetes 1.18和1.19版、請使用v3.0.3



如果在GKE環境中設定隨需磁碟區快照、請勿建立快照控制器。GKE使用內建的隱藏式快照控制器。

V5.0控制器

```
kubectl apply -f https://raw.githubusercontent.com/kubernetes-csi/external-snapshotter/release-5.0/deploy/kubernetes/snapshot-controller/rbac-snapshot-controller.yaml
kubectl apply -f https://raw.githubusercontent.com/kubernetes-csi/external-snapshotter/release-5.0/deploy/kubernetes/snapshot-controller/setup-snapshot-controller.yaml
```

v3.0.3控制器

```
kubectl apply -f https://raw.githubusercontent.com/kubernetes-csi/external-snapshotter/v3.0.3/deploy/kubernetes/snapshot-controller/rbac-snapshot-controller.yaml
kubectl apply -f https://raw.githubusercontent.com/kubernetes-csi/external-snapshotter/v3.0.3/deploy/kubernetes/snapshot-controller/setup-snapshot-controller.yaml
```



「csi Snapshotter」提供 ["正在驗證Webhook"](#) 協助使用者驗證現有的v1Beta1快照、並確認它們是有效的資源物件。驗證Webhook會自動標示無效的快照物件、並防止未來建立無效物件。驗證Webhook是由Kubernetes Orchestrator部署。請參閱指示以手動部署驗證Webhook ["請按這裡"](#)。尋找無效快照資訊清單的範例 ["請按這裡"](#)。

以下範例說明使用快照所需的架構、並說明如何建立及使用快照。

步驟1：設定 VolumeSnapshotClass

在建立Volume Snapshot之前、請先設定 ["d7ca7162c394dee752c35d07a92823da"](#)。

```
$ cat snap-sc.yaml
#Use apiVersion v1 for Kubernetes 1.20 and above. For Kubernetes 1.18 and
1.19, use apiVersion v1beta1.
apiVersion: snapshot.storage.k8s.io/v1
kind: VolumeSnapshotClass
metadata:
  name: csi-snapclass
driver: csi.trident.netapp.io
deletionPolicy: Delete
```

《driver》指的是Astra Trident的csi驅動程式。「刪除原則」可以是「刪除」或「保留」。當設定為「保留」時、即使刪除「Volume Snapshot」物件、也會保留儲存叢集上的基礎實體快照。

步驟2：建立現有PVC的快照

```
$ cat snap.yaml
#Use apiVersion v1 for Kubernetes 1.20 and above. For Kubernetes 1.18 and
1.19, use apiVersion v1beta1.
apiVersion: snapshot.storage.k8s.io/v1
kind: VolumeSnapshot
metadata:
  name: pvc1-snap
spec:
  volumeSnapshotClassName: csi-snapclass
  source:
    persistentVolumeClaimName: pvc1
```

快照是針對名為「pvc1」的永久虛擬資料建立、快照名稱則設為「pvc1-snap」。

```
$ kubectl create -f snap.yaml
volumesnapshot.snapshot.storage.k8s.io/pvc1-snap created

$ kubectl get volumesnapshots
NAME                AGE
pvc1-snap           50s
```

這會建立一個「Volume Snapshot」物件。Volume Snapshot類似於一個PVC、並與代表實際快照的「Volume SnapshotContent」物件相關聯。

您可以透過描述「pvc1-snap」Volume Snapshot的「Volume SnapshotContent」物件來識別該物件。

```

$ kubectl describe volumesnapshots pvcl-snap
Name:          pvcl-snap
Namespace:    default
.
.
.
Spec:
  Snapshot Class Name:  pvcl-snap
  Snapshot Content Name: snapcontent-e8d8a0ca-9826-11e9-9807-525400f3f660
  Source:
    API Group:
    Kind:      PersistentVolumeClaim
    Name:      pvcl
Status:
  Creation Time:  2019-06-26T15:27:29Z
  Ready To Use:  true
  Restore Size:  3Gi
.
.

```

「快照內容名稱」可識別提供此快照的Volume SnapshotContent物件。「Ready to use」（準備使用）參數表示Snapshot可用於建立新的PVC。

步驟3：從Volume Snapshot建立PVCS

請參閱下列範例、瞭解如何使用快照建立永久虛擬資料：

```

$ cat pvc-from-snap.yaml
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: pvc-from-snap
spec:
  accessModes:
    - ReadWriteOnce
  storageClassName: golden
  resources:
    requests:
      storage: 3Gi
  dataSource:
    name: pvcl-snap
    kind: VolumeSnapshot
    apiGroup: snapshot.storage.k8s.io

```

「Data來源」顯示、必須使用名為「pvc1-snap」的Volume Snapshot建立PVC作為資料來源。這會指示Astra Trident從快照建立一個永久虛擬資料。建立好永久虛擬基礎架構之後、就能將它附加到Pod上、就像使用任何其他永久虛擬基礎架構一樣使用。



刪除具有相關快照的持續Volume時、對應的Trident Volume會更新為「刪除狀態」。若要刪除Astra Trident磁碟區、則應移除該磁碟區的快照。

如需詳細資訊、請參閱

- ["Volume快照"](#)
- ["d7ca7162c394dee752c35d07a92823da"](#)

展開Volume

Astra Trident可讓Kubernetes使用者在建立磁碟區之後擴充磁碟區。尋找擴充iSCSI和NFS磁碟區所需組態的相關資訊。

展開iSCSI Volume

您可以使用「SCSI資源配置程式」來擴充iSCSI持續磁碟區（PV）。



iSCSI磁碟區擴充支援「ontap-san」、「ONTAP-san經濟」、「Poolidfire-san」等驅動程式、需要Kubernetes 1.16及更新版本。

總覽

擴充iSCSI PV包括下列步驟：

- 編輯StorageClass定義、將「owvolVolume Expansion」欄位設定為「true」。
- 編輯PVC定義並更新「s.pec.estices.storage」、以反映新的所需大小、此大小必須大於原始大小。
- 必須將PV附加至Pod、才能調整其大小。調整iSCSI PV的大小有兩種情況：
 - 如果PV附加至Pod、Astra Trident會在儲存後端擴充磁碟區、重新掃描裝置、並重新調整檔案系統的大小。
 - 嘗試調整未附加PV的大小時、Astra Trident會在儲存後端上擴充磁碟區。在將永久虛擬磁碟綁定至Pod之後、Trident會重新掃描裝置並重新調整檔案系統的大小。然後、Kubernetes會在擴充作業成功完成後、更新PVC大小。

以下範例顯示擴充iSCSI PV的運作方式。

步驟1：設定StorageClass以支援Volume擴充

```

$ cat storageclass-ontapsan.yaml
---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontap-san
provisioner: csi.trident.netapp.io
parameters:
  backendType: "ontap-san"
allowVolumeExpansion: True

```

對於已存在的StorageClass、請編輯此類以包含「allowVolumeExpansion」參數。

步驟2：使用您建立的StorageClass建立一個永久虛擬儲存設備

```

$ cat pvc-ontapsan.yaml
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: san-pvc
spec:
  accessModes:
  - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: ontap-san

```

Astra Trident會建立持續磁碟區 (PV)、並將其與此持續磁碟區宣告 (PVC) 建立關聯。

```

$ kubectl get pvc
NAME          STATUS    VOLUME                                     CAPACITY
ACCESS MODES  STORAGECLASS  AGE
san-pvc      Bound      pvc-8a814d62-bd58-4253-b0d1-82f2885db671  1Gi
RWX
ontap-san    8s

$ kubectl get pv
NAME          CAPACITY  ACCESS MODES
RECLAIM POLICY  STATUS    CLAIM                                STORAGECLASS  REASON  AGE
pvc-8a814d62-bd58-4253-b0d1-82f2885db671  1Gi      RWO
Delete          Bound      default/san-pvc                      ontap-san    10s

```


步驟3：定義一個連接至PVC的Pod

在此範例中、會建立使用「shan -PVC」的Pod。

```
$ kubectl get pod
NAME          READY   STATUS    RESTARTS   AGE
centos-pod    1/1     Running   0           65s

$ kubectl describe pvc san-pvc
Name:          san-pvc
Namespace:     default
StorageClass:  ontap-san
Status:        Bound
Volume:        pvc-8a814d62-bd58-4253-b0d1-82f2885db671
Labels:        <none>
Annotations:   pv.kubernetes.io/bind-completed: yes
               pv.kubernetes.io/bound-by-controller: yes
               volume.beta.kubernetes.io/storage-provisioner:
               csi.trident.netapp.io
Finalizers:    [kubernetes.io/pvc-protection]
Capacity:      1Gi
Access Modes:  RWO
VolumeMode:    Filesystem
Mounted By:    centos-pod
```

步驟4：展開PV

若要調整從1Gi建立至2Gi的PV大小、請編輯PVC定義、並將「sec.resumes.requests.storage」更新為2Gi。

```
$ kubectl edit pvc san-pvc
# Please edit the object below. Lines beginning with a '#' will be
ignored,
# and an empty file will abort the edit. If an error occurs while saving
this file will be
# reopened with the relevant failures.
#
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  annotations:
    pv.kubernetes.io/bind-completed: "yes"
    pv.kubernetes.io/bound-by-controller: "yes"
    volume.beta.kubernetes.io/storage-provisioner: csi.trident.netapp.io
  creationTimestamp: "2019-10-10T17:32:29Z"
  finalizers:
  - kubernetes.io/pvc-protection
  name: san-pvc
  namespace: default
  resourceVersion: "16609"
  selfLink: /api/v1/namespaces/default/persistentvolumeclaims/san-pvc
  uid: 8a814d62-bd58-4253-b0d1-82f2885db671
spec:
  accessModes:
  - ReadWriteOnce
  resources:
    requests:
      storage: 2Gi
  ...
```

步驟5：驗證擴充

您可以檢查PVC、PV和Astra Trident Volume的大小、以正確驗證擴充作業：

```

$ kubectl get pvc san-pvc
NAME          STATUS    VOLUME                                     CAPACITY
ACCESS MODES  STORAGECLASS  AGE
san-pvc      Bound      pvc-8a814d62-bd58-4253-b0d1-82f2885db671  2Gi
RWO           ontap-san    11m
$ kubectl get pv
NAME          CAPACITY  ACCESS MODES
RECLAIM POLICY  STATUS    CLAIM          STORAGECLASS  REASON  AGE
pvc-8a814d62-bd58-4253-b0d1-82f2885db671  2Gi      RWO
Delete         Bound     default/san-pvc  ontap-san    12m
$ tridentctl get volumes -n trident
+-----+-----+-----+
+-----+-----+-----+-----+
|          NAME          |  SIZE  | STORAGE CLASS |
PROTOCOL |          BACKEND UUID          |  STATE  |  MANAGED  |
+-----+-----+-----+-----+
+-----+-----+-----+-----+
| pvc-8a814d62-bd58-4253-b0d1-82f2885db671 | 2.0 GiB | ontap-san    |
block    | a9b7bfff-0505-4e31-b6c5-59f492e02d33 | online | true    |
+-----+-----+-----+-----+
+-----+-----+-----+-----+

```

展開NFS Volume

Astra Trident支援在「ONTAP-NAS」、「ONTAP-NAS-P節約」、「ONTAP-NAS-flexgroup」、「GCP-CVS」和「azure-NetApp-fils」後端配置NFS PV的Volume擴充。

步驟1：設定StorageClass以支援Volume擴充

若要調整NFS PV的大小、管理員必須先將「allowVolumeExpansion」欄位設定為「true」、以設定儲存類別以允許磁碟區擴充：

```

$ cat storageclass-ontapnas.yaml
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontapnas
provisioner: csi.trident.netapp.io
parameters:
  backendType: ontap-nas
allowVolumeExpansion: true

```

如果您已經建立了沒有此選項的儲存類別、只要使用「kubectl edit storageclass」來編輯現有的儲存類別、即可進行磁碟區擴充。

步驟2：使用您建立的StorageClass建立一個永久虛擬儲存設備

```
$ cat pvc-ontapnas.yaml
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: ontapnas20mb
spec:
  accessModes:
  - ReadWriteOnce
  resources:
    requests:
      storage: 20Mi
  storageClassName: ontapnas
```

Astra Trident應為此PVC建立20MiB NFS PV：

```
$ kubectl get pvc
NAME                STATUS    VOLUME
CAPACITY            ACCESS MODES  STORAGECLASS  AGE
ontapnas20mb       Bound      pvc-08f3d561-b199-11e9-8d9f-5254004dfdb7  20Mi
RWO                 ontapnas      9s

$ kubectl get pv pvc-08f3d561-b199-11e9-8d9f-5254004dfdb7
NAME                CAPACITY  ACCESS MODES
RECLAIM POLICY     STATUS    CLAIM                STORAGECLASS  REASON
AGE
pvc-08f3d561-b199-11e9-8d9f-5254004dfdb7  20Mi     RWO
Delete            Bound     default/ontapnas20mb  ontapnas
2m42s
```

步驟3：展開PV

若要將新建立的20MiB PV調整至1GiB、請編輯PVC並將「spec.resumes.requests.storage」設為1GB：

```
$ kubectl edit pvc ontapnas20mb
# Please edit the object below. Lines beginning with a '#' will be
ignored,
# and an empty file will abort the edit. If an error occurs while saving
this file will be
# reopened with the relevant failures.
#
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  annotations:
    pv.kubernetes.io/bind-completed: "yes"
    pv.kubernetes.io/bound-by-controller: "yes"
    volume.beta.kubernetes.io/storage-provisioner: csi.trident.netapp.io
  creationTimestamp: 2018-08-21T18:26:44Z
  finalizers:
  - kubernetes.io/pvc-protection
  name: ontapnas20mb
  namespace: default
  resourceVersion: "1958015"
  selfLink: /api/v1/namespaces/default/persistentvolumeclaims/ontapnas20mb
  uid: c1bd7fa5-a56f-11e8-b8d7-fa163e59eaab
spec:
  accessModes:
  - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  ...
```

步驟4：驗證擴充

您可以檢查PVC、PV和Astra Trident Volume的大小、以正確驗證調整大小：

```

$ kubectl get pvc ontapnas20mb
NAME          STATUS    VOLUME
CAPACITY     ACCESS MODES   STORAGECLASS   AGE
ontapnas20mb Bound      pvc-08f3d561-b199-11e9-8d9f-5254004dfdb7 1Gi
RWO          ontapnas      4m44s

$ kubectl get pv pvc-08f3d561-b199-11e9-8d9f-5254004dfdb7
NAME          CAPACITY   ACCESS MODES
RECLAIM POLICY STATUS    CLAIM          STORAGECLASS   REASON
AGE
pvc-08f3d561-b199-11e9-8d9f-5254004dfdb7 1Gi       RWO
Delete      Bound    default/ontapnas20mb  ontapnas
5m35s

$ tridentctl get volume pvc-08f3d561-b199-11e9-8d9f-5254004dfdb7 -n
trident
+-----+-----+-----+-----+
+-----+-----+-----+-----+
|          NAME          | SIZE  | STORAGE CLASS |
PROTOCOL |          BACKEND UUID          | STATE | MANAGED |
+-----+-----+-----+-----+
+-----+-----+-----+-----+
| pvc-08f3d561-b199-11e9-8d9f-5254004dfdb7 | 1.0 GiB | ontapnas      |
file      | c5a6f6a4-b052-423b-80d4-8fb491a14a22 | online | true     |
+-----+-----+-----+-----+
+-----+-----+-----+-----+

```

匯入磁碟區

您可以使用「tridentctl匯入」將現有的儲存磁碟區匯入為Kubernetes PV。

支援Volume匯入的驅動程式

下表說明支援匯入磁碟區的驅動程式、以及這些磁碟區所引進的版本。

驅動程式	版本
「ONTAP-NAS」	19.04
「ONTAP-NAA-flexgroup」	19.04
「olidfire - san」	19.04
《azure-NetApp-filiform》	19.04

驅動程式	版本
《GCP—CVS》	19.04
「ONTAP-SAN」	19.04

為什麼要匯入磁碟區？

將Volume匯入Trident的使用案例有多種：

- 容器化應用程式、並重新使用現有的資料集
- 將資料集的複本用於暫時性應用程式
- 重建故障的Kubernetes叢集
- 在災難恢復期間移轉應用程式資料

匯入如何運作？

Volume匯入程序會使用持續磁碟區宣告 (PVC) 檔案來建立PVC。至少、PVC檔案應包含名稱、命名空間、存取模式及storageClassName欄位、如下例所示。

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: my_claim
  namespace: my_namespace
spec:
  accessModes:
    - ReadWriteOnce
  storageClassName: my_storage_class
```

「tridentctl」用戶端用於匯入現有的儲存磁碟區。Trident會持續儲存Volume中繼資料並建立PVC和PV、以匯入Volume。

```
$ tridentctl import volume <backendName> <volumeName> -f <path-to-pvc-
file>
```

若要匯入儲存磁碟區、請指定包含該磁碟區的Astra Trident後端名稱、以及唯一識別儲存區上該磁碟區的名稱（例如ONTAP FlexVol：Wsel, Element Volume、CVS Volume path）。儲存磁碟區必須允許讀取/寫入存取、且可由指定的Astra Trident後端存取。需要「-f」字串引數、並指定Yaml或Json PVC檔案的路徑。

當Astra Trident收到匯入磁碟區要求時、現有的磁碟區大小會在PVC中決定及設定。儲存驅動程式匯入磁碟區之後、PV會以PVC的ClaimRef建立。在PV中、回收原則一開始設定為「需要」。Kubernetes成功繫結了PVC和PV之後、系統會更新回收原則以符合儲存類別的回收原則。如果儲存類別的回收原則為「刪除」、則儲存磁碟區會在刪除PV時刪除。

當使用「-NO-Manage」引數匯入磁碟區時、Trident不會在物件生命週期的永久虛擬磁碟或PV上執行任何額外作業。由於Trident會忽略「無管理」物件的PV和PVC事件、因此在刪除PV時不會刪除儲存磁碟區。此外、也會忽略其他作業、例如Volume Clone和Volume resize。如果您想要將Kubernetes用於容器化工作負載、但想要管理Kubernetes以外儲存磁碟區的生命週期、則此選項非常實用。

將註釋新增至PVC和PV、這有兩種用途、表示已匯入磁碟區、以及是否管理了PVC和PV。不應修改或移除此附註。

Trident 19.07及更新版本可處理PV的附加元件、並在匯入磁碟區時掛載磁碟區。對於使用舊版Astra Trident的匯入、資料路徑不會有任何作業、而且磁碟區匯入不會驗證是否可以掛載磁碟區。如果在Volume匯入時發生錯誤（例如StorageClass不正確）、您可以將PV上的回收原則變更為「保留」、刪除PVC和PV、然後重新嘗試Volume匯入命令來恢復。

ontap-nas 和 ontap-nas-flexgroup 匯入

每個使用「ONTAP-NAS」驅動程式建立的Volume FlexVol、都是ONTAP一個在整個叢集上的功能。使用「ONTAP-NAS」驅動程式匯入FlexVols的運作方式相同。已存在於某個叢集上的一個功能、可以匯入為「ONTAP-NAS」的PVC。FlexVol ONTAP同樣地FlexGroup、也可以將此數據匯入為「ONTAP-NAS-Flexgroup」PVCS。



若要以Trident匯入某個類型的Rw。ONTAP如果磁碟區是DP類型、則它是SnapMirror目的地磁碟區；在將磁碟區匯入Trident之前、您應該先中斷鏡射關係。



「ONTAP-NAS」驅動程式無法匯入及管理qtree。「ONTAP-NAS」和「ONTAP-NAS-flexgroup」驅動程式不允許重複的Volume名稱。

例如、若要在名為「ONTAP_NAS」的後端匯入名為「managed_volume」的磁碟區、請使用下列命令：

```
$ tridentctl import volume ontap_nas managed_volume -f <path-to-pvc-file>
```

PROTOCOL	NAME	BACKEND UUID	SIZE	STATE	STORAGE CLASS	MANAGED
file	pvc-bf5ad463-afbb-11e9-8d9f-5254004dfdb7	c5a6f6a4-b052-423b-80d4-8fb491a14a22	1.0 GiB	online	standard	true

若要匯入名為「Unmanaged_volume」（在「ONTAP_NAS後端」上）的磁碟區（Trident無法管理）、請使用下列命令：


```
$ tridentctl import volume nas_blog unmanaged_volume -f <path-to-pvc-file>
--no-manage
```

PROTOCOL	NAME	BACKEND UUID	SIZE	STORAGE CLASS	STATE	MANAGED
file	pvc-df07d542-afbc-11e9-8d9f-5254004dfdb7	c5a6f6a4-b052-423b-80d4-8fb491a14a22	1.0 GiB	standard	online	false

使用「-no-管理」引數時、Trident不會重新命名磁碟區、也不會驗證磁碟區是否已掛載。如果未手動掛載磁碟區、則磁碟區匯入作業會失敗。



已修正先前使用自訂Unix權限 匯入磁碟區的錯誤。您可以在您的PVC定義或後端組態中指定unixPermissions、並指示Astra Trident依此匯入磁碟區。

ontap-san 匯入

Astra Trident也能匯入ONTAP 包含單一LUN的SAN FlexVols。這與「ONTAP-SAN」驅動程式一致、因為它會為FlexVol 每個實體磁碟和FlexVol 一個LUN在整個過程中建立一個版本。您可以使用「tridentctl import」命令、方法與其他情況相同：

- 請加入「ONTAP-SAN」後端名稱。
- 請提供FlexVol 需要匯入的名稱。請記住FlexVol、這個功能只包含一個必須匯入的LUN。
- 提供必須搭配「-f」旗標使用的PVC定義路徑。
- 您可以選擇管理或不受管理的永久虛擬網路。根據預設、Trident會管理PVC、並在FlexVol 後端重新命名該LUN。若要匯入為非託管磁碟區、請傳遞「-no-管理」旗標。



匯入未受管理的「ONTAP-SAN」Volume時、您應確定FlexVol 將位於該功能的LUN命名為「lun0」、並對應至具有所需啟動器的igroup。Astra Trident會自動處理這項作業、以便進行託管匯入。

然後Astra Trident會匯入FlexVol 該等物件、並將其與PVC定義建立關聯。Astra Trident也將FlexVol 該等數據重新命名為「PVC-<uuid>」格式、FlexVol 並將LUN重新命名為「lun0」。



建議匯入沒有現有作用中連線的磁碟區。如果您要匯入使用中的Volume、請先複製該Volume、然後再執行匯入。

範例

若要匯入FlexVol 在「ONTAP_san預設」後端上出現的「onta-san託管」功能、請執行「tridentctl匯入」命令：

```
$ tridentctl import volume ontapsan_san_default ontap-san-managed -f pvc-
basic-import.yaml -n trident -d
```

```
+-----+-----+-----+-----+
+-----+-----+-----+-----+
|          NAME          |  SIZE  | STORAGE CLASS |
PROTOCOL |          BACKEND UUID          |  STATE  | MANAGED |
+-----+-----+-----+-----+
+-----+-----+-----+-----+
| pvc-d6ee4f54-4e40-4454-92fd-d00fc228d74a | 20 MiB | basic          |
block   | cd394786-ddd5-4470-adc3-10c5ce4ca757 | online | true          |
+-----+-----+-----+-----+
+-----+-----+-----+-----+
```



若要由Astra Trident匯入某個類型的RW磁碟區。ONTAP如果磁碟區為DP類型、則為SnapMirror目的地磁碟區；您應該先中斷鏡射關係、再將磁碟區匯入Astra Trident。

element 匯入

您可以使用NetApp Element Trident將支援功能的軟體/NetApp HCI磁碟區匯入Kubernetes叢集。您需要Astra Trident後端的名稱、以及磁碟區的獨特名稱和Pvc檔案、作為「tridentctl import」命令的引數。

```
$ tridentctl import volume element_default element-managed -f pvc-basic-
import.yaml -n trident -d
```

```
+-----+-----+-----+-----+
+-----+-----+-----+-----+
|          NAME          |  SIZE  | STORAGE CLASS |
PROTOCOL |          BACKEND UUID          |  STATE  | MANAGED |
+-----+-----+-----+-----+
+-----+-----+-----+-----+
| pvc-970ce1ca-2096-4ecd-8545-ac7edc24a8fe | 10 GiB | basic-element |
block   | d3ba047a-ea0b-43f9-9c42-e38e58301c49 | online | true          |
+-----+-----+-----+-----+
+-----+-----+-----+-----+
```



Element驅動程式支援重複的Volume名稱。如果有重複的Volume名稱、Trident的Volume匯入程序會傳回錯誤。因應措施是複製磁碟區、並提供唯一的磁碟區名稱。然後匯入複製的Volume。

gcp-cvs 匯入



若要匯入以NetApp Cloud Volumes Service 支援的GCP磁碟區、請使用磁碟區路徑來識別該磁碟區、而非其名稱。

若要在後端匯入名為「gcpcvs_YEppr」的「gcp-CVS」磁碟區、並將磁碟區路徑匯入為「adj統 法社-jolly Swift」、請使用下列命令：

```
$ tridentctl import volume gcpcvs_YEppr adroit-jolly-swift -f <path-to-pvc-file> -n trident
```

PROTOCOL	NAME	BACKEND UUID	SIZE	STATE	STORAGE CLASS	MANAGED
	pvc-a46ccab7-44aa-4433-94b1-e47fc8c0fa55	e1a6e65b-299e-4568-ad05-4f0a105c888f	93 GiB	online	gcp-storage	true



Volume路徑是Volume匯出路徑的一部分、位於:/之後。例如、如果匯出路徑為「10.0.0.1:/統 法社-jolly Swift」、則Volume路徑為「adj統 法社-jolly Swift」。

azure-netapp-files 匯入

若要在後端匯入名為「azurenetafiles_40517」的「azure-netapp-fil」磁碟區、並使用磁碟區路徑「importvol1」、請執行下列命令：

```
$ tridentctl import volume azurenetafiles_40517 importvol1 -f <path-to-pvc-file> -n trident
```

PROTOCOL	NAME	BACKEND UUID	SIZE	STATE	STORAGE CLASS	MANAGED
file	pvc-0ee95d60-fd5c-448d-b505-b72901b3a4ab	1c01274f-d94b-44a3-98a3-04c953c9a51e	100 GiB	online	anf-storage	true



anf磁碟區的磁碟區路徑會出現在裝載路徑中的:/之後。例如、如果掛載路徑為「10.0.0.2:/importvol1」、則磁碟區路徑為「importvol1」。

準備工作節點

Kubernetes叢集中的所有工作節點都必須能夠掛載您已為Pod配置的磁碟區。如果您使用的是「ONTAP-NAS」、「ONTAP-NAS-節約」或「ONTAP-NAS-Flexgroup」驅動程式來支援其中一個後端、則您的工作節點需要NFS工具。否則他們需要iSCSI工具。

最新版本的RedHat CoreOS預設會同時安裝NFS和iSCSI。



安裝NFS或iSCSI工具之後、您應該一律重新啟動工作節點、否則將磁碟區附加至容器可能會失敗。

NFS磁碟區

傳輸協定	作業系統	命令
NFS	RHEL/CentOS	「Udo yum install -y ns-utils」
NFS	Ubuntu / DEBIAN	「Udo apt-Get install -y nfs通用」



您應確保NFS服務在開機期間啟動。


iSCSI磁碟區

使用iSCSI磁碟區時、請考量下列事項：

- Kubernetes叢集中的每個節點都必須具有唯一的IQN。這是必要的先決條件。
- 如果使用帶有「Poolidfire - san」驅動程式的RHcos 4.5或更新版本、或RHEL或CentOS 8.2或更新版本、請確定在「/etc/iscsi/iscsid.conf」中將CHAP驗證演算法設定為「md5」。

```
sudo sed -i 's/^\(node.session.auth.chap_algs\) .*\/\1 = MD5/'  
/etc/iscsi/iscsid.conf
```

- 使用執行RHEL/RedHat CoreOS搭配iSCSI PV的工作節點時、請務必在StorageClass中指定「discard」掛載選項、以執行即時空間回收。請參閱 "[RedHat的文件](#)"。

傳輸協定	作業系統	命令
iSCSI	RHEL/CentOS	<p>1. 安裝下列系統套件：</p> <pre>Udo yum install -y lsscsi iscsi-initiator-utils sg3_utils device-mapper-multipath'</pre> <p>2. 檢查iscsite-initier-utils版本是否為6.6.0.874-2.el7或更新版本：</p> <pre>rpm -q iscsisiscsiscs-initier- utils」</pre> <p>3. 將掃描設為手動：</p> <pre>Udo sed -i 's/^((node.session.scan).*\1 = manual /) /etc/iscsi/iscsid.conf)</pre> <p>4. 啟用多重路徑：</p> <pre>Udo mpathconf --enable --with _docathd y --fisco_multipaths n」</pre> <div style="border: 1px solid gray; padding: 5px; margin: 10px 0;">  <p>確保在"default" (錯誤) 下"etc/multipath. conf"包 含"fappe_multip aths no"。</p> </div> <p>5. 確保運行的是"iscsid" 和"multipathd"：</p> <pre>Udo systemctl -現在 是iscsid多路徑」</pre> <p>6. 啟用並啟動「iSCSI」：</p> <pre>Udo systemctl enable - NOW iSCSI」</pre>

傳輸協定	作業系統	命令
iSCSI	Ubuntu / DEBIAN	<p>1. 安裝下列系統套件：</p> <pre>Udo apt-Get install -y open-iscsi sg3-utils multipath-tools scsitols</pre> <p>2. 檢查開放式iSCSI版本是否為2.0.874-5ubuntu2.10或更新版本（適用於雙聲網路）或2.0.874-7.1ubuntu6.1或更新版本（適用於焦點）：</p> <pre>Udo dpkg -l open-iscsi</pre> <p>3. 將掃描設為手動：</p> <pre>Udo sed -i 's/^(node.session.scan).*\1 = manual /' /etc/iscsi/iscsid.conf</pre> <p>4. 啟用多重路徑：</p> <pre>Udo tee /etc/multipath.conf <<-'EOF'預設值 {user_易 名稱yes fops_multipaths no } EOF Sudo systemctl enable --NOW multipath-tools.service Sudo服務multipath-ttools restart</pre> <div style="border: 1px solid gray; padding: 5px; margin: 10px 0;"> <p> 確保在"default"（錯誤）下"etc/multipath.conf"包含"fappe_multipaths no"。</p> </div> <p>5. 確保已啟用並執行「open-iscsi」和「多路徑工具」：</p> <pre>Udo systemctl狀態multipath-ttools/ Udo show systemctl enable -現在open-iscsi.service Udo show systemctl狀態open-iscsi</pre>



對於Ubuntu 18.04、您必須先使用「iscsiadm」探索目標連接埠、然後再啟動「open-iscsi」、iSCSI精靈才能啟動。您也可以修改「iSCSI」服務、以自動啟動「iscsid」。



如果您想要深入瞭解自動工作節點準備（這是試用版功能）、請參閱 ["請按這裡"](#)。

自動準備工作節點

Astra Trident可自動在Kubernetes叢集中的節點上安裝所需的「NFS」和「iSCSI」工具。這是*試用版功能*、不適用於*正式作業叢集*。目前、此功能適用於執行CentOS、RHEL及Ubuntu*的節點。

針對這項功能、Astra Trident提供新的安裝旗標：「-enable節點準備」、適用於以「tridentctl」部署的安裝。對於使用Trident運算子的部署、請使用布林選項「enableNodePrep」。



「啟用節點準備」安裝選項可讓Astra Trident安裝、並確保在工作節點上掛載磁碟區時、NFS和iSCSI套件及/或服務會執行。這是*試用版功能*、適用於*不符合*正式作業使用資格的開發/測試環境。

當使用「tridentctl」部署的Astra Trident安裝包含「-enable節點準備」旗標時、會發生以下情況：

1. 在安裝過程中、Astra Trident會登錄其執行的節點。
2. 當提出持續磁碟區宣告（PVC）要求時、Astra Trident會從其管理的其中一個後端建立PV。
3. 在Pod中使用永久虛擬磁碟時、需要使用Astra Trident將磁碟區掛載到執行Pod的節點上。Astra Trident會嘗試安裝所需的NFS/iSCSI用戶端公用程式、並確保所需的服務處於作用中狀態。這是在掛載磁碟區之前完成的。

在第一次嘗試掛載磁碟區時、只需準備一次工作節點。只要Astra Trident以外的任何變更都不會接觸到「NFS」和「iSCSI」公用程式、所有後續的Volume掛載就會成功。

如此一來、Astra Trident就能確保Kubernetes叢集中的所有節點都具備必要的公用程式、以掛載及附加磁碟區。對於NFS磁碟區、匯出原則也應該允許掛載磁碟區。Trident可以自動管理每個後端的匯出原則、也可以管理頻外的匯出原則。

監控Astra Trident

Astra Trident提供一組Prometheus指標端點、可用來監控Astra Trident的效能。

Astra Trident提供的指標可讓您執行下列作業：

- 隨時掌握Astra Trident的健全狀況與組態。您可以檢查作業的成功程度、以及是否能如預期般與後端進行通訊。
- 檢查後端使用資訊、並瞭解後端上配置的磁碟區數量、以及所耗用的空間量等。
- 維護可用後端配置的磁碟區數量對應。
- 追蹤效能。您可以查看Astra Trident與後端及執行作業所需的時間。



根據預設、Trident的指標會顯示在「/度量」端點的目標連接埠「8001」上。安裝Trident時*預設會啟用這些度量。

您需要的產品

- 安裝Astra Trident的Kubernetes叢集。

- Prometheus執行個體。這可以是 ["容器化Prometheus部署"](#) 或者、您也可以選擇以執行Prometheus ["原生應用程式"](#)。

步驟1：定義Prometheus目標

您應該定義Prometheus目標、以收集指標並取得有關後端Astra Trident管理的資訊、以及其建立的磁碟區等資訊。這 ["部落格"](#) 說明如何使用Prometheus和Grafana搭配Astra Trident來擷取指標。部落格說明如何在Kubernetes叢集中以營運者的形式執行Prometheus、以及建立ServiceMonitor來取得Astra Trident的指標。

步驟2：建立Prometheus ServiceMonitor

若要使用Trident指標、您應該建立Prometheus ServiceMonitor、以監控「Trident - csi」服務、並在「metrics」連接埠上聆聽。ServiceMonitor範例如下所示：

```
apiVersion: monitoring.coreos.com/v1
kind: ServiceMonitor
metadata:
  name: trident-sm
  namespace: monitoring
  labels:
    release: prom-operator
spec:
  jobLabel: trident
  selector:
    matchLabels:
      app: controller.csi.trident.netapp.io
  namespaceSelector:
    matchNames:
      - trident
  endpoints:
    - port: metrics
      interval: 15s
```

此ServiceMonitor定義會擷取由「Trident -csi」服務傳回的度量、並特別尋找服務的「metrics」端點。因此、Prometheus現在已設定為瞭解Astra Trident的指標。

除了可直接從Astra Trident取得的指標之外、Kubelet還透過自己的指標端點、公開許多「kubelet_volume」*指標。Kubelet可提供有關所附加磁碟區、Pod及其處理的其他內部作業的資訊。請參閱 ["請按這裡"](#)。

步驟3：使用PromQL查詢Trident度量

PromQL適用於建立傳回時間序列或表格資料的運算式。

以下是一些您可以使用的PromQL查詢：

取得Trident健全狀況資訊

- 來自Astra Trident的HTTP 2XX回應百分比*

```
(sum (trident_rest_ops_seconds_total_count{status_code=~"2.."} OR on()  
vector(0)) / sum (trident_rest_ops_seconds_total_count)) * 100
```

- 透過狀態代碼*來自Astra Trident的休息回應百分比

```
(sum (trident_rest_ops_seconds_total_count) by (status_code) / scalar  
(sum (trident_rest_ops_seconds_total_count))) * 100
```

- 由Astra Trident執行的平均營運持續時間

```
sum by (operation)  
(trident_operation_duration_milliseconds_sum{success="true"}) / sum by  
(operation)  
(trident_operation_duration_milliseconds_count{success="true"})
```

取得Astra Trident使用資訊

- 平均Volume大小*

```
trident_volume_allocated_bytes/trident_volume_count
```

- 每個後端配置的Volume空間總計*

```
sum (trident_volume_allocated_bytes) by (backend_uuid)
```

取得個別Volume使用量



只有同時收集kubelet度量時、才會啟用此功能。

- 每個Volume的已用空間百分比*

```
kubelet_volume_stats_used_bytes / kubelet_volume_stats_capacity_bytes *  
100
```

深入瞭解Astra Trident AutoSupport 遙測技術

依預設、Astra Trident會每日傳送Prometheus指標和基本後端資訊給NetApp。

- 若要停止Astra Trident將Prometheus指標和基本後端資訊傳送給NetApp、請在Astra Trident安裝期間、傳遞「-Silence-autosupport」旗標。
- Astra Trident也可透過「tridentctl傳送AutoSupport 功能」、隨需將容器記錄傳送給NetApp Support。您需要觸發Astra Trident來上傳記錄。在您提交記錄之前、您應該接受NetApp的<https://www.netapp.com/company/legal/privacy-policy/>["隱私權政策"]。
- 除非另有說明、Astra Trident會從過去24小時擷取記錄。
- 您可以使用「-自」旗標來指定記錄保留時間範圍。例如：「tridentctl傳送AutoSupport 功能不全、自一小時起」。此資訊是透過Astra Trident一起安裝的「Trident -autosupport」容器來收集和傳送。您可以從取得Container映像 "[Trident AutoSupport 的](#)"。
- Trident AutoSupport 無法收集或傳輸個人識別資訊 (PII) 或個人資訊。隨附a "[EULA](#)" 這不適用於Trident Container映像本身。您可以深入瞭解NetApp對資料安全性與信任的承諾 "[請按這裡](#)"。

Astra Trident傳送的有效負載範例如下：

```
{
  "items": [
    {
      "backendUUID": "ff3852e1-18a5-4df4-b2d3-f59f829627ed",
      "protocol": "file",
      "config": {
        "version": 1,
        "storageDriverName": "ontap-nas",
        "debug": false,
        "debugTraceFlags": null,
        "disableDelete": false,
        "serialNumbers": [
          "nwkvzfanek_SN"
        ],
        "limitVolumeSize": ""
      },
      "state": "online",
      "online": true
    }
  ]
}
```

- 此資訊將傳送至NetApp的「不只是」端點。AutoSupport AutoSupport如果您使用私有登錄來儲存容器映像、可以使用「-image-registry」旗標。
- 您也可以產生安裝Yaml檔案來設定Proxy URL。您可以使用「tridentctl install -generate-custom-yaml」來建立Yaml檔案、並在「trident部署.yaml」中新增「trident -autosupport」容器的「-proxy-URL」引數。

停用Astra Trident度量

要使指標不被報告，您應該生成自定義YAML（使用"-generame-custom-yaml"標誌）並進行編輯，以刪除對"trident-main"容器所調用的"-mication"標誌。

版權資訊

Copyright © 2024 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。