



設定及管理後端 Astra Trident

NetApp
January 14, 2026

目錄

設定及管理後端	1
設定後端	1
Azure NetApp Files	1
設定Azure NetApp Files 一個靜態後端	1
準備設定Azure NetApp Files 一個功能完善的後端	4
列舉後端組態選項與範例Azure NetApp Files	7
Google Cloud NetApp Volumes	17
設定 Google Cloud NetApp Volumes 後端	17
準備設定 Google Cloud NetApp Volumes 後端	17
Google Cloud NetApp Volumes 後端組態選項和範例	18
設定Cloud Volumes Service 適用於Google Cloud後端的功能	28
Google Cloud 驅動程式詳細資料	28
深入瞭解Astra Trident對Cloud Volumes Service Google Cloud的支援	28
後端組態選項	29
Volume資源配置選項	30
CVS效能服務類型範例	30
CVS服務類型範例	36
接下來呢？	38
設定NetApp HCI 一個不只是功能的SolidFire 後端	39
元素驅動程式詳細資料	39
開始之前	39
後端組態選項	39
範例 1：具有三種磁碟區類型的驅動程式後端組態 solidfire-san	40
範例 2：具有虛擬集區之驅動程式的後端和儲存類別組態 solidfire-san	41
如需詳細資訊、請參閱	45
支援SAN驅動程式ONTAP	45
ONTAP SAN 驅動程式概觀	45
準備使用ONTAP 不完善的SAN驅動程式來設定後端	47
列舉SAN組態選項與範例ONTAP	53
ONTAP NAS 驅動程式	67
ONTAP NAS 驅動程式概觀	67
準備使用ONTAP 不含NAS的驅動程式來設定後端	69
列舉NAS組態選項與範例ONTAP	77
Amazon FSX for NetApp ONTAP 產品	93
使用Astra Trident搭配Amazon FSX for NetApp ONTAP 解決方案	93
建立 IAM 角色和 AWS 密碼	94
安裝Astra Trident	95
設定儲存後端	101
設定儲存類別和 PVC	109

部署範例應用程式	114
在 EKS 叢集上設定 Astra Trident EKS 附加元件	115
使用 kubectl 建立後端	121
TridentBackendConfig	121
步驟總覽	122
步驟 1：建立 Kubernetes 機密	123
步驟 2：建立 TridentBackendConfig CR	124
步驟 3：確認 CR 的狀態 TridentBackendConfig	125
(選用) 步驟 4：取得更多詳細資料	126
管理後端	127
以 KECBECVL 執行後端管理	128
使用 tridentctl 執行後端管理	129
在後端管理選項之間切換	130

設定及管理後端

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。瞭解如何設定儲存系統的後端。

- ["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 功能不一的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"](#)

Azure NetApp Files

設定Azure NetApp Files 一個靜態後端

您可以將 Azure NetApp Files 設定為 Astra Trident 的後端。您可以使用 Azure NetApp Files 後端連接 NFS 和 SMB 磁碟區。Astra Trident 也支援使用 Azure Kubernetes Services (aks) 叢集的託管身分識別來進行認證管理。

Azure NetApp Files 驅動程式詳細資料

Astra Trident 提供下列 Azure NetApp Files 儲存驅動程式來與叢集通訊。支援的存取模式包括：
ReadWriteOnce (rwo) 、 *ReadOnlyMany* (ROX) 、 *_ReadWriteMany* (rwx) 、 *_ReadWriteOncePod* (RWOP) 。

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
azure-netapp-files	NFS SMB	檔案系統	Rwo 、 ROX 、 rwx 、 RWOP	nfs 、 smb

考量

- 此支援服務不支援小於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-GiB 磁碟區。
- Astra Trident僅支援安裝在Windows節點上執行的Pod上的SMB磁碟區。

管理的身分識別

Astra Trident 支援 ["託管身分識別"](#)Azure Kubernetes 服務叢集。若要善用託管身分識別所提供的簡化認證管理功能、您必須具備：

- 使用 aks 部署的 Kubernetes 叢集
- 在 aks Kubernetes 叢集上設定的託管身分識別
- 安裝了 Astra Trident ，其中包括 `cloudProvider to specify "Azure"` 。

Trident 運算子

若要使用 Trident 運算子安裝 Astra Trident 、請編輯 `tridentorchestrator_cr.yaml` 以設定為 `cloudProvider "Azure"` 。例如：

```
apiVersion: trident.netapp.io/v1
kind: TridentOrchestrator
metadata:
  name: trident
spec:
  debug: true
  namespace: trident
  imagePullPolicy: IfNotPresent
  cloudProvider: "Azure"
```

掌舵

以下範例使用環境變數將 Astra Trident Set 安裝 `cloudProvider` 至 `Azure` `$CP` ：

```
helm install trident trident-operator-100.2406.0.tgz --create
--namespace --namespace <trident-namespace> --set cloudProvider=$CP
```

`<code>` 取向 `</code>`

以下範例安裝 Astra Trident 並將旗標設定 `cloudProvider` 為 `Azure` ：

```
tridentctl install --cloud-provider="Azure" -n trident
```

雲端身分識別

雲端身分識別可讓 Kubernetes Pod 以工作負載身分驗證來存取 Azure 資源、而非提供明確的 Azure 認證。

若要在 Azure 中使用雲端身分識別、您必須具備：

- 使用 aks 部署的 Kubernetes 叢集
- 在 OKS Kubernetes 叢集上設定的工作負載識別和 `oidc-c` 發行者
- 已安裝 Astra Trident 、其中包含 `cloudProvider` 指定 `"Azure"` 及 `cloudIdentity` 指定工作負載身分識別的

Trident 運算子

若要使用 Trident 運算子安裝 Astra Trident、請編輯 `tridentorchestrator_cr.yaml` 以設定為 `cloudProvider "Azure"`、並設定 `cloudIdentity` 為 `azure.workload.identity/client-id: xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxx`。

例如：

```
apiVersion: trident.netapp.io/v1
kind: TridentOrchestrator
metadata:
  name: trident
spec:
  debug: true
  namespace: trident
  imagePullPolicy: IfNotPresent
  cloudProvider: "Azure"
  *cloudIdentity: 'azure.workload.identity/client-id: xxxxxxxx-xxxx-
  xxx-xxxx-xxxxxxxxxxxxx'*
```

掌舵

使用下列環境變數設定 * 雲端供應商 (CP) * 和 * 雲端身分識別 (CI) * 旗標的值：

```
export CP="Azure"
export CI="azure.workload.identity/client-id: xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxxx"
```

以下範例安裝 Astra Trident、並使用環境變數設定 `cloudProvider` 為 `Azure $CP`、並使用環境變數 `$CI` 設定 `cloudIdentity`：

```
helm install trident trident-operator-100.2406.0.tgz --set
cloudProvider=$CP --set cloudIdentity=$CI
```

<code> 取向 </code>

使用下列環境變數設定 * 雲端供應商 * 和 * 雲端 IDENTITY * 旗標的值：

```
export CP="Azure"
export CI="azure.workload.identity/client-id: xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxxx"
```

以下範例安裝 Astra Trident 並將旗標設定 `cloud-provider` 為 `$CP`、和 `cloud-identity $CI`：

```
tridentctl install --cloud-provider=$CP --cloud-identity="$CI" -n
trident
```

準備設定 Azure NetApp Files 一個功能完善的後端

在您設定 Azure NetApp Files 完後端功能之前、您必須確保符合下列要求。

NFS 和 SMB 磁碟區的必要條件

如果您是第一次使用 Azure NetApp Files、或是在新位置使用、則必須先進行一些初始設定、才能設定 Azure NetApp Files 並建立 NFS Volume。請參閱 ["Azure：設定 Azure NetApp Files 功能以建立 NFS Volume"](#)。

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



- `clientID`` 在 AKS 叢集上使用託管身分識別時、``subscriptionID``、`tenantID``、``location`` 和 ``clientSecret`` 是選用的。
- `tenantID``、`clientID`` 和 ``clientSecret`` 是在 AKS 叢集上使用雲端身分識別時的選用功能。

- 容量集區。請參閱 ["Microsoft：為 Azure NetApp Files 建立容量集區"](#)。
- 委派給 Azure NetApp Files 的子網路。請參閱 ["Microsoft：將子網路委派給 Azure NetApp Files"](#)。
- ``subscriptionID`` 從啟用 Azure NetApp Files 的 Azure 訂閱中取得。
- 和 `clientSecret`` 來自 Azure Active Directory 的 ["應用程式註冊"](#)、`tenantID``、`clientID`` 具有 Azure NetApp Files 服務的足夠權限。應用程式登錄應使用下列其中一項：
 - 擁有者或貢獻者角色 ["由 Azure 預先定義"](#)。
 - a ["自訂貢獻者角色"](#) 在訂閱層級 (`assignableScopes`)、具有下列權限、僅限於 Astra Trident 所需的權限。建立自訂角色之後 ["使用 Azure 入口網站指派角色"](#)，。

```
{
  "id": "/subscriptions/<subscription-
id>/providers/Microsoft.Authorization/roleDefinitions/<role-
definition-id>",
  "properties": {
    "roleName": "custom-role-with-limited-perms",
    "description": "custom role providing limited
permissions",
    "assignableScopes": [
      "/subscriptions/<subscription-id>"
    ],
    "permissions": [
      {
        "actions": [
          "Microsoft.NetApp/netAppAccounts/capacityPools/read",
          "Microsoft.NetApp/netAppAccounts/capacityPools/write",
          "Microsoft.NetApp/netAppAccounts/capacityPools/volumes/read",
          "Microsoft.NetApp/netAppAccounts/capacityPools/volumes/write",
          "Microsoft.NetApp/netAppAccounts/capacityPools/volumes/delete",
          "Microsoft.NetApp/netAppAccounts/capacityPools/volumes/snapshots/
read",
          "Microsoft.NetApp/netAppAccounts/capacityPools/volumes/snapshots/
write",
          "Microsoft.NetApp/netAppAccounts/capacityPools/volumes/snapshots/
delete",
          "Microsoft.NetApp/netAppAccounts/capacityPools/volumes/MountTarge
ts/read",
          "Microsoft.Network/virtualNetworks/read",
          "Microsoft.Network/virtualNetworks/subnets/read",
          "Microsoft.Features/featureProviders/subscriptionFeatureRegistrat
ions/read",
```

```

"Microsoft.Features/featureProviders/subscriptionFeatureRegistrations/write",

"Microsoft.Features/featureProviders/subscriptionFeatureRegistrations/delete",

    "Microsoft.Features/features/read",
    "Microsoft.Features/operations/read",
    "Microsoft.Features/providers/features/read",

"Microsoft.Features/providers/features/register/action",

"Microsoft.Features/providers/features/unregister/action",

"Microsoft.Features/subscriptionFeatureRegistrations/read"
    ],
    "notActions": [],
    "dataActions": [],
    "notDataActions": []
  }
]
}
}

```

- 包含至少一個 ["委派的子網路"](#)的 Azure location。從 Trident 22.01 開始、此 `location` 參數是後端組態檔案最上層的必填欄位。會忽略虛擬資源池中指定的位置值。
- 要使用 Cloud Identity，請從 ["使用者指派的託管身分識別"](#)獲取 client ID、並在中指定該 ID `azure.workload.identity/client-id: xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx`。

SMB磁碟區的其他需求

若要建立 SMB Volume、您必須具備：

- Active Directory 已設定並連線至 Azure NetApp Files。請參閱 ["Microsoft：建立及管理 Azure NetApp Files 的 Active Directory 連線"](#)。
- Kubernetes叢集具備Linux控制器節點、以及至少一個執行Windows Server 2022的Windows工作節點。Astra Trident僅支援安裝在Windows節點上執行的Pod上的SMB磁碟區。
- 至少有一個 Astra Trident 秘密、內含您的 Active Directory 認證、以便 Azure NetApp Files 能夠驗證至 Active Directory。產生機密 smbcreds：

```

kubectl create secret generic smbcreds --from-literal username=user
--from-literal password='password'

```

- 設定為Windows服務的SCSI Proxy。若要設定 csi-proxy、請參閱或["GitHub：適用於Windows的SCSI Proxy"](#)、瞭["GitHub：csi Proxy"](#)解在 Windows 上執行的 Kubernetes 節點。

列舉後端組態選項與範例 Azure NetApp Files

瞭解 Azure NetApp Files 的 NFS 和 SMB 後端組態選項、並檢閱組態範例。

後端組態選項

Astra Trident 使用您的後端組態（子網路、虛擬網路、服務層級和位置）、在所要求位置的可用容量集區上建立 Azure NetApp Files Volume、並符合所要求的服務層級和子網路。



Astra Trident 不支援手動 QoS 容量集區。

Azure NetApp Files 後端提供這些組態選項。

參數	說明	預設
version		永遠為 1
storageDriverName	儲存驅動程式名稱	「Azure - NetApp-Files」
backendName	自訂名稱或儲存後端	驅動程式名稱 + 「_」 + 隨機字元
subscriptionID	當在 AKS 叢集上啟用託管身分識別時、Azure 訂閱的訂閱 ID 為選用。	
tenantID	在 AKS 叢集上使用託管身分識別或雲端身分識別時、應用程式登錄的租戶 ID 為選用。	
clientID	在 AKS 叢集上使用託管身分識別或雲端身分識別時、應用程式登錄的用戶端 ID 為選用。	
clientSecret	在 AKS 叢集上使用託管身分識別或雲端身分識別時、應用程式登錄的用戶端機密為選用。	
serviceLevel	Premium、或之 Ultra、或 Standard	"" (隨機)
location	在 AKS 叢集上啟用託管身分識別時、將在其中建立新磁碟區的 Azure 位置名稱為選用。	
resourceGroups	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)
capacityPools	用於篩選已探索資源的容量集區清單	「[]」 (無篩選器、隨機)
virtualNetwork	具有委派子網路的虛擬網路名稱	"
subnet	委派給的子網路名稱 Microsoft.Netapp/volumes	"

參數	說明	預設
networkFeatures	Volume 的 vnet 功能集可以是 Basic 或 Standard。並非所有地區都提供網路功能、可能必須在訂閱中啟用。指定 networkFeatures 功能未啟用的時間會導致 Volume 資源配置失敗。	"
nfsMountOptions	精細控制NFS掛載選項。SMB磁碟區已忽略。若要使用 NFS 4.1 版掛載磁碟區、請在以逗號分隔的掛載選項清單中加入 nfsvers=4、以選擇 NFS v4.1。儲存類別定義中設定的掛載選項會覆寫在後端組態中設定的掛載選項。	"nfsvers=3"
limitVolumeSize	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": false, "method": true, "discovery": 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
nasType	設定NFS或SMB磁碟區建立。選項為 nfs、smb 或 null。NFS磁碟區的預設值設為null。	nfs
supportedTopologies	代表此後端所支援的區域和區域清單。如需詳細資訊、請 "使用「csi拓撲」" 參閱。	



如需網路功能的詳細資訊、請["設定Azure NetApp Files 適用於某個聲音量的網路功能"](#)參閱。

必要的權限與資源

如果您在建立 PVC 時收到「找不到容量集區」錯誤、您的應用程式註冊可能沒有相關的必要權限和資源（子網路、虛擬網路、容量集區）。如果啟用偵錯、Astra Trident會記錄在建立後端時探索到的Azure資源。確認使用的角色是否適當。

、netappAccounts capacityPools、virtualNetwork和 subnet 的值 resourceGroups 可以使用短名稱或全限定名稱來指定。在大多數情況下、建議使用完整名稱、因為短名稱可以符合多個名稱相同的資源。

resourceGroups netappAccounts 和 capacityPools 值是將一組探索到的資源限制於此儲存後端可用資源的篩選器、可以任意組合指定。完整名稱格式如下：

類型	格式
資源群組	<資源群組>

類型	格式
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>

Volume資源配置

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Volume資源配置。如 [\[組態範例\]](#) 需詳細資訊、請參閱。

參數	說明	預設
exportRule	匯出新磁碟區的規則。 `exportRule`必須是以逗號分隔的清單、其中列出任何以 CIDR 表示法表示的 IPv4 位址或 IPv4 子網路組合。SMB磁碟區已忽略。	「0.0.0.0/0」
snapshotDir	控制.snapshot目錄的可見度	"假"
size	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）。SMB磁碟區已忽略。	""（預覽功能、訂閱時需要白名單）

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。

最小組態

這是絕對最低的后端組態。有了這項組態、Astra Trident 會探索您在設定位置中委派給 Azure NetApp Files 的所有 NetApp 帳戶、容量集區和子網路、並隨機將新磁碟區放在其中一個集區和子網路上。由於省略、因此 `nasType nfs` 會套用預設值、而後端會為 NFS 磁碟區進行資源配置。

當您剛開始使用 Azure NetApp Files 並試用時、這項組態是理想的選擇、但實際上您會想要為您所配置的磁碟區提供額外的範圍。

```
---
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-anf-1
  namespace: trident
spec:
  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
```

管理的身分識別

這種后端組態 `subscriptionID` 會省略、`tenantID`、`clientID` 和 `clientSecret`、這些是使用託管身分識別時的選用功能。

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-anf-1
  namespace: trident
spec:
  version: 1
  storageDriverName: azure-netapp-files
  capacityPools: ["ultra-pool"]
  resourceGroups: ["aks-ami-eastus-rg"]
  netappAccounts: ["smb-na"]
  virtualNetwork: eastus-prod-vnet
  subnet: eastus-anf-subnet
```

雲端身分識別

這種後端組態會省略、`clientID`和`clientSecret`、這`tenantID`是使用雲端身分識別時的選用功能。

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-anf-1
  namespace: trident
spec:
  version: 1
  storageDriverName: azure-netapp-files
  capacityPools: ["ultra-pool"]
  resourceGroups: ["aks-ami-eastus-rg"]
  netappAccounts: ["smb-na"]
  virtualNetwork: eastus-prod-vnet
  subnet: eastus-anf-subnet
  location: eastus
  subscriptionID: 9f87c765-4774-fake-ae98-a721add45451
```

具有容量集區篩選器的特定服務層級組態

此後端組態會將磁碟區放置在 Azure 的位置、並置於`eastus`容量集區中`Ultra`。Astra Trident 會自動探索該位置中委派給 Azure NetApp Files 的所有子網路、並隨機在其中一個子網路上放置新的磁碟區。

```
---
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
```

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
networkFeatures: Standard
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'
```

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。虛擬池標籤用於根據區分池 performance。

```
---
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
networkFeatures: Basic
nfsMountOptions: vers=3,proto=tcp,timeo=600
labels:
  cloud: azure
storage:
- labels:
  performance: gold
  serviceLevel: Ultra
  capacityPools:
  - ultra-1
  - ultra-2
  networkFeatures: Standard
- labels:
  performance: silver
  serviceLevel: Premium
  capacityPools:
  - premium-1
- labels:
  performance: bronze
  serviceLevel: Standard
  capacityPools:
  - standard-1
  - standard-2
```

Astra Trident 可根據區域和可用性區域、為工作負載提供更多資源。`supportedTopologies` 此後端組態中的區塊用於提供每個後端的區域和區域清單。此處指定的區域和區域值必須符合每個 Kubernetes 叢集節點上標籤的區域和區域值。這些區域和區域代表可在儲存類別中提供的允許值清單。對於包含後端所提供區域和區域子集的儲存類別、Astra Trident 會在所述區域和區域中建立磁碟區。如需詳細資訊、請 [使用「csi拓撲」](#) 參閱。

```
---
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
supportedTopologies:
- topology.kubernetes.io/region: eastus
  topology.kubernetes.io/zone: eastus-1
- topology.kubernetes.io/region: eastus
  topology.kubernetes.io/zone: eastus-2
```

儲存類別定義

下列 `StorageClass` 定義是指上述儲存資源池。

使用欄位的定義範例 `parameter.selector`

使用 `parameter.selector` 您可以為用於裝載 Volume 的每個虛擬集區指定 `StorageClass`。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

```

SMB磁碟區的定義範例

使用 `nasType`、`node-stage-secret-name` 和 `node-stage-secret-namespace`，您可以指定 SMB 磁碟區並提供所需的 Active Directory 認證。

預設命名空間的基本組態

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: anf-sc-smb
provisioner: csi.trident.netapp.io
parameters:
  backendType: "azure-netapp-files"
  trident.netapp.io/nasType: "smb"
  csi.storage.k8s.io/node-stage-secret-name: "smbcreds"
  csi.storage.k8s.io/node-stage-secret-namespace: "default"
```

每個命名空間使用不同的機密

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: anf-sc-smb
provisioner: csi.trident.netapp.io
parameters:
  backendType: "azure-netapp-files"
  trident.netapp.io/nasType: "smb"
  csi.storage.k8s.io/node-stage-secret-name: "smbcreds"
  csi.storage.k8s.io/node-stage-secret-namespace: ${pvc.namespace}
```

每個磁碟區使用不同的機密

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: anf-sc-smb
provisioner: csi.trident.netapp.io
parameters:
  backendType: "azure-netapp-files"
  trident.netapp.io/nasType: "smb"
  csi.storage.k8s.io/node-stage-secret-name: ${pvc.name}
  csi.storage.k8s.io/node-stage-secret-namespace: ${pvc.namespace}
```



nasType: smb`支援 SMB 磁碟區的集區篩選器。`nasType: nfs`或`nasType: null` NFS 集區的篩選器。

建立後端

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。

Google Cloud NetApp Volumes

設定 Google Cloud NetApp Volumes 後端

您現在可以將 Google Cloud NetApp Volumes 設定為 Astra Trident 的後端。您可以使用 Google Cloud NetApp Volumes 後端來附加 NFS 磁碟區。

Google Cloud NetApp Volumes is a tech preview feature in Astra Trident 24.06.

Google Cloud NetApp Volumes 驅動程式詳細資料

Astra Trident 提供 `google-cloud-netapp-volumes` 與叢集通訊的驅動程式。支援的存取模式包括：
ReadWriteOnce (rwo)、*ReadOnlyMany* (ROX)、*_ReadWriteMany* (rwx)、*_ReadWriteOncePod* (RWOP)。

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
<code>google-cloud-netapp-volumes</code>	NFS	檔案系統	Rwo、ROX、rwx、RWOP	nfs

準備設定 Google Cloud NetApp Volumes 後端

在您設定 Google Cloud NetApp Volumes 後端之前、您必須確保符合下列需求。

NFS Volume 的必要條件

如果您是第一次使用 Google Cloud NetApp Volumes、或是在新位置使用、則需要進行一些初始設定、才能設定 Google Cloud NetApp Volumes 並建立 NFS Volume。請參閱 ["開始之前"](#)。

在設定 Google Cloud NetApp Volumes 後端之前、請先確認您擁有下列項目：

- 使用 Google Cloud NetApp Volumes 服務設定的 Google Cloud 帳戶。請參閱 "[Google Cloud NetApp Volumes](#)"。
- Google Cloud 帳戶的專案編號。請參閱 "[識別專案](#)"。
- 具有 Volumes Admin (NetApp Volume 管理) 角色的 Google Cloud 服務帳戶 (netappcloudvolumes.admin。請參閱 "[身分識別與存取管理角色與權限](#)"。
- 您的 GCNV 帳戶的 API 金鑰檔案。請參閱 "[使用 API 金鑰進行驗證](#)"
- 儲存池。請參閱 "[儲存資源池總覽](#)"。

如需如何設定 Google Cloud NetApp Volumes 存取權限的詳細資訊、請 "[設定 Google Cloud NetApp Volumes 的存取權](#)"參閱。

Google Cloud NetApp Volumes 後端組態選項和範例

瞭解 Google Cloud NetApp Volumes 的 NFS 後端組態選項、並檢閱組態範例。

後端組態選項

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。

參數	說明	預設
version		永遠為1
storageDriverName	儲存驅動程式名稱	的值 storageDriverName 必須指定為「googoogle 雲端 -NetApp-Volumes」。
backendName	(選用) 儲存後端的自訂名稱	驅動程式名稱+「_」+ API 金鑰的一部分
storagePools	選用參數、用於指定用於建立磁碟區的儲存資源池。	
projectNumber	Google Cloud帳戶專案編號。此值可在Google Cloud 入口網站首頁找到。	
location	Astra Trident 建立 GCNV Volume 的 Google Cloud 位置。建立跨區域 Kubernetes 叢集時、在中建立的磁碟區 location 可用於跨多個 Google Cloud 區域的節點上排程的工作負載。跨區域流量會產生額外成本。	
apiKey	具有此角色的 Google Cloud 服務帳戶的 API 金鑰 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。apiKey`必須包含下列金鑰的金鑰值配對： `type project_id`、`client_email`、`client_id`、`auth_uri`、`token_uri`、`auth_provider_x509_cert_url`、和 `client_x509_cert_url`。	
nfsMountOptions	精細控制NFS掛載選項。	"nfsves=3"

參數	說明	預設
limitVolumeSize	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)
serviceLevel	儲存池及其磁碟區的服務層級。這些值包括 flex、standard、premium 或 extreme。	
network	用於 GCNV Volume 的 Google Cloud 網路。	
debugTraceFlags	疑難排解時要使用的偵錯旗標。範例： { "api": false, "method": 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
supportedTopologies	代表此後端所支援的區域和區域清單。如需詳細資訊、請 "使用「csi拓撲」" 參閱。例如： supportedTopologies: - topology.kubernetes.io/region: europe-west6 topology.kubernetes.io/zone: europe-west6-b	

Volume 資源配置選項

您可以在組態檔案的區段中控制預設的 Volume 資源配置 defaults。

參數	說明	預設
exportRule	新磁碟區的匯出規則。必須是以逗號分隔的任何 IPv4 位址組合清單。	「0.0.0.0/0」
snapshotDir	存取`.snapshot`目錄	"假"
snapshotReserve	保留給快照的磁碟區百分比	" (接受預設值 0)
unixPermissions	新磁碟區的UNIX權限 (4個八進位數字)。	"

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。

-----END PRIVATE KEY-----

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-gcnv
spec:
  version: 1
  storageDriverName: google-cloud-netapp-volumes
  projectNumber: '123455380079'
  location: europe-west6
  serviceLevel: premium
  apiKey:
    type: service_account
    project_id: my-gcnv-project
    client_email: myproject-prod@my-gcnv-
project.iam.gserviceaccount.com
    client_id: '103346282737811234567'
    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/myproject-prod%40my-
gcnv-project.iam.gserviceaccount.com
  credentials:
    name: backend-tbc-gcnv-secret
```



```
version: 1
storageDriverName: google-cloud-netapp-volumes
projectNumber: '123455380079'
location: europe-west6
serviceLevel: premium
storagePools:
- premium-pool1-europe-west6
- premium-pool2-europe-west6
apiKey:
  type: service_account
  project_id: my-gcnv-project
  client_email: myproject-prod@my-gcnv-
project.iam.gserviceaccount.com
  client_id: '103346282737811234567'
  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/myproject-prod%40my-
gcnv-project.iam.gserviceaccount.com
  credentials:
    name: backend-tbc-gcnv-secret
```



```
znHczZsrtrtHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m
XsYg6gyxy4zq7OlwWgLwGa==
-----END PRIVATE KEY-----
```

```
---
```

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-gcnv
spec:
  version: 1
  storageDriverName: google-cloud-netapp-volumes
  projectNumber: '123455380079'
  location: europe-west6
  apiKey:
    type: service_account
    project_id: my-gcnv-project
    client_email: myproject-prod@my-gcnv-
project.iam.gserviceaccount.com
    client_id: '103346282737811234567'
    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/myproject-prod%40my-
gcnv-project.iam.gserviceaccount.com
  credentials:
    name: backend-tbc-gcnv-secret
  defaults:
    snapshotReserve: '10'
    exportRule: 10.0.0.0/24
  storage:
    - labels:
        performance: extreme
        serviceLevel: extreme
      defaults:
        snapshotReserve: '5'
        exportRule: 0.0.0.0/0
    - labels:
        performance: premium
        serviceLevel: premium
    - labels:
        performance: standard
        serviceLevel: standard
```

接下來呢？

建立後端組態檔之後、請執行下列命令：

```
kubectl create -f <backend-file>
```

若要確認後端已成功建立、請執行下列命令：

```
kubectl get tridentbackendconfig
```

NAME	BACKEND NAME	BACKEND UUID
PHASE	STATUS	
backend-tbc-gcnv	backend-tbc-gcnv	b2fd1ff9-b234-477e-88fd-713913294f65
Bound	Success	

如果後端建立失敗、表示後端組態有問題。您可以使用命令來描述後端 `kubectl get tridentbackendconfig <backend-name>`、或是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、您可以刪除後端、然後再次執行 `create` 命令。

更多範例

儲存類別定義範例

以下是上述後端的基本 `StorageClass` 定義。

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: gcnv-nfs-sc
provisioner: csi.trident.netapp.io
parameters:
  backendType: "google-cloud-netapp-volumes"
```

- 使用欄位的範例定義 `parameter.selector`：`*`

使用、`parameter.selector` 您可以為用於裝載 Volume 的每個指定 `StorageClass` ["虛擬集區"](#)。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: extreme-sc
provisioner: csi.trident.netapp.io
parameters:
  selector: "performance=extreme"
  backendType: "google-cloud-netapp-volumes"
---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: premium-sc
provisioner: csi.trident.netapp.io
parameters:
  selector: "performance=premium"
  backendType: "google-cloud-netapp-volumes"
---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: standard-sc
provisioner: csi.trident.netapp.io
parameters:
  selector: "performance=standard"
  backendType: "google-cloud-netapp-volumes"

```

如需儲存類別的詳細資訊、請 ["建立儲存類別"](#) 參閱。

PVC 定義範例

```

kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: gcnv-nfs-pvc
spec:
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 100Gi
  storageClassName: gcnv-nfs-sc

```

若要驗證 PVC 是否受限、請執行下列命令：

```
kubectl get pvc gcnv-nfs-pvc
```

NAME	STATUS	VOLUME	CAPACITY
gcnv-nfs-pvc	Bound	pvc-b00f2414-e229-40e6-9b16-ee03eb79a213	100Gi
ACCESS MODES	STORAGECLASS	AGE	
RWX	gcnv-nfs-sc	1m	

設定Cloud Volumes Service 適用於Google Cloud後端的功能

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

Google Cloud 驅動程式詳細資料

Astra Trident 提供 `gcp-cvs` 與叢集通訊的驅動程式。支援的存取模式包括：`ReadWriteOnce` (`rwo`)、`ReadOnlyMany` (`ROX`)、`_ReadWriteMany` (`rwx`)、`_ReadWriteOncePod` (`RWOP`)。

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
<code>gcp-cvs</code>	NFS	檔案系統	<code>Rwo</code> 、 <code>ROX</code> 、 <code>rwx</code> 、 <code>RWOP</code>	<code>nfs</code>

深入瞭解Astra Trident對Cloud Volumes Service Google Cloud的支援

Astra Trident 可以在"[服務類型](#)"以下兩種中的一種中建立 Cloud Volumes Service Volume：

- *** CVS效能***：預設的Astra Trident服務類型。這種效能最佳化的服務類型最適合重視效能的正式作業工作負載。CVS效能服務類型是一種硬體選項、可支援最小100 GiB大小的磁碟區。您可以選擇"[三種服務層級](#)"下列其中一項：
 - `standard`
 - `premium`
 - `extreme`
- *** CVS**：CVS服務類型提供高分區可用度、但效能等級僅限於中度。CVS服務類型是一種軟體選項、使用儲存資源池來支援小至1 GiB的磁碟區。儲存資源池最多可包含50個磁碟區、其中所有磁碟區都會共用資源池的容量和效能。您可以選擇"[兩種服務層級](#)"下列其中一項：
 - `standardsw`
 - `zoneredundantstandardsw`

您需要的產品

若要設定及使用 "[適用於 Google Cloud Cloud Volumes Service](#)"後端、您需要下列項目：

- Google Cloud帳戶已設定NetApp Cloud Volumes Service 功能
- Google Cloud帳戶的專案編號

- 具有此角色的 Google Cloud Service 帳戶 `netappcloudvolumes.admin`
- API金鑰檔案、供Cloud Volumes Service 您的I方面 帳戶使用

後端組態選項

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。

參數	說明	預設
<code>version</code>		永遠為1
<code>storageDriverName</code>	儲存驅動程式名稱	「GCP-CVS」
<code>backendName</code>	自訂名稱或儲存後端	驅動程式名稱+「_」+ API 金鑰的一部分
<code>storageClass</code>	用於指定CVS服務類型的選用參數。用於 <code>software`</code> 選擇 CVS 服務類型。否則，Astra Trident 將採用 <code>CVS-Performance</code> 服務類型 (<code>`hardware</code>)。	
<code>storagePools</code>	僅限CVS服務類型。選用參數、用於指定用於建立磁碟區的儲存資源池。	
<code>projectNumber</code>	Google Cloud帳戶專案編號。此值可在Google Cloud 入口網站首頁找到。	
<code>hostProjectNumber</code>	如果使用共享VPC網路、則為必要項目。在此案例中 <code>projectNumber`</code> 、是服務專案、 <code>`hostProjectNumber`</code> 也是主機專案。	
<code>apiRegion</code>	Astra Trident在Google Cloud區域建立Cloud Volumes Service 了各種不全的功能。建立跨區域 Kubernetes 叢集時、在中建立的磁碟區 <code>`apiRegion`</code> 可用於跨多個 Google Cloud 區域的節點上排程的工作負載。跨區域流量會產生額外成本。	
<code>apiKey</code>	具有此角色的 Google Cloud 服務帳戶的 API 金鑰 <code>netappcloudvolumes.admin</code> 。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。	
<code>proxyURL</code>	Proxy URL (如果需要Proxy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。	
<code>nfsMountOptions</code>	精細控制NFS掛載選項。	"nfsves=3"
<code>limitVolumeSize</code>	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)
<code>serviceLevel</code>	適用於新磁碟區的CVS效能或CVS服務層級。CVS-Performance 值包括 <code>standard`</code> 、 <code>premium`</code> 或 <code>`extreme</code> 。CVS 值為 <code>standardsw`</code> 或 <code>`zoneredundantstandardsw</code> 。	CVS效能預設為「標準」。CVS預設為「標準」。
<code>network</code>	Google Cloud網路用於Cloud Volumes Service 解決資料不整的問題。	"預設"

參數	說明	預設
debugTraceFlags	疑難排解時要使用的偵錯旗標。範例： \{"api":false, "method":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
allowedTopologies	若要啟用跨區域存取、的 StorageClass 定義 allowedTopologies`必須包含所有地區。例如： `- key: topology.kubernetes.io/region values: - us-east1 - europe-west1	

Volume資源配置選項

您可以在組態檔案的區段中控制預設的 Volume 資源配置 defaults。

參數	說明	預設
exportRule	新磁碟區的匯出規則。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。	「0.0.0.0/0」
snapshotDir	存取`.snapshot`目錄	"假"
snapshotReserve	保留給快照的磁碟區百分比	"" (接受CVS預設值為0)
size	新磁碟區的大小。CVS效能最低為100 GiB。CVS最低為1 GiB。	CVS效能服務類型預設為「100GiB」。CVS服務類型並未設定預設值、但至少需要1 GiB。

CVS效能服務類型範例

下列範例提供CVS效能服務類型的範例組態。

範例1：最低組態

這是使用預設「標準」服務層級的預設CVS效能服務類型的最低後端組態。

```
---
version: 1
storageDriverName: gcp-cvs
projectNumber: '012345678901'
apiRegion: us-west2
apiKey:
  type: service_account
  project_id: my-gcp-project
  private_key_id: "<id_value>"
  private_key: |
    -----BEGIN PRIVATE KEY-----
    <key_value>
    -----END PRIVATE KEY-----
  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：服務層級組態

本範例說明後端組態選項、包括服務層級和Volume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: "<id_value>"
  private_key: |
    -----BEGIN PRIVATE KEY-----
    <key_value>
    -----END PRIVATE KEY-----
  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
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 3：虛擬集區組態

此範例使用 `storage` 來設定虛擬集區、以及 `StorageClasses` 可重新參照它們的。請參閱[\[儲存類別定義\]](#)以瞭解儲存類別的定義方式。

此處會針對所有虛擬集區設定特定的預設值、將設為 5%、將設 `snapshotReserve` 為 `exportRule 0.0.0.0/0`。虛擬集區是在一節中定義 `storage`。每個個別虛擬集區都會自行定義 `serviceLevel`、有些集區會覆寫預設值。虛擬池標籤用於根據和 `protection` 區分池 `performance`。

```
---
version: 1
storageDriverName: gcp-cvs
projectNumber: '012345678901'
apiRegion: us-west2
apiKey:
  type: service_account
  project_id: my-gcp-project
  private_key_id: "<id_value>"
  private_key: |
    -----BEGIN PRIVATE KEY-----
    <key_value>
    -----END PRIVATE KEY-----
  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
```

儲存類別定義

下列StorageClass定義適用於虛擬集區組態範例。使用 `parameters.selector` 時、您可以為每個 StorageClass 指定用於主控磁碟區的虛擬集區。該磁碟區會在所選的資源池中定義各個層面。

儲存類別範例

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

```
allowVolumeExpansion: true
---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: cvs-extra-protection
provisioner: csi.trident.netapp.io
parameters:
  selector: "protection=extra"
allowVolumeExpansion: true
```

- First StorageClass (cvs-extreme-extra-protection) 映射到第一個虛擬池。這是唯一提供極致效能、快照保留率為10%的資源池。
- Last StorageClass (cvs-extra-protection (最後一個 StorageClass) 調用任何提供 10% 快照保留的儲存池。Astra Trident決定選取哪個虛擬集區、並確保符合快照保留需求。

CVS服務類型範例

下列範例提供CVS服務類型的範例組態。

範例1：最低組態

這是用來指定 CVS 服務類型和預設服務層級的 `standardsw` 最低後端組態 `storageClass`。

```
---
version: 1
storageDriverName: gcp-cvs
projectNumber: '012345678901'
storageClass: software
apiRegion: us-east4
apiKey:
  type: service_account
  project_id: my-gcp-project
  private_key_id: "<id_value>"
  private_key: |
    -----BEGIN PRIVATE KEY-----
    <key_value>
    -----END PRIVATE KEY-----
  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
serviceLevel: standardsw
```

範例2：儲存資源池組態

此後端組態範例用於 `storagePools` 設定儲存池。

```
---
version: 1
storageDriverName: gcp-cvs
backendName: gcp-std-so-with-pool
projectNumber: '531265380079'
apiRegion: europe-west1
apiKey:
  type: service_account
  project_id: cloud-native-data
  private_key_id: "<id_value>"
  private_key: |-
    -----BEGIN PRIVATE KEY-----
    <key_value>
    -----END PRIVATE KEY-----
  client_email: cloudvolumes-admin-sa@cloud-native-
data.iam.gserviceaccount.com
  client_id: '107071413297115343396'
  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%40cloud-native-data.iam.gserviceaccount.com
storageClass: software
zone: europe-west1-b
network: default
storagePools:
- 1bc7f380-3314-6005-45e9-c7dc8c2d7509
serviceLevel: Standardsw
```

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 Astra Trident 安裝中建立和使用元素後端。

元素驅動程式詳細資料

Astra Trident 提供 `solidfire-san` 儲存驅動程式來與叢集通訊。支援的存取模式包括：`ReadWriteOnce`（`rwo`）、`ReadOnlyMany`（`ROX`）、`_ReadWriteMany`（`rwx`）、`_ReadWriteOncePod`（`RWOP`）。

`solidfire-san` 儲存驅動程式支援 `_file_` 和 `_block_` 磁碟區模式。對於 `Filesystem` `volemode`、Astra Trident 會建立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
solidfire-san	iSCSI	區塊	Rwo、ROX、rwx、RWOP	無檔案系統。原始區塊裝置。
solidfire-san	iSCSI	檔案系統	RWO、RWOP	xfs ext3、ext4

開始之前

在建立元素後端之前、您需要下列項目。

- 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"](#)。

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
version		永遠為1
storageDriverName	儲存驅動程式名稱	永遠是「solidfire-san」
backendName	自訂名稱或儲存後端	「S指_」+儲存設備（iSCSI）IP位址SolidFire

參數	說明	預設
Endpoint	MVIP、適用於SolidFire 採用租戶認證的不含用戶身分證明的叢集	
SVIP	儲存設備 (iSCSI) IP位址和連接埠	
labels	套用到磁碟區的任意JSON-格式化標籤集。	「」
TenantName	要使用的租戶名稱 (如果找不到、請建立)	
InitiatorIFace	將iSCSI流量限制在特定的主機介面	「預設」
UseCHAP	使用 CHAP 驗證 iSCSI。Astra Trident 使用 CHAP。	是的
AccessGroups	要使用的存取群組ID清單	尋找名為「Trident」的存取群組ID
Types	QoS規格	
limitVolumeSize	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。

範例 1：具有三種磁碟區類型的驅動程式後端組態 solidfire-san

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。您很可能會使用儲存類別參數來定義儲存類別、以使用其中的每個 `IOPS` 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：具有虛擬集區之驅動程式的後端和儲存類別組態 solidfire-san

此範例顯示使用虛擬資源池設定的後端定義檔、以及參照這些資源池的StorageClass。

Astra Trident會在資源配置時、將儲存資源池上的標籤複製到後端儲存LUN。為了方便起見、儲存管理員可以針對每個虛擬資源池定義標籤、並依標籤將磁碟區分組。

在下面顯示的範例後端定義檔中、會針對所有儲存池設定特定的預設值、這些儲存池會將設 type 為 Silver。虛擬集區是在一節中定義 storage。在此範例中、有些儲存資源池會自行設定類型、有些資源池則會覆寫上述預設值。

```

---
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定義是指上述虛擬資源池。使用此 `parameters.selector` 欄位、每個 StorageClass 都會呼叫哪些虛擬集區可用於主控磁碟區。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 (solidfire-gold-four) 將映射到第一個虛擬池。這是唯一提供黃金級效能的集區

Volume Type QoS ◦ Last StorageClass (solidfire-silver (最後一個 StorageClass) 調用任何提供銀牌性能的存儲池。Astra Trident將決定選取哪個虛擬集區、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"](#)

支援SAN驅動程式ONTAP

ONTAP SAN 驅動程式概觀

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

ONTAP SAN 驅動程式詳細資料

Astra Trident 提供下列 SAN 儲存驅動程式、可與 ONTAP 叢集通訊。支援的存取模式包括：*ReadWriteOnce* (rwo)、*ReadOnlyMany* (ROX)、*_ReadWriteMany* (rwx)、*_ReadWriteOncePod* (RWOP)。



如果您使用 Astra Control 來保護、恢復和移動、請參閱[Astra Control 驅動程式相容性](#)。

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
ontap-san	iSCSI	區塊	Rwo、ROX、rwx、RWOP	無檔案系統；原始區塊裝置
ontap-san	iSCSI	檔案系統	RWO、RWOP 檔案系統磁碟區模式中無法使用 Rox 和 rwx。	xfs ext3、ext4
ontap-san	NVMe / TCP 請參閱 NVMe / TCP 的其他考量事項 。	區塊	Rwo、ROX、rwx、RWOP	無檔案系統；原始區塊裝置
ontap-san	NVMe / TCP 請參閱 NVMe / TCP 的其他考量事項 。	檔案系統	RWO、RWOP 檔案系統磁碟區模式中無法使用 Rox 和 rwx。	xfs ext3、ext4
ontap-san-economy	iSCSI	區塊	Rwo、ROX、rwx、RWOP	無檔案系統；原始區塊裝置

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
ontap-san-economy	iSCSI	檔案系統	RWO、RWOP 檔案系統磁碟區模式中無法使用 Rox 和 rwx。	xfs ext3、ext4

Astra Control 驅動程式相容性

Astra Control 可針對使用、和 `ontap-san`` 驅動程式建立的磁碟區、``ontap-nas-flexgroup`` 提供無縫保護、災難恢復和行動性（在 Kubernetes 叢集之間移動磁碟區）``ontap-nas``。如 "[Astra Control 複寫先決條件](#)" 需詳細資訊、請參閱。



- ``ontap-san-economy`` 僅當持續容量使用量計數預期高於 "[支援的 ONTAP Volume 限制](#)" 時才使用。
- 僅在持續磁碟區使用量計數預期高於且 `ontap-san-economy`` 無法使用驅動程式時才 "[支援的 ONTAP Volume 限制](#)" 使用 ``ontap-nas-economy``。
- 如果您預期需要資料保護、災難恢復或行動力、請勿使用 `ontap-nas-economy``。

使用者權限

Astra Trident 預期會以 ONTAP 或 SVM 管理員的身分執行、通常使用叢集使用者或 `vsadmin`` SVM 使用者、或是使用 `admin`` 具有相同角色的不同名稱的使用者。對於用於 NetApp ONTAP 部署的 Amazon FSX、Astra Trident 預期會以 ONTAP 或 SVM 管理員的身分、使用叢集使用者或 ``vsadmin`` SVM 使用者、或是具有相同角色的不同名稱的使用者執行 `fsxadmin``。``fsxadmin`` 使用者只能有限地取代叢集管理使用者。



如果您使用此 `limitAggregateUsage`` 參數、則需要叢集管理權限。搭配 Astra Trident 使用 Amazon FSX for NetApp ONTAP 時、此參數將無法搭配 ``vsadmin`` 和 ``fsxadmin`` 使用者帳戶使用 ``limitAggregateUsage``。如果您指定此參數、組態作業將會失敗。

雖然可以在 ONTAP 中建立更具限制性的角色、讓 Trident 驅動程式可以使用、但我們不建議這樣做。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。

NVMe / TCP 的其他考量事項

Astra Trident 支援使用驅動程式的非揮發性記憶體高速（NVMe）傳輸協定 `ontap-san``、包括：

- IPv6
- NVMe 磁碟區的快照和複本
- 調整 NVMe 磁碟區大小
- 匯入 Astra Trident 外部建立的 NVMe Volume、以便 Astra Trident 管理其生命週期
- NVMe 原生多重路徑
- K8s 節點正常或不正常關機（24.06）

Astra Trident 不支援：

- NVMe 原生支援的 DH-HMAC-CHAP
- 裝置對應工具（DM）多重路徑
- LUKS加密

準備使用ONTAP 不完善的SAN驅動程式來設定後端

瞭解使用 ONTAP SAN 驅動程式設定 ONTAP 後端的需求和驗證選項。

需求

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 `san-dev`、使用驅動程式的類別 `ontap-san`、以及 `san-default`、使用該類別的類別 `ontap-san-economy`。

您所有的Kubernetes工作節點都必須安裝適當的iSCSI工具。如 "[準備工作節點](#)" 需詳細資訊、請參閱。

驗證 ONTAP 後端

Astra Trident提供兩種驗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。建議您使用預先定義的安全登入角色、例如 `admin` 或 `vsadmin` 以確保與 ONTAP 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。

您可以更新現有的後端、以便在認證型和憑證型方法之間移動。不過、一次只支援一種驗證方法。若要切換至不同的驗證方法、您必須從後端組態中移除現有方法。



如果您嘗試同時提供*認證與認證*、後端建立將會失敗、並在組態檔中提供多種驗證方法。

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。建議您使用標準的預先定義角色、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：

YAML

```
---
version: 1
backendName: ExampleBackend
storageDriverName: ontap-san
managementLIF: 10.0.0.1
svm: svm_nfs
username: vsadmin
password: password
```

JSON

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

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。建立或更新後端是唯一需要具備認證知識的步驟。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 安全登入角色支援 `cert` 驗證方法。

```
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",
  "svm": "vserver_test",
  "clientCertificate": "Faaaakkkkeeee...Vaaalllluuueeee",
  "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、您必須移除現有的驗證方法、然後新增驗證方法。然後使用包含執行所需參數的更新後端 .json 檔案 `tridentctl backend update`。

```

cat cert-backend-updated.json
{
"version": 1,
"storageDriverName": "ontap-san",
"backendName": "SanBackend",
"managementLIF": "1.2.3.4",
"svm": "vserver_test",
"username": "vsadmin",
"password": "password",
"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。

使用雙向CHAP驗證連線

Astra Trident 可以使用和 `ontap-san-economy` 驅動程式的雙向 CHAP 驗證 iSCSI 工作階段 `ontap-san`。這需要在後端定義中啟用 `useCHAP` 選項。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密碼。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: password
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。

此外、`chapInitiatorSecret`、`chapTargetInitiatorSecret`、`chapTargetUsername` 和 `chapUsername` 欄位也 `useCHAP=true` 必須包含在後端定義中。通過運行創建後端後，可以更改機密 `tridentctl update`。

運作方式

儲存管理員會將設定 `useCHAP` 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（預設為「無」） * 且 * 磁碟區中沒有預先存在的 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密碼。
 - 如果SVM包含LUN、Astra Trident將不會在SVM上啟用CHAP。這可確保不限制對 SVM 上已存在的 LUN 的存取。
- 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）。

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

您可以更新檔案中的 CHAP 參數來更新 CHAP 認證 `backend.json`。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": "password",
  "chapInitiatorSecret": "cl9qxUpDaTeD",
  "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。

列舉SAN組態選項與範例ONTAP

瞭解如何在 Astra Trident 安裝中建立及使用 ONTAP SAN 驅動程式。本節提供後端組態範例及將後端對應至 StorageClasses 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
version		永遠為1

參數	說明	預設
storageDrive rName	儲存驅動程式名稱	ontap-nas、ontap-nas- economy ontap-nas-flexgroup ontap-san、ontap-san- economy
backendName	自訂名稱或儲存後端	驅動程式名稱 + "_" + dataLIF
managementLI F	叢集或 SVM 管理 LIF 的 IP 位址。您可以指定完整網 域名稱 (FQDN)。如果使用 IPv6 旗標安裝 Astra Trident、則可以設定為使用 IPv6 位址。IPv6 位址必 須以方括弧定義，例如 [28e8:d9fb:a825:b7bf:69a8:d02f:9e7b:3555]。如需無縫 MetroCluster 之間的互通性[mcc-best]、 請參閱。	「10.0.0.1」、 「[2001:1234:abcd::: fefo]」
dataLIF	傳輸協定LIF的IP位址。* 請勿指定 iSCSI。 Astra Trident 使用"可選擇的LUN對應ONTAP"來探索建立多 重路徑工作階段所需的 iSCSI 生命。如果明確定義、就 會產生警告 dataLIF 。 MetroCluster 省略。 *請參 閱[mcc-best]。	源自SVM
svm	儲存虛擬機器使用 * 略過 MetroCluster。 *請參 閱[mcc-best]。	如果指定 SVM 則衍生 managementLIF
useCHAP	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]。設為 true for Astra Trident、以 設定和使用雙向 CHAP 做為後端所指定 SVM 的預設驗 證。如 "準備使用ONTAP 不完善的SAN驅動程式來設 定後端" 需詳細資訊、請參閱。	false
chapInitiato rSecret	CHAP啟動器密碼。必要條件 useCHAP=true	"
labels	套用到磁碟區的任意JSON-格式化標籤集	"
chapTargetIn itiatorSecre t	CHAP目標啟動器機密。必要條件 useCHAP=true	"
chapUsername	傳入使用者名稱。必要條件 useCHAP=true	"
chapTargetUs ername	目標使用者名稱。必要條件 useCHAP=true	"
clientCertif icate	用戶端憑證的Base64編碼值。用於憑證型驗證	"
clientPrivat eKey	用戶端私密金鑰的Base64編碼值。用於憑證型驗證	"
trustedCACer tificate	受信任CA憑證的Base64編碼值。選用。用於憑證型驗 證。	"
username	與ONTAP 該叢集通訊所需的使用者名稱。用於認證型 驗證。	"
password	與ONTAP 該叢集通訊所需的密碼。用於認證型驗證。	"

參數	說明	預設
svm	要使用的儲存虛擬機器	如果指定 SVM 則衍生 managementLIF
storagePrefix	在SVM中配置新磁碟區時所使用的前置碼。稍後無法修改。若要更新此參數、您需要建立新的後端。	trident
limitAggregateUsage	如果使用率高於此百分比、則無法進行資源配置。如果您使用 Amazon FSX for NetApp ONTAP 後端、請勿指定 limitAggregateUsage。提供的 `fsxadmin` 和 `vsadmin` 不包含擷取 Aggregate 使用量並使用 Astra Trident 加以限制所需的權限。	"" (預設不強制執行)
limitVolumeSize	如果要求的磁碟區大小高於此值、則資源配置失敗。也會限制其管理的qtree和LUN磁碟區大小上限。	"" (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。例如、除非您正在進行疑難排解並需要詳細的記錄傾印、否則 {"api" : false、"method" : true} 不使用。	null
useREST	使用ONTAP Isrest API的布林參數。 useREST 設為 true`時、 Astra Trident 將使用 ONTAP REST API 與後端通訊；設為 `false`時、 Astra Trident 將使用 ONTAP ZAPI 呼叫與後端通訊。此功能需要ONTAP 使用更新版本的版本。此外、使用的 ONTAP 登入角色必須具有應用程式存取權 `ontap`。這是預先定義的和角色所滿足 vsadmin cluster-admin 的。從 Astra Trident 24.06 版本開始、 ONTAP 9.15.1 或更新版本 useREST 預設為 true ；變更 useREST 為 false 使用 ONTAP ZAPI 呼叫。 useREST 完全符合 NVMe / TCP 的資格。	true 對於 ONTAP 9.15.1 或更高版本，否則 false。
sanType	用於選擇 iscsi iSCSI 或 nvme NVMe / TCP 。	`iscsi` 如果空白

用於資源配置磁碟區的后端組態選項

您可以使用組態區段中的這些選項來控制預設資源配置 defaults。如需範例、請參閱下列組態範例。

參數	說明	預設
spaceAllocation	LUN的空間分配	"對"
spaceReserve	空間保留模式；「無」（精簡）或「Volume」（粗）	"無"
snapshotPolicy	要使用的Snapshot原則	"無"

參數	說明	預設
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。搭配Astra Trident使用QoS原則群組需要ONTAP 使用更新版本的版本。我們建議使用非共用的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。	"
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	"
snapshotReserve	保留給快照的磁碟區百分比	如果為「無」、則為「0 snapshotPolicy」、否則為「」
splitOnClone	建立複本時、從其父複本分割複本	"假"
encryption	在新磁碟區上啟用 NetApp Volume Encryption (NVE)；預設為 false。必須在叢集上授權並啟用NVE、才能使用此選項。如果在後端啟用NAE、則Astra Trident中配置的任何磁碟區都會啟用NAE。如需更多資訊、請參閱" Astra Trident如何與NVE和NAE搭配運作 "：。	"假"
luksEncryption	啟用LUKS加密。請參閱 " 使用Linux統一金鑰設定 (LUKS) "。NVMe / TCP 不支援 LUKS 加密。	"
securityStyle	新磁碟區的安全樣式	unix
tieringPolicy	分層原則以使用「無」	「僅限快照」適用於 ONTAP 9.5 之前的 SVM-DR 組態
nameTemplate	建立自訂磁碟區名稱的範本。	"
limitVolumePoolSize	在 ONTAP SAN 經濟型後端中使用 LUN 時、可要求的最大 FlexVol 大小。	"" (預設不強制執行)

Volume資源配置範例

以下是定義預設值的範例：

```

---
version: 1
storageDriverName: ontap-san
managementLIF: 10.0.0.1
svm: trident_svm
username: admin
password: <password>
labels:
  k8scluster: dev2
  backend: dev2-sanbackend
storagePrefix: alternate-trident
debugTraceFlags:
  api: false
  method: true
defaults:
  spaceReserve: volume
  qosPolicy: standard
  spaceAllocation: 'false'
  snapshotPolicy: default
  snapshotReserve: '10'

```



對於使用驅動程式建立的所有磁碟區 ontap-san、Astra Trident 會為 FlexVol 額外增加 10% 的容量、以容納 LUN 中繼資料。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。

對於定義的後端 snapshotReserve、Astra Trident 會依照下列公式計算體積大小：

$$\text{Total volume size} = [(\text{PVC requested size}) / (1 - (\text{snapshotReserve percentage} / 100))] * 1.1$$

1.1 是額外 10% 的 Astra Trident 加入 FlexVol 到支援 LUN 中繼資料的功能。若 snapshotReserve = 5%、且 PVC 要求 = 5GiB、則總 Volume 大小為 5.79GiB、可用大小為 5.50GiB。`volume show` 命令應顯示類似於此範例的結果：

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 上搭配 Astra Trident 使用 Amazon FSX、建議您指定生命的 DNS 名稱、而非 IP 位址。

ONTAP SAN 範例

這是使用驅動程式的基本組態 `ontap-san`。

```
---
version: 1
storageDriverName: ontap-san
managementLIF: 10.0.0.1
svm: svm_iscsi
labels:
  k8scluster: test-cluster-1
  backend: testcluster1-sanbackend
username: vsadmin
password: <password>
```

ONTAP SAN 經濟效益範例

```
---
version: 1
storageDriverName: ontap-san-economy
managementLIF: 10.0.0.1
svm: svm_iscsi_eco
username: vsadmin
password: <password>
```

1. 範例

您可以設定後端"SVM 複寫與還原"、以避免在切換後手動更新後端定義、並在期間切換。

若要無縫切換和切換、請使用並省略 `dataLIF`` 和 ``svm`` 參數來指定 `SVM `managementLIF``。例如：

```
---
version: 1
storageDriverName: ontap-san
managementLIF: 192.168.1.66
username: vsadmin
password: password
```

憑證型驗證範例

在這個基本組態範例中 `clientCertificate``、`clientPrivateKey`` 和 ``trustedCACertificate`` (如果使用信任的 CA、則為選用) 會分別填入 ``backend.json`` 用戶端憑證、私密金鑰和信任的 CA 憑證的 base64 編碼值。

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

雙向 CHAP 範例

這些範例會建立後端、並 useCHAP 將設為 true。

ONTAP SAN CHAP 範例

```
---
version: 1
storageDriverName: ontap-san
managementLIF: 10.0.0.1
svm: svm_iscsi
labels:
  k8scluster: test-cluster-1
  backend: testcluster1-sanbackend
useCHAP: true
chapInitiatorSecret: cl9qxIm36DKyawxy
chapTargetInitiatorSecret: rqxigXgkesIpwxyz
chapTargetUsername: iJF4heBRT0TCwxyz
chapUsername: uh2aNCLSD6cNwxyz
username: vsadmin
password: <password>
```

ONTAP SAN 經濟 CHAP 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
username: vsadmin
password: <password>
```

NVMe / TCP 範例

您必須在 ONTAP 後端上設定 NVMe 的 SVM 。這是適用於 NVMe / TCP 的基本後端組態。

```
---
version: 1
backendName: NVMeBackend
storageDriverName: ontap-san
managementLIF: 10.0.0.1
svm: svm_nvme
username: vsadmin
password: password
sanType: nvme
useREST: true
```

名稱範本的后端組態範例

```
---
version: 1
storageDriverName: ontap-san
backendName: ontap-san-backend
managementLIF: <ip address>
svm: svm0
username: <admin>
password: <password>
defaults: {
  "nameTemplate":
    "{{.volume.Name}}_{{.labels.cluster}}_{{.volume.Namespace}}_{{.volume.RequestName}}"
},
"labels": {"cluster": "ClusterA", "PVC":
  "{{.volume.Namespace}}_{{.volume.RequestName}}"}
}
```

虛擬集區的后端範例

在這些後端定義檔範例中、會針對所有儲存池設定特定的預設值、例如 `spaceReserve` 「無」、`spaceAllocation` 「假」和 `encryption` 「假」。虛擬資源池是在儲存區段中定義的。

Astra Trident 會在「意見」欄位中設定資源配置標籤。請在 FlexVol The 過程中提出意見。Astra Trident 會在資源配置時、將虛擬資源池上的所有標籤複製到儲存磁碟區。為了方便起見、儲存管理員可以針對每個虛擬資源池定義標籤、並依標籤將磁碟區分組。

在這些範例中、有些儲存資源池會自行設定 `spaceReserve`、和 `encryption` 值、`spaceAllocation` 有些資源池則會覆寫預設值。



```
---
version: 1
storageDriverName: ontap-san
managementLIF: 10.0.0.1
svm: svm_iscsi
useCHAP: true
chapInitiatorSecret: cl9qxIm36DKyawxy
chapTargetInitiatorSecret: rqxigXgkesIpwxyz
chapTargetUsername: iJF4heBRT0TCwxyz
chapUsername: uh2aNCLSD6cNwxyz
username: vsadmin
password: <password>
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'
```

```

---
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
username: vsadmin
password: <password>
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'
- labels:
  department: legal
  creditpoints: '5000'

```

```
zone: us_east_1c
defaults:
  spaceAllocation: 'true'
  encryption: 'false'
```

NVMe / TCP 範例

```
---
version: 1
storageDriverName: ontap-san
sanType: nvme
managementLIF: 10.0.0.1
svm: nvme_svm
username: vsadmin
password: <password>
useREST: true
defaults:
  spaceAllocation: 'false'
  encryption: 'true'
storage:
- labels:
  app: testApp
  cost: '20'
  defaults:
    spaceAllocation: 'false'
    encryption: 'false'
```

將後端對應至StorageClass

以下 StorageClass 定義請參閱[\[虛擬集區的后端範例\]](#)。使用此 `parameters.selector` 欄位、每個 StorageClass 都會呼叫哪些虛擬集區可用於主控磁碟區。磁碟區將會在所選的虛擬資源池中定義各個層面。

- `protection-gold` StorageClass 會對應至後端的第一個虛擬集區 `ontap-san`。這是唯一提供金級保護的集區。

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: protection-gold
provisioner: csi.trident.netapp.io
parameters:
  selector: "protection=gold"
  fsType: "ext4"
```

- protection-not-gold`StorageClass 會對應至後端的第二個和第三個虛擬集區 `ontap-san。這是唯一提供金級以外保護層級的集區。

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: protection-not-gold
provisioner: csi.trident.netapp.io
parameters:
  selector: "protection!=gold"
  fsType: "ext4"
```

- app-mysqldb`StorageClass 會對應至後端的第三個虛擬集區 `ontap-san-economy。這是唯一為mysqldb 類型應用程式提供儲存池組態的集區。

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: app-mysqldb
provisioner: csi.trident.netapp.io
parameters:
  selector: "app=mysqldb"
  fsType: "ext4"
```

- protection-silver-creditpoints-20k`StorageClass 會對應至後端的第二個虛擬集區 `ontap-san。這是唯一提供銀級保護和 20000 個信用點數的資源池。

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: protection-silver-creditpoints-20k
provisioner: csi.trident.netapp.io
parameters:
  selector: "protection=silver; creditpoints=20000"
  fsType: "ext4"
```

- creditpoints-5k StorageClass 會對應至後端的第三個虛擬集區、以及後端的第 `ontap-san` 四個虛擬集區 `ontap-san-economy`。這是唯一擁有 5000 個信用點數的集區方案。

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: creditpoints-5k
provisioner: csi.trident.netapp.io
parameters:
  selector: "creditpoints=5000"
  fsType: "ext4"
```

- my-test-app-sc StorageClass 將使用映射到 `testAPP` 驅動程序 `sanType: nvme` 中的虛擬池 `ontap-san`。這是唯一提供的資源池 testApp。

```
---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: my-test-app-sc
provisioner: csi.trident.netapp.io
parameters:
  selector: "app=testApp"
  fsType: "ext4"
```

Astra Trident將決定選取哪個虛擬集區、並確保符合儲存需求。

ONTAP NAS 驅動程式

ONTAP NAS 驅動程式概觀

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

ONTAP NAS 驅動程式詳細資料

Astra Trident 提供下列 NAS 儲存驅動程式、可與 ONTAP 叢集通訊。支援的存取模式包括：*ReadWriteOnce*（*rwo*）、*ReadOnlyMany*（*ROX*）、*_ReadWriteMany*（*rwX*）、*_ReadWriteOncePod*（*RWOP*）。



如果您使用 Astra Control 來保護、恢復和移動、請參閱[Astra Control 驅動程式相容性](#)。

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
ontap-nas	NFS SMB	檔案系統	Rwo、ROX、rwX、RWOP	"nfs、smb"
ontap-nas-economy	NFS SMB	檔案系統	Rwo、ROX、rwX、RWOP	"nfs、smb"
ontap-nas-flexgroup	NFS SMB	檔案系統	Rwo、ROX、rwX、RWOP	"nfs、smb"

Astra Control 驅動程式相容性

Astra Control 可針對使用、和 `ontap-san`` 驅動程式建立的磁碟區、`ontap-nas-flexgroup`` 提供無縫保護、災難恢復和行動性（在 Kubernetes 叢集之間移動磁碟區）`ontap-nas``。如 "[Astra Control 複寫先決條件](#)" 需詳細資訊、請參閱。



- `ontap-san-economy`` 僅當持續容量使用量計數預期高於 "[支援的 ONTAP Volume 限制](#)" 時才使用。
- 僅在持續磁碟區使用量計數預期高於且 `ontap-san-economy`` 無法使用驅動程式時才 "[支援的 ONTAP Volume 限制](#)" 使用 `ontap-nas-economy``。
- 如果您預期需要資料保護、災難恢復或行動力、請勿使用 `ontap-nas-economy``。

使用者權限

Astra Trident 預期會以 ONTAP 或 SVM 管理員的身分執行、通常使用叢集使用者或 `vsadmin` SVM 使用者、或是使用 `admin`` 具有相同角色的不同名稱的使用者。

對於用於 NetApp ONTAP 部署的 Amazon FSX、Astra Trident 預期會以 ONTAP 或 SVM 管理員的身分、使用叢集使用者或 `vsadmin` SVM 使用者、或是具有相同角色的不同名稱的使用者執行 `fsxadmin``。`fsxadmin`` 使用者只能有限地取代叢集管理使用者。



如果您使用此 `limitAggregateUsage`` 參數、則需要叢集管理權限。搭配 Astra Trident 使用 Amazon FSX for NetApp ONTAP 時、此參數將無法搭配 `vsadmin`` 和 `fsxadmin`` 使用者帳戶使用 `limitAggregateUsage``。如果您指定此參數、組態作業將會失敗。

雖然可以在 ONTAP 中建立更具限制性的角色、讓 Trident 驅動程式可以使用、但我們不建議這樣做。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。

準備使用ONTAP 不含NAS的驅動程式來設定後端

瞭解使用 ONTAP NAS 驅動程式設定 ONTAP 後端的需求、驗證選項和匯出原則。

需求

- 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。
- 您可以執行多個驅動程式、並建立指向其中一個或另一個的儲存類別。例如、您可以設定使用驅動程式的 Gold 類別 `ontap-nas`、以及使用該類別的 Bronze 類別 `ontap-nas-economy`。
- 您所有的Kubernetes工作節點都必須安裝適當的NFS工具。如["請按這裡"](#)需詳細資訊、請參閱。
- Astra Trident僅支援安裝在Windows節點上執行的Pod上的SMB磁碟區。如 [準備配置SMB磁碟區](#) 需詳細資訊、請參閱。

驗證 ONTAP 後端

Astra Trident提供兩種驗證ONTAP 證功能來驗證支援的後端。

- 認證型：此模式需要對 ONTAP 後端擁有足夠的權限。建議您使用與預先定義的安全登入角色相關聯的帳戶、例如 ``admin``或、``vsadmin``以確保與 ONTAP 版本的最大相容性。
- 憑證型：此模式需要在後端安裝憑證、Astra Trident 才能與 ONTAP 叢集通訊。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。

您可以更新現有的後端、以便在認證型和憑證型方法之間移動。不過、一次只支援一種驗證方法。若要切換至不同的驗證方法、您必須從後端組態中移除現有方法。



如果您嘗試同時提供*認證與憑證*、後端建立將會失敗、並在組態檔中提供多種驗證方法。

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。建議您使用標準的預先定義角色、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：

YAML

```
---
version: 1
backendName: ExampleBackend
storageDriverName: ontap-nas
managementLIF: 10.0.0.1
dataLIF: 10.0.0.2
svm: svm_nfs
username: vsadmin
password: password
```

JSON

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

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 安全登入角色支援 `cert` 驗證方法。

```
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 default-data-management。

```
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...Vaaalllluuueeee",
  "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、您必須移除現有的驗證方法、然後新增驗證方法。然後使用包含執行所需參數的更新後端 .json 檔案 `tridentctl update backend`。

```

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": "password",
"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。

Astra Trident 提供動態管理 ONTAP 後端匯出原則的能力。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、這有助於限制只有在指定範圍內有IP的工作者節點才能存取儲存叢集、以支援精細且自動化的管理。



使用動態匯出原則時、請勿使用網路位址轉譯（NAT）。使用 NAT 時、儲存控制器會看到前端 NAT 位址、而非實際 IP 主機位址、因此在匯出規則中找不到相符項目時、就會拒絕存取。

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: password
autoExportCIDRs:
- 192.168.0.0/24
autoExportPolicy: true
```



使用此功能時、您必須確保SVM中的根連接點具有先前建立的匯出原則、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、將 SVM 專門用於 Astra Trident 。

以下是使用上述範例說明此功能的運作方式：

- `autoExportPolicy`` 設定為 ``true``。這表示 Astra Trident 將為 SVM 建立匯出原則 `svm1`、並使用位址區塊處理規則的新增和刪除 `autoExportCIDRs`。例如、UUID 為 `403b5326-8482-40der-96d0-d83fb3f4daec` 的後端、並 `autoExportPolicy`` 設為 ``true`` 在 SVM 上建立名為的匯出原則 ``trident-403b5326-8482-40db-96d0-d83fb3f4daec``。
- ``autoExportCIDRs`` 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 `192.168.0.0/24`、會提供位址空間。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、它會擷取節點的 IP 位址、並對照中提供的位址區塊進行檢查 `autoExportCIDRs`。在篩選 IP 之後、Astra Trident 會為其探索到的用戶端 IP 建立匯出原則規則、並針對其識別的每個節點建立一個規則。

您可以在建立後更新 `autoExportPolicy`` 及 ``autoExportCIDRs`` 以供後端使用。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。您也可以選擇針對後端停用 ``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、您可以使用或對應的 `tridentbackend` CRD 來檢查後端 `tridentctl`：

```
./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（前提是它們位於為後端指定的位址範圍內 `autoExportCIDRs`）。

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、使用更新後端 `tridentctl update backend` 可確保 Astra Trident 自動管理匯出原則。如此將會建立以後端 UUID 命名的新匯出原則、並在重新掛載後端上的磁碟區時、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。

準備配置 SMB 磁碟區

只需稍加準備、您就能使用驅動程式來配置 SMB 磁碟區 `ontap-nas`。



您必須在 SVM 上同時設定 NFS 和 SMB/CIFS 通訊協定、才能為內部部署的 ONTAP 建立 `ontap-nas-economy` SMB 磁碟區。若未設定上述任一種通訊協定、將導致 SMB 磁碟區建立失敗。

開始之前

在配置 SMB 磁碟區之前、您必須具備下列項目。

- Kubernetes叢集具備Linux控制器節點、以及至少一個執行Windows Server 2022的Windows工作節點。Astra Trident僅支援安裝在Windows節點上執行的Pod上的SMB磁碟區。
- 至少有一個Astra Trident機密、其中包含您的Active Directory認證資料。產生機密 `smbcreds`：

```
kubectl create secret generic smbcreds --from-literal username=user  
--from-literal password='password'
```

- 設定為Windows服務的SCSI Proxy。若要設定 `csi-proxy`、請參閱或"[GitHub：適用於Windows的SCSI Proxy](#)"、瞭"[GitHub：csi Proxy](#)"解在 Windows 上執行的 Kubernetes 節點。

步驟

1. 對於內部部署 ONTAP、您可以選擇性地建立 SMB 共用、或是 Astra Trident 可以為您建立一個。



Amazon FSX for ONTAP 需要 SMB 共享。

您可以使用共用資料夾嵌入式管理單元或使用 ONTAP CLI、以兩種方式之一建立 SMB 管理員共用"[Microsoft管理主控台](#)"。若要使用ONTAP CLI建立SMB共用：

- a. 如有必要、請建立共用的目錄路徑結構。

命令會 `vserver cifs share create` 在建立共用時檢查 `-path` 選項中指定的路徑。如果指定的路徑不存在、則命令會失敗。

- b. 建立與指定SVM相關的SMB共用區：

```
vserver cifs share create -vserver vserver_name -share-name  
share_name -path path [-share-properties share_properties,...]  
[other_attributes] [-comment text]
```

- c. 確認共用區已建立：

```
vserver cifs share show -share-name share_name
```



如需完整詳細資料、請參閱"[建立SMB共用區](#)"。

2. 建立後端時、您必須設定下列項目以指定SMB Volume。有關 ONTAP 後端組態選項的所有 FSX "[FSX提供ONTAP 各種組態選項和範例](#)"、請參閱。

參數	說明	範例
smbShare	您可以指定下列其中一項：使用 Microsoft 管理主控台或 ONTAP CLI 建立的 SMB 共用名稱；允許 Astra Trident 建立 SMB 共用的名稱；或將參數保留空白以防止共用磁碟區存取。對於內部部署 ONTAP、此參數為選用項目。Amazon FSX 需要此參數才能支援 ONTAP 後端、且不可為空白。	smb-share
nasType	* 必須設定為 smb.*如果為 null，則默認為 nfs。	smb
securityStyle	新磁碟區的安全樣式。* 必須設定為 ntfs SMB Volume 或 mixed。*	ntfs`或`mixed SMB Volume
unixPermissions	新磁碟區的模式。SMB磁碟區*必須保留為空白。*	"

列舉NAS組態選項與範例ONTAP

瞭解如何在 Astra Trident 安裝中建立及使用 ONTAP NAS 驅動程式。本節提供後端組態範例及將後端對應至 StorageClasses 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
version		永遠為1
storageDriverName	儲存驅動程式名稱	「ONTAP - NAS」、「ONTAP - NAS - 經濟」、「ONTAP - NAS - Flexgroup」、「ONTAP - SAN」、「ONTAP - SAN 經濟」
backendName	自訂名稱或儲存後端	驅動程式名稱 + "_" + dataLIF
managementLIF	叢集或 SVM 管理 LIF 的 IP 位址可以指定完整網域名稱 (FQDN)。如果使用 IPv6 旗標安裝 Astra Trident、則可以設定為使用 IPv6 位址。IPv6 位址必須以方括弧定義，例如 [28e8:d9fb:a825:b7bf:69a8:d02f:9e7b:3555]。如需無縫 MetroCluster 之間的互通性[mcc-best]、請參閱。	「10.0.0.1」、「[2001:1234:abcd::fefe]」
dataLIF	傳輸協定LIF的IP位址。建議您指定 dataLIF。如果未提供、Astra Trident會從SVM擷取資料lifs。您可以指定要用於NFS掛載作業的完整網域名稱 (FQDN)、讓您建立循環配置資源DNS、以便在多個資料生命期之間達到負載平衡。可在初始設定之後變更。請參閱。如果使用 IPv6 旗標安裝 Astra Trident、則可以設定為使用 IPv6 位址。IPv6 位址必須以方括弧定義，例如 [28e8:d9fb:a825:b7bf:69a8:d02f:9e7b:3555]。* MetroCluster 省略。*請參閱[mcc-best]。	指定位址或從SVM衍生（若未指定）（不建議使用）

參數	說明	預設
svm	儲存虛擬機器使用 * 略過 MetroCluster 。 *請參閱[mcc-best] 。	如果指定 SVM 則衍生 managementLIF
autoExportPolicy	啟用自動匯出原則建立及更新[布林值]。Astra Trident 可使用 `autoExportPolicy` 和 `autoExportCIDRs` 選項、自動管理匯出原則。	錯
autoExportCIDRs	將 Kubernetes 節點 IP 篩選在啟用時的 CIDR 清單 autoExportPolicy。Astra Trident 可使用 `autoExportPolicy` 和 `autoExportCIDRs` 選項、自動管理匯出原則。	["0.0.0/0"、"/0"]
labels	套用到磁碟區的任意JSON-格式化標籤集	"
clientCertificate	用戶端憑證的Base64編碼值。用於憑證型驗證	"
clientPrivateKey	用戶端私密金鑰的Base64編碼值。用於憑證型驗證	"
trustedCACertificate	受信任CA憑證的Base64編碼值。選用。用於憑證型驗證	"
username	連線至叢集/ SVM的使用者名稱。用於認證型驗證	
password	連線至叢集/ SVM的密碼。用於認證型驗證	
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 "
limitAggregateUsage	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)
limitVolumeSize	如果要求的磁碟區大小高於此值、則資源配置失敗。也會限制其管理 qtree 和 LUN 的最大磁碟區大小、而且此 `qtreesPerFlexvol` 選項可讓您自訂每個 FlexVol 的最大 qtree 數量。	"" (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。例如、除非您正在進行疑難排解並需要詳細的記錄傾印、否則 { "api" : false、"method" : true} 不使用 debugTraceFlags。	null
nasType	設定NFS或SMB磁碟區建立。選項為 nfs、`smb` 或 null。NFS磁碟區的預設值設為null。	nfs
nfsMountOptions	以逗號分隔的NFS掛載選項清單。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、Astra Trident將不會在相關的持續磁碟區上設定任何掛載選項。	"
qtreesPerFlexvol	每FlexVol 個邊的最大qtree數、必須在範圍內[50、300]	"200"

參數	說明	預設
smbShare	您可以指定下列其中一項：使用 Microsoft 管理主控台或 ONTAP CLI 建立的 SMB 共用名稱；允許 Astra Trident 建立 SMB 共用的名稱；或將參數保留空白以防止共用磁碟區存取。對於內部部署 ONTAP、此參數為選用項目。Amazon FSX 需要此參數才能支援 ONTAP 後端、且不可為空白。	smb-share
useREST	使用ONTAP Isrest API的布林參數。useREST 設為 true`時、Astra Trident 將使用 ONTAP REST API 與後端通訊；設為 `false`時、Astra Trident 將使用 ONTAP ZAPI 呼叫與後端通訊。此功能需要ONTAP 使用更新版本的版本。此外、使用的 ONTAP 登入角色必須具有應用程式存取權 `ontap`。這是預先定義的和角色所滿足 vsadmin cluster-admin 的。從 Astra Trident 24.06 版本開始、ONTAP 9.15.1 或更新版本 useREST 預設為 true ；變更 useREST 為 false 使用 ONTAP ZAPI 呼叫。	true 對於 ONTAP 9.15.1 或更高版本，否則 false。
limitVolumePoolSize	在 ONTAP NAS 經濟型後端使用 qtree 時、可要求的 FlexVol 大小上限。	"" (預設不強制執行)

用於資源配置磁碟區的后端組態選項

您可以使用組態區段中的這些選項來控制預設資源配置 defaults。如需範例、請參閱下列組態範例。

參數	說明	預設
spaceAllocation	LUN的空間分配	"對"
spaceReserve	空間保留模式；「無」（精簡）或「Volume」（粗）	"無"
snapshotPolicy	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。	"
snapshotReserve	保留給快照的磁碟區百分比	如果為「無」、則為「0 snapshotPolicy」、否則為「」
splitOnClone	建立複本時、從其父複本分割複本	"假"
encryption	在新磁碟區上啟用 NetApp Volume Encryption (NVE) ；預設為 false。必須在叢集上授權並啟用NVE、才能使用此選項。如果在後端啟用NAE、則Astra Trident中配置的任何磁碟區都會啟用NAE。如需更多資訊、請參閱 "Astra Trident如何與NVE和NAE搭配運作" ：。	"假"

參數	說明	預設
tieringPolicy	分層原則以使用「無」	「僅限快照」適用於 ONTAP 9.5 之前的 SVM-DR 組態
unixPermissions	新磁碟區的模式	"777" 表示 NFS 磁碟區；SMB 磁碟區為空的（不適用）
snapshotDir	控制對目錄的存取 .snapshot	"假"
exportPolicy	要使用的匯出原則	"預設"
securityStyle	新磁碟區的安全樣式。NFS 支援 `mixed` 和 `unix` 安全樣式。SMB 支援 `mixed` 和 `ntfs` 安全樣式。	NFS 預設值為 unix。SMB 預設為 ntfs。
nameTemplate	建立自訂磁碟區名稱的範本。	"



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。

Volume 資源配置範例

以下是定義預設值的範例：

```

---
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 現在使用新的計算方式、確保 FlexVol 的大小正確、並使用 snapshotReserve 百分比和 PVC。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。這是因為使用者所要求的是整個 Volume、而且 snapshotReserve 是其中的百分比。使用 Trident 21.07 時、使用者要求的是可寫入空間、而 Astra Trident 則將該數量定義 snapshotReserve 為整個 Volume 的百分比。這不適用於 ontap-nas-economy。請參閱下列範例以瞭解此功能的運作方式：

計算方式如下：

```
Total volume size = (PVC requested size) / (1 - (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 命令應顯示類似於此範例的結果：

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、使用較早版本的 2GiB PVC 會產生一個提供 1GiB snapshotReserve=50 可寫入空間的 Volume。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 經濟效益範例

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

ONTAP NAS FlexGroup 範例

```
---  
version: 1  
storageDriverName: ontap-nas-flexgroup  
managementLIF: 10.0.0.1  
dataLIF: 10.0.0.2  
svm: svm_nfs  
username: vsadmin  
password: password
```

MetroCluster 範例

您可以設定後端"[SVM 複寫與還原](#)"、以避免在切換後手動更新後端定義、並在期間切換。

若要無縫切換和切換、請使用並省略 dataLIF 和 `svm` 參數來指定 SVM `managementLIF`。例如：

```
---  
version: 1  
storageDriverName: ontap-nas  
managementLIF: 192.168.1.66  
username: vsadmin  
password: password
```

SMB Volume 範例

```
---  
version: 1  
backendName: ExampleBackend  
storageDriverName: ontap-nas  
managementLIF: 10.0.0.1  
nasType: smb  
securityStyle: ntfs  
unixPermissions: ""  
dataLIF: 10.0.0.2  
svm: svm_nfs  
username: vsadmin  
password: password
```

憑證型驗證範例

這是最小的後端組態範例。`clientCertificate`和`trustedCACertificate`（如果使用信任的CA、`clientPrivateKey`則為選用）會分別填入`backend.json`用戶端憑證、私密金鑰和信任的CA憑證的base64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_
```

自動匯出原則範例

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。和`ontap-nas-flexgroup`驅動程式的運作方式相同`ontap-nas-economy`。

```
---
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: password
nfsMountOptions: nfsvers=4
```

IPv6 位址範例

此範例顯示 `managementLIF` 使用 IPv6 位址。

```
---
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: password
```

Amazon FSX for ONTAP 使用 SMB Volume 範例

使用 SMB 磁碟區的 ONTAP 適用的 FSX 需要此 `smbShare` 參數。

```
---
version: 1
backendName: SMBBackend
storageDriverName: ontap-nas
managementLIF: example.mgmt.fqdn.aws.com
nasType: smb
dataLIF: 10.0.0.15
svm: nfs_svm
smbShare: smb-share
clientCertificate: ZXR0ZXJwYXB...ICMgJ3BhcGVyc2
clientPrivateKey: vciwKIyAgZG...0cnksIGRlc2NyaX
trustedCACertificate: zcyBbaG...b3Igb3duIGNsYXNz
storagePrefix: myPrefix_
```

名稱範本的后端組態範例

```
---
version: 1
storageDriverName: ontap-nas
backendName: ontap-nas-backend
managementLIF: <ip address>
svm: svm0
username: <admin>
password: <password>
defaults: {
  "nameTemplate":
    "{{.volume.Name}}_{{.labels.cluster}}_{{.volume.Namespace}}_{{.volume.R
    equestName}}"
  },
  "labels": {"cluster": "ClusterA", "PVC":
    "{{.volume.Namespace}}_{{.volume.RequestName}}"}
}
```

虛擬集區的后端範例

在下面顯示的后端定義檔範例中、會針對所有儲存池設定特定的預設值、例如 `spaceReserve` 「無」、`spaceAllocation` 「假」和 `encryption` 「假」。虛擬資源池是在儲存區段中定義的。

Astra Trident 會在「意見」欄位中設定資源配置標籤。註解是在 `FlexVol for` 或 `FlexGroup for ontap-nas-flexgroup` 上設定 `ontap-nas`。Astra Trident 會在資源配置時、將虛擬資源池上的所有標籤複製到儲存磁碟區。為了方便起見、儲存管理員可以針對每個虛擬資源池定義標籤、並依標籤將磁碟區分組。

在這些範例中、有些儲存資源池會自行設定 `spaceReserve`、和 `encryption` 值、`spaceAllocation` 有些資源池則會覆寫預設值。

```

---
version: 1
storageDriverName: ontap-nas
managementLIF: 10.0.0.1
svm: svm_nfs
username: admin
password: <password>
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:
  department: legal
  creditpoints: '5000'
  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 範例

```
---
version: 1
storageDriverName: ontap-nas-flexgroup
managementLIF: 10.0.0.1
svm: svm_nfs
username: vsadmin
password: <password>
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
svm: svm_nfs
username: vsadmin
password: <password>
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:
  protection: bronze
  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

以下 StorageClass 定義請參閱[\[虛擬集區的後端範例\]](#)。使用此 `parameters.selector` 欄位、每個 StorageClass 都會呼叫哪些虛擬集區可用於主控磁碟區。磁碟區將會在所選的虛擬資源池中定義各個層面。

- protection-gold`StorageClass 會對應至後端的第一個和第二個虛擬集區 `ontap-nas-flexgroup。這是唯一提供金級保護的資源池。

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: protection-gold
provisioner: csi.trident.netapp.io
parameters:
  selector: "protection=gold"
  fsType: "ext4"
```

- protection-not-gold`StorageClass 會對應至後端的第三個和第四個虛擬集區 `ontap-nas-flexgroup。這是唯一提供金級以外保護層級的資源池。

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: protection-not-gold
provisioner: csi.trident.netapp.io
parameters:
  selector: "protection!=gold"
  fsType: "ext4"
```

- app-mysqldb`StorageClass 會對應至後端的第四個虛擬集區 `ontap-nas。這是唯一為 mysqldb 類型應用程式提供儲存池組態的集區。

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: app-mysqldb
provisioner: csi.trident.netapp.io
parameters:
  selector: "app=mysqldb"
  fsType: "ext4"
```

- `tprotection-silver-creditpoints-20k` StorageClass 會對應至後端的第三個虛擬集區 `ontap-nas-flexgroup`。這是唯一提供銀級保護和 20000 個信用點數的資源池。

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: protection-silver-creditpoints-20k
provisioner: csi.trident.netapp.io
parameters:
  selector: "protection=silver; creditpoints=20000"
  fsType: "ext4"
```

- `creditpoints-5k` StorageClass 會對應至後端的第三個虛擬集區、以及後端的第二個虛擬集區 `ontap-nas-economy`。這是唯一擁有 5000 個信用點數的集區方案。

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: creditpoints-5k
provisioner: csi.trident.netapp.io
parameters:
  selector: "creditpoints=5000"
  fsType: "ext4"
```

Astra Trident將決定選取哪個虛擬集區、並確保符合儲存需求。

在初始組態後更新 dataLIF

您可以在初始組態後變更資料LIF、方法是執行下列命令、以更新資料LIF提供新的後端Json檔案。

```
tridentctl update backend <backend-name> -f <path-to-backend-json-file-with-updated-dataLIF>
```



如果將PVCS附加至一或多個Pod、您必須關閉所有對應的Pod、然後將其重新啟動、新的資料LIF才會生效。

Amazon FSX for NetApp ONTAP 產品

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 NetApp ONTAP 儲存作業系統所支援的檔案系統。FSX for ONTAP VMware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 for ONTAP Sfor支援ONTAP Isf供 檔案系統功能和管理API。

您可以將Amazon FSX for NetApp ONTAP 的支援文件系統與Astra Trident整合、以確保在Amazon Elastic Kubernetes Service (EKS) 中執行的Kubernetes叢集能夠配置區塊並以ONTAP 支援的方式歸檔持續Volume。

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。

需求

此外"[Astra Trident的需求](#)"、若要將適用於 ONTAP 的 FSX 與 Astra Trident 整合、您還需要：

- 現有的 Amazon EKS 叢集或已安裝的自我管理 Kubernetes 叢集 `kubectl`。
- 可從叢集工作節點存取的現有 Amazon FSX for NetApp ONTAP 檔案系統和儲存虛擬機器 (SVM)。
- 準備好用於的工作節點"[NFS或iSCSI](#)"。



根據您的 EKS AMI 類型、確保您遵循 Amazon Linux 和 Ubuntu (Amis) 所需的節點準備步驟 "[Amazon機器映像](#)"。

考量

- SMB Volume：
 - SMB 磁碟區僅支援使用 ``ontap-nas`` 驅動程式。
 - Astra Trident EKS 附加元件不支援 SMB Volume。
 - Astra Trident僅支援安裝在Windows節點上執行的Pod上的SMB磁碟區。如 "[準備配置SMB磁碟區](#)" 需詳細資訊、請參閱。
- 在 Astra Trident 24.02 之前、Trident 無法刪除在 Amazon FSX 檔案系統上建立且已啟用自動備份的磁碟區。若要在 Astra Trident 24.02 或更新版本中避免此問題、請在 AWS FSX for ONTAP 的後端組態檔案中指定 `fsxFilesystemID`、`AWS apiRegion`、`AWS apikey` 和 `AWS secretKey`。



如果您要指定 Astra Trident 的 IAM 角色、則可以省略將 ``apiKey`` 和 ``secretKey`` 欄位明確指定 ``apiRegion`` 給 Astra Trident。如需詳細資訊、請 ["FSX提供ONTAP 各種組態選項和範例"](#) 參閱。

驗證

Astra Trident提供兩種驗證模式。

- 認證型（建議）：在 AWS Secrets Manager 中安全地儲存認證。您可以將使用者用於檔案系統、或是使用 `fsxadmin vsadmin` 為 SVM 設定的使用者。



Astra Trident 應以 SVM 使用者或具有相同角色之不同名稱的使用者身分執行 `vsadmin`。
◦ Amazon FSX for NetApp ONTAP 的 `fsxadmin`` 使用者僅能有限地取代 ONTAP ``admin` 叢集使用者。強烈建議搭配 Astra Trident 使用 `vsadmin`。

- 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。

如需啟用驗證的詳細資訊、請參閱您的驅動程式類型驗證：

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

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"](#)

建立 IAM 角色和 AWS 密碼

您可以將 Kubernetes Pod 設定為以 AWS IAM 角色進行驗證、而非提供明確的 AWS 認證、以存取 AWS 資源。



若要使用 AWS IAM 角色進行驗證、您必須使用 EKS 部署 Kubernetes 叢集。

建立 AWS Secret Manager 機密

此範例建立 AWS Secret Manager 機密以儲存 Astra Trident CSI 認證：

```
aws secretsmanager create-secret --name trident-secret --description "Trident CSI credentials" --secret-string "{\"user\":\"vsadmin\",\"password\":\"<svmpassword>\"}"
```

建立 IAM 原則

下列範例使用 AWS CLI 建立 IAM 原則：

```
aws iam create-policy --policy-name AmazonFSxNCSIDriverPolicy --policy-document file://policy.json --description "This policy grants access to Trident CSI to FSxN and Secret manager"
```

• 政策 JSON 檔案 * :

```
policy.json:
{
  "Statement": [
    {
      "Action": [
        "fsx:DescribeFileSystems",
        "fsx:DescribeVolumes",
        "fsx:CreateVolume",
        "fsx:RestoreVolumeFromSnapshot",
        "fsx:DescribeStorageVirtualMachines",
        "fsx:UntagResource",
        "fsx:UpdateVolume",
        "fsx:TagResource",
        "fsx>DeleteVolume"
      ],
      "Effect": "Allow",
      "Resource": "*"
    },
    {
      "Action": "secretsmanager:GetSecretValue",
      "Effect": "Allow",
      "Resource": "arn:aws:secretsmanager:<aws-region>:<aws-account-id>:secret:<aws-secret-manager-name>"
    }
  ],
  "Version": "2012-10-17"
}
```

為服務帳戶建立和 **IAM** 角色

以下範例為 EKS 中的服務帳戶建立 IAM 角色：

```
eksctl create iamserviceaccount --name trident-controller --namespace trident
--cluster <my-cluster> --role-name <AmazonEKS_FSxN_CSI_DriverRole> --role-only
--attach-policy-arn arn:aws:iam::aws:policy/service-
role/AmazonFSxNCSI_DriverPolicy --approve
```

安裝 **Astra Trident**

Astra Trident 簡化了在 Kubernetes 進行 NetApp ONTAP 儲存管理的 Amazon FSX、讓開發人員和管理員能夠專注於應用程式部署。

您可以使用下列其中一種方法來安裝 Astra Trident：

- 掌舵
- EKS 附加元件

If you want to make use of the snapshot functionality, install the CSI snapshot controller add-on. Refer to <https://docs.aws.amazon.com/eks/latest/userguide/csi-snapshot-controller.html>.

透過 helm 安裝 Astra Trident

1. 下載 Astra Trident 安裝程式套件

Astra Trident 安裝程式套件包含部署 Trident 操作員及安裝 Astra Trident 所需的一切。從 GitHub 的 Assets 區段下載並解壓縮 Astra Trident 安裝程式的最新版本。

```
wget https://github.com/NetApp/trident/releases/download/v24.06.0/trident-installer-24.06.0.tar.gz
tar -xf trident-installer-24.06.0.tar.gz
cd trident-installer
```

2. 使用下列環境變數設定 * 雲端供應商 * 和 * 雲端 IDENTITY * 旗標的值：

```
export CP="AWS"
export CI="'eks.amazonaws.com/role-arn:arn:aws:iam::<accountID>:role/<AmazonEKS_FSxN_CSI_DriverRole>'"
```

以下範例安裝 Astra Trident 並將旗標設定 cloud-provider 為 ` \$CP ` 和 cloud-identity \$CI：

```
helm install trident trident-operator-100.2406.0.tgz --set
cloudProvider=$CP --set cloudIdentity=$CI --namespace trident
```

您可以使用 `helm list` 命令檢閱安裝詳細資料，例如名稱，命名空間，圖表，狀態，應用程式版本和修訂版編號。

```
helm list -n trident
```

NAME	NAMESPACE	REVISION	UPDATED
STATUS	CHART		APP VERSION
trident-operator	trident	1	2024-10-14 14:31:22.463122
+0300 IDT	deployed	trident-operator-100.2406.1	24.06.1

透過 EKS 附加元件安裝 Astra Trident

Astra Trident EKS 附加元件包含最新的安全性修補程式、錯誤修正程式、並經過 AWS 驗證、可與 Amazon EKS 搭配使用。EKS 附加元件可讓您持續確保 Amazon EKS 叢集安全穩定、並減少安裝、設定及更新附加元件所需的工作量。

先決條件

在設定 AWS EKS 的 Astra Trident 附加元件之前、請確定您具有下列項目：

- 具有附加訂閱的 Amazon EKS 叢集帳戶
- AWS 對 AWS 市場的權限：
"aws-marketplace:ViewSubscriptions",
"aws-marketplace:Subscribe",
"aws-marketplace:Unsubscribe"
- AMI 類型：Amazon Linux 2 (AL2_x86_64) 或 Amazon Linux 2 ARM (ARM_64)
- 節點類型：AMD 或 ARM
- 現有的 Amazon FSX for NetApp ONTAP 檔案系統

啟用 AWS 的 Astra Trident 附加元件

EKS 叢集

下列命令範例會安裝 Astra Trident EKS 附加元件：

```
eksctl create addon --cluster clusterName --name netapp_trident-operator  
--version v24.6.1-eksbuild  
eksctl create addon --cluster clusterName --name netapp_trident-operator  
--version v24.6.1-eksbuild.1 (使用專用版本)
```



當您設定選用參數 `cloudIdentity` 時，請確保在使用 EKS 附加元件安裝 Trident 時指定 `cloudProvider`。

管理主控台

1. 開啟 Amazon EKS 主控台：<https://console.aws.amazon.com/eks/home#/clusters>。
2. 在左導覽窗格中、按一下 * 叢集 *。
3. 按一下您要設定 NetApp Trident CSI 附加元件的叢集名稱。
4. 按一下 * 附加元件 *、然後按一下 * 取得更多附加元件 *。
5. 在 *S*SELECT 附加元件 * 頁面上、執行下列步驟：
 - a. 在 AWS Marketplace EKS-addons 區段中、選取 *Astra Trident by NetApp* 核取方塊。
 - b. 單擊 * 下一步 *。
6. 在 * 設定選取的附加元件 * 設定頁面上、執行下列步驟：
 - a. 選擇您要使用的 * 版本 *。
 - b. 對於 * 選取 IAM 角色 *、請保留 * 未設定 *。
 - c. 展開 * 選用組態設定 *、遵循 * 附加元件組態架構 *、並將 * 組態值 * 區段上的組態值參數設定為您在上一個步驟中建立的角色參數（值應採用下列格式：`eks.amazonaws.com/role-arn:arn:aws:iam::464262061435:role/AmazonEKS_FSXN_CSI_DriverRole`）。如果您為衝突解決方法選取「覆寫」、則現有附加元件的一或多個設定可以使用 Amazon EKS 附加元件設定覆寫。如果您未啟用此選項、且與現有設定發生衝突、則作業將會失敗。您可以使用產生的錯誤訊息來疑難排解衝突。選取此選項之前、請確定 Amazon EKS 附加元件不會管理您需要自行管理的設定。



當您設定選用參數 `cloudIdentity` 時，請確保在使用 EKS 附加元件安裝 Trident 時指定 `cloudProvider`。

7. 選擇 * 下一步 *。
8. 在 * 檢閱及新增 * 頁面上、選擇 * 建立 *。

附加元件安裝完成後、您會看到已安裝的附加元件。

AWS CLI

1. 建立 `add-on.json` 檔案：

```
add-on.json
{
    "clusterName": "<eks-cluster>",
    "addonName": "netapp_trident-operator",
    "addonVersion": "v24.6.1-eksbuild.1",
    "serviceAccountRoleArn": "arn:aws:iam::123456:role/astratrident-
role",
    "configurationValues": "{\"cloudIdentity\":
'eks.amazonaws.com/role-arn: arn:aws:iam::123456:role/astratrident-
role'\",
    \"cloudProvider\": \"AWS\"}"
}
```



當您設定選用參數 `cloudIdentity` 時，請確保在使用 EKS 附加元件安裝 Trident 時指定 `AWS` 為 `cloudProvider`。

2. 安裝 Astra Trident EKS 附加元件 "

```
aws eks create-addon --cli-input-json file://add-on.json
```

更新 Astra Trident EKS 附加元件

EKS 叢集

- 檢查 FSxN Trident CSI 附加元件的目前版本。以叢集名稱取代 `my-cluster`。
`eksctl get addon --name netapp_trident-operator --cluster my-cluster`
- 輸出範例：*

```
NAME                                VERSION                                STATUS    ISSUES
IAMROLE    UPDATE AVAILABLE    CONFIGURATION VALUES
netapp_trident-operator    v24.6.1-eksbuild.1    ACTIVE    0
{"cloudIdentity":"'eks.amazonaws.com/role-arn:
arn:aws:iam::139763910815:role/AmazonEKS_FSXN_CSI_DriverRole'"}

```

- 將附加元件更新至上一個步驟輸出中可用更新所傳回的版本。
`eksctl update addon --name netapp_trident-operator --version v24.6.1-eksbuild.1 --cluster my-cluster --force`

如果您移除此 `--force` 選項、且任何 Amazon EKS 附加元件設定與您現有的設定發生衝突、則更新 Amazon EKS 附加元件會失敗；您會收到錯誤訊息、協助您解決衝突。在指定此選項之前、請確定 Amazon EKS 附加元件不會管理您需要管理的設定、因為這些設定會以此選項覆寫。如需此設定的其他選項的詳細資訊，請參閱 "[附加元件](#)"。如需 Amazon EKS Kubernetes 現場管理的詳細資訊、請參閱 "[Kubernetes 現場管理](#)"。

管理主控台

1. 打開 Amazon EKS 控制檯 <https://console.aws.amazon.com/eks/home#/clusters>。
2. 在左導覽窗格中、按一下 * 叢集 *。
3. 按一下您要更新 NetApp Trident CSI 附加元件的叢集名稱。
4. 按一下 * 附加元件 * 索引標籤。
5. 按一下 **Astra Trident by NetApp** *，然後按一下 *Edit*。
6. 在 * 設定 Astra Trident by NetApp * 頁面上、執行下列步驟：
 - a. 選擇您要使用的 * 版本 *。
 - b. (可選) 您可以展開 * 可選配置設置 * 並根據需要進行修改。
 - c. 按一下 *儲存變更*。

AWS CLI

下列範例更新 EKS 附加元件：

```
aws eks update-addon --cluster-name my-cluster netapp_trident-operator vpc-cni
--addon-version v24.6.1-eksbuild.1 \
--service-account-role-arn arn:aws:iam::111122223333:role/role-name
--configuration-values '{}' --resolve-conflicts --preserve

```

您有兩種移除 Amazon EKS 附加元件的選項：

- * 保留叢集上的附加軟體 * –此選項會移除 Amazon EKS 對任何設定的管理。它也會移除 Amazon EKS 通知您更新的功能、並在您啟動更新後自動更新 Amazon EKS 附加元件。不過、它會保留叢集上的附加軟體。此選項可讓附加元件成為自我管理的安裝、而非 Amazon EKS 附加元件。有了這個選項、附加元件就不會停機。保留 `--preserve` 命令中的選項以保留附加元件。
- * 從叢集完全移除附加軟體 * –我們建議您只有在叢集上沒有任何相關資源的情況下、才從叢集移除 Amazon EKS 附加元件。從命令中移除 `--preserve` 選項 `delete` 以移除附加元件。



如果附加元件有相關的 IAM 帳戶、則不會移除 IAM 帳戶。

EKS 叢集

下列命令會解除安裝 Astra Trident EKS 附加元件：

```
eksctl delete addon --cluster K8s-arm --name netapp_trident-operator
```

管理主控台

1. 開啟 Amazon EKS 主控台：<https://console.aws.amazon.com/eks/home#/clusters>。
2. 在左導覽窗格中、按一下 * 叢集 *。
3. 按一下您要移除 NetApp Trident CSI 附加元件的叢集名稱。
4. 單擊 **Add-ons** 選項卡，然後單擊 *Astra Trident by NetApp*。
5. 按一下「移除」。
6. 在 * 移除 NetApp_trident 操作員確認 * 對話方塊中、執行下列步驟：
 - a. 如果您想要 Amazon EKS 停止管理附加元件的設定、請選取 * 保留在叢集 * 上。如果您想要保留叢集上的附加軟體、以便自行管理附加元件的所有設定、請執行此動作。
 - b. 輸入 **NetApp_trident - operer**。
 - c. 按一下「移除」。

AWS CLI

以叢集名稱取代 `my-cluster`、然後執行下列命令。

```
aws eks delete-addon --cluster-name my-cluster --addon-name netapp_trident-operator --preserve
```

設定儲存後端

整合SAN和NAS驅動程式ONTAP

您可以使用儲存在 AWS Secret Manager 中的 SVM 認證（使用者名稱和密碼）來建立後端檔案、如以下範例所示：

YAML

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-ontap-nas
spec:
  version: 1
  storageDriverName: ontap-nas
  backendName: tbc-ontap-nas
  svm: svm-name
  aws:
    fsxFileSystemID: fs-xxxxxxxxxx
  credentials:
    name: "arn:aws:secretsmanager:us-west-2:xxxxxxx:secret:secret-
name"
    type: awsarn
```

JSON

```
{
  "apiVersion": "trident.netapp.io/v1",
  "kind": "TridentBackendConfig",
  "metadata": {
    "name": "backend-tbc-ontap-nas"
  },
  "spec": {
    "version": 1,
    "storageDriverName": "ontap-nas",
    "backendName": "tbc-ontap-nas",
    "svm": "svm-name",
    "aws": {
      "fsxFileSystemID": "fs-xxxxxxxxxx"
    },
    "managementLIF": null,
    "credentials": {
      "name": "arn:aws:secretsmanager:us-west-2:xxxxxxx:secret:secret-
name",
      "type": "awsarn"
    }
  }
}
```

如需建立後端的相關資訊、請參閱下列頁面：

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

適用於 **ONTAP** 驅動程式詳細資料的 **FSX**

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

- `ontap-san`：配置的每個 PV 都是其各自 Amazon FSX 中的 LUN（用於 NetApp ONTAP Volume）。建議用於區塊儲存。
- `ontap-nas`：配置的每個 PV 都是 NetApp ONTAP Volume 的完整 Amazon FSX。建議用於 NFS 和 SMB。
- `ontap-san-economy`：每個 PV 配置的 LUN 都是 NetApp ONTAP Volume 的每個 Amazon FSX 具有可設定數量的 LUN。
- `ontap-nas-economy`：每個已配置的 PV 都是 `qtree`、每個 Amazon FSX 的 NetApp ONTAP Volume 可設定數量的 `qtree`。
- `ontap-nas-flexgroup`：配置的每個 PV 都是 NetApp ONTAP FlexGroup Volume 的完整 Amazon FSX。

如需驅動程式詳細資料、請參閱["NAS 驅動程式"](#)和["SAN 驅動程式"](#)。

組態範例

搭配加密管理程式的 **AWS FSX for ONTAP** 組態

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-ontap-nas
spec:
  version: 1
  storageDriverName: ontap-nas
  backendName: tbc-ontap-nas
  svm: svm-name
  aws:
    fsxFileSystemID: fs-xxxxxxxxxx
  managementLIF:
  credentials:
    name: "arn:aws:secretsmanager:us-west-2:xxxxxxx:secret:secret-
name"
    type: awsarn
```

SMB 磁碟區的儲存類別組態

使用 `nasType`、`node-stage-secret-name` 和 `node-stage-secret-namespace`，您可以指定 SMB 磁碟區並提供所需的 Active Directory 認證。SMB 磁碟區僅支援使用 `ontap-nas` 驅動程式。

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: nas-smb-sc
provisioner: csi.trident.netapp.io
parameters:
  backendType: "ontap-nas"
  trident.netapp.io/nasType: "smb"
  csi.storage.k8s.io/node-stage-secret-name: "smbcreds"
  csi.storage.k8s.io/node-stage-secret-namespace: "default"
```

後端進階組態和範例

如需後端組態選項、請參閱下表：

參數	說明	範例
<code>version</code>		永遠為1
<code>storageDriverName</code>	儲存驅動程式名稱	<code>ontap-nas</code> 、 <code>ontap-nas-economy</code> 、 <code>ontap-nas-flexgroup</code> 、 <code>ontap-san</code> 、 <code>ontap-san-economy</code>
<code>backendName</code>	自訂名稱或儲存後端	驅動程式名稱 + 「_」 + <code>dataLIF</code>
<code>managementLIF</code>	叢集或 SVM 管理 LIF 的 IP 位址可以指定完整網域名稱 (FQDN)。如果使用 IPv6 旗標安裝 Astra Trident、則可以設定為使用 IPv6 位址。IPv6 位址必須以方括弧來定義、例如 <code>[28e8:d9fb:a825:b7bf:69a8:d02f:9e7b:3555]</code> 。如果您提供的是「下方」欄位、則 <code>fsxFilesystemID</code> <code>aws</code> 不需要提供、 <code>managementLIF</code> 因為 Astra Trident 會從 AWS 擷取 SVM <code>managementLIF</code> 資訊。因此、您必須在 SVM 下提供使用者的認證 (例如: <code>vsadmin</code>)、且使用者必須具有該 <code>vsadmin</code> 角色。	「10.0.0.1」、 <code>[2001:1234:abcd::fefe]</code>

參數	說明	範例
dataLIF	傳輸協定LIF的IP位址。不適用 NAS 驅動程式：建議您指定dataLIF ONTAP。如果未提供、Astra Trident會從SVM擷取資料lifs。您可以指定要用於NFS掛載作業的完整網域名稱（FQDN）、讓您建立循環配置資源DNS、以便在多個資料生命期之間達到負載平衡。可在初始設定之後變更。請參閱。《 SAN 驅動程式：請勿指定用於iSCSI》 ONTAP。Astra Trident使用ONTAP「選擇性LUN地圖」來探索建立多重路徑工作階段所需的iSCSI lifs。如果明確定義dataLIF、就會產生警告。如果使用 IPv6 旗標安裝 Astra Trident、則可以設定為使用 IPv6 位址。IPv6位址必須以方括弧來定義、例如[28e8:d9fb:a825:b7bf:69a8:d02f:9e7b:3555]。	
autoExportPolicy	啟用自動匯出原則建立及更新[布林值]。Astra Trident 可使用 `autoExportPolicy` 和 `autoExportCIDRs` 選項、自動管理匯出原則。	false
autoExportCIDRs	將 Kubernetes 節點 IP 篩選在啟用時的 CIDR 清單 autoExportPolicy。Astra Trident 可使用 `autoExportPolicy` 和 `autoExportCIDRs` 選項、自動管理匯出原則。	[「0.00.0.0/0」、 「:/0」]
labels	套用到磁碟區的任意JSON-格式化標籤集	"
clientCertificate	用戶端憑證的Base64編碼值。用於憑證型驗證	"
clientPrivateKey	用戶端私密金鑰的Base64編碼值。用於憑證型驗證	"
trustedCACertificate	受信任CA憑證的Base64編碼值。選用。用於憑證型驗證。	"
username	連線至叢集或SVM的使用者名稱。用於認證型驗證。例如、vsadmin。	
password	連線至叢集或SVM的密碼。用於認證型驗證。	
svm	要使用的儲存虛擬機器	指定SVM管理LIF時衍生。

參數	說明	範例
storagePrefix	在SVM中配置新磁碟區時所使用的前置碼。無法在建立後修改。若要更新此參數、您需要建立新的後端。	trident
limitAggregateUsage	* 請勿指定 Amazon FSX for NetApp ONTAP。*提供的`fsxadmin`和`vsadmin`不包含擷取 Aggregate 使用量並使用 Astra Trident 加以限制所需的權限。	請勿使用。
limitVolumeSize	如果要求的磁碟區大小高於此值、則資源配置失敗。也會限制其管理 qtree 和 LUN 的最大磁碟區大小、而且此`qtreesPerFlexvol`選項可讓您自訂每個 FlexVol 的最大 qtree 數量。	「」（預設不強制執行）
lunsPerFlexvol	每FlexVol 個LUN的最大LUN數量、範圍必須為[50、200]。僅限 SAN。	“100”
debugTraceFlags	疑難排解時要使用的偵錯旗標。例如、除非您正在進行疑難排解並需要詳細的記錄傾印、否則 {API} : false、「方法」: true } 不會使用 debugTraceFlags。	null
nfsMountOptions	以逗號分隔的NFS掛載選項清單。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、Astra Trident將不會在相關的持續磁碟區上設定任何掛載選項。	''
nasType	設定NFS或SMB磁碟區建立。選項包括`nfs`、`smb`或`null`。*SMB Volume 必須設為`smb`。*NFS磁碟區的預設值設為`null`。	nfs
qtreesPerFlexvol	每FlexVol 個邊的最大qtree數、必須在範圍內[50、300]	"200"
smbShare	您可以指定下列其中一項：使用 Microsoft 管理主控台或 ONTAP CLI 建立的 SMB 共用名稱、或是允許 Astra Trident 建立 SMB 共用的名稱。ONTAP 後端的 Amazon FSX 需要此參數。	smb-share

參數	說明	範例
useREST	使用ONTAP Isrest API的布林參數。 * 技術預覽 * useREST 是以建議用於測試環境、而非正式作業工作負載的「技術預覽」形式提供。設為 true 時、Astra Trident 將使用 ONTAP REST API 與後端通訊。此功能需要ONTAP 使用更新版本的版本。此外、使用的 ONTAP 登入角色必須具有應用程式存取權 `ontap`。這是預先定義的和角色所滿足 vsadmin cluster-admin 的。	false
aws	您可以在 AWS FSX for ONTAP 的組態檔中指定下列項目： - fsxFilesystemID：指定 AWS FSX 檔案系統的 ID。 - apiRegion：AWS API 區域名稱。 - apikey：AWS API 金鑰。 - secretKey：AWS 秘密金鑰。	"" "" ""
credentials	指定要儲存在 AWS Secret Manager 中的 FSX SVM 認證。 - name：機密的 Amazon 資源名稱（ARN）、其中包含 SVM 的認證。 - type：設為 awsarn。如需詳細資訊、請參閱 "建立 AWS Secrets Manager 密碼" 。	

用於資源配置磁碟區的後端組態選項

您可以使用組態區段中的這些選項來控制預設資源配置 defaults。如需範例、請參閱下列組態範例。

參數	說明	預設
spaceAllocation	LUN的空間分配	true
spaceReserve	空間保留模式；「無」（精簡）或「Volume」（完整）	none
snapshotPolicy	要使用的Snapshot原則	none
qosPolicy	要指派給所建立磁碟區的QoS原則群組。選擇每個儲存集區或後端的其中一個qosPolicy 或adaptiveQosPolicy。搭配Astra Trident使用QoS原則群組需要ONTAP 使用更新版本的版本。我們建議使用非共用的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。	「」

參數	說明	預設
adaptiveQosPolicy	要指派給所建立磁碟區的調適性QoS原則群組。選擇每個儲存集區或後端的其中一個qosPolicy或adaptiveQosPolicy。不受ONTAP-NAS-經濟支援。	「」
snapshotReserve	保留給快照「0」的磁碟區百分比	如果 snapshotPolicy 是 `none` else 「」
splitOnClone	建立複本時、從其父複本分割複本	false
encryption	在新磁碟區上啟用 NetApp Volume Encryption (NVE)；預設為 false。必須在叢集上授權並啟用NVE、才能使用此選項。如果在後端啟用NAE、則Astra Trident中配置的任何磁碟區都會啟用NAE。如需更多資訊、請參閱" Astra Trident 如何與NVE和NAE搭配運作 "：。	false
luksEncryption	啟用LUKS加密。請參閱 " 使用Linux 統一金鑰設定 (LUKS) "。僅限 SAN。	"
tieringPolicy	要使用的分層原則 none	`snapshot-only` 適用於 ONTAP 9 前的 SVM-DR 組態
unixPermissions	新磁碟區的模式。如果是SMB磁碟區、請保留空白。	「」
securityStyle	新磁碟區的安全樣式。NFS 支援 `mixed` 和 `unix` 安全樣式。SMB 支援 `mixed` 和 `ntfs` 安全樣式。	NFS 預設值為 unix。SMB 預設為 ntfs。

準備配置SMB磁碟區

您可以使用驅動程式來配置 SMB 磁碟區 `ontap-nas`。完成下列步驟之前[整合SAN和NAS驅動程式ONTAP](#)。

開始之前

在您使用驅動程式來配置 SMB 磁碟區之前 `ontap-nas`、您必須具備下列項目。

- Kubernetes叢集具備Linux控制器節點、以及至少一個執行Windows Server 2019的Windows工作節點。Astra Trident僅支援安裝在Windows節點上執行的Pod上的SMB磁碟區。
- 至少有一個Astra Trident機密、其中包含您的Active Directory認證資料。產生機密 `smbcreds`：

```
kubectl create secret generic smbcreds --from-literal username=user
--from-literal password='password'
```

- 設定為Windows服務的SCSI Proxy。若要設定 `csi-proxy`、請參閱或"[GitHub：適用於Windows的SCSI Proxy](#)"、瞭"[GitHub：csi Proxy](#)"解在 Windows 上執行的 Kubernetes 節點。

步驟

1. 建立SMB共用區。您可以使用共用資料夾嵌入式管理單元或使用 ONTAP CLI、以兩種方式之一建立 SMB 管理員共用"[Microsoft管理主控台](#)"。若要使用ONTAP CLI建立SMB共用：

- a. 如有必要、請建立共用的目錄路徑結構。

命令會 `vserver cifs share create` 在建立共用時檢查 `-path` 選項中指定的路徑。如果指定的路徑不存在、則命令會失敗。

- b. 建立與指定SVM相關的SMB共用區：

```
vserver cifs share create -vserver vserver_name -share-name
share_name -path path [-share-properties share_properties,...]
[other_attributes] [-comment text]
```

- c. 確認共用區已建立：

```
vserver cifs share show -share-name share_name
```



如需完整詳細資料、請參閱"[建立SMB共用區](#)"。

2. 建立後端時、您必須設定下列項目以指定SMB Volume。有關 ONTAP 後端組態選項的所有 FSX "[FSX提供ONTAP 各種組態選項和範例](#)"、請參閱。

參數	說明	範例
smbShare	您可以指定下列其中一項：使用 Microsoft 管理主控台或 ONTAP CLI 建立的 SMB 共用名稱、或是允許 Astra Trident 建立 SMB 共用的名稱。ONTAP 後端的 Amazon FSX 需要此參數。	smb-share
nasType	* 必須設定為 smb.*如果為 null，則默認為 nfs。	smb
securityStyle	新磁碟區的安全樣式。* 必須設定為 ntfs SMB Volume 或 mixed。 *	ntfs`或 `mixed SMB Volume
unixPermissions	新磁碟區的模式。SMB磁碟區*必須保留為空白。*	"

設定儲存類別和 PVC

設定 Kubernetes StorageClass 物件並建立儲存類別、以指示 Astra Trident 如何配置 Volume。建立 PersistentVolume (PV) 和 PersistentVolume Claim (PVC)、使用設定的 Kubernetes StorageClass 來要求存取 PV。然後、您可以將 PV 掛載至 Pod。

建立儲存類別

設定 Kubernetes StorageClass 物件

<https://kubernetes.io/docs/concepts/storage/storage-classes/> ["Kubernetes StorageClass 物件"] 將 Astra Trident 識別為該類別所使用的資源配置程式、指示 Astra Trident 如何資源配置 Volume 。例如：

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontap-gold
provisioner: csi.trident.netapp.io
parameters:
  backendType: "ontap-nas"
  media: "ssd"
  provisioningType: "thin"
  snapshots: "true"
```

如需儲存類別如何與互動的詳細資訊 PersistentVolumeClaim、以及控制 Astra Trident 配置磁碟區的參數、請參閱["Kubernetes和Trident物件"](#)。

建立儲存類別

步驟

1. 這是 Kubernetes 物件、因此請使用 `kubectl` 在 Kubernetes 中建立。

```
kubectl create -f storage-class-ontapnas.yaml
```

2. 現在您應該會看到Kubernetes和Astra Trident中的* basic、csi *儲存類別、而Astra Trident應該已經在後端探索集區。

```
kubectl get sc basic-csi
NAME          PROVISIONER          AGE
basic-csi     csi.trident.netapp.io 15h
```

建立 PV 和 PVC

["PersistentVolumer"](#) (PV) 是叢集管理員在 Kubernetes 叢集上配置的實體儲存資源。 ["_PersistentVolume Claim"](#) (PVC) 是存取叢集上 PersistentVolume 的要求。

可將 PVC 設定為要求儲存特定大小或存取模式。叢集管理員可以使用相關的 StorageClass 來控制超過 PersistentVolume 大小和存取模式的權限、例如效能或服務層級。

建立 PV 和 PVC 之後、您可以將磁碟區裝入 Pod 。

範例資訊清單

PersistentVolume 範例資訊清單

此範例資訊清單顯示與 StorageClass 相關的 10Gi 基本 PV `basic-csi` 。

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: pv-storage
  labels:
    type: local
spec:
  storageClassName: basic-csi
  capacity:
    storage: 10Gi
  accessModes:
    - ReadWriteMany
  hostPath:
    path: "/my/host/path"
```

PersistentVolume Claim 範例資訊清單

這些範例顯示基本的 PVC 組態選項。

可存取 **RWO** 的 **PVC**

此範例顯示具有 `rwX` 存取權的基本 PVC、與名稱為的 StorageClass 相關聯 `basic-csi`。

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: pvc-storage
spec:
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 1Gi
  storageClassName: basic-csi
```

採用 **NVMe / TCP** 的 **PVC**

此範例顯示 NVMe / TCP 的基本 PVC、並提供與名稱為的 StorageClass 相關聯的 `rwO` 存取 `protection-gold`。

```
---
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: pvc-san-nvme
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 300Mi
  storageClassName: protection-gold
```

建立 PV 和 PVC

步驟

1. 建立 PV。

```
kubectl create -f pv.yaml
```

2. 確認 PV 狀態。

```
kubectl get pv
NAME          CAPACITY  ACCESS MODES  RECLAIM POLICY  STATUS  CLAIM
STORAGECLASS REASON    AGE
pv-storage   4Gi      RWO           Retain          Available
7s
```

3. 建立 PVC。

```
kubectl create -f pvc.yaml
```

4. 確認 PVC 狀態。

```
kubectl get pvc
NAME          STATUS  VOLUME          CAPACITY  ACCESS MODES  STORAGECLASS  AGE
pvc-storage  Bound  pv-name 2Gi      RWO
5m
```

如需儲存類別如何與互動的詳細資訊 PersistentVolumeClaim、以及控制 Astra Trident 配置磁碟區的參數、請參閱"[Kubernetes和Trident物件](#)"。

Astra Trident 屬性

這些參數決定應使用哪些 Astra Trident 管理的儲存資源池來配置指定類型的磁碟區。

屬性	類型	價值	優惠	申請	支援者
媒體 ^{1^}	字串	HDD、混合式、SSD	資源池包含此類型的媒體、混合式表示兩者	指定的媒體類型	ONTAP-NAS、ONTAP-NAS-經濟型、ONTAP-NAS-flexgroup、ONTAP-SAN、solidfire-san
資源配置類型	字串	纖薄、厚實	Pool支援此資源配置方法	指定的資源配置方法	厚：全ONTAP 是邊、薄：全ONTAP 是邊、邊、邊、邊、邊、邊、邊、邊、邊、邊

屬性	類型	價值	優惠	申請	支援者
後端類型	字串	ONTAP-NAS 、ONTAP-NAS- 經濟 型、ONTAP- NAS-flexgroup 、ONTAP- SAN、solidfire- san、GCP- CVS、azure- NetApp-Files 、ONTAP-san經 濟	集區屬於此類型 的後端	指定後端	所有驅動程式
快照	布爾	對、錯	集區支援具有快 照的磁碟區	已啟用快照 的Volume	ONTAP-NAS 、ONTAP- SAN、Solidfire- SAN、GCP-CVS
複製	布爾	對、錯	資源池支援複製 磁碟區	已啟用複本 的Volume	ONTAP-NAS 、ONTAP- SAN、Solidfire- SAN、GCP-CVS
加密	布爾	對、錯	資源池支援加密 磁碟區	已啟用加密 的Volume	ONTAP-NAS 、ONTAP-NAS- 經濟型、ONTAP- NAS- FlexGroups、ON TAP-SAN
IOPS	內部	正整數	集區能夠保證此 範圍內的IOPS	Volume保證這 些IOPS	solidfire-san

¹：ONTAP Select 不受支援

部署範例應用程式

部署範例應用程式。

步驟

1. 將磁碟區裝入 Pod。

```
kubectl create -f pv-pod.yaml
```

這些範例顯示將 PVC 附加至 Pod 的基本組態：`* 基本組態 *`：

```

kind: Pod
apiVersion: v1
metadata:
  name: pv-pod
spec:
  volumes:
    - name: pv-storage
      persistentVolumeClaim:
        claimName: basic
  containers:
    - name: pv-container
      image: nginx
      ports:
        - containerPort: 80
          name: "http-server"
      volumeMounts:
        - mountPath: "/my/mount/path"
          name: pv-storage

```



您可以使用監控進度 `kubectl get pod --watch`。

2. 驗證是否已將磁碟區掛載到上 `/my/mount/path`。

```
kubectl exec -it task-pv-pod -- df -h /my/mount/path
```

```

Filesystem                                Size
Used Avail Use% Mounted on
192.168.188.78:/trident_pvc_ae45ed05_3ace_4e7c_9080_d2a83ae03d06 1.1G
320K 1.0G 1% /my/mount/path

```

1. 您現在可以刪除 Pod。Pod 應用程式將不再存在、但該磁碟區仍會保留。

```
kubectl delete pod task-pv-pod
```

在 EKS 叢集上設定 Astra Trident EKS 附加元件

Astra Trident 簡化了在 Kubernetes 進行 NetApp ONTAP 儲存管理的 Amazon FSX、讓開發人員和管理員能夠專注於應用程式部署。Astra Trident EKS 附加元件包含最新的安全性修補程式、錯誤修正程式、並經過 AWS 驗證、可與 Amazon EKS 搭配使用。EKS 附加元件可讓您持續確保 Amazon EKS 叢集安全穩定、並減少安裝、設定及更新附加元件所需的

工作量。

先決條件

在設定 AWS EKS 的 Astra Trident 附加元件之前、請確定您具有下列項目：

- 具有附加訂閱的 Amazon EKS 叢集帳戶
- AWS 對 AWS 市場的權限：
"aws-marketplace:ViewSubscriptions",
"aws-marketplace:Subscribe",
"aws-marketplace:Unsubscribe"
- AMI 類型：Amazon Linux 2 (AL2_x86_64) 或 Amazon Linux 2 ARM (ARM_64)
- 節點類型：AMD 或 ARM
- 現有的 Amazon FSX for NetApp ONTAP 檔案系統

步驟

1. 在您的 EKS Kubernetes 叢集上、瀏覽至 * 附加元件 * 索引標籤。

The screenshot displays the AWS EKS console interface for a cluster named 'tri-env-eks'. At the top right, there are buttons for 'Delete cluster' and 'Upgrade version'. Below this, a notification banner states: 'End of standard support for Kubernetes version 1.30 is July 28, 2025. On that date, your cluster will enter the extended support period with additional fees. For more information, see the pricing page.' A 'Upgrade now' button is present on the right of this banner. The main content area is titled 'Cluster info' and includes a table with the following data:

Status	Kubernetes version	Support period	Provider
Active	1.30	Standard support until July 28, 2025	EKS

Below the cluster info, there is a navigation bar with tabs: Overview, Resources, Compute, Networking, Add-ons (1), Access, Observability, Upgrade insights, Update history, and Tags. The 'Add-ons' tab is selected. A notification banner below the tabs says: 'New versions are available for 3 add-ons.' The 'Add-ons (3)' section includes a search bar with the placeholder 'Find add-on', filters for 'Any category' and 'Any status', and indicates '3 matches'. There are buttons for 'View details', 'Edit', 'Remove', and 'Get more add-ons'.

2. 前往 * AWS Marketplace 附加元件 * 並選擇 _storage 類別。

AWS Marketplace add-ons (1)

Discover, subscribe to and configure EKS add-ons to enhance your EKS clusters.

Filtering options

Any category ▼ NetApp, Inc. ▼ Any pricing model ▼ Clear filters

NetApp, Inc. X < 1 >

NetApp **NetApp Trident**

NetApp Trident streamlines Amazon FSx for NetApp ONTAP storage management in Kubernetes to let your developers and administrators focus on application deployment. FSx for ONTAP flexibility, scalability, and integration capabilities make it the ideal choice for organizations seeking efficient containerized storage workflows. [Product details](#)

Standard Contract

Category	Listed by	Supported versions	Pricing starting at
storage	NetApp, Inc.	1.30, 1.29, 1.28, 1.27, 1.26, 1.25, 1.24, 1.23	View pricing details

Cancel **Next**

3. 找到 *Astra NetApp Trident Trident 附加元件* 並選取該核取方塊。
4. 選擇所需版本的附加元件。

NetApp Trident Remove add-on

Listed by NetApp	Category storage	Status ✔ Ready to install
----------------------------	---------------------	------------------------------

You're subscribed to this software View subscription ×

You can view the terms and pricing details for this product or choose another offer if one is available.

Version
Select the version for this add-on.

v24.6.1-eksbuild.1

Select IAM role
Select an IAM role to use with this add-on. To create a new custom role, follow the instructions in the [Amazon EKS User Guide](#).

Not set ↻

▶ **Optional configuration settings**

Cancel Previous Next

5. 選取 IAM 角色選項以從節點繼承。

Review and add

Step 1: Select add-ons

Edit

Selected add-ons (1)

Find add-on

< 1 >

Add-on name



Type



Status

netapp_trident-operator

storage

Ready to install

Step 2: Configure selected add-ons settings

Edit

Selected add-ons version (1)

< 1 >

Add-on name



Version



IAM role for service account (IRSA)

netapp_trident-operator

v24.6.1-eksbuild.1

Not set

Cancel

Previous

Create

- (可選) 根據需要配置任何可選的配置設置，然後選擇 * 下一步 *。

遵循 *Add-on 組態架構*，並將 *組態值* 區段上的組態值參數設定為您在上一個步驟中建立的角色參數（值應採用下列格式：`eks.amazonaws.com/role-arn:`

`arn:aws:iam::464262061435:role/AmazonEKS_FSXN_CSI_DriverRole`）。如果您為衝突解決方法選取「覆寫」、則現有附加元件的一或多個設定可以使用 Amazon EKS 附加元件設定覆寫。如果您未啟用此選項、且與現有設定發生衝突、則作業將會失敗。您可以使用產生的錯誤訊息來疑難排解衝突。選取此選項之前、請確定 Amazon EKS 附加元件不會管理您需要自行管理的設定。



當您設定選用參數 `cloudIdentity` 時，請確保在使用 EKS 附加元件安裝 Trident 時指定 `AWS` 為 `cloudProvider`。

Select IAM role
 Select an IAM role to use with this add-on. To create a new custom role, follow the instructions in the [Amazon EKS User Guide](#).

Not set ▼ ↻

Optional configuration settings

Add-on configuration schema
 Refer to the JSON schema below. The configuration values entered in the code editor will be validated against this schema.

```
{
  "$id": "http://example.com/example.json",
  "$schema": "https://json-schema.org/draft/2019-09/schema",
  "default": {},
  "examples": [
    {
      "cloudIdentity": ""
    }
  ],
  "properties": {
    "cloudIdentity": {
      "default": "",
      "examples": [

```

Configuration values [Info](#)
 Specify any additional JSON or YAML configurations that should be applied to the add-on.

```
1 {
2   "cloudIdentity": "'eks.amazonaws.com/role-arn: arn:aws
   :iam::139763910815:role
   /AmazonEKS_FSXN_CSI_DriverRole'",
3   "cloudProvider": "AWS"
4 }
```

7. 選擇* Create (建立)。
8. 確認附加元件的狀態為 *Active*。

Add-ons (1) [Info](#) View details Edit Remove Get more add-ons

netapp × Any category Any status 1 match < 1 >

NetApp **Astra Trident by NetApp**

Astra Trident streamlines Amazon FSx for NetApp ONTAP storage management in Kubernetes to let your developers and administrators focus on application deployment. FSx for ONTAP flexibility, scalability, and integration capabilities make it the ideal choice for organizations seeking efficient containerized storage workflows. [Product details](#)

Category	Status	Version	IAM role for service account (IRSA)	Listed by
storage	Active	v24.6.1-eksbuild.1	Not set	NetApp, Inc.

View subscription

使用 CLI 安裝 / 解除安裝 Astra Trident EKS 附加元件

使用 CLI 安裝 Astra Trident EKS 附加元件：

下列範例命令會安裝 Astra Trident EKS 附加元件：

```
eksctl create addon --cluster K8s-arm --name netapp_trident-operator --version
```

```
v24.6.1-eksbuild
eksctl create addon --cluster clusterName --name netapp_trident-operator
--version v24.6.1-eksbuild.1 (含專用版本)
```



當您設定選用參數 `cloudIdentity` 時，請確保在使用 EKS 附加元件安裝 Trident 時指定 `cloudProvider`。

使用 CLI 解除安裝 Astra Trident EKS 附加元件：

下列命令會解除安裝 Astra Trident EKS 附加元件：

```
eksctl delete addon --cluster K8s-arm --name netapp_trident-operator
```

使用 kubectl 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。您可以使用或等效的 CLI 工具來 `kubectl` 進行 Kubernetes 發佈。

TridentBackendConfig

TridentBackendConfig (tbc、tbconfig、tbackendconfig) 為前端、命名 CRD、可讓您使用管理 Astra Trident 後端 kubectl。Kubernetes 和儲存管理員現在可以直接透過 Kubernetes CLI 建立和管理後端 (tridentctl、而不需要專用的命令列公用程式)。

建立物件時 TridentBackendConfig、會發生下列情況：

- Astra Trident 會根據您提供的組態自動建立後端。這在內部表示為 A TridentBackend (tbe、tridentbackend) CR。
- TridentBackendConfig 與 Astra Trident 所建立的有獨特的連結 `TridentBackend`。

每個都 TridentBackendConfig 使用維護一對一對應 `TridentBackend`。前者是提供給使用者的介面、用於設計和設定後端；後者是 Trident 代表實際後端物件的方式。



TridentBackend`CRS 由 Astra Trident 自動建立。您*不應該*修改這些項目。如果您想要更新後端、請修改物件以進行更新 `TridentBackendConfig`。

請參閱下列 CR 格式範例 TridentBackendConfig：

```

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

```

您也可以查看目錄中的 "[Trident安裝程式](#)" 範例、以取得所需儲存平台 / 服務的範例組態。

`spec` 採用後端特定的組態參數。在此範例中、後端會使用 `ontap-san` 儲存驅動程式、並使用此處列出的組態參數。如需所需儲存驅動程式的組態選項清單、請參閱 [link:backends.html\["儲存驅動程式的後端組態資訊"\]](#)。

這 `spec` 一節也包含 `credentials` 和 `deletionPolicy` 欄位、這些欄位是在 CR 中新推出 `TridentBackendConfig` 的：

- `credentials`：此參數為必填欄位、包含用於驗證儲存系統 / 服務的認證。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。
- `deletionPolicy`：此字段定義刪除時應發生的 `TridentBackendConfig` 情況。可能需要兩種可能的值之一：
 - `delete`：這會同時刪除 `TridentBackendConfig` CR 和相關的後端。這是預設值。
 - `retain`：`TridentBackendConfig` 刪除 CR 後、後端定義仍會存在、可透過管理 `tridentctl`。將刪除原則設定為 `retain` 允許使用者降級至較早版本（2004 年 1 月 21 日之前）、並保留建立的後端。此欄位的值可在建立後更新 `TridentBackendConfig`。



後端的名稱是使用設定 `spec.backendName` 的。如果未指定、則後端的名稱會設為物件名稱 `TridentBackendConfig (metadata.name)`。建議使用明確設定後端名稱 `spec.backendName`。



使用建立的後端 `tridentctl` 沒有關聯的 `TridentBackendConfig` 物件。您可以建立 CR 來 `TridentBackendConfig` 選擇管理此類後端 `kubectl`。必須注意指定相同的組態參數（例如 `spec.backendName`、`spec.storagePrefix`、`spec.storageDriverName` 等）。Astra Trident 會自動連結新建立的 `TridentBackendConfig` 與先前存在的後端。

步驟總覽

若要使用建立新的後端 `kubectl`、您應該執行下列動作：

1. 建立 "Kubernetes機密"。密碼包含 Astra Trident 與儲存叢集 / 服務通訊所需的認證。
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、您可以使用觀察後端的狀態 `kubectl get tbc <tbc-name> -n <trident-namespace>`、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。範例如下：

```
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: password
```

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

儲存平台機密欄位說明	秘密	欄位說明
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-economy
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-economy
ONTAP	chapTargetUsername	目標使用者名稱。如果useCHAP=true則需要。適用於ontap-san`和`ontap-san-economy
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-economy

此步驟中建立的秘密將會在下一個步驟中建立的物件欄位 `TridentBackendConfig`` 中參照 ``spec.credentials``。

步驟 2：建立 `TridentBackendConfig` CR

您現在已準備好建立 `TridentBackendConfig` CR 了。在此範例中、使用驅動程式的後端 `ontap-san`` 是使用下列物件建立的 ``TridentBackendConfig``：

```
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 CR 的狀態 TridentBackendConfig

現在您已建立 TridentBackendConfig CR、可以驗證狀態。請參閱下列範例：

```

kubectl -n trident get tbc backend-tbc-ontap-san

```

NAME	PHASE	STATUS	BACKEND NAME	BACKEND UUID
backend-tbc-ontap-san			ontap-san-backend	8d24fce7-6f60-4d4a-8ef6-
bab2699e6ab8	Bound	Success		

已成功建立後端並繫結至 TridentBackendConfig CR。

階段可以採用下列其中一個值：

- **Bound**：TridentBackendConfig CR 與後端相關聯、且後端包含 configRef、設定為 `TridentBackendConfig` CR 的 uid 的項目。
- **Unbound**：使用表示 ""。`TridentBackendConfig` 物件未繫結至後端。根據預設、所有新建立的 CRS 都 `TridentBackendConfig` 處於此階段。階段變更之後、就無法再恢復為 Unbound（未綁定）。
- **Deleting**：TridentBackendConfig CR deletionPolicy 已設定為刪除。刪除 CR 後 TridentBackendConfig、它會轉換至「刪除」狀態。
 - 如果後端不存在持續磁碟區宣告（PVCS）、刪除 TridentBackendConfig、將會導致 Astra Trident 刪除後端和 `TridentBackendConfig` CR。
 - 如果後端上有一個或多個 PVCS、則會進入刪除狀態。TridentBackendConfig`CR 隨後也會進入刪除階段。只有刪除所有 PVCS 之後、才會刪除後端和 `TridentBackendConfig`。
- **Lost**：與 CR 相關的後端 TridentBackendConfig、被意外或刻意刪除、`TridentBackendConfig` CR 仍有已刪除後端的參照。TridentBackendConfig`無論值為何、仍可刪除 CR `deletionPolicy`。
- **Unknown**：Astra Trident 無法判斷與 CR 相關聯的後端狀態或存在 TridentBackendConfig。例如、如果 API 伺服器沒有回應、或 tridentbackends.trident.netapp.io CRD 遺失。這可能需要介入。

在此階段、成功建立後端！還有幾項作業可以額外處理，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、您也可以取得的 YAML/JSON 傾印 TridentBackendConfig。

```
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 包含回應 CR 所建立後端 TridentBackendConfig 的 backendName 和 backendUUID。此 lastOperationStatus 欄位代表 CR 上次操作的狀態 TridentBackendConfig 可由使用者觸發（例如、使用者在中變更項目 spec）、或由 Astra Trident 觸發（例如 Astra Trident 重新啟動期間）。可能是「成功」或「失敗」。phase 代表 CR 與後端之間關係的狀態 TridentBackendConfig。在上述範例中、phase 有值界限、表示 TridentBackendConfig CR 與後端相關聯。

您可以執行 `kubectl -n trident describe tbc <tbc-cr-name>` 命令以取得事件記錄的詳細資料。



您無法使用更新或刪除包含相關物件 tridentctl 的後端 TridentBackendConfig。瞭解在和 TridentBackendConfig 之間切換所涉及的步驟 tridentctl、[請參閱此處](#)。

管理後端

以KECBECVL執行後端管理

瞭解如何使用執行後端管理作業 `kubectl`。

刪除後端

刪除 `TridentBackendConfig` 後、您會指示 Astra Trident 刪除 / 保留後端（根據 `deletionPolicy`）。若要刪除後端、請確定已 `deletionPolicy` 設定為刪除。若要僅刪除 `TridentBackendConfig`、請確定已 `deletionPolicy` 設定為保留。這將確保後端仍然存在，並且可以使用進行管理 `tridentctl`。

執行下列命令：

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

Astra Trident 不會刪除使用中的 Kubernetes 機密 `TridentBackendConfig`。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` 取得存在的所有後端清單。此清單也會包含使用建立的後端 `tridentctl`。

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、必須更新物件中使用的 Kubernetes Secret `TridentBackendConfig`。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）。
 - 您可以使用下列命令、直接透過 Kubernetes 更新 `TridentBackendConfig` 物件：

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

- 或者、您也可以使用下列命令變更現有的 `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。您可以執行或 `kubectl describe tbc <tbc-name> -n trident` 來檢視記錄、以判斷原因、`kubectl get tbc <tbc-name> -o yaml -n trident`。
- 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

瞭解如何使用執行後端管理作業 `tridentctl`。

建立後端

建立之後"後端組態檔"，請執行下列命令：

```
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、只要再次執行命令即可 `create`。

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、只要再次執行命令即可 update。

識別使用後端的儲存類別

這是您可以使用 JSON 來回答的問題類型範例、此問題可 `tridentctl` 為後端物件輸出。這會使用 `jq` 您需要安裝的公用程式。

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

這也適用於使用建立的後端 TridentBackendConfig。

在後端管理選項之間切換

瞭解Astra Trident管理後端的不同方法。

管理後端的選項

隨着的推出 TridentBackendConfig，管理員現在有兩種獨特的方法來管理後端。這會提出下列問題：

- 使用建立的後端是否可以 `tridentctl` 與一起管理 `TridentBackendConfig`？
- 使用建立的後端可以 `TridentBackendConfig` 使用管理 `tridentctl` 嗎？

使用管理 `tridentctl` 後端 `TridentBackendConfig`

本節說明透過建立物件、直接透過 Kubernetes 介面來管理建立 `TridentBackendConfig` 的後端所需的步驟 `tridentctl`。

這將適用於下列案例：

- 預先存在的後端、因為是使用建立而 `tridentctl` 沒有 `TridentBackendConfig`。
- 使用建立的新後端 `tridentctl`、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。
- 將使用建立的後端連結 `tridentctl` 至新 `TridentBackendConfig` 物件。這樣做將意味着後端將使用而非 `tridentctl` 來管理 `kubectl`。

若要使用管理預先存在的後端 `kubectl`、您必須建立 `TridentBackendConfig` 與現有後端連結的。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。必須注意指定相同的組態參數（例如 `spec.backendName`、`spec.storagePrefix`、`spec.storageDriverName` 等）。`spec.backendName` 必須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 CR、該 CR 會自動連結至預先存在的 ontap-nas-backend（如本範例所示）。確保符合下列要求：

- 中定義了相同的後端名稱 `spec.backendName`。
- 組態參數與原始後端相同。
- 虛擬資源池（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 **CR** 的狀態 `TridentBackendConfig`

建立之後 `TridentBackendConfig`、其階段必須是 `Bound`。它也應反映與現有後端相同的後端名稱和UUID。

```

kubect1 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 |
+-----+-----+-----+-----+
+-----+-----+-----+-----+

```

後端現在將使用物件完全管理 tbc-ontap-nas-backend TridentBackendConfig。

使用管理 TridentBackendConfig 後端 `tridentctl`

```

`tridentctl` 可用於列出使用建立的後端
`TridentBackendConfig`。此外，系統管理員也可以選擇透過刪除並確定
`spec.deletionPolicy` 設定為 `retain` 來 `TridentBackendConfig` 完全管理這類後端
`tridentctl`。

```

步驟0：識別後端

例如、假設下列後端是使用建立的 TridentBackendConfig：

```
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、會看到 `TridentBackendConfig` 已成功建立並繫結至後端 [觀察後端的 UUID] 。

步驟 1：確認 `deletionPolicy` 設定為 `retain`

讓我們來看看的價值 `deletionPolicy`。這需要設為 `retain`。如此可確保刪除 CR 時 `TridentBackendConfig`、後端定義仍會存在、並可透過進行管理 `tridentctl`。

```
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
```



除非設定為 `retain`、否則請勿繼續執行下一個步驟 `deletionPolicy`。

步驟 2：刪除 TridentBackendConfig CR

最後一步是刪除 TridentBackendConfig CR。確認設定為 retain 之後，deletionPolicy，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 |
+-----+-----+-----+-----+
+-----+-----+-----+-----+
```

刪除物件後 TridentBackendConfig、Astra Trident 只是將其移除、而不會實際刪除後端本身。

版權資訊

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

由 NetApp 版權資料衍伸之軟體必須遵守下列授權和免責聲明：

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

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

本手冊所述之產品受到一項（含）以上的美國專利、國外專利或申請中專利所保障。

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

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

商標資訊

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