支援SAN驅動程式ONTAP

Astra Trident

NetApp
March 05, 2026

目錄

使用ONTAP SAN驅動程式設定後端

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

- "準備"
- "組態與範例"

Astra Control可為使用建立的磁碟區提供無縫保護、災難恢復和移動性（在Kubernetes叢集之間移動磁碟區） `ontap-nas`、 `ontap-nas-flexgroup`和 `ontap-san` 驅動程式：請參閱 "Astra Control複寫先決條件" 以取得詳細資料。

- 您必須使用 `ontap-nas` 適用於需要資料保護、災難恢復和行動力的正式作業工作負載。
- 使用 `ontap-san-economy` 當預期的Volume使用量將遠高於ONTAP 支援的容量時。
- 使用 `ontap-nas-economy` 只有在預期的Volume使用量會比ONTAP 支援的高出許多、以及 `ontap-san-economy` 無法使用驅動程式。
- 請勿使用 `ontap-nas-economy` 如果您預期需要資料保護、災難恢復或行動性、

使用者權限

Astra Trident希望以ONTAP 支援的形式執行、通常是以支援的方式執行 `admin` 叢集使用者或 `vsadmin` SVM使用者、或具有相同角色之不同名稱的使用者。對於Amazon FSX for NetApp ONTAP 支援的NetApp功能、Astra Trident預期會以ONTAP 使用叢集的形式執行、以執行支援或SVM管理員的身分 `fsxadmin` 使用者或 `vsadmin` SVM使用者、或具有相同角色之不同名稱的使用者。。 `fsxadmin` 使用者是叢集管理使用者的有限替代。

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

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

準備使用ONTAP 支援的SAN驅動程式來設定後端

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

請記住、您也可以執行多個驅動程式、並建立指向一個或多個驅動程式的儲存類別。例如、您可以設定 `san-dev` 使用的類別 `ontap-san` 驅動程式與 `san-default` 使用的類別 `ontap-san-economy` 一、

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

驗證

Astra Trident提供兩種驗ONTAP 證功能來驗證支援的後端。

- 認證型：ONTAP 對具備所需權限的使用者名稱和密碼。建議使用預先定義的安全登入角色、例如 `admin` 或 `vsadmin` 以確保與ONTAP 更新版本的最大相容性。
- 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。建議使用預先定義的標準角色、例如 `admin` 或 `vsadmin`。這可確保與未來ONTAP 的支援版本保持前瞻相容、因為未來的Astra Trident版本可能會使用功能API。您可以建立自訂的安全登入角色、並與Astra Trident搭配使用、但不建議使用。

後端定義範例如下所示：

YAML

版本： 1 個後端名稱： ExampleBackend storageDriverName ： ONTAP-SAN 管理 LIF ： 10.0.0.1 SVM ： SVM_NFS 使用者名稱： vsadmin 密碼：密碼

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CACertifate：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
```

++ 注意：執行此指令後，ONTAP 會提示輸入憑證。貼上步驟 1 中產生的 `k8senv.pem`檔案內容，然後輸入 `END`以完成安裝。

4. 確認ONTAP 支援的不安全登入角色 `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...Vaaalllluuuueeee",
"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。然後使用更新的backend.json檔案、其中包含要執行的必要參數 `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。

指定igroup

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

- Astra Trident可自動建立及管理每個後端的igroup。如果 `igroupName` 未包含在後端定義中、Astra Trident會建立名為的igroup `trident-<backend-UUID>` 在SVM上。如此可確保每個後端都有專屬的igroup、並處理Kubernetes節點IQN的自動新增/刪除作業。
- 或者、也可以在後端定義中提供預先建立的igroup。您可以使用來完成此作業 `igroupName` 組態參數。Astra Trident會將Kubernetes節點IQN新增/刪除至預先存在的igroup。

適用於具有的後端 `igroupName` 定義 `igroupName` 可以使用刪除 `tridentctl backend update` 使用Astra Trident自動處理igroup。這不會中斷對已附加至工作負載之磁碟區的存取。未來的連線將使用建立的igroup Astra Trident來處理。

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

如果將Astra Trident設定為使用「csi資源配置程式」、則Kubernetes節點IQN會自動新增至igroup或從其中移除。當節點新增至Kubernetes叢集時、 `trident-csi` 示範集部署Pod (`trident-csi-xxxxx` 在23.01或之前的版本中 `trident-node<operating system>-xxxx` 在23.01及更新版本中）、登錄新增的節點、然後登錄可附加磁碟區的新節點。節點IQN也會新增至後端的igroup。當節點封鎖、排放及從Kubernetes刪除時、類似的一組步驟可處理刪除IQN。

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

使用雙向**CHAP**驗證連線

Astra Trident可以使用雙向CHAP驗證iSCSI工作階段 `ontap-san` 和 `ontap-san-economy` 驅動程式：這需要啟用 `useCHAP` 選項。設定為時 `true`Astra Trident將SVM的預設啟動器安全性設定為雙向CHAP、並從後端檔案設定使用者名稱和機密。NetApp建議使用雙向CHAP來驗證連線。請參閱下列組態範例：

```
---
version: 1
storageDriverName: ontap-san
backendName: ontap_san_chap
managementLIF: 192.168.0.135
svm: ontap_iscsi_svm
useCHAP: true
username: vsadmin
password: password
igroupName: trident
chapInitiatorSecret: cl9qxIm36DKyawxy
chapTargetInitiatorSecret: rqxigXgkesIpwxyz
chapTargetUsername: iJF4heBRT0TCwxyz
chapUsername: uh2aNCLSd6cNwxyz
```

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

此外 `useCHAP=true`、 `chapInitiatorSecret`、 `chapTargetInitiatorSecret`、 `chapTargetUsername`和 `chapUsername` 欄位必須包含在後端定義中。執行建立後端後端之後、即可變更機密資訊 `tridentctl update`。

運作方式

透過設定 `useCHAP` 為真、儲存管理員指示Astra Trident在儲存後端上設定CHAP。這包括下列項目：

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

建立後端之後、Astra Trident會建立對應的 `tridentbackend` 將CHAP機密與使用者名稱儲存為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

您可以更新中的CHAP參數來更新CHAP認證 `backend.json` 檔案：這需要更新CHAP機密並使用 `tridentctl update` 命令以反映這些變更。

更新後端的CHAP機密時、您必須使用 `tridentctl` 以更新後端。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",
    "igroupName": "trident",
    "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。

SAN組態選項與範例ONTAP

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

後端組態選項

如需後端組態選項、請參閱下表：

參數	說明	預設
`version`		永遠為1

參數	說明	預設
`storageDriverName`	儲存驅動程式名稱	「ONTAP-NAS」、「ONTAP-NAS-節約 型」、「ONTAP-NAS-flexgroup」、「ONTAP-SAN」、「ONTAP-san經濟型」
`backendName`	自訂名稱或儲存後端	驅動程式名稱+「_」+ dataLIF
`managementLIF`	叢集的IP位址或SVM管理LIF若要順暢MetroCluster 切換、您必須指定SVM管理LIF。您可以指定完整網域名稱（FQDN）。如果使用安裝Astra Trident、則可設定使用IPv6位址 `--use-ipv6` 旗標。IPv6位址必須以方括弧來定義、例如[28e8：d9fb：a825：b7bf：69a8：d02f：9e7b：3555]。	「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的用途 "可選擇的LUN對應ONTAP" 探索建立多重路徑工作階段所需的iSCI LIF。如果發生此情況、將會產生警告 `dataLIF` 已明確定義。	源自SVM
`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]。設定為 `true` 用於Astra Trident設定及使用雙向CHAP做為後端SVM的預設驗證。請參閱 "準備使用ONTAP 支援的SAN驅動程式來設定後端" 以取得詳細資料。	錯
`chapInitiatorSecret`	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	「」
`chapTargetInitiatorSecret`	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`	「」
`chapTargetUsername`	目標使用者名稱。必要條件 `useCHAP=true`	「」
`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。	「」

參數	說明	預設
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。	「」
svm	要使用的儲存虛擬機器	如果是SVM則衍生 managementLIF 已指定
igroupName	要使用之SAN磁碟區的igroup名稱。請參閱 以取得更多資訊。	「Trident -<後端-UUID>」
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 Sendbackend、請勿指定 limitAggregateUsage。提供的 fsxadmin 和 vsadmin 請勿包含擷取Aggregate使用量所需的權限、並使用Astra Trident加以限制。	「」（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、除非您正在疑難排解並需要詳細的記錄傾印、否則請勿使用 {"API"：假、「method」：true } 。	null
useREST	使用ONTAP Isrest API的布林參數。技術預覽 useREST 以*技術預覽*的形式提供、建議用於測試環境、而非用於正式作業工作負載。設定為時 true、Astra Trident將使用ONTAP靜止API與後端進行通訊。此功能需要ONTAP 使用更新版本的版本。此外ONTAP 、所使用的登入角色必須能夠存取 ontap 應用程式：這是預先定義的 vsadmin 和 cluster-admin 角色： useREST 不支援MetroCluster 使用支援。	錯

詳細資料 `igroupName`

`igroupName` 可設定為ONTAP 已在叢集上建立的igroup。如果未指定、Astra Trident會自動建立名為的igroup `trident-<backend-UUID>`。

如果提供預先定義的igroupName、我們建議每個Kubernetes叢集使用一個igroup、如果要在不同環境之間共用SVM。這是Astra Trident自動維護IQN新增與刪除作業所必需的。

- `igroupName` 可更新以指向在Astra Trident以外的SVM上建立及管理的新igroup。
- `igroupName` 可省略。在此案例中、Astra Trident將建立並管理名為的igroup `trident-<backend-UUID>` 自動：

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

用於資源配置磁碟區的後端組態選項

您可以使用中的這些選項來控制預設資源配置 `defaults` 組態區段。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	「真的」
`spaceReserve`	空間保留模式；「無」（精簡）或「Volume」（完整）	「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。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「0」的磁碟區百分比	如果 `snapshotPolicy` 為「無」、否則為「」
`splitOnClone`	建立複本時、從其父複本分割複本	「假」

參數	說明	預設
`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。如需詳細資訊、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）"。	"
`securityStyle`	新磁碟區的安全樣式	`unix`
`tieringPolicy`	分層原則以使用「無」	ONTAP 9.5之前的SVM-DR組態為「純快照」

Volume資源配置範例

以下是已定義預設值的範例：

```
---
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
igroupName: custom
debugTraceFlags:
  api: false
  method: true
defaults:
  spaceReserve: volume
  qosPolicy: standard
  spaceAllocation: 'false'
  snapshotPolicy: default
  snapshotReserve: '10'
```

針對使用建立的所有Volume `ontap-san` 驅動程式Astra Trident在FlexVol 支援LUN中繼資料的過程中、額外增加10%的容量。LUN的配置大小與使用者在PVc中要求的大小完全相同。Astra Trident在FlexVol 整個過程中增加10%的速度（顯示ONTAP 在畫面上可用的尺寸）。使用者現在可以取得所要求的可用容量。此變更也可防止LUN成為唯讀、除非可用空間已充分利用。這不適用於ONTAP-san經濟型。

用於定義的後端 `snapshotReserve`、Astra Trident會依照下列方式計算Volume大小：

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

1.1是額外10%的Astra Trident加入FlexVol 到the支援LUN中繼資料的功能。適用於 `snapshotReserve` = 5%、而PVC要求= 5GiB、磁碟區總大小為5.79GiB、可用大小為5.5GiB。。 `volume show` 命令應顯示類似以下範例的結果：

```
Vserver   Volume       Aggregate    State      Type       Size  Available Used%
--------- ------------ ------------ ---------- ---- ---------- ---------- -----
          _pvc_89f1c156_3801_4de4_9f9d_034d54c395f4
                       online     RW         10GB     5.00GB    0%
          _pvc_e42ec6fe_3baa_4af6_996d_134adbbb8e6d
                       online     RW       5.79GB     5.50GB    0%
          _pvc_e8372153_9ad9_474a_951a_08ae15e1c0ba
                       online     RW          1GB    511.8MB    0%
3 entries were displayed.
```

目前、只有調整大小、才能將新計算用於現有的Volume。

最低組態範例

下列範例顯示基本組態、讓大部分參數保留預設值。這是定義後端最簡單的方法。

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

`ontap-san` 具有憑證型驗證的驅動程式

這是最小的後端組態範例。 `clientCertificate`、 `clientPrivateKey`和 `trustedCACertificate`（選用、如果使用信任的CA）會填入 `backend.json` 並分別取得用戶端憑證、私密金鑰及信任CA憑證的基礎64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
igroupName: trident
clientCertificate: ZXR0ZXJwYXB...ICMgJ3BhcGVyc2
clientPrivateKey: vciwKIyAgZG...0cnksIGRlc2NyaX
trustedCACertificate: zcyBbaG...b3Igb3duIGNsYXNz
```

`ontap-san` 使用雙向**CHAP**的驅動程式

這是最小的後端組態範例。此基本組態會建立 `ontap-san` 後端 `useCHAP` 設定為 `true`。

```
---
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
igroupName: trident
username: vsadmin
password: password
```

`ontap-san-economy` 驅動程式

```
---
version: 1
storageDriverName: ontap-san-economy
managementLIF: 10.0.0.1
svm: svm_iscsi_eco
useCHAP: true
chapInitiatorSecret: cl9qxIm36DKyawxy
chapTargetInitiatorSecret: rqxigXgkesIpwxyz
chapTargetUsername: iJF4heBRT0TCwxyz
chapUsername: uh2aNCLSd6cNwxyz
igroupName: trident
username: vsadmin
password: password
```

虛擬集區的後端範例

在下圖所示的範例後端定義檔案中、會針對所有儲存資源池設定特定的預設值、例如 `spaceReserve` 無、`spaceAllocation` 假、和 `encryption` 錯。虛擬資源池是在儲存區段中定義的。

Astra Trident會在「Comments」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。

在此範例中、有些儲存資源池會自行設定 `spaceReserve`、 `spaceAllocation`和 `encryption` 值、部分集區會覆寫上述設定的預設值。

```
---
version: 1
storageDriverName: ontap-san
managementLIF: 10.0.0.1
svm: svm_iscsi
useCHAP: true
chapInitiatorSecret: cl9qxIm36DKyawxy
chapTargetInitiatorSecret: rqxigXgkesIpwxyz
chapTargetUsername: iJF4heBRT0TCwxyz
chapUsername: uh2aNCLSd6cNwxyz
igroupName: trident
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'
```

以下是的iSCSI範例 `ontap-san-economy` 驅動程式：

```
---
version: 1
storageDriverName: ontap-san-economy
managementLIF: 10.0.0.1
svm: svm_iscsi_eco
useCHAP: true
chapInitiatorSecret: cl9qxIm36DKyawxy
chapTargetInitiatorSecret: rqxigXgkesIpwxyz
chapTargetUsername: iJF4heBRT0TCwxyz
chapUsername: uh2aNCLSd6cNwxyz
igroupName: trident
username: vsadmin
password: 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'
```

將後端對應至StorageClass

下列StorageClass定義是指上述虛擬資源池。使用 `parameters.selector` 欄位中、每個StorageClass會呼叫哪些虛擬資源池可用於裝載Volume。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 (`protection-gold`）將對應至中的第一個、第二個虛擬集區 `ontap-nas-flexgroup` 後端和中的第一個虛擬集區 `ontap-san` 後端：這是唯一提供金級保護的資源池。
- 第二個StorageClass (`protection-not-gold`）將對應至中的第三、第四個虛擬集區 `ontap-nas-flexgroup` 中的後端和第二個、第三個虛擬集區 `ontap-san` 後端：這是唯一提供金級以外保護層級的資源池。
- 第三個StorageClass (`app-mysqldb`）將對應至中的第四個虛擬資源池 `ontap-nas` 中的後端和第三個虛擬集區 `ontap-san-economy` 後端：這些是唯一提供mysqldb類型應用程式儲存池組態的集區。
- 第四個StorageClass (`protection-silver-creditpoints-20k`）將對應至中的第三個虛擬集區 `ontap-nas-flexgroup` 中的後端和第二個虛擬集區 `ontap-san` 後端：這些資源池是唯一能以20000個信用點數提供金級保護的資源池。
- 第五個StorageClass (`creditpoints-5k`）將對應至中的第二個虛擬資源池 `ontap-nas-economy` 中的後端和第三個虛擬集區 `ontap-san` 後端：這些是唯一提供5000個信用點數的資源池產品。

Astra Trident將決定選取哪個虛擬集區、並確保符合儲存需求。

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