使用 ONTAP 或 Cloud Volumes ONTAP SAN 驱动程序配置后端

Astra Trident

NetApp
April 16, 2024

目录

使用 ONTAP SAN 驱动程序配置后端

了解如何使用 ONTAP 或 Cloud Volumes ONTAP SAN 驱动程序配置 ONTAP 后端。

- "准备"
- "配置和示例"

用户权限

Astra Trident 应以 ONTAP 或 SVM 管理员身份运行，通常使用 `admin` cluster 用户或 `vsadmin` SVM 用户，或者使用具有相同角色的其他名称的用户。对于适用于 NetApp ONTAP 的 Amazon FSX 部署， Astra Trident 应使用集群 `fsxadmin` user 或 `vsadmin` SVM 用户或具有相同角色的其他名称的用户作为 ONTAP 或 SVM 管理员运行。`fsxadmin` 用户是集群管理员用户的有限替代用户。

> 如果使用 `limitAggregateUsage` 参数，则需要集群管理员权限。将适用于 NetApp ONTAP 的 Amazon FSx 与 Astra Trident 结合使用时， `limitAggregateUsage` 参数不适用于 `vsadmin` 和 `fsxadmin` 用户帐户。如果指定此参数，配置操作将失败。

准备

了解如何准备使用 ONTAP SAN 驱动程序配置 ONTAP 后端。对于所有 ONTAP 后端， Astra Trident 需要至少为 SVM 分配一个聚合。

请记住，您还可以运行多个驱动程序，并创建指向其中一个驱动程序的存储类。例如，您可以配置使用 `ontap-san` 驱动程序的 `san-dev` 类和使用 `ontap-san-economy-one` 的 `san-default` 类。

所有 Kubernetes 工作节点都必须安装适当的 iSCSI 工具。请参见 "此处" 有关详细信息：

身份验证

Astra Trident 提供了两种对 ONTAP 后端进行身份验证的模式。

- Credential Based ：具有所需权限的 ONTAP 用户的用户名和密码。建议使用 `admin` 或 `vsadmin` 等预定义的安全登录角色，以确保与 ONTAP 版本的最大兼容性。
- 基于证书： Astra Trident 还可以使用后端安装的证书与 ONTAP 集群进行通信。此处，后端定义必须包含客户端证书，密钥和可信 CA 证书的 Base64 编码值（如果使用）（建议）。

用户还可以选择更新现有后端，选择从基于凭据迁移到基于证书，反之亦然。如果 * 同时提供了凭据和证书 * ，则 Astra Trident 将在发出警告以从后端定义中删除凭据时默认使用证书。

启用基于凭据的身份验证

Astra Trident 需要 SVM 范围 / 集群范围的管理员的凭据才能与 ONTAP 后端进行通信。建议使用标准的预定义角色，例如 `admin` 或 `vsadmin` 。这样可以确保与未来的 ONTAP 版本向前兼容，这些版本可能会使功能 API 公开供未来的 Astra Trident 版本使用。可以创建自定义安全登录角色并将其用于 Astra Trident ，但不建议使用。

后端定义示例如下所示：

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

请注意，后端定义是凭据以纯文本格式存储的唯一位置。创建后端后，用户名 / 密码将使用 Base64 进行编码并存储为 Kubernetes 密钥。创建 / 更新后端是唯一需要了解凭据的步骤。因此，这是一项仅由管理员执行的操作，由 Kubernetes 或存储管理员执行。

启用基于证书的身份验证

新的和现有的后端可以使用证书并与 ONTAP 后端进行通信。后端定义需要三个参数。

- clientCertificate： 客户端证书的 Base64 编码值。
- clientPrivateKey： 关联私钥的 Base64 编码值。
- trustedCACertifate： 受信任 CA 证书的 Base64 编码值。如果使用可信 CA ，则必须提供此参数。如果不使用可信 CA ，则可以忽略此设置。

典型的工作流包括以下步骤。

步骤

1. 生成客户端证书和密钥。生成时，将公用名（ Common Name ， CN ）设置为要作为身份验证的 ONTAP 用户。

   ```
   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. 将可信 CA 证书添加到 ONTAP 集群。此问题可能已由存储管理员处理。如果未使用可信 CA ，则忽略。

   ```
   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 集群上安装客户端证书和密钥（从步骤 1 开始）。

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

4. 确认 ONTAP 安全登录角色支持 `cert` 身份验证方法。

```
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. 使用生成的证书测试身份验证。将 <SVM 管理 LIF> 和 <SVM 名称 > 替换为管理 LIF IP 和 ONTAP 名称。

```
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 对证书，密钥和可信 CA 证书进行编码。

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

7. 使用从上一步获得的值创建后端。

```
$ cat cert-backend.json
{
"version": 1,
"storageDriverName": "ontap-san",
"backendName": "SanBackend",
"managementLIF": "1.2.3.4",
"dataLIF": "1.2.3.8",
"svm": "vserver_test",
"clientCertificate": "Faaaakkkkeeee...Vaaallllluuuueeee",
"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 |
+------------+----------------+--------------------------------------
+--------+---------+
```

更新身份验证方法或轮换凭据

您可以更新现有后端以使用其他身份验证方法或轮换其凭据。这两种方式都适用：使用用户名 / 密码的后端可以更新为使用证书；使用证书的后端可以更新为基于用户名 / 密码的后端。为此，请使用更新后的 `backend.json` 文件，该文件包含执行 `tridentctl` **后端更新** 所需的参数。

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

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

> 轮换密码时，存储管理员必须先在 ONTAP 上更新用户的密码。然后进行后端更新。轮换证书时，可以向用户添加多个证书。之后，后端将更新以使用新证书，然后可以从 ONTAP 集群中删除旧证书。

更新后端不会中断对已创建卷的访问，也不会影响在之后建立的卷连接。成功的后端更新表明， Astra Trident 可以与 ONTAP 后端进行通信并处理未来的卷操作。

指定 igroup

Astra Trident 使用 igroup 来控制对其配置的卷（ LUN ）的访问。在为后端指定 igroup 时，管理员有两种选择：

- Astra Trident 可以自动为每个后端创建和管理 igroup 。如果后端定义中未包含 `igroupName` ，则 Astra Trident 会在 SVM 上创建一个名为 `trident -<backender-UUUUUID>` 的 igroup 。这将确保每个后端都有一个专用的 igroup ，并处理 Kubernetes 节点 IQN 的自动添加 / 删除。
- 或者，也可以在后端定义中提供预先创建的 igroup 。可以使用 `igroupName` config 参数来执行此操作。Astra Trident 会将 Kubernetes 节点 IQN 添加 / 删除到已有的 igroup 中。

对于已定义 `igroupName` 的后端，可以使用 `tridentctl` **后端更新** 删除 `igroupName` ，以使 Astra Trident

自动处理 igroup 。这样不会中断对已连接到工作负载的卷的访问。未来的连接将使用创建的 igroup Astra Trident 进行处理。

> 为 Astra Trident 的每个唯一实例指定一个 igroup 是一个最佳实践，对 Kubernetes 管理员和存储管理员都很有用。CSI Trident 可自动向 igroup 添加和删除集群节点 IQN ，从而极大地简化了其管理。在 Kubernetes 环境（以及 Astra Trident 安装）中使用相同的 SVM 时，使用专用的 igroup 可确保对一个 Kubernetes 集群所做的更改不会影响与另一个 Kubernetes 集群关联的 igroup 。此外，还必须确保 Kubernetes 集群中的每个节点都具有唯一的 IQN 。如上所述，Astra Trident 会自动处理 IQN 的添加和删除。在多个主机之间重复使用 IQN 可能会导致出现主机相互错误并拒绝访问 LUN 的不希望出现的情况。

如果将 Astra Trident 配置为充当 CSI 配置程序，则 Kubernetes 节点 IQN 会自动添加到 igroup 中或从 igroup 中删除。将节点添加到 Kubernetes 集群后， `trident - CSI` DemonSet 会在新添加的节点上部署一个 Pod （`trident - CSI - xxxxx` ），并注册可将卷连接到的新节点。节点 IQN 也会添加到后端的 igroup 中。在对节点进行隔离，清空并从 Kubernetes 中删除时，可以执行一组类似的步骤来删除 IQN 。

如果 Astra Trident 未作为 CSI 配置程序运行，则必须手动更新 igroup ，以包含 Kubernetes 集群中每个工作节点的 iSCSI IQN 。需要将加入 Kubernetes 集群的节点的 IQN 添加到 igroup 中。同样，必须从 igroup 中删除从 Kubernetes 集群中删除的节点的 IQN 。

使用双向 CHAP 对连接进行身份验证

Astra Trident 可以使用双向 CHAP 对 `ontap-san` 和 `ontap-san-economy-sn` 驱动程序的 iSCSI 会话进行身份验证。这需要在后端定义中启用 `useCHAP` 选项。如果设置为 `true` ，则 Astra Trident 会将 SVM 的默认启动程序安全性配置为双向 CHAP ，并从后端文件设置用户名和密码。NetApp 建议使用双向 CHAP 对连接进行身份验证。请参见以下配置示例：

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

> `useCHAP` 参数是一个布尔选项，只能配置一次。默认情况下，此参数设置为 false 。将其设置为 true 后，无法将其设置为 false 。

除了 `useCHAP=true` 之外，后端定义还必须包括 `chapInitiatorSecret` ， `chapTargetInitiatorSecret` ， `chapTargetUsername` 和 `chapUsername` 字段。通过运行

`tridentctl update` 创建后端，可以更改这些密钥。

工作原理

通过将 `useCHAP` 设置为 true ，存储管理员指示 Astra Trident 在存储后端配置 CHAP 。其中包括：

- 在 SVM 上设置 CHAP ：
  - 如果 SVM 的默认启动程序安全类型为 none （默认设置） * 和 * 卷中没有已存在的 LUN ，则 Astra Trident 会将默认安全类型设置为 `CHAP` ，然后继续配置 CHAP 启动程序以及目标用户名和密码。
  - 如果 SVM 包含 LUN ，则 Astra Trident 不会在 SVM 上启用 CHAP 。这样可以确保对 SVM 上已存在的 LUN 的访问不受限制。
- 配置 CHAP 启动程序以及目标用户名和密码；必须在后端配置中指定这些选项（如上所示）。
- 管理向后端提供的 `igroupName` 添加启动程序的操作。如果未指定，则默认为 `trident` 。

创建后端后， Astra Trident 会创建相应的 `tridentbackend` CRD ，并将 CHAP 密码和用户名存储为 Kubernetes 密码。此后端由 Astra Trident 创建的所有 PV 都将通过 CHAP 进行挂载和连接。

轮换凭据并更新后端

您可以通过更新 `backend.json` 文件中的 CHAP 参数来更新 CHAP 凭据。这需要更新 CHAP 密码并使用 `tridentctl update` 命令反映这些更改。

> 更新后端的 CHAP 密码时，必须使用 `tridentctl` 来更新后端。请勿通过 CLI/ONTAP UI 更新存储集群上的凭据，因为 Astra Trident 将无法选取这些更改。

```
$ cat backend-san.json
{
    "version": 1,
    "storageDriverName": "ontap-san",
    "backendName": "ontap_san_chap",
    "managementLIF": "192.168.0.135",
    "svm": "ontap_iscsi_svm",
    "useCHAP": true,
    "username": "vsadmin",
    "password": "FaKePaSsWoRd",
    "igroupName": "trident",
    "chapInitiatorSecret": "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 |
+----------------+----------------+--------------------------------------
+--------+---------+
```

现有连接将不受影响；如果凭据由 SVM 上的 Astra Trident 更新，则这些连接将继续保持活动状态。新连接将使用更新后的凭据，现有连接将继续保持活动状态。断开并重新连接旧的 PV 将导致它们使用更新后的凭据。

配置选项和示例

了解如何在您的 Astra Trident 安装中创建和使用 ONTAP SAN 驱动程序。本节提供了后端配置示例以及有关如何将后端映射到 StorageClasses 的详细信息。

后端配置选项

有关后端配置选项，请参见下表：

参数	Description	Default
`ve`版本		始终为 1

参数	Description	Default
storageDriverName	存储驱动程序的名称	"ontap-nas"， "ontap-nas-economy-"， "ontap-nas-flexgroup"， "ontap-san "， "ontap-san-economy-"
backendName	自定义名称或存储后端	驱动程序名称 + "_" + dataLIF
m年 月 日	集群或 SVM 管理 LIF 的 IP 地址	"10.0.0.1 "， "2001： 1234： abcd：：： fefe] "
dataLIF	协议 LIF 的 IP 地址。对于 IPv6，请使用方括号。设置后无法更新	由 SVM 派生，除非另有说明
使用 CHAP	使用 CHAP 对 iSCSI 的 ONTAP SAN 驱动程序进行身份验证 [布尔值]	false
chapInitiatorSecret	CHAP 启动程序密钥。如果为 useCHAP=true，则为必需项	"
标签	要应用于卷的一组任意 JSON 格式的标签	"
chapTargetInitiatorSecret	CHAP 目标启动程序密钥。如果为 useCHAP=true，则为必需项	"
chapUsername	入站用户名。如果为 useCHAP=true，则为必需项	"
chapTargetUsername	目标用户名。如果为 useCHAP=true，则为必需项	"
客户端证书	客户端证书的 Base64 编码值。用于基于证书的身份验证	"
clientPrivateKey	客户端专用密钥的 Base64 编码值。用于基于证书的身份验证	"
trustedCACertifate	受信任 CA 证书的 Base64 编码值。可选。用于基于证书的身份验证	"
用户名	用于连接到集群 /SVM 的用户名。用于基于凭据的身份验证	"
密码	连接到集群 /SVM 的密码。用于基于凭据的身份验证	"
sVM	要使用的 Storage Virtual Machine	如果指定了 SVM managementLIF，则派生
igroupName	要使用的 SAN 卷的 igroup 的名称	"trident — < 后端 UUID >"
s存储前缀	在 SVM 中配置新卷时使用的前缀。设置后无法更新	Trident
limitAggregateUsage	如果使用量超过此百分比，则配置失败。* 不适用于适用于 ONTAP 的 Amazon FSx *	" （默认情况下不强制实施）

参数	Description	Default
limitVolumeSize	如果请求的卷大小超过此值、则配置失败。	" （默认情况下不强制实施）
lunsPerFlexvol	每个 FlexVol 的最大 LUN 数，必须在 50 ， 200 范围内	100
debugTraceFlags	故障排除时要使用的调试标志。示例 ｛"api"： false ， "method "： true｝	空
useREST	用于使用 ONTAP REST API 的布尔参数。* 技术预览 *	false

> useREST 提供了一个 * 技术预览 * ，建议用于测试环境，而不是生产工作负载。如果设置为 true ，则 Astra Trident 将使用 ONTAP REST API 与后端进行通信。此功能需要使用 ONTAP 9.9 及更高版本。此外，使用的 ONTAP 登录角色必须能够访问 ONTAP 应用程序。这一点可通过预定义的 vsadmin 和 cluster-admin 角色来满足。

要与 ONTAP 集群通信，您应提供身份验证参数。这可以是安全登录的用户名 / 密码，也可以是已安装的证书。

> 如果您使用适用于 NetApp ONTAP 后端的 Amazon FSX ，请勿指定 limitAggregateUsage 参数。Amazon FSX for NetApp ONTAP 提供的 fsxadmin 和 vsadmin 角色不包含检索聚合使用情况并通过 Astra Trident 对其进行限制所需的访问权限。

> 请勿使用 debugTraceFlags ，除非您正在进行故障排除并需要详细的日志转储。

对于 ontap-san 驱动程序，默认使用 SVM 中的所有数据 LIF IP 并使用 iSCSI 多路径。为 ontap-san 驱动程序的 dataLIF 指定 IP 地址会强制其禁用多路径并仅使用指定的地址。

> 创建后端时，请记住，创建后无法修改 dataLIF 和 storagePrefix 。要更新这些参数，您需要创建一个新的后端。

可以将 igroupName 设置为已在 ONTAP 集群上创建的 igroup 。如果未指定，则 Astra Trident 会自动创建一个名为 trident -<backender-UUUUUID> 的 igroup 。如果要在环境之间共享 SVM ，则如果提供预定义的 igroupName ， NetApp 建议为每个 Kubernetes 集群使用一个 igroup 。这对于 Astra Trident 自动保持 IQN 添加 / 删除是必需的。

后端也可以在创建后更新 igroup ：

- 可以更新 igroupName 以指向在 Astra Trident 之外的 SVM 上创建和管理的新 igroup 。
- 可以省略 igroupName 。在这种情况下， Astra Trident 将自动创建和管理 trident -<backend-UUUUID> igroup 。

在这两种情况下，仍可访问卷附件。未来的卷附件将使用更新后的 igroup 。此更新不会中断对后端卷的访问。

可以为 managementLIF 选项指定完全限定域名 （ FQDN ）。

对于所有 ONTAP 驱动程序，也可以将 managementLIF 设置为 IPv6 地址。请务必使用 ` -use-ipv6` 标志安装 Trident 。必须注意在方括号内定义 managementLIF IPv6 地址。

> 使用 IPv6 地址时，请确保在方括号内定义 `managementLIF` 和 `dataLIF` （如果包含在后端定义中），例如 [28e8： d9fb： a825： b7bf： 69a8： d02f： 9e7b： 3555] 。如果未提供 `dataLIF` ，则 Astra Trident 将从 SVM 提取 IPv6 数据 LIF 。

要使 ontap-san 驱动程序能够使用 CHAP ，请在后端定义中将 `useCHAP` 参数设置为 `true` 。然后， Astra Trident 将配置双向 CHAP 并将其用作后端给定 SVM 的默认身份验证。请参见 "此处" 了解其工作原理。

对于 `ontap-san-economi` 驱动程序， `limitVolumeSize` 选项还会限制它为 qtree 和 LUN 管理的卷的最大大小。

> Astra Trident 会在使用 `ontap-san` 驱动程序创建的所有卷的 "Comments" 字段中设置配置标签。对于创建的每个卷， FlexVol 上的 "Comments" 字段将使用其所在存储池上的所有标签填充。存储管理员可以为每个存储池定义标签，并对存储池中创建的所有卷进行分组。这样，您就可以根据后端配置中提供的一组可自定义标签来方便地区分卷了。

用于配置卷的后端配置选项

您可以在配置的特殊部分中使用这些选项来控制默认配置每个卷的方式。有关示例，请参见以下配置示例。

参数	Description	Default
`spaceAllocation`	LUN 的空间分配	true
`s`页面预留	空间预留模式； " 无 " （精简）或 " 卷 " （厚）	无
`sSnapshot` 策略	要使用的 Snapshot 策略	无
`qosPolicy`	要为创建的卷分配的 QoS 策略组。选择每个存储池 / 后端的 qosPolicy 或 adaptiveQosPolicy 之一	"
`adaptiveQosPolicy`	要为创建的卷分配的自适应 QoS 策略组。选择每个存储池 / 后端的 qosPolicy 或 adaptiveQosPolicy 之一	"
`sSnapshot` 预留	为快照预留的卷百分比为 "0"	如果 `snapshotPolicy` 为 " 无 " ，则为 " 无 " ，否则为 " "
`splitOnClone`	创建克隆时，从其父级拆分该克隆	false
`splitOnClone`	创建克隆时，从其父级拆分该克隆	false
加密	启用 NetApp 卷加密	false
`securityStyle`	新卷的安全模式	"unix"
分层策略	使用 " 无 " 的分层策略	适用于 ONTAP 9.5 SVM-DR 之前的配置的 " 仅快照 "

> 在 Astra Trident 中使用 QoS 策略组需要 ONTAP 9.8 或更高版本。建议使用非共享 QoS 策略组，并确保策略组分别应用于每个成分卷。共享 QoS 策略组将对所有工作负载的总吞吐量实施上限。

下面是定义了默认值的示例：

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

> 对于使用 `ontap-san` 驱动程序创建的所有卷， Astra Trident 会向 FlexVol 额外添加 10% 的容量，以容纳 LUN 元数据。LUN 将使用用户在 PVC 中请求的确切大小进行配置。Astra Trident 将 FlexVol 增加 10% （在 ONTAP 中显示为可用大小）。用户现在将获得所请求的可用容量。此更改还可防止 LUN 变为只读状态，除非已充分利用可用空间。这不适用于 ontap-san-economy.

对于定义 `snapshotReserve` 的后端， Astra Trident 将按如下方式计算卷的大小：

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

1.1 是 Astra Trident 向 FlexVol 额外添加 10% 以容纳 LUN 元数据。对于 `snapshotReserve` = 5% ， PVC 请求 = 5GiB ，卷总大小为 5.79GiB ，可用大小为 5.5GiB 。`volume show` 命令应显示与以下示例类似的结果：

```
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，调整大小是对现有卷使用新计算的唯一方法。

最低配置示例

以下示例显示了将大多数参数保留为默认值的基本配置。这是定义后端的最简单方法。

> 如果您正在将 NetApp ONTAP 上的 Amazon FSx 与 Astra Trident 结合使用，建议为 LIF 指定 DNS 名称，而不是 IP 地址。

`ontap-san` 具有基于证书的身份验证的驱动程序

这是一个最低后端配置示例。`clientCertificate` ， `clientPrivateKey` 和 `trustedCACertifate` （如果使用可信 CA ，则可选）分别填充在 `backend.json` 中，并采用客户端证书，私钥和可信 CA 证书的 base64 编码值。

```
{
    "version": 1,
    "storageDriverName": "ontap-san",
    "backendName": "DefaultSANBackend",
    "managementLIF": "10.0.0.1",
    "dataLIF": "10.0.0.3",
    "svm": "svm_iscsi",
    "useCHAP": true,
    "chapInitiatorSecret": "cl9qxIm36DKyawxy",
    "chapTargetInitiatorSecret": "rqxigXgkesIpwxyz",
    "chapTargetUsername": "iJF4heBRT0TCwxyz",
    "chapUsername": "uh2aNCLSd6cNwxyz",
    "igroupName": "trident",
    "clientCertificate": "ZXR0ZXJwYXB...ICMgJ3BhcGVyc2",
    "clientPrivateKey": "vciwKIyAgZG...0cnksIGRlc2NyaX",
    "trustedCACertificate": "zcyBbaG...b3Igb3duIGNsYXNz"
}
```

`ontap-san` 具有双向**CHAP**的驱动程序

这是一个最低后端配置示例。此基本配置将创建一个 `ontap-san` 后端，并将 `useCHAP` 设置为 `true` 。

```
{
    "version": 1,
    "storageDriverName": "ontap-san",
    "managementLIF": "10.0.0.1",
    "dataLIF": "10.0.0.3",
    "svm": "svm_iscsi",
    "labels": {"k8scluster": "test-cluster-1", "backend": "testcluster1-
sanbackend"},
    "useCHAP": true,
    "chapInitiatorSecret": "cl9qxIm36DKyawxy",
    "chapTargetInitiatorSecret": "rqxigXgkesIpwxyz",
    "chapTargetUsername": "iJF4heBRT0TCwxyz",
    "chapUsername": "uh2aNCLSd6cNwxyz",
    "igroupName": "trident",
    "username": "vsadmin",
    "password": "secret"
}
```

`ontap-san-economy` 驱动程序

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

虚拟存储池后端示例

在下面显示的示例后端定义文件中，会为所有存储池设置特定的默认值，例如 `spaceReserve` at none ， `spaceAllocation` at false 和 `encryption` at false 。虚拟存储池在存储部分中进行定义。

在此示例中，某些存储池会设置自己的 `spaceReserve` ， `spaceAllocation` 和 `encryption` 值，而某些池会覆盖上述设置的默认值。

```
{
```

```
    "version": 1,
    "storageDriverName": "ontap-san",
    "managementLIF": "10.0.0.1",
    "dataLIF": "10.0.0.3",
    "svm": "svm_iscsi",
    "useCHAP": true,
    "chapInitiatorSecret": "cl9qxIm36DKyawxy",
    "chapTargetInitiatorSecret": "rqxigXgkesIpwxyz",
    "chapTargetUsername": "iJF4heBRT0TCwxyz",
    "chapUsername": "uh2aNCLSd6cNwxyz",
    "igroupName": "trident",
    "username": "vsadmin",
    "password": "secret",

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

```
                "encryption": "false"
            }
        }
    ]
}
```

以下是 `ontap-san-economy`-经济 驱动程序的 iSCSI 示例：

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

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

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

将后端映射到 StorageClasses

以下 StorageClass 定义引用了上述虚拟存储池。使用 `parameters.selector` 字段，每个 StorageClass 都会调用可用于托管卷的虚拟池。卷将在选定虚拟池中定义各个方面。

- 第一个 StorageClass （`protection-gold`）将映射到 `ontap-nas-flexgroup` 后端的第一个，第二个虚拟存储池以及 `ontap-san` 后端的第一个虚拟存储池。这是唯一一个提供黄金级保护的池。
- 第二个 StorageClass （`protection-not-gold`）将映射到 `ontap-nas-flexgroup` 后端的第三个，第四个虚拟存储池以及 `ontap-san` 后端的第二个，第三个虚拟存储池。这些池是唯一提供黄金级以外保护级别的池。
- 第三个 StorageClass （`app-mysqldb`）将映射到 `ontap-NAS` 后端的第四个虚拟存储池和 `ontap-san-economy-backend` 的第三个虚拟存储池。这些池是唯一为 mysqldb 类型的应用程序提供存储池配置的池。
- 第四个存储类（`protection-silver-creditpoins-20k`）将映射到 `ontap-nas-flexgroup` 后端的第三个虚拟存储池和 `ontap-san` 后端的第二个虚拟存储池。这些池是唯一以 20000 个信用点提供黄金级保护的池。
- 第五个存储类（`credits-5k`）将映射到 `ontap-nas-economy-backend` 中的第二个虚拟存储池和 `ontap-san` 后端的第三个虚拟存储池。这些是唯一一款具有 5000 个信用点的池产品。

Astra Trident 将决定选择哪个虚拟存储池，并确保满足存储要求。

```
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"
```