



ONTAP SAN 驱动程序

Astra Trident

NetApp
November 14, 2025

This PDF was generated from <https://docs.netapp.com/zh-cn/trident-2301/trident-use/ontap-san-prep.html> on November 14, 2025. Always check docs.netapp.com for the latest.

目录

使用 ONTAP SAN 驱动程序配置后端	1
用户权限	1
准备使用ONTAP SAN驱动程序配置后端	1
身份验证	1
指定 igrup	5
使用双向 CHAP 对连接进行身份验证	6
ONTAP SAN配置选项和示例	8
后端配置选项	8
用于配置卷的后端配置选项	11
最低配置示例	13
虚拟池后端示例	14
将后端映射到 StorageClasses	17

使用 ONTAP SAN 驱动程序配置后端

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

- "准备"
- "配置和示例"

Astra Control 可为使用创建的卷提供无缝保护、灾难恢复和移动性(在Kubernetes集群之间移动卷) ontap-nas, ontap-nas-flexgroup, 和 ontap-san 驱动程序。请参见 "[Astra Control复制前提条件](#)" 了解详细信息。

-  • 您必须使用 ontap-nas 适用于需要数据保护、灾难恢复和移动性的生产工作负载。
- 使用 ... ontap-san-economy 预期的卷使用量应远远高于ONTAP 支持的容量。
- 使用 ... ontap-nas-economy 仅当预期的卷使用量应远高于ONTAP 支持的容量时、以及 ontap-san-economy 无法使用驱动程序。
- 请勿使用 ontap-nas-economy 预测数据保护、灾难恢复或移动性的需求。

用户权限

Astra Trident 应以ONTAP 或SVM管理员身份运行、通常使用 admin 集群用户或 vsadmin SVM用户或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 应使用集群以ONTAP 或SVM管理员身份运行 fsxadmin 用户或 vsadmin SVM用户或具有相同角色的其他名称的用户。。fsxadmin 用户是集群管理员用户的有限替代用户。

 如果您使用 limitAggregateUsage 参数、需要集群管理员权限。在将适用于NetApp ONTAP 的Amazon FSx与Astra Trident结合使用时、会显示 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。

虽然可以在 ONTAP 中创建 Trident 驱动程序可以使用的限制性更强的角色，但我们不建议这样做。大多数新版本的 Trident 都会调用需要考虑的其他 API，从而使升级变得困难且容易出错。

准备使用ONTAP SAN驱动程序配置后端

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 san-dev 使用的类 ontap-san 驱动程序和 A san-default 使用的类 ontap-san-economy 一个。

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。建议使用预定义的安全登录角色、例如

admin 或 vsadmin 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）。

您可以更新现有后端、以便在基于凭据的方法和基于证书的方法之间移动。但是、一次仅支持一种身份验证方法。要切换到其他身份验证方法、必须从后端配置中删除现有方法。



如果您尝试同时提供*凭据和证书*、则后端创建将失败、并显示一条错误、指出配置文件中提供了多种身份验证方法。

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：

YAML

```
版本: 1
backendName: ExampleBackend
storageDriverName: ONTAP SAN管理
LIF: 10.0.0.1 SVM
: SVM_NFS
用户名: vsadmin
密码: password
```

JSON

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

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。创建或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。

- trustedCACertificate：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",  
  "svm": "vserver_test",  
  "clientCertificate": "Faaaakkkeeee...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，您必须删除现有身份验证方法并添加新的身份验证方法。然后、使用更新后的backend.json文件、该文件包含要执行的所需参数 `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 | 
+-----+-----+-----+
+-----+-----+

```

 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。条件 igroupName 不包含在后端定义中、Astra Trident 将创建一个名为的 igroup trident-<backend-UUID> 在 SVM 上。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。可以使用完成此操作 igroupName config 参数。Astra Trident 会将 Kubernetes 节点 IQN 添加 / 删除到已有的 igroup 中。

用于具有后端 igroupName 定义的、igroupName 可以使用删除 tridentctl backend update 使用 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 igrup 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 igrup 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 igrup 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 igrup。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 igrup 中或从 igrup 中删除。将节点添加到 Kubernetes 集群后、trident-csi DemonSet 部署 POD (trident-csi-xxxxx 在 23.01 或之前的版本中 trident-node<operating system>-xxxx 在 23.01 及更高版本中) 在新添加的节点上注册可将卷连接到的新节点。节点 IQN 也会添加到后端的 igrup 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 igrup，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 igrup 中。同样，必须从 igrup 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 iSCSI 会话进行身份验证 ontap-san 和 ontap-san-economy 驱动程序。这需要启用 useCHAP 选项。设置为时 true、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: password  
igrupName: trident  
chapInitiatorSecret: c19qxIm36DKyawxy  
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。

此外 useCHAP=true, chapInitiatorSecret, chapTargetInitiatorSecret, chapTargetUsername，和 chapUsername 后端定义中必须包含字段。在创建后端后、可以运行来更改这些密码 tridentctl update。

工作原理

通过设置 `useCHAP` 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）。
- 管理向添加启动程序的操作 `igroupName` 在后端提供。如果未指定，则默认为 `trident`。

创建后端后、Astra Trident 将创建相应的 `tridentbackend` CRD 并将 CHAP 密钥和用户名存储为 Kubernetes 密钥。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

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



更新后端的 CHAP 密码时，必须使用 `tridentctl` 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": "password",
    "igroupName": "trident",
    "chapInitiatorSecret": "c19qxUpDaTeD",
    "chapTargetInitiatorSecret": "rqxigXgkeUpDaTeD",
    "chapTargetUsername": "iJF4heBRT0TCwxyz",
    "chapUsername": "uh2aNCLSd6cNwxyz",
}
./tridentctl update backend ontap_san_chap -f backend-san.json -n trident
+-----+-----+
+-----+-----+
|   NAME          | STORAGE DRIVER |           UUID           |
STATE | VOLUMES |
+-----+-----+
+-----+-----+
| ontap_san_chap | ontap-san      | aa458f3b-ad2d-4378-8a33-1a472ffbeb5c |
online |       7 |
+-----+-----+
+-----+-----+

```

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。

ONTAP SAN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
version		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
managementLIF	集群或SVM管理LIF的IP地址要进行无缝MetroCluster 切换、必须指定SVM管理LIF。可以指定完全限定域名(FQDN)。如果使用安装了Astra Trident、则可以将设置为使用IPv6 地址 --use-ipv6 标志。IPv6地址必须用方括号定义、例如：[28e8 : d9fb: a825: b7bf: 69a8: d02f : 9e7b: 3555]。	"10.0.0.1" , "2001: 1234 : abcd :: fefe]"
dataLIF	协议 LIF 的 IP 地址。请勿为 <i>iSCSI</i> 指定。Astra Trident使用 " ONTAP 选择性LUN映射 " 发现建立多路径会话所需的 <i>iSCI</i> LIF。如果出现、则会生成警告 dataLIF 已明确定义。	由SVM派生
useCHAP	使用CHAP对 <i>iSCSI</i> 的ONTAP SAN 驱动程序进行身份验证[布尔值]。设置为 true 让Astra Trident为后端给定的SVM配置并使用双向CHAP作为默认身份验证。请参见 " 准备使用ONTAP SAN驱动程序配置后端 " 了解详细信息。	false
chapInitiatorSecret	CHAP 启动程序密钥。如果为、则为必需项 useCHAP=true	"
labels	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、则为必需项 useCHAP=true	"
chapUsername	入站用户名。如果为、则为必需项 useCHAP=true	"
chapTargetUsername	目标用户名。如果为、则为必需项 useCHAP=true	"
clientCertificate	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	"
trustedCACertificate	受信任 CA 证书的 Base64 编码值。可选。用于基于证书的身份验证。	"
username	与ONTAP 集群通信所需的用户名。用于基于凭据的身份验证。	"

参数	Description	Default
password	与ONTAP 集群通信所需的密码。用于基于凭据的身份验证。	"
svm	要使用的 Storage Virtual Machine	如果是SVM、则派生 managementLIF 已指定
igroupName	要使用的SAN卷的igroup的名称。请参见 有关详细信息 ...	"trident — <后端 UUID>"
storagePrefix	在 SVM 中配置新卷时使用的前缀。无法稍后修改。要更新此参数、您需要创建一个新的后端。	Trident
limitAggregateUsage	如果使用量超过此百分比，则配置失败。如果您使用适用于NetApp ONTAP 后端的Amazon FSX、请勿指定 limitAggregateUsage。提供的 fsxadmin 和 vsadmin 请勿包含检索聚合使用情况所需的权限、并使用Astra Trident对其进行限制。	" (默认情况下不强制实施)
limitVolumeSize	如果请求的卷大小超过此值、则配置失败。此外、还会限制它为qtree 和LUN管理的卷的最大大小。	" (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。例如、除非您正在进行故障排除并需要详细的日志转储、否则 {"api": false, "method": true} 请勿使用。	空
useREST	用于使用 ONTAP REST API 的布尔参数。* 技术预览 * useREST 作为一个*技术预览版提供、建议用于测试环境、而不是生产工作负载。设置为时 true 、Astra Trident将使用ONTAP REST API与后端进行通信。此功能需要使用ONTAP 9.11.1及更高版本。此外、使用的ONTAP 登录角色必须有权访问 ontap 应用程序。这一点可通过预定义来满足 vsadmin 和 cluster-admin 角色。useREST MetroCluster 不支持。	false

详细信息 ingroupName

ingroupName 可以设置为已在ONTAP 集群上创建的igroup。如果未指定、则Astra Trident会自动创建名为的igroup trident-<backend-UUID>。

如果要在环境之间共享SVM、则如果要提供预定义的ingroupName、建议为每个Kubernetes集群使用一个

igroup。这对于Astra Trident自动维护IQN添加和删除是必需的。

- igrup Name 可以更新为指向在Astra Trident之外的SVM上创建和管理的新igroup。
- igrup Name 可以省略。在这种情况下、Astra Trident将创建并管理名为的igroup trident-<backend-UUID> 自动。

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 igrup 。此更新不会中断对后端卷的访问。

用于配置卷的后端配置选项

您可以在中使用这些选项控制默认配置 defaults 配置部分。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
spaceReserve	空间预留模式； "无"（精简）或 "卷"（厚）	无
snapshotPolicy	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。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。	"
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	"
snapshotReserve	为快照预留的卷百分比为 "0"	条件 snapshotPolicy 为"无"、否则为""
splitOnClone	创建克隆时，从其父级拆分该克隆	false
encryption	在新卷上启用NetApp卷加密(NVE)；默认为 false。要使用此选项，必须在集群上获得 NVE 的许可并启用 NVE。如果在后端启用了NAE，则在Astra Trident中配置的任何卷都将启用NAE。有关详细信息、请参见： " Astra Trident如何与NVE和NAE配合使用 "。	false
luksEncryption	启用LUKS加密。请参见 " 使用Linux统一密钥设置(LUKS) "。	""
securityStyle	新卷的安全模式	unix
tieringPolicy	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"

卷配置示例

下面是定义了默认值的示例：

```
---
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'
```



用于使用创建的所有卷 ontap-san 驱动程序、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.

用于定义的后端 snapshotReserve、Astra Trident将按如下所示计算卷大小：

```
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。适用于 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_134adb8e6d		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 trustedCACertificate (如果使用可信CA，则可选)将填充 backend.json 和分别采用客户端证书、专用密钥和可信CA证书的base64 编码值。

```
---
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 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 ontap-san 后端 useCHAP 设置为 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: c19qxIm36DKyawxy
chapTargetInitiatorSecret: rqxigXgkesIpwxyz
chapTargetUsername: iJF4heBRT0TCwxyz
chapUsername: uh2aNCLSd6cNwxyz
igroupName: trident
username: vsadmin
password: password

```

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: c19qxIm36DKyawxy
chapTargetInitiatorSecret: rqxigXgkesIpwxyz
chapTargetUsername: iJF4heBRT0TCwxyz
chapUsername: uh2aNCLSd6cNwxyz
igroupName: trident
username: vsadmin
password: password

```

虚拟池后端示例

在下面所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 无、spaceAllocation 为false、和 encryption 为false。虚拟池在存储部分中进行定义。

Astra Trident会在"Comments"字段中设置配置标签。注释在FlexVol 上设置。在配置时、Astra Trident会将虚拟池上的所有标签复制到存储卷。为了方便起见、存储管理员可以按标签为每个虚拟池和组卷定义标签。

在此示例中、某些存储池会设置自己的存储池 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
svm: svm_iscsi
useCHAP: true
chapInitiatorSecret: c19qxIm36DKyawxy
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示例 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: 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: mysqladb
    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池。使用 `parameters.selector` 字段中、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`)将映射到中的第一个、第二个虚拟池 `ontap-nas-flexgroup` 中的后端和第一个虚拟池 `ontap-san` 后端。这是唯一一个提供黄金级保护的池。
- 第二个StorageClass (`protection-not-gold`)将映射到中的第三个、第四个虚拟池 `ontap-nas-flexgroup` 中的后端和第二个、第三个虚拟池 `ontap-san` 后端。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`)将映射到中的第四个虚拟池 `ontap-nas` 中的后端和第三个虚拟池 `ontap-san-economy` 后端。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StorageClass (`protection-silver-creditpoints-20k`)将映射到中的第三个虚拟池 `ontap-nas-flexgroup` 中的后端和第二个虚拟池 `ontap-san` 后端。这些池是唯一以 20000 个信用点提供黄金级保护的池。
- 第五个StorageClass (`creditpoints-5k`)将映射到中的第二个虚拟池 `ontap-nas-economy` 中的后端和第三个虚拟池 `ontap-san` 后端。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"

```

版权信息

版权所有 © 2025 NetApp, Inc.。保留所有权利。中国印刷。未经版权所有者事先书面许可，本文档中受版权保护的任何部分不得以任何形式或通过任何手段（图片、电子或机械方式，包括影印、录音、录像或存储在电子检索系统中）进行复制。

从受版权保护的 NetApp 资料派生的软件受以下许可和免责声明的约束：

本软件由 NetApp 按“原样”提供，不含任何明示或暗示担保，包括但不限于适销性以及针对特定用途的适用性的隐含担保，特此声明不承担任何责任。在任何情况下，对于因使用本软件而以任何方式造成的任何直接性、间接性、偶然性、特殊性、惩罚性或后果性损失（包括但不限于购买替代商品或服务；使用、数据或利润方面的损失；或者业务中断），无论原因如何以及基于何种责任理论，无论出于合同、严格责任或侵权行为（包括疏忽或其他行为），NetApp 均不承担责任，即使已被告知存在上述损失的可能性。

NetApp 保留在不另行通知的情况下随时对本文档所述的任何产品进行更改的权利。除非 NetApp 以书面形式明确同意，否则 NetApp 不承担因使用本文档所述产品而产生的任何责任或义务。使用或购买本产品不表示获得 NetApp 的任何专利权、商标权或任何其他知识产权许可。

本手册中描述的产品可能受一项或多项美国专利、外国专利或正在申请的专利的保护。

有限权利说明：政府使用、复制或公开本文档受 DFARS 252.227-7013（2014 年 2 月）和 FAR 52.227-19（2007 年 12 月）中“技术数据权利 — 非商用”条款第 (b)(3) 条规定的限制条件的约束。

本文档中所含数据与商业产品和/或商业服务（定义见 FAR 2.101）相关，属于 NetApp, Inc. 的专有信息。根据本协议提供的所有 NetApp 技术数据和计算机软件具有商业性质，并完全由私人出资开发。美国政府对这些数据的使用权具有非排他性、全球性、受限且不可撤销的许可，该许可既不可转让，也不可再许可，但仅限在与交付数据所依据的美国政府合同有关且受合同支持的情况下使用。除本文档规定的情形外，未经 NetApp, Inc. 事先书面批准，不得使用、披露、复制、修改、操作或显示这些数据。美国政府对国防部的授权仅限于 DFARS 的第 252.227-7015(b)（2014 年 2 月）条款中明确的权利。

商标信息

NetApp、NetApp 标识和 <http://www.netapp.com/TM> 上所列的商标是 NetApp, Inc. 的商标。其他公司和产品名称可能是其各自所有者的商标。