



# 配置后端

## Astra Trident

NetApp  
November 20, 2023

This PDF was generated from <https://docs.netapp.com/zh-cn/trident-2201/trident-use/anf.html> on November 20, 2023. Always check [docs.netapp.com](https://docs.netapp.com) for the latest.

# 目录

配置后端 . . . . .	1
配置 Azure NetApp Files 后端 . . . . .	1
为 GCP 后端配置 CVS . . . . .	7
配置 NetApp HCI 或 SolidFire 后端 . . . . .	18
使用 ONTAP 或 Cloud Volumes ONTAP SAN 驱动程序配置后端 . . . . .	24
使用 ONTAP NAS 驱动程序配置后端 . . . . .	42
将 Astra Trident 与适用于 NetApp ONTAP 的 Amazon FSX 结合使用 . . . . .	61

# 配置后端

后端定义了 Astra Trident 与存储系统之间的关系。它告诉 Astra Trident 如何与该存储系统进行通信，以及 Astra Trident 如何从该存储系统配置卷。Astra Trident 将自动从后端提供符合存储类定义的要求的存储池。了解有关根据您拥有的存储系统类型配置后端的更多信息。

- "配置 Azure NetApp Files 后端"
- "配置适用于 Google 云平台的 Cloud Volumes Service 后端"
- "配置 NetApp HCI 或 SolidFire 后端"
- "使用 ONTAP 或 Cloud Volumes ONTAP NAS 驱动程序配置后端"
- "使用 ONTAP 或 Cloud Volumes ONTAP SAN 驱动程序配置后端"
- "将 Astra Trident 与适用于 NetApp ONTAP 的 Amazon FSX 结合使用"

## 配置 Azure NetApp Files 后端

了解如何使用提供的示例配置将 Azure NetApp Files（ANF）配置为 Astra Trident 安装的后端。



Azure NetApp Files 服务不支持小于 100 GB 的卷。如果请求的卷较小，则 Astra Trident 会自动创建 100 GB 的卷。

您需要的内容

配置和使用 "[Azure NetApp Files](#)" 后端，您需要满足以下要求：

- `ssubscriptionID` 来自启用了 Azure NetApp Files 的 Azure 订阅。
- 租户 `ID`，`clientID` 和 `clientSecret` 来自 "[应用程序注册](#)" 在 Azure Active Directory 中，具有足够的 Azure NetApp Files 服务权限。应用程序注册应使用 `Azure 预定义的 owner` 或 `Contributor...` 角色。



要了解有关 Azure 内置角色的详细信息，请参见 "[Azure 文档](#)"。

- 至少包含一个的 Azure 位置 "[委派子网](#)"。从 Trident 22.01 开始，`location` 参数是后端配置文件顶层的必填字段。在虚拟池中指定的位置值将被忽略。
- 如果您是首次使用 Azure NetApp Files 或在新位置使用，则需要进行一些初始配置。请参见 "[《快速入门指南》](#)"。

关于此任务

根据后端配置（子网，虚拟网络，服务级别和位置），Trident 会在请求位置提供的容量池上创建 ANF 卷，并与请求的服务级别和子网匹配。



注意：Astra Trident 不支持手动 QoS 容量池。

## 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
storageDriverName	存储驱动程序的名称	"Azure-netapp-files"
backendName	自定义名称或存储后端	驱动程序名称 + " " + 随机字符
ssubscriptionID。	Azure 订阅中的订阅 ID	
租户 ID。	应用程序注册中的租户 ID	
客户端 ID。	应用程序注册中的客户端 ID	
clientSecret。	应用程序注册中的客户端密钥	
s服务级别	s标准，高级或超级之一	"" (随机)
位置	要创建新卷的 Azure 位置的名称	
s服务级别	s标准，高级或超级之一	"" (随机)
resourceGroups	用于筛选已发现资源的资源组列表	"[]" (无筛选器)
netappAccounts	用于筛选已发现资源的 NetApp 帐户列表	"[]" (无筛选器)
容量池	用于筛选已发现资源的容量池列表	"[]" (无筛选器，随机)
virtualNetwork	具有委派子网的虚拟网络的名称	""
ssubnet	委派给 Microsoft.Netapp/volumes 的子网的名称	""
nfsMountOptions	精细控制 NFS 挂载选项。	"nfsvers=3"
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)
debugTraceFlags	故障排除时要使用的调试标志。示例，` \ { "api" : false , "method" : true , "discovery" : true } `。除非您正在进行故障排除并需要详细日志转储，否则请勿使用此功能。	空

 如果在尝试创建 PVC 时遇到 "未找到容量池" 错误，则您的应用程序注册可能没有关联的所需权限和资源（子网，虚拟网络，容量池）。启用调试后，Astra Trident 将记录在创建后端时发现的 Azure 资源。请务必检查是否正在使用适当的角色。

 如果要使用 NFS 4.1 挂载卷，可以在逗号分隔的挂载选项列表中包含 nfsver=4 以选择 NFS v4.1。存储类中设置的任何挂载选项都会覆盖后端配置文件中设置的挂载选项。

可以使用短名称或完全限定名称指定 resourceGroups，netappAccounts，capacityPools，virtualNetwork 和 ssubnet 的值。短名称可以与多个同名资源匹配，因此在大多数情况下，建议使用完全限定名称。resourceGroups，netappAccounts 和 capacityPools 值是一种筛选器，可将发现的一组资源限制为此存储后端提供的资源，并可通过任意组合方式指定。完全限定名称的格式如下：

Type	格式。
Resource group	< 资源组 >
NetApp 帐户	< 资源组 >/< NetApp 帐户 >
容量池	< 资源组 >/< NetApp 帐户 >/< 容量池 >
虚拟网络	< 资源组 >/< 虚拟网络 >
Subnet	< 资源组 >/< 虚拟网络 >/< 子网 >

您可以通过在配置文件的特殊部分中指定以下选项来控制默认配置每个卷的方式。请参见以下配置示例。

参数	Description	Default
exportRule	新卷的导出规则	"0.0.0.0/0
snapshotDir	控制 .snapshot 目录的可见性	false
s 大小	新卷的默认大小	"100 克 "
unixPermissions	新卷的 UNIX 权限（4 个八进制数字）	"" （预览功能，需要在订阅中列入白名单）

exportRule 值必须是以 CIDR 表示法表示的 IPv4 地址或 IPv4 子网任意组合的逗号分隔列表。



对于在 ANF 后端创建的所有卷，Astra Trident 会在配置存储池时将存储池上的所有标签复制到该存储卷。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。

## 示例 1：最低配置

这是绝对的最低后端配置。使用此配置，Astra Trident 会发现在已配置位置委派给 ANF 的所有 NetApp 帐户，容量池和子网，并随机将新卷放置在其中一个池和子网上。

当您刚开始使用 ANF 并尝试执行相关操作时，此配置是理想的选择，但实际上，您希望为所配置的卷提供更多范围界定。

```
{
  "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"
}
```

## 示例 2：使用容量池筛选器的特定服务级别配置

此后端配置会将卷放置在 Azure 的 Easus 位置的 超高 容量池中。Astra Trident 会自动发现该位置委派给 ANF 的所有子网，并随机在其中一个子网上放置一个新卷。

```
{  
    "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"  
    ],  
}
```

## 示例 3：高级配置

此后端配置进一步将卷放置范围缩小为一个子网，并修改了某些卷配置默认值。

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

#### 示例 4：虚拟存储池配置

此后端配置可在同一个文件中定义多个存储池。如果您有多个容量池支持不同的服务级别，并且您希望在 Kubernetes 中创建表示这些服务级别的存储类，则此功能非常有用。

```

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

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

```

以下 `s` 存储类 定义是指上述存储池。通过使用 `parameters.selector` 字段，您可以为每个 `StorageClass` 指定用于托管卷的实际池。卷将在选定池中定义各个方面。

```
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
```

## 下一步是什么？

创建后端配置文件后，运行以下命令：

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

如果后端创建失败，则后端配置出现问题。您可以运行以下命令来查看日志以确定发生原因：

```
tridentctl logs
```

确定并更正配置文件中的问题后，您可以再次运行 `create` 命令。

## 为 GCP 后端配置 CVS

了解如何使用提供的示例配置将适用于 Google 云平台（GCP）的 NetApp Cloud Volumes Service（CVS）配置为 Astra Trident 安装的后端。



NetApp Cloud Volumes Service for Google Cloud 不支持大小小于 100 GiB 的 CVS-Performance 卷或小于 300 GiB 的 CVS 卷。如果请求的卷小于最小大小，则 Astra Trident 会自动创建最小大小的卷。

## 您需要的内容

以配置和使用 "适用于 Google Cloud 的 Cloud Volumes Service" 后端，您需要满足以下要求：

- 配置了 NetApp CVS 的 Google Cloud 帐户
- Google Cloud 帐户的项目编号
- 具有 `netappcloudvolumes.admin` 角色的 Google Cloud 服务帐户
- CVS 服务帐户的 API 密钥文件

Astra Trident 现在支持使用默认值的较小卷 "[GCP 上的 CVS 服务类型](#)"。用于使用创建的后端 `storageClass=software` 下、卷的最小配置大小将为 300 GiB。CVS 目前在 "受控可用性" 下提供此功能，不提供技术支持。用户必须注册才能访问低于 1TiB 的卷 "[此处](#)"。NetApp 建议客户对非生产工作负载使用 1 TiB 以下的卷。



使用默认 CVS 服务类型 (`storageClass=software`) 部署后端时，用户必须获得 GCP 上有关项目编号和项目 ID 的 1 TiB 卷功能的访问权限。这对于 Astra Trident 配置低于 1TiB 的卷是必需的。否则，对于小于 600 GiB 的 PVC，卷创建将失败。使用访问低于 1TiB 的卷 "[此表单](#)"。

由 Astra Trident 为默认 CVS 服务级别创建的卷将按以下方式进行配置：

- 小于 300 GiB 的 PVC 将导致 Astra Trident 创建 300 GiB CVS 卷。
- 如果 PVC 介于 300 GiB 到 600 GiB 之间，则 Astra Trident 将创建请求大小的 CVS 卷。
- 介于 600 GiB 和 1 TiB 之间的 PVC 将导致 Astra Trident 创建 1 TiB CVS 卷。
- 如果 PVC 大于 1 TiB，则 Astra Trident 将创建请求大小的 CVS 卷。

## 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
storageDriverName	存储驱动程序的名称	"GCP-CVS"
backendName	自定义名称或存储后端	驱动程序名称 + "_" + API 密钥的一部分
storageClass	存储类型可选择 硬件（性能优化）或 软件（CVS 服务类型）	
projectNumber	Google Cloud 帐户项目编号。该值可在 Google Cloud 门户的主页页面上找到。	
region	CVS 帐户区域。后端将在该区域配置卷。	

参数	Description	Default
apiKey	具有 netappcloudvolumes.admin 角色的 Google Cloud 服务帐户的 API 密钥。它包括 Google Cloud 服务帐户专用密钥文件的 JSON 格式的内容（逐字复制到后端配置文件）。	
代理 URL	代理服务器需要连接到 CVS 帐户时的代理 URL。代理服务器可以是 HTTP 代理，也可以是 HTTPS 代理。对于 HTTPS 代理，将跳过证书验证，以允许在代理服务器中使用自签名证书。不支持启用了身份验证的代理服务器。	
nfsMountOptions	精细控制 NFS 挂载选项。	"nfsvers=3"
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)
s服务级别	新卷的 CVS 服务级别。这些值包括 "standard"， "premer" 和 "Extreme"。	标准
网络	用于CVS卷的GCP网络	default
debugTraceFlags	故障排除时要使用的调试标志。示例，` \ { "api" : false , "method" : true" } `。除非您正在进行故障排除并需要详细的工作负载日志，否则请勿使用此功能。	空

如果使用共享 VPC 网络，则必须同时指定 `projectNumber` 和 `hostProjectNumber`。在这种情况下，`projectNumber` 是服务项目，而 `hostProjectNumber` 是主机项目。

"`apiRegion`" 表示 Astra Trident 在其中创建 CVS 卷的 GCP 区域。在 "跨区域 Kubernetes 集群时，在 "`apiRegion`" 中创建的 CVS 卷可用于在多个 GCP 区域的节点上计划的工作负载。请注意，跨区域流量会产生额外成本。

- 要启用跨区域访问，`allowedTopologies` 的 `StorageClass` 定义必须包括所有地区。例如：



```

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

```

- `storageClass` 是一个可选参数，您可以使用该参数选择所需的 "CVS 服务类型"。您可以从基本 CVS 服务类型 (`storageClass=software`) 或 CVS-Performance 服务类型 (`storageClass=hardware`) 中进行选择，Trident 默认使用此服务类型。请指定一个 "`apiRegion`"，用于在后端定义中提供相应的 CVS `storageClass`。



Astra Trident 与 Google Cloud 上的基本 CVS 服务类型集成是一项 \* 测试版功能 \*，不适用于生产工作负载。在 CVS-Performance 服务类型中，Trident 是 "\* 完全支持 "，默认情况下会使用它。

每个后端都会在一个 Google Cloud 区域中配置卷。要在其他区域创建卷，您可以定义其他后端。

您可以通过在配置文件的特殊部分中指定以下选项来控制默认配置每个卷的方式。请参见以下配置示例。

参数	Description	Default
exportRule	新卷的导出规则	"0.0.0.0/0
snapshotDir	访问 ` .snapshot` 目录	false
sSnapshot 预留	为快照预留的卷百分比	"" (接受 CVS 默认值为 0 )
s大小	新卷的大小	"100Gi"

`exportRule` 值必须是以 CIDR 表示法表示的 IPv4 地址或 IPv4 子网任意组合的逗号分隔列表。



对于在 CVS Google Cloud 后端创建的所有卷，Trident 会在配置存储池时将其上的所有标签复制到该存储卷。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。

### 示例 1：最低配置

这是绝对的最低后端配置。

```
{  
  "version": 1,  
  "storageDriverName": "gcp-cvs",  
  "projectNumber": "012345678901",  
  "apiRegion": "us-west2",  
  "apiKey": {  
    "type": "service_account",  
    "project_id": "my-gcp-project",  
    "private_key_id": "1234567890123456789012345678901234567890",  
    "private_key": "-----BEGIN PRIVATE KEY-----  
\\nznHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlZZE4jK3b1/qp8B4Kws8zX5ojY9m\\nznHczZ  
srrtHisIsAbOguSaPIKeyAZNchRAGzlZZE4jK3b1/qp8B4Kws8zX5ojY9m\\nznHczZsrrtHisIsAbOguSa  
PIKeyAZNchRAGzlZZE4jK3b1/qp8B4Kws8zX5ojY9m\\nznHczZsrrtHisIsAbOguSaPIKeyAZN  
chRAGzlZZE4jK3b1/qp8B4Kws8zX5ojY9m\\nznHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzl  
ZZE4jK3b1/qp8B4Kws8zX5ojY9m\\nznHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlZZE4jK3b1  
/qp8B4Kws8zX5ojY9m\\nznHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlZZE4jK3b1/qp8B4Kw  
s8zX5ojY9m\\nznHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlZZE4jK3b1/qp8B4Kws8zX5ojY  
9m\\nznHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlZZE4jK3b1/qp8B4Kws8zX5ojY9m\\nznH  
czZsrrtHisIsAbOguSaPIKeyAZNchRAGzlZZE4jK3b1/qp8B4Kws8zX5ojY9m\\nznHczZsrrtHi  
sIsAbOguSaPIKeyAZNchRAGzlZZE4jK3b1/qp8B4Kws8zX5ojY9m\\nznHczZsrrtHisIsAbOgu
```

#### 示例 2：基本 CVS 服务类型配置

此示例显示了使用基本 CVS 服务类型的后端定义，该服务类型适用于通用工作负载，可提供轻 / 中性能以及高区域可用性。

```
{  
  "version": 1,  
  "storageDriverName": "gcp-cvs",  
  "projectNumber": "012345678901",  
  "storageClass": "software",  
  "apiRegion": "us-east4",  
  "apiKey": {  
    "type": "service_account",  
    "project_id": "my-gcp-project",  
    "private_key_id": "1234567890123456789012345678901234567890",  
    "private_key": "-----BEGIN PRIVATE KEY-----  
\\nznHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzE4jK3b1/qp8B4Kws8zX5ojY9m\\nznHczZ  
srrtHisIsAbOguSaPIKeyAZNchRAGzlzE4jK3b1/qp8B4Kws8zX5ojY9m\\nznHczZsrrtHisI  
sAbOquSaPIKeyAZNchRAGzlzE4jK3b1/qp8B4Kws8zX5ojY9m\\nznHczZsrrtHisIsAbOquSa
```

```

PIKeyAZNchRAGzlZE4jK3b1/qp8B4Kws8zX5ojY9m\nznHczZsrrtHisIsAbOguSaPIKeyAZN
chRAGzlZE4jK3b1/qp8B4Kws8zX5ojY9m\nznHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzl
ZE4jK3b1/qp8B4Kws8zX5ojY9m\nznHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlZE4jK3b1
/qp8B4Kws8zX5ojY9m\nznHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlZE4jK3b1/qp8B4Kw
s8zX5ojY9m\nznHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlZE4jK3b1/qp8B4Kws8zX5ojY
9m\nznHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlZE4jK3b1/qp8B4Kws8zX5ojY9m\nznHc
zzsrrtHisIsAbOguSaPIKeyAZNchRAGzlZE4jK3b1/qp8B4Kws8zX5ojY9m\nznHczZsrrtHi
sIsAbOguSaPIKeyAZNchRAGzlZE4jK3b1/qp8B4Kws8zX5ojY9m\nznHczZsrrtHisIsAbOgu
SaPIKeyAZNchRAGzlZE4jK3b1/qp8B4Kws8zX5ojY9m\nznHczZsrrtHisIsAbOguSaPIKeyA
ZNchRAGzlZE4jK3b1/qp8B4Kws8zX5ojY9m\nznHczZsrrtHisIsAbOguSaPIKeyAZNchRAGz
lZE4jK3b1/qp8B4Kws8zX5ojY9m\nznHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlZE4jK3
b1/qp8B4Kws8zX5ojY9m\nznHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlZE4jK3b1/qp8B4
Kws8zX5ojY9m\nznHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlZE4jK3b1/qp8B4Kws8zX5o
jY9m\nznHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlZE4jK3b1/qp8B4Kws8zX5ojY9m\nzn
HczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlZE4jK3b1/qp8B4Kws8zX5ojY9m\nznHczZsrrt
HisIsAbOguSaPIKeyAZNchRAGzlZE4jK3b1/qp8B4Kws8zX5ojY9m\nznHczZsrrtHisIsAbO
guSaPIKeyAZNchRAGzlZE4jK3b1/qp8B4Kws8zX5ojY9m\nznHczZsrrtHisIsAbOguSaPIKe
yAZNchRAGzlZE4jK3b1/qp8B4Kws8zX5ojY9m\nznHczZsrrtHisIsAbOguSaPIKeyAZNchRA
GzlZE4jK3b1/qp8B4Kws8zX5ojY9m\nznHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlZE4j
K3b1/qp8B4Kws8zX5ojY9m\nnxsYg6gyxy4zq7OlwWgLwGa==\n----END PRIVATE
KEY----\n",
    "client_email": "cloudvolumes-admin-sa@my-gcp-
project.iam.gserviceaccount.com",
    "client_id": "123456789012345678901",
    "auth_uri": "https://accounts.google.com/o/oauth2/auth",
    "token_uri": "https://oauth2.googleapis.com/token",
    "auth_provider_x509_cert_url":
"https://www.googleapis.com/oauth2/v1/certs",
    "client_x509_cert_url":
"https://www.googleapis.com/robot/v1/metadata/x509/cloudvolumes-admin-
sa%40my-gcp-project.iam.gserviceaccount.com"
}
}

```

### 示例 3：单服务级别配置

此示例显示了一个后端文件，该文件对 Google Cloud us-west2 区域中由 Astra Trident 创建的所有存储应用相同的方面。此示例还显示了后端配置文件中使用的 proxyURL。

```
{
  "version": 1,
  "storageDriverName": "gcp-cvs",
  "projectNumber": "012345678901",
  "apiRegion": "us-west2",
  "apiKey": {

```



```
        "exportRule": "10.0.0.0/24,10.0.1.0/24,10.0.2.100",
        "size": "5Ti"
    }
}
```

#### 示例 4：虚拟存储池配置

此示例显示了使用虚拟存储池配置的后端定义文件以及引用这些池的 StorageClasses。

在下面显示的示例后端定义文件中，为所有存储池设置了特定的默认值，这些默认值会将 `snapshotReserve` 设置为 5%，并将 `exportRule` 设置为 0.0.0.0/0。虚拟存储池在 `s` 存储部分中进行定义。在此示例中，每个存储池都会设置自己的 `serviceLevel`，而某些池会覆盖默认值。

```

KEY----\n",
    "client_email": "cloudvolumes-admin-sa@my-gcp-
project.iam.gserviceaccount.com",
    "client_id": "123456789012345678901",
    "auth_uri": "https://accounts.google.com/o/oauth2/auth",
    "token_uri": "https://oauth2.googleapis.com/token",
    "auth_provider_x509_cert_url":
"https://www.googleapis.com/oauth2/v1/certs",
    "client_x509_cert_url":
"https://www.googleapis.com/robot/v1/metadata/x509/cloudvolumes-admin-
sa%40my-gcp-project.iam.gserviceaccount.com"
},
"nfsMountOptions": "vers=3,proto=tcp,timeo=600",

"defaults": {
    "snapshotReserve": "5",
    "exportRule": "0.0.0.0/0"
},
"labels": {
    "cloud": "gcp"
},
"region": "us-west2",

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

```

```

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

```

以下 StorageClass 定义引用了上述存储池。通过使用 parameters.selector 字段，您可以为每个 StorageClass 指定用于托管卷的虚拟池。卷将在选定池中定义各个方面。

第一个 StorageClass (`cvs-ext-protection`) 映射到第一个虚拟存储池。这是唯一一个可提供极高性能且 Snapshot 预留为 10% 的池。最后一个 StorageClass (`cvs-extra protection`) 调用提供 10% 快照预留的任何存储池。Astra Trident 决定选择哪个虚拟存储池，并确保满足快照预留要求。

```

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

```

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

## 下一步是什么？

创建后端配置文件后，运行以下命令：

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

如果后端创建失败，则后端配置出现问题。您可以运行以下命令来查看日志以确定发生原因：

```
tridentctl logs
```

确定并更正配置文件中的问题后，您可以再次运行 create 命令。

## 配置 NetApp HCI 或 SolidFire 后端

了解如何在 Astra Trident 安装中创建和使用 Element 后端。

您需要的内容

- 运行 Element 软件的受支持存储系统。
- NetApp HCI/SolidFire 集群管理员或租户用户的凭据，可用于管理卷。
- 所有 Kubernetes 工作节点都应安装适当的 iSCSI 工具。请参见 "[工作节点准备信息](#)"。

您需要了解的信息

solidfire-san 存储驱动程序支持两种卷模式：file 和 block。对于 filesystem volumemode，Astra Trident 将创建一个卷并创建一个文件系统。文件系统类型由 StorageClass 指定。

驱动程序	协议	卷模式	支持的访问模式	支持的文件系统
solidfire-san	iSCSI	块	rwo, rox, rwx	无文件系统。原始块设备。
solidfire-san	iSCSI	块	rwo, rox, rwx	无文件系统。原始块设备。
solidfire-san	iSCSI	文件系统	工单, ROX	xfs, ext3, ext4
solidfire-san	iSCSI	文件系统	工单, ROX	xfs, ext3, ext4



Astra Trident 在用作增强型 CSI 配置程序时使用 CHAP。如果您使用的是 CHAP（这是 CSI 的默认设置），则无需进行进一步准备。建议明确设置 UseCHAP 选项，以便对非 CSI Trident 使用 CHAP。否则，请参见 "[此处](#)"。



只有适用于 Astra Trident 的传统非 CSI 框架才支持卷访问组。如果配置为在 CSI 模式下运行，则 Astra Trident 将使用 CHAP。

如果未设置 AccessGroups 或 UseCHAP，则适用以下规则之一：

- 如果检测到默认的 trident 访问组，则会使用访问组。
- 如果未检测到访问组，并且 Kubernetes 版本为 1.7 或更高版本，则会使用 CHAP。

## 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
storageDriverName	存储驱动程序的名称	始终为 "solidfire-san"
backendName	自定义名称或存储后端	SolidFire + 存储（iSCSI）IP 地址
端点	使用租户凭据的 SolidFire 集群的 MVIP	
sVIP	存储（iSCSI）IP 地址和端口	
标签	要应用于卷的一组任意 JSON 格式的标签。	"
租户名称	要使用的租户名称（如果未找到，则创建）	
InitiatorIFace	将 iSCSI 流量限制为特定主机接口	default
UseCHAP	使用 CHAP 对 iSCSI 进行身份验证	true
访问组	要使用的访问组 ID 列表	查找名为 "trident" 的访问组的 ID
类型	QoS 规范	
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）
debugTraceFlags	故障排除时要使用的调试标志。示例 { "api": false , "method" : true }	空



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。



对于创建的所有卷，Astra Trident 会在配置存储池时将存储池上的所有标签复制到备用存储 LUN。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。

### 示例1：的后端配置 solidfire-san 具有三种卷类型的驱动程序

此示例显示了一个后端文件，该文件使用 CHAP 身份验证并使用特定 QoS 保证对三种卷类型进行建模。然后，您很可能会使用 IOPS storage class 参数定义存储类以使用其中的每种类型。

```
{  
    "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}}]  
}
```

## 示例2：的后端和存储类配置 solidfire-san 具有虚拟存储池的驱动程序

此示例显示了使用虚拟存储池配置的后端定义文件以及引用这些池的 StorageClasses。

在下面所示的示例后端定义文件中，为所有存储池设置了特定的默认值，即将 type 设置为 Silver。虚拟存储池在 s 存储部分中进行定义。在此示例中，某些存储池设置了自己的类型，而某些池将覆盖上述默认值。

```

{
  "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"
    }
  ]
}

```

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（`solidfire-gold-四`）将映射到第一个虚拟存储池。这是唯一一个性能卓越的池，

其卷类型 QoS 值为金牌。最后一个 StorageClass (solidfire-silver) 调用可提供银牌性能的任何存储池。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: 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"
```

## 了解更多信息

- "卷访问组"

# 使用 ONTAP 或 Cloud Volumes 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"

## 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。

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

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。
- 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",
  "dataLIF": "1.2.3.8",
  "svm": "vserver_test",
  "clientCertificate": "Faaaakkkeeee...Vaaallluuuueeee",
  "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-<backend-UUID>` 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 igrup。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 igrup 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 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 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 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": "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。

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": "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。

## 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	"
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	"
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。 <i>* 技术预览 *</i>	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日志转储。

对于 ontap-san 驱动程序，默然使用 SVM 中的所有数据 LIF IP 并使用 iSCSI 多路径。为 ontap-san 驱动程序的数据 LIF 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-<backend-UUID> 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-UUID> 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-economy 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_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）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": "c19qxIm36DKyawxy",  
    "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": "c19qxIm36DKyawxy",
  "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": "c19qxIm36DKyawxy",
  "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": "c19qxIm36DKyawxy",
"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"
        }
    }
]
}

```

```

        "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": "c19qxIm36DKyawxy",
  "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-creditpoints-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"
```

# 使用 ONTAP NAS 驱动程序配置后端

了解如何使用 ONTAP 和 Cloud Volumes ONTAP NAS 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。

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

了解如何准备使用 ONTAP NAS 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。

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，您可以配置一个使用 `ontap-nas 驱动程序` 的金牌类和一个使用 `ontap-nas-economy-one` 的铜牌类。`

所有 Kubernetes 工作节点都必须安装适当的 NFS 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-nas",  
    "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。
- 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=vsadmin"
```

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 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。您必须确保 LIF 的服务策略设置为 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 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-updated.json
{
  "version": 1,
  "storageDriverName": "ontap-nas",
  "backendName": "NasBackend",
  "managementLIF": "1.2.3.4",
  "dataLIF": "1.2.3.8",
  "svm": "vserver_test",
  "clientCertificate": "Faaaakkkeeee...Vaaalllluuuueeee",
  "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-nas",
  "backendName": "NasBackend",
  "managementLIF": "1.2.3.4",
  "dataLIF": "1.2.3.8",
  "svm": "vserver_test",
  "username": "vsadmin",
  "password": "secret",
  "storagePrefix": "myPrefix_"
}

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

```



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。

## 管理 NFS 导出策略

Astra Trident 使用 NFS 导出策略来控制对其配置的卷的访问。

使用导出策略时，Astra Trident 提供了两个选项：

- Astra Trident 可以动态管理导出策略本身；在此操作模式下，存储管理员会指定一个表示可接受 IP 地址的 CIDR 块列表。Astra Trident 会自动将属于这些范围的节点 IP 添加到导出策略中。或者，如果未指定任何 CIDR，则在节点上找到的任何全局范围的单播 IP 都将添加到导出策略中。
- 存储管理员可以手动创建导出策略和添加规则。除非在配置中指定了不同的导出策略名称，否则 Astra Trident 将使用默认导出策略。

## 动态管理导出策略

CSI Trident 20.04 版可以动态管理 ONTAP 后端的导出策略。这样，存储管理员就可以为工作节点 IP 指定允许的地址空间，而不是手动定义显式规则。它大大简化了导出策略管理；修改导出策略不再需要手动干预存储集群。此外，这有助于将对存储集群的访问限制为仅允许 IP 位于指定范围内的工作节点访问，从而支持精细的自动化管理。



只有 CSI Trident 才支持动态管理导出策略。请务必确保工作节点未被 NAT 处理。

### 示例

必须使用两个配置选项。下面是一个后端定义示例：

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



使用此功能时，您必须确保 SVM 中的根接合具有预先创建的导出策略，并具有允许节点 CIDR 块的导出规则（例如默认导出策略）。请始终遵循 NetApp 建议的最佳实践，为 Astra Trident 专用 SVM。

以下是使用上述示例对此功能的工作原理进行的说明：

- `autoExportPolicy` 设置为 `true`。这表示 Astra Trident 将为 `svm1` SVM 创建导出策略，并使用 `autoExportCIDRs` 地址块处理规则的添加和删除。例如，UUID 为 `403b5326-8482-40db-96d0-d83fb3f4daec` 且 `autoExportPolicy` 设置为 `true` 的后端会在 SVM 上创建一个名为 `trident-403b5326-8482-40db-96d0-d83fb3f4daec` 的导出策略。
- `autoExportCIDR` 包含地址块列表。此字段为可选字段，默认为 "`0.0.0.0/0`，`"::/0"`"。如果未定义，则 Astra Trident 会添加在工作节点上找到的所有全局范围的单播地址。

在此示例中，提供了 `192.168.0.0/24` 地址空间。这表示此地址范围内的 Kubernetes 节点 IP 将添加到 Astra Trident 创建的导出策略中。当 Astra Trident 注册其运行的节点时，它会检索该节点的 IP 地址，并根据 `autoExportCIDRs` 中提供的地址块对其进行检查。筛选 IP 后，Astra Trident 会为其发现的客户端 IP 创建导出策略规则，并为其标识的每个节点创建一个规则。

创建后，您可以为后端更新 `autoExportPolicy` 和 `autoExportCIDR`。您可以为自动管理的后端附加新的 CIDR，也可以删除现有的 CIDR。删除 CIDR 时请务必小心，以确保现有连接不会断开。您也可以选择对后端禁用 `autoExportPolicy`，并回退到手动创建的导出策略。这需要在后端配置中设置 `exportPolicy` 参数。

在 Astra Trident 创建或更新后端后，您可以使用 `tridentctl` 或相应的 `tridentbackend` CRD 检查后端：

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

当节点添加到 Kubernetes 集群并向 Astra Trident 控制器注册后，现有后端的导出策略将会更新（前提是它们位于后端的 `autoExportCIDR` 中指定的地址范围内）。

删除节点后，Astra Trident 会检查所有联机后端，以删除该节点的访问规则。通过从受管后端的导出策略中删除此节点 IP，Astra Trident 可防止恶意挂载，除非此 IP 可由集群中的新节点重复使用。

对于以前存在的后端，使用 `tridentctl update backend` 更新后端可确保 Astra Trident 自动管理导出策略。这将创建一个以后端 UUID 命名的新导出策略，后端上存在的卷将在重新挂载时使用新创建的导出策略。



删除具有自动管理导出策略的后端将删除动态创建的导出策略。如果重新创建后端，则会将其视为新的后端，并会创建新的导出策略。

如果更新了活动节点的 IP 地址，则必须在此节点上重新启动 Astra Trident Pod。然后，Astra Trident 将更新其管理的后端的导出策略，以反映此 IP 更改。

## 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 NAS 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
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
autosExportPolicy	启用自动创建和更新导出策略 [ 布尔值 ]	false
autosExportCIDR	启用 AutoExportPolicy 时用于筛选 Kubernetes 节点 IP 的 CIDR 列表	[ "0.0.0.0/0" , "::/0" ]
标签	要应用于卷的一组任意 JSON 格式的标签	"
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	"
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	"
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)
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

参数	Description	Default
debugTraceFlags	故障排除时要使用的调试标志。示例 { "api" : false , "method" : true }	空
nfsMountOptions	NFS 挂载选项的逗号分隔列表	"
qtreesPerFlexvol	每个 FlexVol 的最大 qtree 数，必须在 50 , 300 范围内	200
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。

 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。

可以为 managementLIF 选项指定完全限定域名 ( FQDN )。也可以为 dataLIF 选项指定 FQDN , 在这种情况下, FQDN 将用于 NFS 挂载操作。这样, 您就可以创建循环 DNS , 以便在多个数据 LIF 之间实现负载平衡。

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 Astra Trident 。必须在方括号内的 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 。

使用 autosExportPolicy 和 autosExportCIDR 选项, CSI Trident 可以自动管理导出策略。所有 ontap-nas-\* 驱动程序均支持此功能。

对于 ontap-nas-economy 驱动程序, limitVolumeSize 选项还会限制它为 qtree 和 FlexVol 管理的卷的最大大小, 而 qtreesPerFlexvol 选项允许自定义每个的最大 qtree 数。

可以使用 nfsMountOptions 参数指定挂载选项。Kubernetes 永久性卷的挂载选项通常在存储类中指定, 但如果在存储类中未指定挂载选项, 则 Astra Trident 将回退为使用存储后端配置文件中指定的挂载选项。如果在存储类或配置文件中未指定挂载选项, 则 Astra Trident 不会在关联的永久性卷上设置任何挂载选项。



Astra Trident 会在使用 ontap-NAS 和 ontap-nas-flexgroup 创建的所有卷的 "Comments" 字段中设置配置标签。根据所使用的驱动程序，注释将在 FlexVol (ontap-NAS) 或 FlexGroup (ontap-nas-flexgroup) 上进行设置。Astra Trident 会在配置存储池时将存储池上的所有标签复制到该存储卷。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。不受 ontap-nas-economy.	"
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
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"
unixPermissions	新卷的模式	777.
snapshotDir	控制 `snapshot` 目录的可见性	false
导出策略	要使用的导出策略	default
securityStyle	新卷的安全模式	"unix"



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-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"
    }
}

```

对于 ontap-nas 和 ontap-nas-flexgroups，Astra Trident 现在使用新的计算方法来确保 FlexVol 的大小正确，并使用 snapshotReserve 百分比和 PVC。当用户请求 PVC 时，Astra Trident 会使用新计算创建具有更多空间的原始 FlexVol。此计算可确保用户在 PVC 中收到所请求的可写空间，而不是小于所请求的空间。在 v21.07 之前，如果用户请求 PVC（例如，5GiB），并且 snapshotReserve 为 50%，则只会获得 2.5 GiB 的可写空间。这是因为用户请求的是整个卷，而 snapshotReserve 是其中的一个百分比。在 Trident 21.07 中，用户请求的是可写空间，Astra Trident 将 snapshotReserve number 定义为整个卷的百分比。这不适用于 ontap-nas-economy”。请参见以下示例以了解其工作原理：

计算方法如下：

```
Total volume size = (PVC requested size) / (1 - (snapshotReserve percentage) / 100)
```

对于 snapshotReserve = 50%，PVC 请求 = 5GiB，卷总大小为  $2/5 = 10\text{GiB}$ ，可用大小为 5GiB，这是用户在 PVC 请求中请求的大小。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_e8372153_9ad9_474a_951a_08ae15e1c0ba		online	RW	1GB	511.8MB	0%
2 entries were displayed.							

在升级 Astra Trident 时，先前安装的现有后端将按照上述说明配置卷。对于在升级之前创建的卷，您应调整其卷的大小，以便观察到所做的更改。例如，一个 2 GiB PVC，其 `snapshotReserve=50 earlier` 会导致一个卷提供 1 GiB 的可写空间。例如，将卷大小调整为 3GiB 可为应用程序在一个 6 GiB 卷上提供 3GiB 的可写空间。

## 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。



如果在采用 Trident 的 NetApp ONTAP 上使用 Amazon FSx，建议为 LIF 指定 DNS 名称，而不是 IP 地址。

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

ontap-nas 具有自动导出策略的驱动程序

此示例显示了如何指示 Astra Trident 使用动态导出策略自动创建和管理导出策略。这对于 `ontap-nas-economy` 和 `ontap-nas-flexgroup` 驱动程序也是如此。

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

#### ontap-nas-flexgroup 驱动程序

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

#### ontap-nas 支持IPv6的驱动程序

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

## ontap-nas-economy 驱动程序

```
{  
    "version": 1,  
    "storageDriverName": "ontap-nas-economy",  
    "managementLIF": "10.0.0.1",  
    "dataLIF": "10.0.0.2",  
    "svm": "svm_nfs",  
    "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。

## ontap-nas 驱动程序

```
{  
    {  
        "version": 1,  
        "storageDriverName": "ontap-nas",  
        "managementLIF": "10.0.0.1",  
        "dataLIF": "10.0.0.2",  
        "svm": "svm_nfs",  
        "username": "admin",  
        "password": "secret",  
        "nfsMountOptions": "nfsvers=4",  
  
        "defaults": {  
            "spaceReserve": "none",  
            "encryption": "false",  
            "qosPolicy": "standard"  
        },  
        "labels": {"store": "nas_store", "k8scluster": "prod-cluster-1"},  
        "region": "us_east_1",  
        "storage": [  
            {  
                "labels": {"app": "msoffice", "cost": "100"},  
                "zone": "us_east_1a",  
                "defaults": {  
                    "spaceReserve": "volume",  
                    "spaceAllocation": "true",  
                    "encryption": "true"  
                }  
            }  
        ]  
    }  
}
```

```

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

```

### ontap-nas-flexgroup 驱动程序

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

```

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

```

```
{  
    "version": 1,  
    "storageDriverName": "ontap-nas-economy",  
    "managementLIF": "10.0.0.1",  
    "dataLIF": "10.0.0.2",  
    "svm": "svm_nfs",  
    "username": "vsadmin",  
    "password": "secret",  
  
    "defaults": {  
        "spaceReserve": "none",  
        "encryption": "false"  
    },  
    "labels": {"store": "nas_economy_store"},  
    "region": "us_east_1",  
    "storage": [  
        {  
            "labels": {"department": "finance", "creditpoints": "6000"},  
            "zone": "us_east_1a",  
            "defaults": {  
                "spaceReserve": "volume",  
                "encryption": "true",  
                "unixPermissions": "0755"  
            }  
        },  
        {  
            "labels": {"department": "legal", "creditpoints": "5000"},  
            "zone": "us_east_1b",  
            "defaults": {  
                "spaceReserve": "none",  
                "encryption": "true",  
                "unixPermissions": "0755"  
            }  
        },  
        {  
            "labels": {"department": "engineering", "creditpoints": "3000"},  
            "zone": "us_east_1c",  
            "defaults": {  
                "spaceReserve": "none",  
                "encryption": "true",  
                "unixPermissions": "0775"  
            }  
        },  
        {  
    ]  
}
```

```

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

```

## 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-creditpoints-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"
```

# 将 Astra Trident 与适用于 NetApp ONTAP 的 Amazon FSX 结合使用

"适用于 NetApp ONTAP 的 Amazon FSX"是一种完全受管的 AWS 服务，支持客户启动和运行由 NetApp 的 ONTAP 存储操作系统提供支持的文件系统。Amazon FSX for NetApp ONTAP 支持您利用您熟悉的 NetApp 功能，性能和管理功能，同时利用在 AWS 上存储数据的简便性，灵活性，安全性和可扩展性。FSX 支持 ONTAP 的许多文件系统功能和管理 API。

文件系统是 Amazon FSX 中的主要资源，类似于内部部署的 ONTAP 集群。在每个 SVM 中，您可以创建一个或多个卷，这些卷是将文件和文件夹存储在文件系统中的数据容器。借助适用于 NetApp ONTAP 的 Amazon FSX，Data ONTAP 将作为云中的托管文件系统提供。新的文件系统类型称为 \* NetApp ONTAP \*。

通过将 Astra Trident 与适用于 NetApp ONTAP 的 Amazon FSx 结合使用，您可以确保在 Amazon Elastic Kubernetes Service (EKS) 中运行的 Kubernetes 集群可以配置由 ONTAP 备份的块和文件永久性卷。

## 创建适用于 ONTAP 的 Amazon FSX 文件系统

Trident 无法删除在启用了自动备份的 Amazon FSx 文件系统上创建的卷。要删除 PVC，您需要手动删除 PV 和 ONTAP 的 FSX 卷。

要防止此问题描述，请执行以下操作：

- 请勿使用 “快速创建” 来创建适用于 ONTAP 的 FSX 文件系统。快速创建工作流可启用自动备份，但不提供选择退出选项。
- 使用 “标准创建” 时，禁用自动备份。禁用自动备份可以使 Trident 成功删除卷，而无需进一步手动干预。



### ▼ Backup and maintenance - optional

#### Daily automatic backup [Info](#)

Amazon FSx can protect your data through daily backups

- Enabled  
 Disabled

## 了解 Astra Trident

如果您是 Astra Trident 的新用户，请使用下面提供的链接进行熟悉：

- [“常见问题解答”](#)
- [“使用 Astra Trident 的要求”](#)
- [“部署 Astra Trident”](#)
- [“配置适用于 NetApp ONTAP 的 ONTAP，Cloud Volumes ONTAP 和 Amazon FSX 的最佳实践”](#)
- [“集成 Astra Trident”](#)

- "[ONTAP SAN 后端配置](#)"
- "[ONTAP NAS 后端配置](#)"

详细了解驱动程序功能 "[此处](#)"。

适用于 NetApp ONTAP 的 Amazon FSX 使用 "[FabricPool](#)" 以管理存储层。通过它，您可以根据数据是否经常访问来将数据存储在层中。

Astra Trident 应以`vsadmin` SVM 用户或具有相同角色的其他名称的用户的身份运行。适用于 NetApp ONTAP 的 Amazon FSX 具有一个`fsxadmin` 用户，该用户是 ONTAP admin cluster 用户的有限替代。建议不要在 Trident 中使用`fsxadmin` 用户，因为`vsadmin` SVM 用户可以访问更多 Astra Trident 功能。

## 驱动程序

您可以使用以下驱动程序将 Astra Trident 与适用于 NetApp ONTAP 的 Amazon FSX 集成：

- `ontap-san`：配置的每个 PV 都是其自己的 Amazon FSX for NetApp ONTAP 卷中的一个 LUN。
- `ontap-san-economy`：为 NetApp ONTAP 卷配置的每个 PV 都是一个 LUN，每个 Amazon FSX 具有可配置的 LUN 数量。
- `ONONTAP_NAS`：配置的每个 PV 都是适用于 NetApp ONTAP 的完整 Amazon FSX 卷。
- `ontap-nas-economy.`：配置的每个 PV 都是一个 qtree，对于 NetApp ONTAP 卷，每个 Amazon FSX 的 qtree 数量是可配置的。
- `ontap-nas-flexgroup`：配置的每个 PV 都是适用于 NetApp ONTAP FlexGroup 卷的完整 Amazon FSX。

## 身份验证

Astra Trident 提供两种身份验证模式：

- 基于证书：Astra Trident 将使用 SVM 上安装的证书与 FSX 文件系统上的 SVM 进行通信。
- 基于凭据：您可以使用文件系统的 `fsxadmin` 用户或为 SVM 配置的 `vsadmin` 用户。



我们强烈建议使用`vsadmin` user 而非`fsxadmin` 来配置后端。Astra Trident 将使用此用户名和密码与 FSX 文件系统进行通信。

要了解有关身份验证的详细信息，请参见以下链接：

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

## 使用适用于 NetApp ONTAP 的 Amazon FSX 在 EKS 上部署和配置 Astra Trident

### 您需要的内容

- 已安装 `kubectl` 的现有 Amazon EKS 集群或自管理 Kubernetes 集群。
- 可从集群的工作节点访问的适用于 NetApp ONTAP 文件系统和 Storage Virtual Machine (SVM) 的现有 Amazon FSX。

- 为准备工作的节点 "NFS 和 / 或 iSCSI"。



确保按照 Amazon Linux 和 Ubuntu 所需的节点准备步骤进行操作 ["Amazon Machine 映像"（AMIS）](#)，具体取决于您的 EKS AMI 类型。

有关其他 Astra Trident 要求, 请参见 “[此处](#)”。

## 步骤

1. 使用 `./trident` 部署 Astra Trident。Get-started/Kubernetes 部署.html[ 部署方法^ ] 之一部署 Astra Trident。
  2. 按照以下步骤配置 Astra Trident：
    - a. 收集 SVM 的管理 LIF DNS 名称。例如，通过使用 AWS 命令行界面，在运行以下命令后，在 `Endpoints → Management` 下找到 `DNSName` 条目：

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

3. 创建并安装用于身份验证的证书。如果您使用的是 ontap-san 后端，请参见 "[此处](#)"。如果您使用的是 ontap-nas 后端，请参见 "[此处](#)"。



您可以从可以访问文件系统的任何位置使用 SSH 登录到文件系统（例如，安装证书）。使用 `fsxadmin` 用户，创建文件系统时配置的密码以及 `AWS FSx describe` 文件系统中的管理 DNS 名称。

4. 使用您的证书和管理 LIF 的 DNS 名称创建后端文件，如以下示例所示：

{

```
"version": 1,  
"storageDriverName": "ontap-san",  
"backendName": "customBackendName",  
"managementLIF": "svm-XXXXXXXXXXXXXXXXXX.fs-XXXXXXXXXXXXXXXXXX.fsx.us-  
east-2.aws.internal",  
"svm": "svm01",  
"clientCertificate": "ZXR0ZXJwYXB...ICMgJ3BhcGVyc2",  
"clientPrivateKey": "vciwKIyAgZG...0cnksIGRlc2Nyax",  
"trustedCACertificate": "zcyBbaG...b3Igb3duIGNsYXNz",  
}
```

有关创建后端的信息，请参见以下链接：

- "使用ONTAP NAS驱动程序配置后端"
  - "使用ONTAP SAN驱动程序配置后端"



请勿为 ontap-san 和 ontap-san-economy- 驱动程序指定 dataLIF，以允许 Astra Trident 使用多路径。



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。

部署完成后，执行以下步骤以创建 "存储类，配置卷以及将卷挂载到 Pod 中"。

## 了解更多信息

- ["Amazon FSX for NetApp ONTAP 文档"](#)
- ["有关适用于 NetApp ONTAP 的 Amazon FSX 的博客文章"](#)

## 版权信息

版权所有 © 2023 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。