配置后端

Astra Trident

NetApp
September 04, 2024

This PDF was generated from https://docs.netapp.com/zh-cn/trident-2301/trident-use/anf.html on September 04, 2024. Always check docs.netapp.com for the latest.

目录

配置后端 1
Azure NetApp Files 1
为Google Cloud后端配置Cloud Volumes Service 12
配置 NetApp HCI 或 SolidFire 后端 27
使用 ONTAP SAN 驱动程序配置后端 33
配置ONTAP NAS后端 51
适用于 NetApp ONTAP 的 Amazon FSX 74

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会自动从后端提供符合存储类定义的要求的存储池。了解如何为存储系统配置后端。

- "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"

Azure NetApp Files

配置 Azure NetApp Files 后端

您可以将Azure NetApp Files (ANF)配置为Astra Trident的后端。您可以使用ANF后端连接NFS和SMB卷。

- "准备"
- "配置选项和示例"

注意事项

- 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。
- Astra Trident仅支持将SMB卷挂载到Windows节点上运行的Pod。
- Astra Trident不支持Windows ARM 架构。

准备配置Azure NetApp Files 后端

在配置Azure NetApp Files 后端之前、您需要确保满足以下要求。

> 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来设置Azure NetApp Files 并创建NFS卷。请参见 "Azure：设置Azure NetApp Files 并创建NFS卷"。

NFS和SMB卷的前提条件

配置和使用 "Azure NetApp Files" 后端，您需要满足以下要求：

- 一个容量池。请参见 "Microsoft：为Azure NetApp Files 创建容量池"。
- 委派给Azure NetApp Files 的子网。请参见 "Microsoft：将子网委派给Azure NetApp Files"。

- `subscriptionID` 从启用了Azure NetApp Files 的Azure订阅。
- `tenantID`， `clientID`，和 `clientSecret` 从 "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以下任一项：
  - 所有者或贡献者角色 "由Azure预定义"。
  - 答 "自定义贡献者角色" 订阅级别 (`assignableScopes`)、并具有以下权限、这些权限仅限于Astra Trident所需的权限。创建自定义角色后、 "使用Azure门户分配角色"。

```
{
    "id": "/subscriptions/<subscription-
id>/providers/Microsoft.Authorization/roleDefinitions/<role-
definition-id>",
    "properties": {
        "roleName": "custom-role-with-limited-perms",
        "description": "custom role providing limited permissions",
        "assignableScopes": [
            "/subscriptions/<subscription-id>"
        ],
        "permissions": [
            {
                "actions": [

"Microsoft.NetApp/netAppAccounts/capacityPools/read",

"Microsoft.NetApp/netAppAccounts/capacityPools/write",

"Microsoft.NetApp/netAppAccounts/capacityPools/volumes/read",

"Microsoft.NetApp/netAppAccounts/capacityPools/volumes/write",

"Microsoft.NetApp/netAppAccounts/capacityPools/volumes/delete",

"Microsoft.NetApp/netAppAccounts/capacityPools/volumes/snapshots/read
",

"Microsoft.NetApp/netAppAccounts/capacityPools/volumes/snapshots/writ
e",

"Microsoft.NetApp/netAppAccounts/capacityPools/volumes/snapshots/dele
te",

"Microsoft.NetApp/netAppAccounts/capacityPools/volumes/subvolumes/rea
d",

"Microsoft.NetApp/netAppAccounts/capacityPools/volumes/subvolumes/wri
te",
```

```
"Microsoft.NetApp/netAppAccounts/capacityPools/volumes/subvolumes/del
ete",

"Microsoft.NetApp/netAppAccounts/capacityPools/volumes/subvolumes/Get
Metadata/action",

"Microsoft.NetApp/netAppAccounts/capacityPools/volumes/MountTargets/r
ead",
                    "Microsoft.Network/virtualNetworks/read",
                    "Microsoft.Network/virtualNetworks/subnets/read",

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

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

"Microsoft.Features/featureProviders/subscriptionFeatureRegistrations
/delete",
                    "Microsoft.Features/features/read",
                    "Microsoft.Features/operations/read",
                    "Microsoft.Features/providers/features/read",

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

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

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

- Azure `location` 至少包含一个 "委派子网"。自Trident 22.01日开始 `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。

SMB卷的其他要求

要创建SMB卷、您必须具有：

- 已配置Active Directory并连接到Azure NetApp Files。请参见 "Microsoft：创建和管理Azure NetApp Files的Active Directory连接"。

- 一个Kubernetes集群、其中包含一个Linux控制器节点以及至少一个运行Windows Server 2019的Windows工作节点。Astra Trident仅支持将SMB卷挂载到Windows节点上运行的Pod。
- 至少一个包含Active Directory凭据的Astra Trident密钥、以便Azure NetApp Files 可以向Active Directory进行身份验证。以生成密钥 `smbcreds`：

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

- 配置为Windows服务的CSI代理。配置 `csi-proxy`、请参见 "GitHub：CSI代理" 或 "GitHub：适用于Windows的CSI代理" 适用于在Windows上运行的Kubernetes节点。

Azure NetApp Files 后端配置选项和示例

了解ANF的NFS和SMB后端配置选项、并查看配置示例。

Astra Trident会使用后端配置(子网、虚拟网络、服务级别和位置)在请求的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。

> Astra Trident 不支持手动 QoS 容量池。

后端配置选项

ANF后端提供了这些配置选项。

参数	Description	Default
`version`		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
`subscriptionID`	Azure 订阅中的订阅 ID	
`tenantID`	应用程序注册中的租户 ID	
`clientID`	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	
`serviceLevel`	其中一个 `Standard`， `Premium` 或 `Ultra`	"" （随机）
`location`	要创建新卷的 Azure 位置的名称	
`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）
`capacityPools`	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	""

参数	Description	Default
`subnet`	委派给的子网的名称 `Microsoft.Netapp/volumes`	""
`networkFeatures`	一个卷的一组vNet功能可能是 `Basic` 或 `Standard`。网络功能并非在所有地区都可用、可能需要在订阅中启用。指定 `networkFeatures` 如果未启用此功能、则会导致卷配置失败。	""
`nfsMountOptions`	精细控制 NFS 挂载选项。SMB卷已忽略。要使用NFS 4.1挂载卷、请包括 `nfsvers=4` 在逗号分隔的挂载选项列表中选择NFS v4.1。存储类定义中设置的挂载选项会覆盖后端配置中设置的挂载选项。	"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的日志转储，否则请勿使用此功能。	空
`nasType`	配置NFS或SMB卷创建。选项包括 `nfs`， `smb` 或为空。默认情况下、将设置为空会将NFS卷设置为空。	`nfs`

有关网络功能的详细信息、请参见 "配置Azure NetApp Files 卷的网络功能"。

所需权限和资源

如果在创建PVC时收到"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)。如果启用了调试、则Astra Trident将记录创建后端时发现的Azure资源。验证是否正在使用适当的角色。

的值 `resourceGroups`， `netappAccounts`， `capacityPools`， `virtualNetwork`，和 `subnet` 可以使用短名称或完全限定名称来指定。在大多数情况下、建议使用完全限定名称、因为短名称可以与多个同名资源匹配。

。 `resourceGroups`， `netappAccounts`，和 `capacityPools` 值是指筛选器、用于将发现的一组资源限制为此存储后端可用的资源、并且可以以任意组合方式指定。完全限定名称采用以下格式：

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 >

Type	格式。
虚拟网络	< 资源组 >/< 虚拟网络 >
Subnet	< 资源组 >/< 虚拟网络 >/< 子网 >

卷配置

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卷配置。请参见 [示例配置] 了解详细信息。

参数	Description	Default
`exportRule`	新卷的导出规则。 `exportRule` 必须是以CIDR表示法表示的任意IPv4地址或IPv4子网组合的逗号分隔列表。SMB卷已忽略。	"0.0.0.0/0
`snapshotDir`	控制 .snapshot 目录的可见性	false
`size`	新卷的默认大小	"100 克 "
`unixPermissions`	新卷的UNIX权限(4个八进制数字)。SMB卷已忽略。	"" （预览功能，需要在订阅中列入白名单）

示例配置

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。因为 `nasType` 省略 `nfs` 默认情况下适用、后端将为NFS卷配置。

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中 `eastus` 位置 `Ultra` 容量池。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
networkFeatures: Standard
nfsMountOptions: vers=3,proto=tcp,timeo=600
limitVolumeSize: 500Gi
defaults:
  exportRule: 10.0.0.0/24,10.0.1.0/24,10.0.2.100
  snapshotDir: 'true'
  size: 200Gi
  unixPermissions: '0777'
```

示例**4**：虚拟池配置

此后端配置可在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。虚拟池标签用于根据区分池 `performance`。

```
---
version: 1
storageDriverName: azure-netapp-files
subscriptionID: 9f87c765-4774-fake-ae98-a721add45451
tenantID: 68e4f836-edc1-fake-bff9-b2d865ee56cf
clientID: dd043f63-bf8e-fake-8076-8de91e5713aa
clientSecret: SECRET
location: eastus
resourceGroups:
- application-group-1
networkFeatures: Basic
nfsMountOptions: vers=3,proto=tcp,timeo=600
labels:
  cloud: azure
storage:
- labels:
    performance: gold
  serviceLevel: Ultra
  capacityPools:
  - ultra-1
  - ultra-2
  networkFeatures: Standard
- labels:
    performance: silver
  serviceLevel: Premium
  capacityPools:
  - premium-1
- labels:
    performance: bronze
  serviceLevel: Standard
  capacityPools:
  - standard-1
  - standard-2
```

存储类定义

以下内容 `StorageClass` 定义是指上述存储池。

使用的示例定义 `parameter.selector` 字段

使用 `parameter.selector` 您可以为每个指定 `StorageClass` 用于托管卷的虚拟池。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
```

SMB卷的示例定义

使用 `nasType`， `node-stage-secret-name`，和 `node-stage-secret-namespace`、您可以指定SMB卷并提供所需的Active Directory凭据。

示例**1**：默认命名空间上的基本配置

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

示例**2**：每个命名空间使用不同的密钥

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

示例**3**：每个卷使用不同的密钥

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

> `nasType: `smb` 支持SMB卷的池的筛选器。 `nasType: `nfs` 或 `nasType: `null` NFS池的筛选器。

创建后端

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Google Cloud后端配置Cloud Volumes Service

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

了解适用于Google Cloud的Cloud Volumes Service 的Astra Trident支持

Astra Trident可以在两个卷中创建Cloud Volumes Service 卷之一 "服务类型"：

- **CVS-Performance**：默认的Astra Trident服务类型。这种性能优化的服务类型最适合重视性能的生产工作负载。CVS-Performance服务类型是一种硬件选项、支持的卷大小至少为100 GiB。您可以选择一个 "三个服务级别"：
  - `standard`
  - `premium`
  - `extreme`
- * CVS*：CVS服务类型提供高区域可用性、性能级别限制为中等。CVS服务类型是一个软件选项、可使用存储池支持小至1 GiB的卷。存储池最多可包含50个卷、其中所有卷都共享池的容量和性能。您可以选择一个 "两个服务级别"：
  - `standardsw`
  - `zoneredundantstandardsw`

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 Cloud Volumes Service 的Google Cloud帐户
- Google Cloud 帐户的项目编号
- Google Cloud服务帐户 `netappcloudvolumes.admin` role

- Cloud Volumes Service 帐户的API密钥文件

后端配置选项

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。

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`	用于指定CVS服务类型的可选参数。使用 … `software` 选择CVS服务类型。否则、Astra Trident将采用CVS-Performance服务类型（`hardware`）。	
`storagePools`	仅限CVS服务类型。用于指定用于创建卷的存储池的可选参数。	
`projectNumber`	Google Cloud 帐户项目编号。此值可在Google Cloud门户主页上找到。	
`hostProjectNumber`	如果使用共享VPC网络、则为必需项。在此情景中、`projectNumber` 是服务项目、和 `hostProjectNumber` 是主机项目。	
`apiRegion`	Astra Trident创建Cloud Volumes Service 卷的Google云区域。创建跨区域Kubernetes集群时、在中创建的卷 `apiRegion` 可用于在多个Google Cloud地区的节点上计划的工作负载。跨区域流量会产生额外成本。	
`apiKey`	Google Cloud服务帐户的API密钥 `netappcloudvolumes.admin` 角色。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）。	
`proxyURL`	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"

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）
`serviceLevel`	新卷的CVS-Performance或CVS服务级别。CVS-Performance值为 `standard`， `premium`或 `extreme`。CVS值为 `standardsw` 或 `zoneredundantstandardsw`。	CVS-Performance默认值为"standard"。CVS默认值为"standardsw"。
`network`	用于Cloud Volumes Service 卷的Google云网络。	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日志转储，否则请勿使用此功能。	空

`allowedTopologies`	要启用跨区域访问、请定义StorageClass `allowedTopologies` 必须包括所有地区。例如：
	`- key: topology.kubernetes.io/region`
	`values:`
	`- us-east1`
	`- europe-west1`	

卷配置选项

您可以在中控制默认卷配置 `defaults` 部分。

参数	Description	Default
`exportRule`	新卷的导出规则。必须是以 CIDR 表示法表示的任意 IPv4 地址或 IPv4 子网组合的逗号分隔列表。	"0.0.0.0/0
`snapshotDir`	访问 `.snapshot` 目录	false
`snapshotReserve`	为快照预留的卷百分比	"" （接受 CVS 默认值为 0 ）
`size`	新卷的大小。CVS性能最小值为100 GiB。CVS最小值为1 GiB。	CVS-Performance服务类型默认为"100GiB"。CVS服务类型未设置默认值、但至少需要1 GiB。

CVS-Performance服务类型示例

以下示例提供了CVS-Performance服务类型的示例配置。

示例 **1**： 最低配置

这是使用默认CVS-Performance服务类型以及默认"标准"服务级别的最小后端配置。

```
---
version: 1
storageDriverName: gcp-cvs
projectNumber: '012345678901'
apiRegion: us-west2
apiKey:
  type: service_account
  project_id: my-gcp-project
  private_key_id: "<id_value>"
  private_key: |
    -----BEGIN PRIVATE KEY-----
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    XsYg6gyxy4zq7OlwWgLwGa==
    -----END PRIVATE KEY-----
  client_email: cloudvolumes-admin-sa@my-gcp-
project.iam.gserviceaccount.com
  client_id: '123456789012345678901'
  auth_uri: https://accounts.google.com/o/oauth2/auth
```

```
  token_uri: https://oauth2.googleapis.com/token
  auth_provider_x509_cert_url:
https://www.googleapis.com/oauth2/v1/certs
  client_x509_cert_url:
https://www.googleapis.com/robot/v1/metadata/x509/cloudvolumes-admin-
sa%40my-gcp-project.iam.gserviceaccount.com
```

示例2：服务级别配置

此示例说明了后端配置选项、包括服务级别和卷默认值。

```
---
version: 1
storageDriverName: gcp-cvs
projectNumber: '012345678901'
apiRegion: us-west2
apiKey:
  type: service_account
  project_id: my-gcp-project
  private_key_id: "<id_value>"
  private_key: |
    -----BEGIN PRIVATE KEY-----
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    XsYg6gyxy4zq7OlwWgLwGa==
    -----END PRIVATE KEY-----
  client_email: cloudvolumes-admin-sa@my-gcp-
project.iam.gserviceaccount.com
  client_id: '123456789012345678901'
  auth_uri: https://accounts.google.com/o/oauth2/auth
```

```
  token_uri: https://oauth2.googleapis.com/token
  auth_provider_x509_cert_url:
https://www.googleapis.com/oauth2/v1/certs
  client_x509_cert_url:
https://www.googleapis.com/robot/v1/metadata/x509/cloudvolumes-admin-
sa%40my-gcp-project.iam.gserviceaccount.com
proxyURL: http://proxy-server-hostname/
nfsMountOptions: vers=3,proto=tcp,timeo=600
limitVolumeSize: 10Ti
serviceLevel: premium
defaults:
  snapshotDir: 'true'
  snapshotReserve: '5'
  exportRule: 10.0.0.0/24,10.0.1.0/24,10.0.2.100
  size: 5Ti
```

示例3：虚拟池配置

此示例使用 `storage` 配置虚拟池和 `StorageClasses` 这是指它们。请参见 [存储类定义] 以查看存储类的定义方式。

此处为设置的所有虚拟池设置了特定的默认值 `snapshotReserve` 5%和 `exportRule` 到0.0.0.0/0。虚拟池在中进行定义 `storage` 部分。每个虚拟池都定义了自己的虚拟池 `serviceLevel`、并且某些池会覆盖默认值。虚拟池标签用于根据区分池 `performance` 和 `protection`。

```
---
version: 1
storageDriverName: gcp-cvs
projectNumber: '012345678901'
apiRegion: us-west2
apiKey:
  type: service_account
  project_id: my-gcp-project
  private_key_id: "<id_value>"
  private_key: |
    -----BEGIN PRIVATE KEY-----
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    XsYg6gyxy4zq7OlwWgLwGa==
```

```
    -----END PRIVATE KEY-----
  client_email: cloudvolumes-admin-sa@my-gcp-
project.iam.gserviceaccount.com
  client_id: '123456789012345678901'
  auth_uri: https://accounts.google.com/o/oauth2/auth
  token_uri: https://oauth2.googleapis.com/token
  auth_provider_x509_cert_url:
https://www.googleapis.com/oauth2/v1/certs
  client_x509_cert_url:
https://www.googleapis.com/robot/v1/metadata/x509/cloudvolumes-admin-
sa%40my-gcp-project.iam.gserviceaccount.com
nfsMountOptions: vers=3,proto=tcp,timeo=600
defaults:
  snapshotReserve: '5'
  exportRule: 0.0.0.0/0
labels:
  cloud: gcp
region: us-west2
storage:
- labels:
    performance: extreme
    protection: extra
  serviceLevel: extreme
  defaults:
    snapshotDir: 'true'
    snapshotReserve: '10'
    exportRule: 10.0.0.0/24
- labels:
    performance: extreme
    protection: standard
  serviceLevel: extreme
- labels:
    performance: premium
    protection: extra
  serviceLevel: premium
  defaults:
    snapshotDir: 'true'
    snapshotReserve: '10'
- labels:
    performance: premium
    protection: standard
  serviceLevel: premium
- labels:
    performance: standard
  serviceLevel: standard
```

存储类定义

以下StorageClass定义适用于虚拟池配置示例。使用 `parameters.selector`、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。

存储类示例

```
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
```

- 第一个StorageClass (`cvs-extreme-extra-protection`)映射到第一个虚拟池。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决定选择哪个虚拟池、并确保满足快照预留要求。

CVS服务类型示例

以下示例提供了CVS服务类型的示例配置。

示例1：最低配置

这是使用的最低后端配置 `storageClass` 指定CVS服务类型和默认值 `standardsw` 服务级别。

```
---
version: 1
storageDriverName: gcp-cvs
projectNumber: '012345678901'
storageClass: software
apiRegion: us-east4
apiKey:
  type: service_account
  project_id: my-gcp-project
  private_key_id: "<id_value>"
  private_key: |
    -----BEGIN PRIVATE KEY-----
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    XsYg6gyxy4zq7OlwWgLwGa==
    -----END PRIVATE KEY-----
  client_email: cloudvolumes-admin-sa@my-gcp-
project.iam.gserviceaccount.com
  client_id: '123456789012345678901'
```

```
  auth_uri: https://accounts.google.com/o/oauth2/auth
  token_uri: https://oauth2.googleapis.com/token
  auth_provider_x509_cert_url:
https://www.googleapis.com/oauth2/v1/certs
  client_x509_cert_url:
https://www.googleapis.com/robot/v1/metadata/x509/cloudvolumes-admin-
sa%40my-gcp-project.iam.gserviceaccount.com
serviceLevel: standardsw
```

示例2：存储池配置

此示例后端配置使用 `storagePools` 配置存储池。

```
---
version: 1
storageDriverName: gcp-cvs
backendName: gcp-std-so-with-pool
projectNumber: '531265380079'
apiRegion: europe-west1
apiKey:
  type: service_account
  project_id: cloud-native-data
  private_key_id: "<id_value>"
  private_key: |-
    -----BEGIN PRIVATE KEY-----
    MIIEvAIBADANBgkqhkiG9w0BAQEFAASCBKYwggSiAgEAAoIBAQDaT+Oui9FBAw19
    L1AGEkrYU5xd9K5NlO5jMkIFND5wCD+Nv+jd1GvtFRLaLK5RvXyF5wzvztmODNS+
    qtScpQ+5cFpQkuGtv9U9+N6qtuVYYO3b504Kp5CtqVPJCgMJaK2j8pZTIqUiMum/
    5/Y9oTbZrjAHSMgJm2nHzFq2X0rqVMaHghI6ATm4DOuWx8XGWKTGIPlc0qPqJlqS
    LLaWOH4VIZQZCAyW5IUp9CAmwqHgdG0uhFNfCgMmED6PBUvVLsLvcq86X+QSWR9k
    ETqElj/sGCenPF7ti1DhGBFafd9hPnxg9PZY29ArEZwY9G/ZjZQX7WPgs0VvxiNR
    DxZRC3GXAgMBAAECggEACn5c59bG/qnVEVI1CwMAalM5M2z09JFhlLlljKwntNPj
    Vilw2eTW2+UE7HbJru/S7KQgA5Dnn9kvCraEahPRuddUMrD0vG4kTl/IODV6uFuk
    Y0sZfbqd4jMUQ21smvGsqFzwloYWS5qzO1W83ivXH/HW/iqkmY2eW+EPRS/hwSSu
    SscR+SojI7PB0BWSJhlV4yqYf3vcD/D95el2CVHfRCkL85DKumeZ+yHEnpiXGZAE
    t8xSs4a5OOPm6NHhevCw2a/UQ95/foXNUR450HtbjieJo5o+FF6EYZQGfU2ZHZO8
    37FBKuaJkdGW5xqaI9TL7aqkGkFMF4F2qvOZM+vy8QKBgQD4oVuOkJDlhkTHP86W
    esFlw1kpWyJR9ZA7LI0g/rVpslnX+XdDq0WQf4umdLNau5hYEH9LU6ZSGs1Xk3/B
    NHwR6OXFuqEKNiu83d0zSlHhTy7PZpOZdj5a/vVvQfPDMz7OvsqLRd7YCAbdzuQ0
    +Ahq0Ztwvg0HQ64hdW0ukpYRRwKBgQDgyHj98oqswoYuIa+pP1yS0pPwLmjwKyNm
    /HayzCp+Qjiyy7Tzg8AUqlH1Ou83XbV428jvg7kDhO7PCCKFq+mMmfqHmTpb0Maq
    KpKnZg4ipsqP1yHNNEoRmcailXbwIhCLewMqMrggUiLOmCw4PscL5nK+4GKu2XE1
    jLqjWAZFMQKBgFHkQ9XXRAJ1kR3XpGHoGN890pZOkCVSrqju6aUef/5KYlFCt8ew
    F/+aIxM2iQSvmWQYOvVCnhuY/F2GFaQ7d0om3decuwI0CX/xy7PjHMkLXa2uaZs4
    WR17sLduj62RqXRLX0c0QkwBiNFyHbRcpdkZJQujbYMhBa+7j7SxT4BtAoGAWMWT
    UucocRXZm/pdvz9wteNH3YDWnJLMxm1KC06qMXbBoYrliY4sm3ywJWMC+iCd/H8A
    Gecxd/xVu5mA2L2N3KMq18Zhz8Th0G5DwKyDRJgOQ0Q46yuNXOoYEjlo4Wjyk8Me
    +tlQ8iK98E0UmZnhTgfSpSNElbz2AqnzQ3MN9uECgYAqdvdVPnKGfvdtZ2DjyMoJ
    E89UIC41WjjJGmHsd8W65+3X0RwMzKMT6aZc5tK9J5dHvmWIETnbM+lTImdBBFga
    NWOC6f3r2xbGXHhaWSl+nobpTuvlo56ZRJVvVk7lFMsiddzMuHH8pxfgNJemwA4P
    ThDHCejv035NNV6KyoO0tA==
    -----END PRIVATE KEY-----
  client_email: cloudvolumes-admin-sa@cloud-native-
data.iam.gserviceaccount.com
  client_id: '107071413297115343396'
```

```
  auth_uri: https://accounts.google.com/o/oauth2/auth
  token_uri: https://oauth2.googleapis.com/token
  auth_provider_x509_cert_url:
https://www.googleapis.com/oauth2/v1/certs
  client_x509_cert_url:
https://www.googleapis.com/robot/v1/metadata/x509/cloudvolumes-admin-
sa%40cloud-native-data.iam.gserviceaccount.com
storageClass: software
zone: europe-west1-b
network: default
storagePools:
- 1bc7f380-3314-6005-45e9-c7dc8c2d7509
serviceLevel: Standardsw
```

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：文件和块。。 `Filesystem` volumemode、Astra Trident 会创建卷并创建文件系统。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。建议显式设置 UseCHAP 可选择对非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 。

如果两者都不是 AccessGroups 或 UseCHAP 设置后、将应用以下规则之一：

- 如果为默认值 trident 检测到访问组、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
version		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
Endpoint	使用租户凭据的 SolidFire 集群的 MVIP	
SVIP	存储（ iSCSI ） IP 地址和端口	
labels	要应用于卷的一组任意 JSON 格式的标签。	"
TenantName	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
AccessGroups	要使用的访问组 ID 列表	查找名为 "trident " 的访问组的 ID

参数	Description	Default
`Types`	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。

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、您很可能会使用定义存储类来使用其中的每一种 `IOPS` storage class参数。

```
---
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` 具有虚拟池的驱动程序

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

在配置时、Astra Trident会将存储池上的标签复制到后端存储LUN。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、这些存储池设置了 `type` 在Silver。虚拟池在中进行定义 `storage` 部分。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 (`solidfire-gold-four`)将映射到第一个虚拟池。这是唯一一个可通过提供金牌性能的池 `Volume Type QoS` 金牌。最后一个StorageClass (`solidfire-silver`)调用提供银牌性能的任何存储池。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: 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 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。
- trustedCACertifate： 受信任 CA 证书的 Base64 编码值。如果使用可信 CA ，则必须提供此参数。如果不使用可信 CA ，则可以忽略此设置。

典型的工作流包括以下步骤。

步骤

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

```
openssl req -x509 -nodes -days 1095 -newkey rsa:2048 -keyout k8senv.key
-out k8senv.pem -subj "/C=US/ST=NC/L=RTP/O=NetApp/CN=admin"
```

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

```
security certificate install -type server -cert-name <trusted-ca-cert-
name> -vserver <vserver-name>
ssl modify -vserver <vserver-name> -server-enabled true -client-enabled
true -common-name <common-name> -serial <SN-from-trusted-CA-cert> -ca
<cert-authority>
```

3. 在 ONTAP 集群上安装客户端证书和密钥（从步骤 1 开始）。

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

4. 确认ONTAP 安全登录角色支持 `cert` 身份验证方法。

```
security login create -user-or-group-name admin -application ontapi
-authentication-method cert
security login create -user-or-group-name admin -application http
-authentication-method cert
```

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

```
curl -X POST -Lk https://<ONTAP-Management-
LIF>/servlets/netapp.servlets.admin.XMLrequest_filer --key k8senv.key
--cert ~/k8senv.pem -d '<?xml version="1.0" encoding="UTF-8"?><netapp
xmlns="http://www.netapp.com/filer/admin" version="1.21"
vfiler="<vserver-name>"><vserver-get></vserver-get></netapp>'
```

6. 使用 Base64 对证书，密钥和可信 CA 证书进行编码。

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

7. 使用从上一步获得的值创建后端。

```
cat cert-backend.json
{
"version": 1,
"storageDriverName": "ontap-san",
"backendName": "SanBackend",
"managementLIF": "1.2.3.4",
"svm": "vserver_test",
"clientCertificate": "Faaaakkkkeeee...Vaaallllluuuueeee",
"clientPrivateKey": "LS0tFaKE...0VaLuES0tLS0K",
"trustedCACertificate": "QNFinfO...SiqOyN",
"storagePrefix": "myPrefix_"
}

tridentctl create backend -f cert-backend.json -n trident
+------------+----------------+--------------------------------------
+--------+---------+
|    NAME    | STORAGE DRIVER |                  UUID                  |
STATE  | VOLUMES |
+------------+----------------+--------------------------------------
+--------+---------+
| SanBackend | ontap-san      | 586b1cd5-8cf8-428d-a76c-2872713612c1 |
online |       0 |
+------------+----------------+--------------------------------------
+--------+---------+
```

更新身份验证方法或轮换凭据

您可以更新现有后端以使用其他身份验证方法或轮换其凭据。这两种方式都适用：使用用户名 / 密码的后端可以更新为使用证书；使用证书的后端可以更新为基于用户名 / 密码的后端。为此、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。

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

如果将 Astra Trident 配置为充当 CSI 配置程序，则 Kubernetes 节点 IQN 会自动添加到 igroup 中或从 igroup 中删除。将节点添加到Kubernetes集群后、 `trident-csi` DemonSet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 igroup 中。在对节点进行隔离，清空并从 Kubernetes 中删除时，可以执行一组类似的步骤来删除 IQN 。

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

使用双向 **CHAP** 对连接进行身份验证

Astra Trident可以使用双向CHAP对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
igroupName: trident
chapInitiatorSecret: cl9qxIm36DKyawxy
chapTargetInitiatorSecret: rqxigXgkesIpwxyz
chapTargetUsername: iJF4heBRT0TCwxyz
chapUsername: uh2aNCLSd6cNwxyz
```

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

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": "cl9qxUpDaTeD",
    "chapTargetInitiatorSecret": "rqxigXgkeUpDaTeD",
    "chapTargetUsername": "iJF4heBRT0TCwxyz",
    "chapUsername": "uh2aNCLSd6cNwxyz",
}

./tridentctl update backend ontap_san_chap -f backend-san.json -n trident
+----------------+----------------+--------------------------------------
+--------+---------+
|   NAME         | STORAGE DRIVER |                 UUID                 |
STATE  | VOLUMES |
+----------------+----------------+--------------------------------------
+--------+---------+
| ontap_san_chap | ontap-san      | aa458f3b-ad2d-4378-8a33-1a472ffbeb5c |
online |       7 |
+----------------+----------------+--------------------------------------
+--------+---------+
```

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

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**iSCSI**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iSCI 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iSCSI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 `igroupName`

`igroupName` 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igroupName、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。

- `igroupName` 可以更新为指向在Astra Trident之外的SVM上创建和管理的新igroup。
- `igroupName` 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 igroup 。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 "

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_134adbbb8e6d
                                   online     RW       5.79GB     5.50GB    0%
          _pvc_e8372153_9ad9_474a_951a_08ae15e1c0ba
                                   online     RW          1GB    511.8MB    0%
3 entries were displayed.
```

目前，调整大小是对现有卷使用新计算的唯一方法。

最低配置示例

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

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

`ontap-san` 具有基于证书的身份验证的驱动程序

这是一个最低后端配置示例。 `clientCertificate`， `clientPrivateKey`，和 `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: cl9qxIm36DKyawxy
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: cl9qxIm36DKyawxy
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: cl9qxIm36DKyawxy
chapTargetInitiatorSecret: rqxigXgkesIpwxyz
chapTargetUsername: iJF4heBRT0TCwxyz
chapUsername: uh2aNCLSd6cNwxyz
igroupName: trident
username: vsadmin
password: password
defaults:
  spaceAllocation: 'false'
  encryption: 'false'
  qosPolicy: standard
labels:
  store: san_store
  kubernetes-cluster: prod-cluster-1
region: us_east_1
storage:
- labels:
    protection: gold
    creditpoints: '40000'
  zone: us_east_1a
  defaults:
    spaceAllocation: 'true'
    encryption: 'true'
    adaptiveQosPolicy: adaptive-extreme
- labels:
    protection: silver
    creditpoints: '20000'
  zone: us_east_1b
  defaults:
    spaceAllocation: 'false'
    encryption: 'true'
    qosPolicy: premium
- labels:
    protection: bronze
    creditpoints: '5000'
  zone: us_east_1c
  defaults:
    spaceAllocation: 'true'
    encryption: 'false'
```

以下是的iSCSI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: 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"
```

配置ONTAP NAS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"

> 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 NAS驱动程序配置后端

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的黄金类 `ontap-nas` 驱动程序和使用的铜牌类 `ontap-nas-economy` 一个。

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

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

JSON

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

请注意，后端定义是凭据以纯文本格式存储的唯一位置。创建后端后，用户名 / 密码将使用 Base64 进行编码并

存储为 Kubernetes 密钥。创建 / 更新后端是唯一需要了解凭据的步骤。因此，这是一项仅由管理员执行的操作，由 Kubernetes 或存储管理员执行。

启用基于证书的身份验证

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

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

典型的工作流包括以下步骤。

步骤

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

```
openssl req -x509 -nodes -days 1095 -newkey rsa:2048 -keyout k8senv.key
-out k8senv.pem -subj "/C=US/ST=NC/L=RTP/O=NetApp/CN=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": "Faaaakkkkeeee...Vaaallllluuuueeee",
"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、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 update backend`。

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

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

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

> 使用此功能时、您必须确保SVM中的根接合具有先前创建的导出策略、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将为创建导出策略 `svm1` SVM并使用处理规则的添加和删除 `autoExportCIDRs` 地址块。例如、UUID为403b5326-8482-40db-96d0-d83fb3f4daec和的后端 `autoExportPolicy` 设置为 `true` 创建名为的导出策略 `trident-403b5326-8482-40db-96d0-d83fb3f4daec` 在 SVM 上。
- `autoExportCIDRs` 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、将显示 `192.168.0.0/24` 提供了地址空间。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注册其运行所在的节点时、它会检索该节点的IP地址并根据中提供的地址块对其进行检查 `autoExportCIDRs`。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。

您可以更新 `autoExportPolicy` 和 `autoExportCIDRs` 用于后端。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。您也可以选择禁用 `autoExportPolicy` 用于后端、并回退到手动创建的导出策略。这需要设置 `exportPolicy` 参数。

在Astra Trident创建或更新后端之后、您可以使用检查后端 `tridentctl` 或相应的 `tridentbackend` CRD：

```
./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集群并注册到Astra Trident控制器时、现有后端的导出策略会进行更新(前提是它们位于中指定的地址范围内 `autoExportCIDRs` 后端)。

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、请使用更新后端 `tridentctl update backend` 将确保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。

ONTAP NAS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
`version`		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
`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。建议指定 `dataLIF`。如果未提供此参数、则Astra Trident会从SVM提取数据LIF。您可以指定用于NFS挂载操作的完全限定域名(FQDN)、从而可以创建循环DNS、以便在多个数据LIF之间实现负载平衡。可以在初始设置后更改。请参见 。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]。	指定的地址或派生自SVM (如果未指定)(不建议)
`autoExportPolicy`	启用自动创建和更新导出策略[布尔值]。使用 `autoExportPolicy` 和 `autoExportCIDRs` 选项、Astra Trident可以自动管理导出策略。	false
`autoExportCIDRs`	用于筛选Kubernetes节点IP的CIDR列表 `autoExportPolicy` 已启用。使用 `autoExportPolicy` 和 `autoExportCIDRs` 选项、Astra Trident可以自动管理导出策略。	["0.0.0.0/0 "， "：： /0 "]`
`labels`	要应用于卷的一组任意 JSON 格式的标签	"
`clientCertificate`	客户端证书的 Base64 编码值。用于基于证书的身份验证	"

参数	Description	Default
`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`	用于连接到集群 /SVM 的用户名。用于基于凭据的身份验证	
`password`	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	如果是SVM、则派生 `managementLIF` 已指定
`storagePrefix`	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）
`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以及管理的卷的最大大小 `qtreesPerFlexvol` 选项用于自定义每个FlexVol 的最大qtree数。	"（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、｛"api"：false、"method "：true｝不使用 `debugTraceFlags` 除非您正在进行故障排除并需要详细的日志转储。	空
`nfsMountOptions`	NFS挂载选项的逗号分隔列表。Kubernetes持久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将回退到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。	"
`qtreesPerFlexvol`	每个 FlexVol 的最大 qtree 数，必须在 50 ， 300 范围内	200

参数	Description	Default
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

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	"
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.	"
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
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 "
unixPermissions	新卷的模式	对于NFS卷为"777"；对于SMB卷为空(不适用)

参数	Description	Default
snapshotDir	控制的可见性 .snapshot 目录	false
exportPolicy	要使用的导出策略	default
securityStyle	新卷的安全模式。NFS支持 mixed 和 unix 安全模式。SMB支持 mixed 和 ntfs 安全模式。	NFS默认值为 unix。SMB默认值为 ntfs。

> 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。

卷配置示例

下面是一个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'
```

适用于 ontap-nas 和 ontap-nas-flexgroups`现在、Astra Trident会使用新的计算方法来确保FlexVol 的大小与snapshotReserve百分比和PVC相同。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和 `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定义了 `snapshotReserve` 数字表示整个卷的百分比。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 = 10GiB，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、具有的2 GiB PVC `snapshotReserve=50` 之前的结果是、卷可提供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。

示例

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。

上的默认选项 **<code>ontap-nas-economy</code>**

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

基于证书的身份验证

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

自动导出策略

以下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。此操作对于也是如此 `ontap-nas-economy` 和 `ontap-nas-flexgroup` 驱动程序。

ontap-NAS 驱动程序

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

<code>ontap-nas-flexgroup</code> 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: password
```

使用**IPv6**地址

此示例显示了 `managementLIF` 使用IPv6地址。

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

`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: password
```

`ontap-nas` 适用于使用**SMB**卷的**Amazon FSX for ONTAP** 的驱动程序

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

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 上为设置 `ontap-nas` 或FlexGroup `ontap-nas-flexgroup`。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。

<code>ontap-nas</code> 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: password
nfsMountOptions: nfsvers=4
defaults:
  spaceReserve: none
  encryption: 'false'
  qosPolicy: standard
labels:
  store: nas_store
  k8scluster: prod-cluster-1
region: us_east_1
storage:
- labels:
    app: msoffice
    cost: '100'
  zone: us_east_1a
  defaults:
    spaceReserve: volume
    encryption: 'true'
    unixPermissions: '0755'
    adaptiveQosPolicy: adaptive-premium
- labels:
    app: slack
    cost: '75'
  zone: us_east_1b
  defaults:
    spaceReserve: none
    encryption: 'true'
    unixPermissions: '0755'
- labels:
    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'
```

<code>ontap-nas-flexgroup</code> 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: password
defaults:
  spaceReserve: none
  encryption: 'false'
labels:
  store: flexgroup_store
  k8scluster: prod-cluster-1
region: us_east_1
storage:
- labels:
    protection: gold
    creditpoints: '50000'
  zone: us_east_1a
  defaults:
    spaceReserve: volume
    encryption: 'true'
    unixPermissions: '0755'
- labels:
    protection: gold
    creditpoints: '30000'
  zone: us_east_1b
  defaults:
    spaceReserve: none
    encryption: 'true'
    unixPermissions: '0755'
- labels:
    protection: silver
    creditpoints: '20000'
  zone: us_east_1c
  defaults:
    spaceReserve: none
    encryption: 'true'
    unixPermissions: '0775'
- labels:
    protection: bronze
    creditpoints: '10000'
```

```
    zone: us_east_1d
    defaults:
      spaceReserve: volume
      encryption: 'false'
      unixPermissions: '0775'
```

<code>ontap-nas-economy</code> 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: password
defaults:
  spaceReserve: none
  encryption: 'false'
labels:
  store: nas_economy_store
region: us_east_1
storage:
- labels:
    department: finance
    creditpoints: '6000'
  zone: us_east_1a
  defaults:
    spaceReserve: volume
    encryption: 'true'
    unixPermissions: '0755'
- labels:
    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'
```

更新 `dataLIF` 初始配置后

您可以在初始配置后更改数据LIF、方法是运行以下命令、为新的后端JSON文件提供更新的数据LIF。

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

> 如果PVC连接到一个或多个Pod、则必须关闭所有对应Pod、然后将其恢复到、新数据LIF才能生效。

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"
```

适用于 NetApp ONTAP 的 Amazon FSX

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

"适用于 NetApp ONTAP 的 Amazon FSX" 是一种完全托管的AWS服务、可使客户启动和运行由NetApp ONTAP 存储操作系统提供支持的文件系统。借助适用于ONTAP 的FSx、您可以利用您熟悉的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 for ONTAP 支持ONTAP 文件系统功能和管理API。

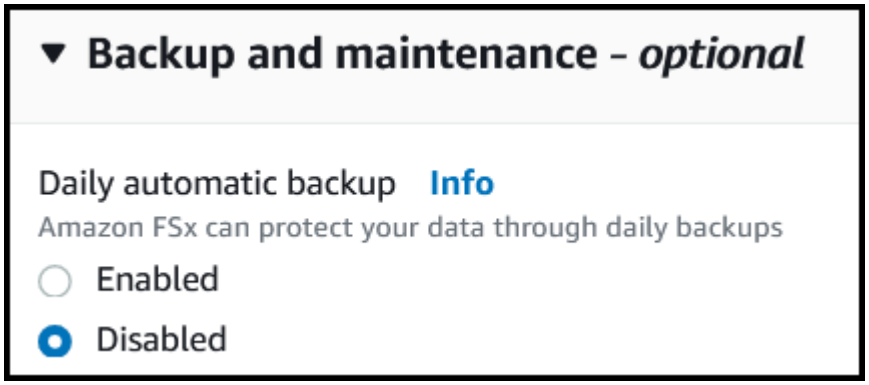
文件系统是 Amazon FSX 中的主要资源，类似于内部部署的 ONTAP 集群。在每个 SVM 中，您可以创建一个或多个卷，这些卷是将文件和文件夹存储在文件系统中的数据容器。借助适用于 NetApp ONTAP 的 Amazon FSX ， Data ONTAP 将作为云中的托管文件系统提供。新的文件系统类型称为 * NetApp ONTAP * 。

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。

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。

注意事项

- SMB卷：
  - SMB卷支持使用 `ontap-nas` 仅限驱动程序。
  - Astra Trident仅支持将SMB卷挂载到Windows节点上运行的Pod。
  - Astra Trident不支持Windows ARM 架构。
- 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。



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`：配置的每个PV都是一个LUN、对于NetApp ONTAP 卷、每个Amazon FSX的LUN数量是可配置的。
- `ontap-nas`：配置的每个PV都是一个适用于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都是一个适用于NetApp ONTAP FlexGroup 的完整Amazon FSX卷。

有关驱动程序详细信息、请参见 "ONTAP 驱动程序"。

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 `fsxadmin` 文件系统或的用户 `vsadmin` 为SVM配置的用户。

> Astra Trident应作为运行 `vsadmin` SVM用户或具有相同角色的其他名称的用户。适用于NetApp ONTAP 的Amazon FSX具有 `fsxadmin` 有限更换ONTAP 的用户 `admin` 集群用户。我们强烈建议使用 `vsadmin` 使用Astra Trident。

您可以更新后端以在基于凭据的方法和基于证书的方法之间移动。但是、如果您尝试提供*凭据和证书*、则后端创建将失败。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。

有关启用身份验证的详细信息、请参阅适用于您的驱动程序类型的身份验证：

- "ONTAP NAS身份验证"
- "ONTAP SAN身份验证"

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"

集成适用于NetApp ONTAP 的Amazon FSX

您可以将适用于NetApp ONTAP 的Amazon FSX文件系统与Astra Trident集成、以确保在Amazon Elastic Kubernetes Service (EKS)中运行的Kubernetes集群可以配置由ONTAP提供支持的块和文件永久性卷。

开始之前

此外 "Astra Trident 要求"要将适用于ONTAP 的FSx与Astra Trident集成、您需要：

- 具有的现有Amazon EKS集群或自管理Kubernetes集群 `kubectl` 已安装。

- 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 。
- 为准备工作的工作节点 "NFS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。

SMB卷的其他要求

- 一个Kubernetes集群、其中包含一个Linux控制器节点以及至少一个运行Windows Server 2019的Windows工作节点。Astra Trident仅支持将SMB卷挂载到Windows节点上运行的Pod。
- 至少一个包含Active Directory凭据的Astra Trident密钥。以生成密钥 `smbcreds`：

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

- 配置为Windows服务的CSI代理。配置 `csi-proxy`、请参见 "GitHub：CSI代理" 或 "GitHub：适用于Windows的CSI代理" 适用于在Windows上运行的Kubernetes节点。

ONTAP SAN和NAS驱动程序集成

> 如果要为SMB卷配置、则必须读取 准备配置SMB卷 创建后端之前。

步骤

1. 使用其中一种部署Astra Trident "部署方法"。
2. 收集SVM管理LIF DNS名称。例如、使用AWS命令行界面查找 `DNSName` 下的条目 `Endpoints` → `Management` 运行以下命令后：

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

3. 创建和安装证书 "NAS后端身份验证" 或 "SAN后端身份验证"。

> 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中的管理DNS名称 `aws fsx describe-file-systems`。

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：

YAML

```
---
version: 1
storageDriverName: ontap-san
backendName: customBackendName
managementLIF: svm-XXXXXXXXXXXXXXXXX.fs-XXXXXXXXXXXXXXXXX.fsx.us-
east-2.aws.internal
svm: svm01
clientCertificate: ZXR0ZXJwYXB...ICMgJ3BhcGVyc2
clientPrivateKey: vciwKIyAgZG...0cnksIGRlc2NyaX
trustedCACertificate: zcyBbaG...b3Igb3duIGNsYXNz
```

JSON

```
{
  "version": 1,
  "storageDriverName": "ontap-san",
  "backendName": "customBackendName",
  "managementLIF": "svm-XXXXXXXXXXXXXXXXX.fs-
XXXXXXXXXXXXXXXXX.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"

结果

部署完成后、您可以创建 "存储类，配置卷以及将卷挂载到 Pod 中"。

准备配置SMB卷

您可以使用配置SMB卷 `ontap-nas` 驱动程序。完成前 ONTAP SAN和NAS驱动程序集成 完成以下步骤。

步骤

1. 创建SMB共享。您可以使用以下两种方式之一创建SMB管理共享 "Microsoft管理控制台" 共享文件夹管理单元或使用ONTAP 命令行界面。要使用ONTAP 命令行界面创建SMB共享、请执行以下操作：
   a. 如有必要，为共享创建目录路径结构。

。 `vserver cifs share create` 命令会在创建共享期间检查-path选项中指定的路径。如果指定路径不存在，则命令将失败。

b. 创建与指定SVM关联的SMB共享：

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

c. 验证是否已创建共享：

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

请参见 "创建 SMB 共享" 了解完整详细信息。

2. 创建后端时、必须配置以下内容以指定SMB卷。有关适用于ONTAP 后端的所有FSX配置选项、请参见 "适用于ONTAP 的FSX配置选项和示例"。

参数	Description	示例
`smbShare`	使用共享文件夹Microsoft管理控制台创建的SMB共享的名称。例如"smb-share"。对于**SMB**卷为必需项。	`smb-share`
`nasType`	*必须设置为 `smb``如果为空、则默认为 `nfs`。	`smb`
`securityStyle`	新卷的安全模式。必须设置为 **`ntfs`** 或 **`mixed`** 用于**SMB**卷。	`ntfs` 或 `mixed` 对于SMB卷
`unixPermissions`	新卷的模式。对于SMB卷、必须留空。	""

适用于**ONTAP** 的**FSX**配置选项和示例

了解适用于ONTAP 的Amazon FSX的后端配置选项。本节提供了后端配置示例。

后端配置选项

有关后端配置选项，请参见下表：

参数	Description	示例
`version`		始终为 1

参数	**Description**	示例
`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。* ONTAP NAS驱动程序*：建议指定dataLIF。如果未提供此参数、则Astra Trident会从SVM提取数据LIF。您可以指定用于NFS挂载操作的完全限定域名(FQDN)、从而可以创建循环DNS、以便在多个数据LIF之间实现负载平衡。可以在初始设置后更改。请参见 。* ONTAP SAN驱动程序*：不为iSCSI指定。Astra Trident使用ONTAP 选择性LUN映射来发现建立多路径会话所需的iSCI LIF。如果明确定义了dataLIF、则会生成警告。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]。	
`autoExportPolicy`	启用自动创建和更新导出策略[布尔值]。使用 `autoExportPolicy` 和 `autoExportCIDRs` 选项、Astra Trident可以自动管理导出策略。	false
`autoExportCIDRs`	用于筛选Kubernetes节点IP的CIDR列表 `autoExportPolicy` 已启用。使用 `autoExportPolicy` 和 `autoExportCIDRs` 选项、Astra Trident可以自动管理导出策略。	"["0.0.0.0/0 "、"：： /0 "]"
`labels`	要应用于卷的一组任意 JSON 格式的标签	""
`clientCertificate`	客户端证书的 Base64 编码值。用于基于证书的身份验证	""

参数	Description	示例
`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`	用于连接到集群或SVM的用户名。用于基于凭据的身份验证。例如、vsadmin。	
`password`	用于连接到集群或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SVM管理LIF则派生。
`igroupName`	要使用的SAN卷的igroup的名称。请参见 。	"trident — < 后端 UUID >"
`storagePrefix`	在 SVM 中配置新卷时使用的前缀。创建后无法修改。要更新此参数、您需要创建一个新的后端。	Trident
`limitAggregateUsage`	*请勿为适用于NetApp ONTAP的Amazon FSX指定。*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。	请勿使用。
`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以及管理的卷的最大大小 `qtreesPerFlexvol` 选项用于自定义每个FlexVol 的最大qtree数。	" （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。仅限SAN。	"100"
`debugTraceFlags`	故障排除时要使用的调试标志。例如、 {"api"：false、"method "：true} 不使用 `debugTraceFlags` 除非您正在进行故障排除并需要详细的日志转储。	空
`nfsMountOptions`	NFS挂载选项的逗号分隔列表。Kubernetes持久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将回退到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。	""

参数	Description	示例
nasType	配置NFS或SMB卷创建。选项包括 nfs， smb`或为空。*必须设置为 `smb 对于SMB卷。*如果设置为空、则默认为NFS卷。	"NFs"
qtreesPerFlexvol	每个 FlexVol 的最大 qtree 数，必须在 50 ， 300 范围内	"200"
smbShare	使用共享文件夹Microsoft管理控制台创建的SMB共享的名称。对于**SMB**卷为必需项。	"smb-share"
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。	false

详细信息 igroupName

igroupName 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igroupName、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。

- igroupName 可以更新为指向在Astra Trident之外的SVM上创建和管理的新igroup。
- igroupName 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 igroup 。此更新不会中断对后端卷的访问。

更新 dataLIF 初始配置后

您可以在初始配置后更改数据LIF、方法是运行以下命令、为新的后端JSON文件提供更新的数据LIF。

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

> 如果PVC连接到一个或多个Pod、则必须关闭所有对应Pod、然后将其恢复到、新数据LIF才能生效。

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。选择每个存储池或后端的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。选择每个存储池或后端的qosPolicy或adaptiveQosPolicy之一。不受 ontap-nas-economy.	"
`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)"。仅限SAN。	""
`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 "
`unixPermissions`	新卷的模式。对于**SMB**卷保留为空。	""
`securityStyle`	新卷的安全模式。NFS支持 `mixed` 和 `unix` 安全模式。SMB支持 `mixed` 和 `ntfs` 安全模式。	NFS默认值为 `unix`。SMB默认值为 `ntfs`。

示例

使用 `nasType`， `node-stage-secret-name`，和 `node-stage-secret-namespace`、您可以指定SMB卷

并提供所需的Active Directory凭据。SMB卷支持使用 `ontap-nas` 仅限驱动程序。

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

版权信息

版权所有 © 2024 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。