# Controladores SAN ONTAP

## Astra Trident

NetApp
January 31, 2025

# Índice

# Controladores SAN ONTAP

## Descrição geral do controlador SAN ONTAP

Saiba mais sobre como configurar um back-end ONTAP com drivers SAN ONTAP e Cloud Volumes ONTAP.

### Detalhes do driver SAN ONTAP

O Astra Trident fornece os seguintes drivers de storage SAN para se comunicar com o cluster ONTAP. Os modos de acesso suportados são: *ReadWriteOnce* (RWO), *ReadOnlyMuy* (ROX), *ReadWriteMany* (RWX), *ReadWriteOncePod* (RWOP).

Se você estiver usando o Astra Control para proteção, recuperação e mobilidade, leia Compatibilidade com driver Astra Control.

| Condutor | Protocolo | VolumeMode | Modos de acesso suportados | Sistemas de arquivos suportados |
|---|---|---|---|---|
| `ontap-san` | ISCSI | Bloco | RWO, ROX, RWX, RWOP | Sem sistema de arquivos; dispositivo de bloco bruto |
| `ontap-san` | ISCSI | Sistema de ficheiros | RWO, RWOP<br><br>ROX e RWX não estão disponíveis no modo de volume do sistema de arquivos. | `xfs` `ext3`, , `ext4` |
| `ontap-san` | NVMe/TCP<br><br>Considerações adicionais para NVMe/TCP Consulte a . | Bloco | RWO, ROX, RWX, RWOP | Sem sistema de arquivos; dispositivo de bloco bruto |
| `ontap-san` | NVMe/TCP<br><br>Considerações adicionais para NVMe/TCP Consulte a . | Sistema de ficheiros | RWO, RWOP<br><br>ROX e RWX não estão disponíveis no modo de volume do sistema de arquivos. | `xfs` `ext3`, , `ext4` |
| `ontap-san-economy` | ISCSI | Bloco | RWO, ROX, RWX, RWOP | Sem sistema de arquivos; dispositivo de bloco bruto |

| Condutor | Protocolo | VolumeMode | Modos de acesso suportados | Sistemas de arquivos suportados |
|---|---|---|---|---|
| `ontap-san-economy` | ISCSI | Sistema de ficheiros | RWO, RWOP<br><br>ROX e RWX não estão disponíveis no modo de volume do sistema de arquivos. | `xfs` `ext3`, , `ext4` |

#### Compatibilidade com driver Astra Control

O Astra Control oferece proteção aprimorada, recuperação de desastres e mobilidade (migrando volumes entre clusters Kubernetes) para volumes criados com os `ontap-nas` drivers , `ontap-nas-flexgroup` e . `ontap-san` "Pré-requisitos de replicação do Astra Control"Consulte para obter detalhes.

- Use `ontap-san-economy` somente se a contagem de uso de volume persistente for esperada ser maior que "Limites de volume ONTAP suportados".
- Use `ontap-nas-economy` somente se a contagem de uso de volume persistente for esperada para ser maior do que "Limites de volume ONTAP suportados" e o `ontap-san-economy` driver não puder ser usado.
- Não use o uso `ontap-nas-economy` se você antecipar a necessidade de proteção de dados, recuperação de desastres ou mobilidade.

## Permissões do usuário

O Astra Trident espera ser executado como administrador da ONTAP ou SVM, normalmente usando o `admin` usuário do cluster ou um `vsadmin` usuário SVM, ou um usuário com um nome diferente que tenha a mesma função. Para implantações do Amazon FSX for NetApp ONTAP, o Astra Trident espera ser executado como administrador do ONTAP ou SVM, usando o usuário do cluster `fsxadmin` ou um `vsadmin` usuário SVM, ou um usuário com um nome diferente que tenha a mesma função. O `fsxadmin` usuário é um substituto limitado para o usuário administrador do cluster.

Se você usar o `limitAggregateUsage` parâmetro, as permissões de administrador do cluster serão necessárias. Ao usar o Amazon FSX for NetApp ONTAP com Astra Trident, o `limitAggregateUsage` parâmetro não funcionará com as `vsadmin` contas de usuário e `fsxadmin`. A operação de configuração falhará se você especificar este parâmetro.

Embora seja possível criar uma função mais restritiva no ONTAP que um driver Trident pode usar, não recomendamos. A maioria das novas versões do Trident chamarão APIs adicionais que teriam que ser contabilizadas, tornando as atualizações difíceis e suscetíveis a erros.

## Considerações adicionais para NVMe/TCP

O Astra Trident dá suporte ao protocolo NVMe (non-volátil Memory Express) usando `ontap-san` o driver, incluindo:

- IPv6
- Snapshots e clones de volumes NVMe

- Redimensionamento de um volume NVMe
- Importação de um volume NVMe que foi criado fora do Astra Trident para que seu ciclo de vida possa ser gerenciado pelo Astra Trident
- Multipathing nativo NVMe
- Desligamento gracioso ou vergonhoso dos K8s nós (23,10)

O Astra Trident não é compatível com:

- DH-HMAC-CHAP que é suportado nativamente pelo NVMe
- Multipathing de mapeador de dispositivos (DM)
- Criptografia LUKS

# Prepare-se para configurar o back-end com drivers SAN ONTAP

Entenda os requisitos e as opções de autenticação para configurar um back-end do ONTAP com drivers de SAN ONTAP.

## Requisitos

Para todos os back-ends ONTAP, o Astra Trident requer pelo menos um agregado atribuído ao SVM.

Lembre-se de que você também pode executar mais de um driver e criar classes de armazenamento que apontam para um ou outro. Por exemplo, você pode configurar uma `san-dev` classe que usa o `ontap-san` driver e uma `san-default` classe que usa a `ontap-san-economy` mesma.

Todos os seus nós de trabalho do Kubernetes devem ter as ferramentas iSCSI apropriadas instaladas. "Prepare o nó de trabalho"Consulte para obter detalhes.

## Autenticar o back-end do ONTAP

O Astra Trident oferece dois modos de autenticação no back-end do ONTAP.

- Baseado em credenciais: O nome de usuário e senha para um usuário do ONTAP com as permissões necessárias. Recomenda-se a utilização de uma função de início de sessão de segurança predefinida, como `admin` ou `vsadmin` para garantir a máxima compatibilidade com as versões do ONTAP.
- Baseado em certificado: O Astra Trident também pode se comunicar com um cluster ONTAP usando um certificado instalado no back-end. Aqui, a definição de back-end deve conter valores codificados em Base64 do certificado de cliente, chave e certificado de CA confiável, se usado (recomendado).

Você pode atualizar os backends existentes para mover entre métodos baseados em credenciais e baseados em certificado. No entanto, apenas um método de autenticação é suportado por vez. Para alternar para um método de autenticação diferente, você deve remover o método existente da configuração de back-end.

Se você tentar fornecer **credenciais e certificados**, a criação de back-end falhará com um erro que mais de um método de autenticação foi fornecido no arquivo de configuração.

### Ative a autenticação baseada em credenciais

O Astra Trident requer as credenciais para um administrador com escopo SVM/cluster para se comunicar com o back-end do ONTAP. Recomenda-se a utilização de funções padrão predefinidas, como `admin` ou `vsadmin`. Isso garante compatibilidade direta com futuras versões do ONTAP que podem expor APIs de recursos a serem usadas por futuras versões do Astra Trident. Uma função de login de segurança personalizada pode ser criada e usada com o Astra Trident, mas não é recomendada.

Uma definição de backend de exemplo será assim:

**YAML**

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

**JSON**

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

Tenha em mente que a definição de back-end é o único lugar onde as credenciais são armazenadas em texto simples. Depois que o back-end é criado, os nomes de usuário/senhas são codificados com Base64 e armazenados como segredos do Kubernetes. A criação ou atualização de um backend é a única etapa que requer conhecimento das credenciais. Como tal, é uma operação somente de administrador, a ser realizada pelo administrador do Kubernetes/storage.

### Ativar autenticação baseada em certificado

Backends novos e existentes podem usar um certificado e se comunicar com o back-end do ONTAP. Três parâmetros são necessários na definição de backend.

- ClientCertificate: Valor codificado base64 do certificado do cliente.
- ClientPrivateKey: Valor codificado em base64 da chave privada associada.
- TrustedCACertificate: Valor codificado base64 do certificado CA confiável. Se estiver usando uma CA

confiável, esse parâmetro deve ser fornecido. Isso pode ser ignorado se nenhuma CA confiável for usada.

Um fluxo de trabalho típico envolve as etapas a seguir.

**Passos**

1. Gerar um certificado e chave de cliente. Ao gerar, defina Nome Comum (CN) para o usuário ONTAP para autenticar como.

```
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. Adicionar certificado de CA confiável ao cluster do ONTAP. Isso pode já ser Tratado pelo administrador do armazenamento. Ignore se nenhuma CA confiável for usada.

```
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. Instale o certificado e a chave do cliente (a partir do passo 1) no cluster do ONTAP.

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

4. Confirme se a função de login de segurança do ONTAP suporta `cert` o método de autenticação.

```
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. Teste a autenticação usando certificado gerado. Substitua o ONTAP Management LIF> e o <vserver name> por IP de LIF de gerenciamento e nome da SVM.

```
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. Codificar certificado, chave e certificado CA confiável com Base64.

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

7. Crie backend usando os valores obtidos na etapa anterior.

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

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

**Atualizar métodos de autenticação ou girar credenciais**

Você pode atualizar um back-end existente para usar um método de autenticação diferente ou para girar suas credenciais. Isso funciona de ambas as maneiras: Backends que fazem uso de nome de usuário / senha podem ser atualizados para usar certificados; backends que utilizam certificados podem ser atualizados para nome de usuário / senha com base. Para fazer isso, você deve remover o método de autenticação existente e adicionar o novo método de autenticação. Em seguida, use o arquivo backend.json atualizado contendo os parâmetros necessários para executar `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 |
+------------+----------------+--------------------------------------
+--------+---------+
```

Ao girar senhas, o administrador de armazenamento deve primeiro atualizar a senha do usuário no ONTAP. Isso é seguido por uma atualização de back-end. Ao girar certificados, vários certificados podem ser adicionados ao usuário. O back-end é então atualizado para usar o novo certificado, seguindo o qual o certificado antigo pode ser excluído do cluster do ONTAP.

A atualização de um back-end não interrompe o acesso a volumes que já foram criados, nem afeta as conexões de volume feitas depois. Uma atualização de back-end bem-sucedida indica que o Astra Trident pode se comunicar com o back-end do ONTAP e lidar com operações de volume futuras.

## Autentique conexões com CHAP bidirecional

O Astra Trident pode autenticar sessões iSCSI com CHAP bidirecional para os `ontap-san` drivers e `ontap-san-economy`. Isso requer a ativação da `useCHAP` opção na definição de backend. Quando definido como `true`, o Astra Trident configura a segurança do iniciador padrão do SVM para CHAP bidirecional e define o nome de usuário e os segredos do arquivo de back-end. O NetApp recomenda o uso de CHAP bidirecional para autenticar conexões. Veja a seguinte configuração de exemplo:

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

O `useCHAP` parâmetro é uma opção booleana que pode ser configurada apenas uma vez. Ele é definido como false por padrão. Depois de configurá-lo como verdadeiro, você não pode configurá-lo como falso.

Além `useCHAP=true` do , os `chapInitiatorSecret` campos , `chapTargetInitiatorSecret`, `chapTargetUsername`, e `chapUsername` devem ser incluídos na definição de back-end. Os segredos podem ser alterados depois que um backend é criado executando `tridentctl update`.

#### Como funciona

Ao definir `useCHAP` como verdadeiro, o administrador de storage instrui o Astra Trident a configurar o CHAP no back-end de storage. Isso inclui o seguinte:

- Configuração do CHAP no SVM:
  - Se o tipo de segurança do iniciador padrão da SVM for nenhum (definido por padrão) **e** não houver LUNs pré-existentes no volume, o Astra Trident definirá o tipo de segurança padrão `CHAP` e continuará configurando o iniciador CHAP e o nome de usuário e os segredos de destino.
  - Se o SVM contiver LUNs, o Astra Trident não ativará o CHAP no SVM. Isso garante que o acesso a LUNs que já estão presentes no SVM não seja restrito.
- Configurando o iniciador CHAP e o nome de usuário e os segredos de destino; essas opções devem ser especificadas na configuração de back-end (como mostrado acima).

Depois que o back-end é criado, o Astra Trident cria um CRD correspondente `tridentbackend` e armazena os segredos e nomes de usuário do CHAP como segredos do Kubernetes. Todos os PVS criados pelo Astra Trident neste back-end serão montados e anexados através do CHAP.

#### Gire credenciais e atualize os backends

Você pode atualizar as credenciais CHAP atualizando os parâmetros CHAP no `backend.json` arquivo. Isso exigirá a atualização dos segredos CHAP e o uso do `tridentctl update` comando para refletir essas alterações.

Ao atualizar os segredos CHAP para um backend, você deve usar `tridentctl` para atualizar o backend. Não atualize as credenciais no cluster de storage por meio da IU da CLI/ONTAP, pois o Astra Trident não conseguirá aceitar essas alterações.

```
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",
    "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 |
+----------------+----------------+--------------------------------------
+--------+---------+
```

As conexões existentes não serão afetadas. Elas continuarão ativas se as credenciais forem atualizadas pelo Astra Trident no SVM. As novas conexões usarão as credenciais atualizadas e as conexões existentes continuam ativas. Desconetar e reconetar PVS antigos resultará em eles usando as credenciais atualizadas.

# Exemplos e opções de configuração de SAN ONTAP

Saiba como criar e usar drivers SAN ONTAP com sua instalação do Astra Trident. Esta seção fornece exemplos de configuração de back-end e detalhes para mapear backends para StorageClasses.

## Opções de configuração de back-end

Consulte a tabela a seguir para obter as opções de configuração de back-end:

| Parâmetro | Descrição | Padrão |
| --- | --- | --- |
| `version` | | Sempre 1 |
| `storageDriverName` | Nome do controlador de armazenamento | `ontap-nas` `ontap-nas-economy`, , `ontap-nas-flexgroup` `ontap-san` , , , `ontap-san-economy` |
| `backendName` | Nome personalizado ou back-end de storage | Nome do driver e dataLIF |
| `managementLIF` | Endereço IP de um cluster ou LIF de gerenciamento de SVM. Um nome de domínio totalmente qualificado (FQDN) pode ser especificado. Pode ser definido para usar endereços IPv6 se o Astra Trident tiver sido instalado usando o sinalizador IPv6. Os endereços IPv6 devem ser definidos entre colchetes, como `[28e8:d9fb:a825:b7bf:69a8:d02f:9e7b:3555]` . Para o switchover MetroCluster otimizado, consulte o Exemplo de MetroCluster. | "10,0.0,1", "[2001:1234:abcd::fefe]" |
| `dataLIF` | Endereço IP do protocolo LIF. **Não especifique para iSCSI.** O Astra Trident usa "Mapa de LUN seletivo da ONTAP" para descobrir os LIFs iSCI necessários para estabelecer uma sessão de vários caminhos. Um aviso é gerado se `dataLIF` for definido explicitamente. **Omita para MetroCluster.** Consulte Exemplo de MetroCluster. | Derivado do SVM |
| `svm` | Máquina virtual de armazenamento para usar **omit for MetroCluster.** Consulte Exemplo de MetroCluster. | Derivado se uma SVM `managementLIF` for especificada |
| `useCHAP` | Use CHAP para autenticar iSCSI para drivers SAN ONTAP [Boolean]. Defina como `true` para o Astra Trident para configurar e usar CHAP bidirecional como a autenticação padrão para o SVM dado no back-end. "Prepare-se para configurar o back-end com drivers SAN ONTAP"Consulte para obter detalhes. | `false` |
| `chapInitiatorSecret` | Segredo do iniciador CHAP. Necessário se `useCHAP=true` | "" |
| `labels` | Conjunto de rótulos arbitrários formatados em JSON para aplicar em volumes | "" |
| `chapTargetInitiatorSecret` | Segredo do iniciador de destino CHAP. Necessário se `useCHAP=true` | "" |
| `chapUsername` | Nome de utilizador de entrada. Necessário se `useCHAP=true` | "" |
| `chapTargetUsername` | Nome de utilizador alvo. Necessário se `useCHAP=true` | "" |
| `clientCertificate` | Valor codificado em base64 do certificado do cliente. Usado para autenticação baseada em certificado | "" |

| Parâmetro | Descrição | Padrão |
| --- | --- | --- |
| `clientPrivateKey` | Valor codificado em base64 da chave privada do cliente. Usado para autenticação baseada em certificado | "" |
| `trustedCACertificate` | Valor codificado em base64 do certificado CA confiável. Opcional. Usado para autenticação baseada em certificado. | "" |
| `username` | Nome de usuário necessário para se comunicar com o cluster ONTAP. Usado para autenticação baseada em credenciais. | "" |
| `password` | Senha necessária para se comunicar com o cluster ONTAP. Usado para autenticação baseada em credenciais. | "" |
| `svm` | Máquina virtual de armazenamento para usar | Derivado se uma SVM `managementLIF` for especificada |
| `storagePrefix` | Prefixo usado ao provisionar novos volumes na SVM. Não pode ser modificado mais tarde. Para atualizar esse parâmetro, você precisará criar um novo backend. | `trident` |
| `limitAggregateUsage` | Falha no provisionamento se o uso estiver acima dessa porcentagem. Se você estiver usando um back-end do Amazon FSX for NetApp ONTAP, não `limitAggregateUsage`especifique . O fornecido `fsxadmin` e `vsadmin` não contém as permissões necessárias para recuperar o uso agregado e limitá-lo usando o Astra Trident. | "" (não aplicado por padrão) |
| `limitVolumeSize` | Falha no provisionamento se o tamanho do volume solicitado estiver acima desse valor. Também restringe o tamanho máximo dos volumes que gerencia para qtrees e LUNs. | "" (não aplicado por padrão) |
| `lunsPerFlexvol` | Máximo de LUNs por FlexVol, tem de estar no intervalo [50, 200] | `100` |
| `debugTraceFlags` | Debug flags para usar ao solucionar problemas. Por exemplo, não use a menos que você esteja solucionando problemas e exija um despejo de log detalhado. | `null` |

| Parâmetro | Descrição | Padrão |
|---|---|---|
| `useREST` | Parâmetro booleano para usar APIs REST do ONTAP. **A visualização técnica** `useREST` é fornecida como uma **prévia técnica** que é recomendada para ambientes de teste e não para cargas de trabalho de produção. Quando definido como `true`, o Astra Trident usará as APIS REST do ONTAP para se comunicar com o back-end. Esse recurso requer o ONTAP 9.11,1 e posterior. Além disso, a função de login do ONTAP usada deve ter acesso ao `ontap` aplicativo. Isso é satisfeito com as funções e `cluster-admin` predefinidas `vsadmin`. `useREST` Não é suportado com MetroCluster. `useREST` É totalmente qualificado para NVMe/TCP. | `false` |
| `sanType` | Utilize para selecionar `iscsi` para iSCSI ou `nvme` para NVMe/TCP. | `iscsi` se estiver em branco |

## Opções de configuração de back-end para volumes de provisionamento

Você pode controlar o provisionamento padrão usando essas opções na `defaults` seção da configuração. Para obter um exemplo, consulte os exemplos de configuração abaixo.

| Parâmetro | Descrição | Padrão |
|---|---|---|
| `spaceAllocation` | Alocação de espaço para LUNs | "verdadeiro" |
| `spaceReserve` | Modo de reserva de espaço; "nenhum" (fino) ou "volume" (grosso) | "nenhum" |
| `snapshotPolicy` | Política de instantâneos a utilizar | "nenhum" |
| `qosPolicy` | Grupo de políticas de QoS a atribuir aos volumes criados. Escolha uma das qosPolicy ou adaptiveQosPolicy por pool de armazenamento/backend. O uso de grupos de política de QoS com o Astra Trident requer o ONTAP 9.8 ou posterior. Recomendamos o uso de um grupo de políticas de QoS não compartilhado e garantir que o grupo de políticas seja aplicado individualmente a cada componente. Um grupo de política de QoS compartilhado aplicará o limite máximo da taxa de transferência total de todos os workloads. | "" |
| `adaptiveQosPolicy` | Grupo de políticas de QoS adaptável a atribuir para volumes criados. Escolha uma das qosPolicy ou adaptiveQosPolicy por pool de armazenamento/backend | "" |
| `snapshotReserve` | Porcentagem de volume reservado para snapshots | "0" se `snapshotPolicy` for "nenhum", caso contrário "" |
| `splitOnClone` | Divida um clone de seu pai na criação | "falso" |

| Parâmetro | Descrição | Padrão |
| --- | --- | --- |
| `encryption` | Ative a criptografia de volume do NetApp (NVE) no novo volume; o padrão é `false`. O NVE deve ser licenciado e habilitado no cluster para usar essa opção. Se o NAE estiver ativado no back-end, qualquer volume provisionado no Astra Trident será o NAE ativado. Para obter mais informações, consulte: "Como o Astra Trident funciona com NVE e NAE". | "falso" |
| `luksEncryption` | Ativar encriptação LUKS. "Usar a configuração de chave unificada do Linux (LUKS)"Consulte a . A criptografia LUKS não é compatível com NVMe/TCP. | "" |
| `securityStyle` | Estilo de segurança para novos volumes | `unix` |
| `tieringPolicy` | Política de disposição em camadas para usar "nenhuma" | "Somente snapshot" para configuração pré-ONTAP 9.5 SVM-DR |

#### Exemplos de provisionamento de volume

Aqui está um exemplo com padrões definidos:

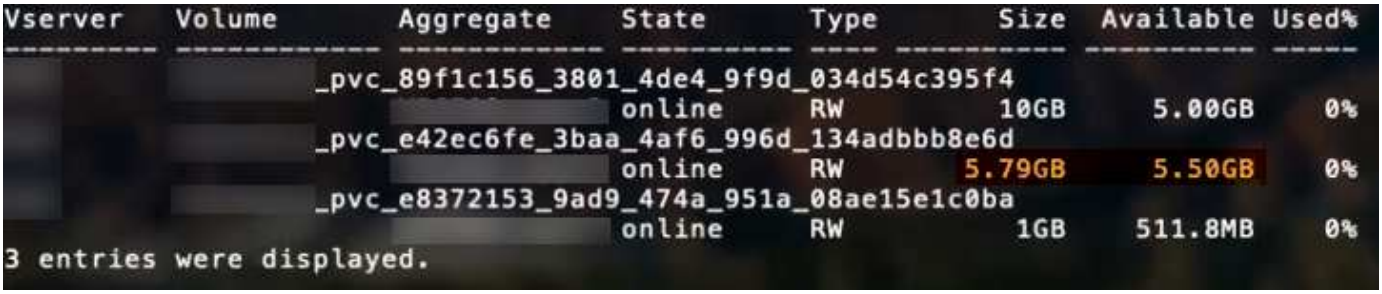
```
---
version: 1
storageDriverName: ontap-san
managementLIF: 10.0.0.1
svm: trident_svm
username: admin
password: <password>
labels:
  k8scluster: dev2
  backend: dev2-sanbackend
storagePrefix: alternate-trident
debugTraceFlags:
  api: false
  method: true
defaults:
  spaceReserve: volume
  qosPolicy: standard
  spaceAllocation: 'false'
  snapshotPolicy: default
  snapshotReserve: '10'
```

Para todos os volumes criados com `ontap-san` o driver, o Astra Trident adiciona uma capacidade extra de 10% ao FlexVol para acomodar os metadados do LUN. O LUN será provisionado com o tamanho exato que o usuário solicita no PVC. O Astra Trident adiciona 10% ao FlexVol (mostra como tamanho disponível no ONTAP). Os usuários agora terão a capacidade utilizável que solicitaram. Essa alteração também impede que LUNs fiquem somente leitura, a menos que o espaço disponível seja totalmente utilizado. Isto não se aplica à ONTAP-san-economia.

Para backends que definem `snapshotReserve`o , o Astra Trident calcula o tamanho dos volumes da seguinte forma:

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

O 1,1 é o 10% adicional que o Astra Trident adiciona ao FlexVol para acomodar os metadados do LUN. Para `snapshotReserve` 5%, e o pedido de PVC é de 5GiB, o tamanho total do volume é de 5,79GiB e o tamanho disponível é de 5,5GiB. O `volume show` comando deve mostrar resultados semelhantes a este exemplo:




Atualmente, o redimensionamento é a única maneira de usar o novo cálculo para um volume existente.

## Exemplos mínimos de configuração

Os exemplos a seguir mostram configurações básicas que deixam a maioria dos parâmetros padrão. Esta é a maneira mais fácil de definir um backend.

Se você estiver usando o Amazon FSX no NetApp ONTAP com Astra Trident, recomendamos que você especifique nomes DNS para LIFs em vez de endereços IP.

#### Exemplo de SAN ONTAP

Esta é uma configuração básica usando `ontap-san` o driver.

```
---
version: 1
storageDriverName: ontap-san
managementLIF: 10.0.0.1
svm: svm_iscsi
labels:
  k8scluster: test-cluster-1
  backend: testcluster1-sanbackend
username: vsadmin
password: <password>
```

#### Exemplo de economia de SAN ONTAP

```
---
version: 1
storageDriverName: ontap-san-economy
managementLIF: 10.0.0.1
svm: svm_iscsi_eco
username: vsadmin
password: <password>
```

#### Exemplo de MetroCluster

Você pode configurar o back-end para evitar ter que atualizar manualmente a definição do back-end após o switchover e o switchback durante "Replicação e recuperação da SVM"o .

Para comutação e switchback contínuos, especifique o SVM usando `managementLIF` e omite os `dataLIF` parâmetros e. `svm` Por exemplo:

```
---
version: 1
storageDriverName: ontap-san
managementLIF: 192.168.1.66
username: vsadmin
password: password
```

**Exemplo de autenticação baseada em certificado**

Neste exemplo de configuração básica `clientCertificate` , `clientPrivateKey` e `trustedCACertificate` (opcional, se estiver usando CA confiável) são preenchidos `backend.json` e recebem os valores codificados em base64 do certificado do cliente, da chave privada e do certificado de CA confiável, respetivamente.

```
---
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
clientCertificate: ZXR0ZXJwYXB...ICMgJ3BhcGVyc2
clientPrivateKey: vciwKIyAgZG...0cnksIGRlc2NyaX
trustedCACertificate: zcyBbaG...b3Igb3duIGNsYXNz
```

#### Exemplos CHAP bidirecional

Esses exemplos criam um backend com `useCHAP` definido como `true`.

##### Exemplo de ONTAP SAN CHAP

```
---
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
username: vsadmin
password: <password>
```

##### Exemplo de CHAP de economia de SAN ONTAP

```
---
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
username: vsadmin
password: <password>
```

**Exemplo de NVMe/TCP**

Você precisa ter um SVM configurado com NVMe no back-end do ONTAP. Esta é uma configuração básica de back-end para NVMe/TCP.

```
---
version: 1
backendName: NVMeBackend
storageDriverName: ontap-san
managementLIF: 10.0.0.1
svm: svm_nvme
username: vsadmin
password: password
sanType: nvme
useREST: true
```

## Exemplos de backends com pools virtuais

Nesses arquivos de definição de back-end de exemplo, padrões específicos são definidos para todos os pools de armazenamento, como `spaceReserve` em nenhum, `spaceAllocation` em falso e `encryption` em falso. Os pools virtuais são definidos na seção armazenamento.

O Astra Trident define rótulos de provisionamento no campo "Comentários". Os comentários são definidos no FlexVol. O Astra Trident copia todas as etiquetas presentes em um pool virtual para o volume de storage no provisionamento. Por conveniência, os administradores de storage podem definir rótulos por pool virtual e volumes de grupo por rótulo.

Nesses exemplos, alguns dos pools de armazenamento definem seus próprios `spaceReserve` `spaceAllocation` valores , e `encryption` , e alguns pools substituem os valores padrão.

**Exemplo de SAN ONTAP**

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

**Exemplo de economia de SAN ONTAP**

```
---
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
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'
- labels:
    department: legal
    creditpoints: '5000'
  zone: us_east_1c
```

```
  defaults:
    spaceAllocation: 'true'
    encryption: 'false'
```

**Exemplo de NVMe/TCP**

```
---
version: 1
storageDriverName: ontap-san
sanType: nvme
managementLIF: 10.0.0.1
svm: nvme_svm
username: vsadmin
password: <password>
useREST: true
defaults:
  spaceAllocation: 'false'
  encryption: 'true'
storage:
- labels:
    app: testApp
    cost: '20'
  defaults:
    spaceAllocation: 'false'
    encryption: 'false'
```

## Mapeie os backends para StorageClasses

As seguintes definições do StorageClass referem-se ao Exemplos de backends com pools virtuais. Usando o `parameters.selector` campo, cada StorageClass chama quais pools virtuais podem ser usados para hospedar um volume. O volume terá os aspetos definidos no pool virtual escolhido.

- O `protection-gold` StorageClass será mapeado para o primeiro pool virtual `ontap-san` no back-end. Esta é a única piscina que oferece proteção de nível dourado.

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: protection-gold
provisioner: csi.trident.netapp.io
parameters:
  selector: "protection=gold"
  fsType: "ext4"
```

- O `protection-not-gold` StorageClass será mapeado para o segundo e terceiro pool virtual no `ontap-san` back-end. Estas são as únicas piscinas que oferecem um nível de proteção diferente do ouro.

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: protection-not-gold
provisioner: csi.trident.netapp.io
parameters:
  selector: "protection!=gold"
  fsType: "ext4"
```

- O `app-mysqldb` StorageClass será mapeado para o terceiro pool virtual no `ontap-san-economy` back-end. Este é o único pool que oferece configuração de pool de armazenamento para o aplicativo tipo mysqldb.

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: app-mysqldb
provisioner: csi.trident.netapp.io
parameters:
  selector: "app=mysqldb"
  fsType: "ext4"
```

- O `protection-silver-creditpoints-20k` StorageClass será mapeado para o segundo pool virtual no `ontap-san` back-end. Esta é a única piscina que oferece proteção de nível de prata e 20000 pontos de crédito.

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: protection-silver-creditpoints-20k
provisioner: csi.trident.netapp.io
parameters:
  selector: "protection=silver; creditpoints=20000"
  fsType: "ext4"
```

- O `creditpoints-5k` StorageClass será mapeado para o terceiro pool virtual no `ontap-san` back-end e o quarto pool virtual no `ontap-san-economy` back-end. Estas são as únicas ofertas de pool com 5000 pontos de crédito.

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: creditpoints-5k
provisioner: csi.trident.netapp.io
parameters:
  selector: "creditpoints=5000"
  fsType: "ext4"
```

- O `my-test-app-sc` StorageClass será mapeado para o `testAPP` pool virtual no `ontap-san` driver com `sanType: nvme`o . Esta é a única piscina que oferece `testApp`.

```
---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: my-test-app-sc
provisioner: csi.trident.netapp.io
parameters:
  selector: "app=testApp"
  fsType: "ext4"
```

O Astra Trident decidirá qual pool virtual está selecionado e garantirá que o requisito de storage seja atendido.

## Informações sobre direitos autorais



## Informações sobre marcas comerciais