



Use o Trident

Trident

NetApp
January 15, 2026

Índice

Use o Trident	1
Prepare o nó de trabalho	1
Selecionar as ferramentas certas	1
Descoberta de serviços de nó	1
Volumes NFS	2
volumes iSCSI	2
Volumes NVMe/TCP	6
SCSI sobre volumes FC	7
Configurar e gerenciar backends	10
Configurar backends	10
Azure NetApp Files	10
Google Cloud NetApp Volumes	30
Configure um Cloud Volumes Service para o backend do Google Cloud.	47
Configure um backend NetApp HCI ou SolidFire.	59
Motoristas ONTAP SAN	64
Drivers ONTAP NAS	94
Amazon FSx for NetApp ONTAP	130
Crie backends com kubectl	166
Gerenciar back-ends	173
Criar e gerenciar classes de armazenamento	183
Criar uma classe de armazenamento	183
Gerenciar classes de armazenamento	186
Provisionar e gerenciar volumes	188
Forneça um volume	188
Expandir volumes	192
Volumes de importação	203
Personalize os nomes e rótulos dos volumes.	211
Compartilhar um volume NFS entre namespaces	214
Clonar volumes entre namespaces	218
Replicar volumes usando o SnapMirror	221
Utilizar a topologia CSI	227
Trabalhar com instantâneos	235
Trabalhar com snapshots de grupos de volumes	243

Use o Trident

Prepare o nó de trabalho

Todos os nós de trabalho no cluster Kubernetes devem ser capazes de montar os volumes que você provisionou para seus pods. Para preparar os nós de trabalho, você deve instalar as ferramentas NFS, iSCSI, NVMe/TCP ou FC, dependendo da sua seleção de driver.

Selecionar as ferramentas certas

Se você estiver usando uma combinação de drivers, deverá instalar todas as ferramentas necessárias para seus drivers. As versões recentes do Red Hat Enterprise Linux CoreOS (RHCOS) vêm com essas ferramentas instaladas por padrão.

Ferramentas NFS

"[Instale as ferramentas NFS](#)" Se você estiver usando: `ontap-nas`, `ontap-nas-economy`, `ontap-nas-flexgroup`, `azure-netapp-files`, `gcp-cvs`.

Ferramentas iSCSI

"[Instale as ferramentas iSCSI](#)" Se você estiver usando: `ontap-san`, `ontap-san-economy`, `solidfire-san`.

Ferramentas NVMe

"[Instale as ferramentas NVMe](#)" se você estiver usando `ontap-san` para o protocolo de memória não volátil expressa (NVMe) sobre TCP (NVMe/TCP).



A NetApp recomenda o ONTAP 9.12 ou posterior para NVMe/TCP.

Ferramentas SCSI sobre FC

Consulte "[Formas de configurar hosts SAN FC e FC-NVMe](#)" Para obter mais informações sobre como configurar seus hosts SAN FC e FC-NVMe.

"[Instale as ferramentas FC](#)" se você estiver usando `ontap-san` com `sanType fcp` (SCSI sobre FC).

Pontos a considerar: * SCSI sobre FC é suportado em ambientes OpenShift e KubeVirt. * SCSI sobre FC não é suportado no Docker. * A autorrecuperação do iSCSI não se aplica ao SCSI sobre FC.

Descoberta de serviços de nó

O Trident tenta detectar automaticamente se o nó pode executar serviços iSCSI ou NFS.



A descoberta de serviços de nós identifica os serviços descobertos, mas não garante que estejam configurados corretamente. Por outro lado, a ausência de um serviço detectado não garante que a montagem do volume falhará.

Revisar eventos

O Trident cria eventos para o nó identificar os serviços descobertos. Para rever esses eventos, execute:

```
kubectl get event -A --field-selector involvedObject.name=<Kubernetes node name>
```

Análise dos serviços descobertos

O Trident identifica os serviços habilitados para cada nó no CR do nó Trident . Para visualizar os serviços detectados, execute:

```
tridentctl get node -o wide -n <Trident namespace>
```

Volumes NFS

Instale as ferramentas NFS usando os comandos para o seu sistema operacional. Certifique-se de que o serviço NFS seja iniciado durante a inicialização do sistema.

RHEL 8+

```
sudo yum install -y nfs-utils
```

Ubuntu

```
sudo apt-get install -y nfs-common
```



Reinicie os nós de trabalho após instalar as ferramentas NFS para evitar falhas ao conectar volumes aos contêineres.

volumes iSCSI

O Trident pode estabelecer automaticamente uma sessão iSCSI, analisar LUNs, descobrir dispositivos multipath, formatá-los e montá-los em um pod.

Capacidades de autorrecuperação do iSCSI

Para sistemas ONTAP , o Trident executa a autorrecuperação iSCSI a cada cinco minutos para:

1. **Identifique** o estado desejado da sessão iSCSI e o estado atual da sessão iSCSI.
2. **Compare** o estado desejado com o estado atual para identificar os reparos necessários. A Trident determina as prioridades de reparo e quando antecipar os reparos.
3. **Executar os reparos** necessários para retornar o estado atual da sessão iSCSI ao estado desejado.



Os registros da atividade de autocura estão localizados em `trident-main` contêiner no respectivo pod do Daemonset. Para visualizar os registros, você precisa ter configurado debug para "verdadeiro" durante a instalação do Trident .

Os recursos de autorrecuperação do Trident iSCSI podem ajudar a prevenir:

- Sessões iSCSI obsoletas ou com problemas de funcionamento que podem ocorrer após uma falha de conectividade de rede. No caso de uma sessão inativa, o Trident aguarda sete minutos antes de encerrar a sessão para restabelecer a conexão com um portal.



Por exemplo, se os segredos CHAP fossem rotacionados no controlador de armazenamento e a rede perdesse a conectividade, os segredos CHAP antigos (obsoletos) poderiam persistir. A função de autorrecuperação consegue reconhecer isso e restabelecer automaticamente a sessão para aplicar os segredos CHAP atualizados.

- Sessões iSCSI ausentes
- LUNs ausentes

Pontos a considerar antes de atualizar o Trident

- Se apenas os grupos de interfaces (igroups) por nó (introduzidos na versão 23.04+) estiverem em uso, a autorrecuperação do iSCSI iniciará novas varreduras SCSI para todos os dispositivos no barramento SCSI.
- Se apenas grupos de interface (igroups) com escopo de backend (obsoletos a partir da versão 23.04) estiverem em uso, a autorrecuperação do iSCSI iniciará novas varreduras SCSI para obter os IDs de LUN exatos no barramento SCSI.
- Se uma combinação de igroups por nó e igroups com escopo de backend estiver em uso, a autorrecuperação do iSCSI iniciará novas varreduras SCSI para obter os IDs de LUN exatos no barramento SCSI.

Instale as ferramentas iSCSI

Instale as ferramentas iSCSI usando os comandos para o seu sistema operacional.

Antes de começar

- Cada nó no cluster Kubernetes deve ter um IQN único. **Este é um pré-requisito necessário.**
- Se estiver usando o RHCOS versão 4.5 ou posterior, ou outra distribuição Linux compatível com RHEL, com o `solidfire-san` Para drivers e Element OS 12.5 ou anterior, certifique-se de que o algoritmo de autenticação CHAP esteja definido como MD5 em `/etc/iscsi/iscsid.conf`. Os algoritmos CHAP seguros e compatíveis com FIPS, SHA1, SHA-256 e SHA3-256, estão disponíveis no Element 12.7.

```
sudo sed -i 's/^\(node.session.auth.chap_algs\) .*/\1 = MD5/'
/etc/iscsi/iscsid.conf
```

- Ao usar nós de trabalho que executam RHEL/Red Hat Enterprise Linux CoreOS (RHCOS) com PVs iSCSI, especifique o `discard` A opção `mountOption` na `StorageClass` permite realizar a recuperação de espaço em linha. Consulte ["Documentação da Red Hat"](#).
- Certifique-se de ter atualizado para a versão mais recente do `multipath-tools`.

RHEL 8+

1. Instale os seguintes pacotes do sistema:

```
sudo yum install -y lsscsi iscsi-initiator-utils device-mapper-multipath
```

2. Verifique se a versão do iscsi-initiator-utils é 6.2.0.874-2.el7 ou posterior:

```
rpm -q iscsi-initiator-utils
```

3. Configure a digitalização para manual:

```
sudo sed -i 's/^\(node.session.scan\) .*/\1 = manual/'  
/etc/iscsi/iscsid.conf
```

4. Ativar multipathing:

```
sudo mpathconf --enable --with_multipathd y --find_multipaths n
```



Garantir /etc/multipath.conf contém find_multipaths no sob defaults .

5. Certifique-se de que iscsid e multipathd estão em execução:

```
sudo systemctl enable --now iscsid multipathd
```

6. Ative e inicie iscsi :

```
sudo systemctl enable --now iscsi
```

Ubuntu

1. Instale os seguintes pacotes do sistema:

```
sudo apt-get install -y open-iscsi lsscsi sg3-utils multipath-tools  
scsitools
```

2. Verifique se a versão do open-iscsi é 2.0.874-5ubuntu2.10 ou posterior (para bionic) ou 2.0.874-7.1ubuntu6.1 ou posterior (para focal):

```
dpkg -l open-iscsi
```

3. Configure a digitalização para manual:

```
sudo sed -i 's/^\(node.session.scan\).*\/\1 = manual/'  
/etc/iscsi/iscsid.conf
```

4. Ativar multipathing:

```
sudo tee /etc/multipath.conf <<-EOF  
defaults {  
    user_friendly_names yes  
    find_multipaths no  
}  
EOF  
sudo systemctl enable --now multipath-tools.service  
sudo service multipath-tools restart
```



Garantir `/etc/multipath.conf` contém `find_multipaths no` sob `defaults`.

5. Certifique-se de que `open-iscsi` e `multipath-tools` estão habilitados e em execução:

```
sudo systemctl status multipath-tools  
sudo systemctl enable --now open-iscsi.service  
sudo systemctl status open-iscsi
```



Para o Ubuntu 18.04, você deve descobrir as portas de destino com `iscsiadm` antes de começar `open-iscsi` para que o daemon iSCSI seja iniciado. Você também pode modificar o `iscsi` serviço para iniciar `iscsid` automaticamente.

Configure ou desative a autorrecuperação do iSCSI.

Você pode configurar as seguintes opções de autorrecuperação do Trident iSCSI para corrigir sessões obsoletas:

- **Intervalo de autorrecuperação do iSCSI:** Determina a frequência com que a autorrecuperação do iSCSI é acionada (padrão: 5 minutos). Você pode configurá-lo para ser executado com mais frequência, definindo um número menor, ou com menos frequência, definindo um número maior.



Definir o intervalo de autorrecuperação do iSCSI para 0 interrompe completamente a autorrecuperação do iSCSI. Não recomendamos desativar a autorrecuperação do iSCSI; ela só deve ser desativada em cenários específicos, quando a autorrecuperação do iSCSI não estiver funcionando conforme o esperado ou para fins de depuração.

- **Tempo de espera de auto-recuperação do iSCSI:** Determina a duração que a auto-recuperação do iSCSI aguarda antes de encerrar uma sessão com problemas e tentar entrar novamente (padrão: 7 minutos). Você pode configurá-lo para um número maior, de forma que as sessões identificadas como não íntegras tenham que esperar mais tempo antes de serem desconectadas e, em seguida, uma nova tentativa de login seja feita, ou para um número menor, para desconectar e entrar mais cedo.

Leme

Para configurar ou alterar as configurações de autorrecuperação do iSCSI, passe o `iscsiSelfHealingInterval` e `iscsiSelfHealingWaitTime` parâmetros durante a instalação ou atualização do Helm.

O exemplo a seguir define o intervalo de autorrecuperação do iSCSI para 3 minutos e o tempo de espera para autorrecuperação para 6 minutos:

```
helm install trident trident-operator-100.2506.0.tgz --set
iscsiSelfHealingInterval=3m0s --set iscsiSelfHealingWaitTime=6m0s -n
trident
```

tridentctl

Para configurar ou alterar as configurações de autorrecuperação do iSCSI, passe o `iscsi-self-healing-interval` e `iscsi-self-healing-wait-time` parâmetros durante a instalação ou atualização do `tridentctl`.

O exemplo a seguir define o intervalo de autorrecuperação do iSCSI para 3 minutos e o tempo de espera para autorrecuperação para 6 minutos:

```
tridentctl install --iscsi-self-healing-interval=3m0s --iscsi-self
-healing-wait-time=6m0s -n trident
```

Volumes NVMe/TCP

Instale as ferramentas NVMe usando os comandos específicos do seu sistema operacional.



- O NVMe requer RHEL 9 ou posterior.
- Se a versão do kernel do seu nó Kubernetes for muito antiga ou se o pacote NVMe não estiver disponível para a sua versão do kernel, talvez seja necessário atualizar a versão do kernel do seu nó para uma que inclua o pacote NVMe.

RHEL 9

```
sudo yum install nvme-cli
sudo yum install linux-modules-extra-$(uname -r)
sudo modprobe nvme-tcp
```

Ubuntu

```
sudo apt install nvme-cli
sudo apt -y install linux-modules-extra-$(uname -r)
sudo modprobe nvme-tcp
```

Verifique a instalação

Após a instalação, verifique se cada nó no cluster Kubernetes possui um NQN exclusivo usando o comando:

```
cat /etc/nvme/hostnqn
```



Trident modifica o `ctrl_device_tmo` Valor para garantir que o NVMe não abandone o caminho caso ele falhe. Não altere esta configuração.

SCSI sobre volumes FC

Agora você pode usar o protocolo Fibre Channel (FC) com o Trident para provisionar e gerenciar recursos de armazenamento no sistema ONTAP .

Pré-requisitos

Configure as definições de rede e de nó necessárias para a FC.

Configurações de rede

1. Obtenha o WWPN das interfaces de destino. Consulte ["exibição de interface de rede"](#) para mais informações.
2. Obtenha o WWPN para as interfaces no iniciador (Host).

Consulte os utilitários correspondentes do sistema operacional do host.

3. Configure o zoneamento no switch FC usando os WWPNs do host e do destino.

Consulte a documentação do respectivo fornecedor do switch para obter informações.

Consulte a seguinte documentação do ONTAP para obter detalhes:

- ["Visão geral do zoneamento Fibre Channel e FCoE"](#)
- ["Formas de configurar hosts SAN FC e FC-NVMe"](#)

Instale as ferramentas FC

Instale as ferramentas FC usando os comandos para o seu sistema operacional.

- Ao usar nós de trabalho que executam RHEL/Red Hat Enterprise Linux CoreOS (RHCOS) com PVs FC, especifique o `discard`. A opção `mountOption` na `StorageClass` permite realizar a recuperação de espaço em linha. Consulte ["Documentação da Red Hat"](#).

RHEL 8+

1. Instale os seguintes pacotes do sistema:

```
sudo yum install -y lsscsi device-mapper-multipath
```

2. Ativar multipathing:

```
sudo mpathconf --enable --with_multipathd y --find_multipaths n
```



Garantir /etc/multipath.conf contém find_multipaths no **sob** defaults.

3. Certifique-se de que multipathd está em execução:

```
sudo systemctl enable --now multipathd
```

Ubuntu

1. Instale os seguintes pacotes do sistema:

```
sudo apt-get install -y lsscsi sg3-utils multipath-tools scsiboot
```

2. Ativar multipathing:

```
sudo tee /etc/multipath.conf <<-EOF
defaults {
    user_friendly_names yes
    find_multipaths no
}
EOF
sudo systemctl enable --now multipath-tools.service
sudo service multipath-tools restart
```



Garantir /etc/multipath.conf contém find_multipaths no **sob** defaults.

3. Certifique-se de que multipath-tools Está habilitado e em execução:

```
sudo systemctl status multipath-tools
```

Configurar e gerenciar backends

Configurar backends

Um backend define a relação entre o Trident e um sistema de armazenamento. Ele informa ao Trident como se comunicar com esse sistema de armazenamento e como o Trident deve provisionar volumes a partir dele.

O Trident oferece automaticamente pools de armazenamento de backends que correspondem aos requisitos definidos por uma classe de armazenamento. Aprenda como configurar o backend do seu sistema de armazenamento.

- ["Configure um backend do Azure NetApp Files"](#)
- ["Configure um backend do Google Cloud NetApp Volumes"](#)
- ["Configure um Cloud Volumes Service para o backend do Google Cloud Platform."](#)
- ["Configure um backend NetApp HCI ou SolidFire."](#)
- ["Configure um backend com drivers ONTAP ou Cloud Volumes ONTAP NAS."](#)
- ["Configure um backend com drivers ONTAP SAN ou Cloud Volumes ONTAP."](#)
- ["Use o Trident com o Amazon FSx for NetApp ONTAP"](#)

Azure NetApp Files

Configure um backend do Azure NetApp Files

Você pode configurar o Azure NetApp Files como backend para o Trident. Você pode conectar volumes NFS e SMB usando um backend do Azure NetApp Files . O Trident também oferece suporte ao gerenciamento de credenciais usando identidades gerenciadas para clusters do Azure Kubernetes Services (AKS).

Detalhes do driver Azure NetApp Files

O Trident fornece os seguintes drivers de armazenamento do Azure NetApp Files para comunicação com o cluster. Os modos de acesso suportados são: *ReadWriteOnce* (RWO), *ReadOnlyMany* (ROX), *ReadWriteMany* (RWX), *ReadWriteOncePod* (RWOP).

Motorista	Protocolo	modo de volume	Modos de acesso suportados	Sistemas de arquivos suportados
azure-netapp-files	NFS SMB	Sistema de arquivos	RWO, ROX, RWX, RWOP	nfs, smb

Considerações

- O serviço Azure NetApp Files não suporta volumes menores que 50 GiB. O Trident cria automaticamente volumes de 50 GiB se um volume menor for solicitado.
- O Trident suporta volumes SMB montados em pods executados apenas em nós Windows.

Identities gerenciadas para AKS

Suporte Trident "identidades gerenciadas" para clusters do Azure Kubernetes Services. Para aproveitar o gerenciamento simplificado de credenciais oferecido pelas identidades gerenciadas, você precisa ter:

- Um cluster Kubernetes implantado usando o AKS.
- Identidades gerenciadas configuradas no cluster Kubernetes do AKS
- Trident instalado que inclui o `cloudProvider` para especificar "Azure" .

Operador do Trident

Para instalar o Trident usando o operador Trident , edite `tridentorchestrator_cr.yaml` para definir `cloudProvider` para "Azure" . Por exemplo:

```
apiVersion: trident.netapp.io/v1
kind: TridentOrchestrator
metadata:
  name: trident
spec:
  debug: true
  namespace: trident
  imagePullPolicy: IfNotPresent
  cloudProvider: "Azure"
```

Leme

O exemplo a seguir instala conjuntos Trident. `cloudProvider` para o Azure usando a variável de ambiente `$CP` :

```
helm install trident trident-operator-100.2506.0.tgz --create
--namespace --namespace <trident-namespace> --set cloudProvider=$CP
```

`tridentctl`

O exemplo a seguir instala o Trident e configura o `cloudProvider` bandeira para Azure :

```
tridentctl install --cloud-provider="Azure" -n trident
```

Identidade na nuvem para AKS

A identidade na nuvem permite que os pods do Kubernetes acessem recursos do Azure autenticando-se como uma identidade de carga de trabalho, em vez de fornecer credenciais explícitas do Azure.

Para aproveitar as vantagens da identidade na nuvem no Azure, você precisa ter:

- Um cluster Kubernetes implantado usando o AKS.

- Identidade de carga de trabalho e emissor OIDC configurados no cluster Kubernetes do AKS
- Trident instalado que inclui o `cloudProvider` para especificar "Azure" e `cloudIdentity` especificando a identidade da carga de trabalho

Operador do Trident

Para instalar o Trident usando o operador Trident, edite `tridentorchestrator_cr.yaml` para definir `cloudProvider` para "Azure" e definir `cloudIdentity` para `azure.workload.identity/client-id: xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx`.

Por exemplo:

```
apiVersion: trident.netapp.io/v1
kind: TridentOrchestrator
metadata:
  name: trident
spec:
  debug: true
  namespace: trident
  imagePullPolicy: IfNotPresent
  cloudProvider: "Azure"
  cloudIdentity: 'azure.workload.identity/client-id: xxxxxxxx-xxxx-
xxxx-xxxx-xxxxxxxxxxxx' # Edit
```

Leme

Defina os valores para os parâmetros **cloud-provider (CP)** e **cloud-identity (CI)** usando as seguintes variáveis de ambiente:

```
export CP="Azure"
export CI="'azure.workload.identity/client-id: xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx'"
```

O exemplo a seguir instala o Trident e configura `cloudProvider` para o Azure usando a variável de ambiente `$CP` e define o `cloudIdentity` usando a variável de ambiente `$CI`:

```
helm install trident trident-operator-100.6.0.tgz --set
cloudProvider=$CP --set cloudIdentity="$CI"
```

<code>tridentctl</code>

Defina os valores para os parâmetros **provedor de nuvem** e **identidade de nuvem** usando as seguintes variáveis de ambiente:

```
export CP="Azure"
export CI="azure.workload.identity/client-id: xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx"
```

O exemplo a seguir instala o Trident e configura o `cloud-provider` bandeira para `$CP`, e `cloud-identity` para `$CI`:

```
tridentctl install --cloud-provider=$CP --cloud-identity="$CI" -n
trident
```

Prepare-se para configurar um backend do Azure NetApp Files.

Antes de configurar o backend do Azure NetApp Files , você precisa garantir que os seguintes requisitos sejam atendidos.

Pré-requisitos para volumes NFS e SMB

Se você estiver usando o Azure NetApp Files pela primeira vez ou em um novo local, será necessária alguma configuração inicial para configurar o Azure NetApp Files e criar um volume NFS. Consulte ["Azure: Configure o Azure NetApp Files e crie um volume NFS."](#) .

Para configurar e usar um ["Azure NetApp Files"](#) Para o backend, você precisa do seguinte:



- `subscriptionID`, `tenantID` , `clientID` , `location` , e `clientSecret` São opcionais ao usar identidades gerenciadas em um cluster AKS.
- `tenantID`, `clientID` , e `clientSecret` São opcionais ao usar uma identidade de nuvem em um cluster AKS.

- Uma piscina de capacidade máxima. Consulte ["Microsoft: Criar um pool de capacidade para o Azure NetApp Files"](#) .
- Uma sub-rede delegada ao Azure NetApp Files. Consulte ["Microsoft: Delegar uma sub-rede ao Azure NetApp Files"](#) .
- `subscriptionID` de uma assinatura do Azure com o Azure NetApp Files ativado.
- `tenantID`, `clientID` , e `clientSecret` de um ["Registro do aplicativo"](#) no Azure Active Directory com permissões suficientes para o serviço Azure NetApp Files . O cadastro no aplicativo deve usar um dos seguintes métodos:
 - O papel de Proprietário ou Colaborador ["predefinido pelo Azure"](#) .
 - UM ["Função de Colaborador personalizada"](#) no nível de assinatura(`assignableScopes`) com as seguintes permissões, que se limitam apenas ao que o Trident exige. Após criar a função personalizada, ["Atribua a função usando o portal do Azure."](#) .

Função de colaborador personalizado

```
{
  "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/write",

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

          "Microsoft.NetApp/netAppAccounts/capacityPools/volumes/MountTargets/read",
          "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": []
  }
]
}

```

- O Azure location que contém pelo menos um ["sub-rede delegada"](#) . A partir da versão 22.01 do Trident , o location O parâmetro é um campo obrigatório no nível superior do arquivo de configuração do backend. Os valores de localização especificados em pools virtuais são ignorados.
- Para usar Cloud Identity , pegue o client ID de um ["identidade gerenciada atribuída pelo usuário"](#) e especifique esse ID em azure.workload.identity/client-id: xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx .

Requisitos adicionais para volumes de PME

Para criar um volume SMB, você precisa ter:

- Active Directory configurado e conectado ao Azure NetApp Files. Consulte ["Microsoft: Criar e gerenciar conexões do Active Directory para o Azure NetApp Files"](#) .
- Um cluster Kubernetes com um nó controlador Linux e pelo menos um nó de trabalho Windows executando o Windows Server 2022. O Trident suporta volumes SMB montados em pods executados apenas em nós Windows.
- É necessário pelo menos um segredo Trident contendo suas credenciais do Active Directory para que o Azure NetApp Files possa se autenticar no Active Directory. Para gerar segredos smbcreds :

```

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

```

- Um proxy CSI configurado como um serviço do Windows. Para configurar um csi-proxy , consulte ["GitHub: Proxy CSI"](#) ou ["GitHub: CSI Proxy para Windows"](#) para nós do Kubernetes executados no Windows.

Opções e exemplos de configuração do backend do Azure NetApp Files

Saiba mais sobre as opções de configuração de back-end NFS e SMB para o Azure NetApp Files e veja exemplos de configuração.

Opções de configuração do backend

O Trident usa sua configuração de back-end (sub-rede, rede virtual, nível de serviço e localização) para criar volumes do Azure NetApp Files em pools de capacidade disponíveis na localização solicitada e que correspondam ao nível de serviço e à sub-rede solicitados.



* A partir da versão NetApp Trident 25.06, pools de capacidade de QoS manuais são suportados como uma prévia técnica.*

Os back-ends do Azure NetApp Files oferecem essas opções de configuração.

Parâmetro	Descrição	Padrão
version		Sempre 1
storageDriverName	Nome do driver de armazenamento	"azure-netapp-files"
backendName	Nome personalizado ou o backend de armazenamento	Nome do motorista + "_" + caracteres aleatórios
subscriptionID	O ID da assinatura da sua assinatura do Azure. Opcional quando as identidades gerenciadas estão habilitadas em um cluster do AKS.	
tenantID	O ID do locatário de um registro de aplicativo é opcional quando identidades gerenciadas ou identidade na nuvem são usadas em um cluster AKS.	
clientID	O ID do cliente de um registro de aplicativo é opcional quando identidades gerenciadas ou identidade na nuvem são usadas em um cluster AKS.	
clientSecret	O segredo do cliente de um registro de aplicativo é opcional quando identidades gerenciadas ou identidade na nuvem são usadas em um cluster AKS.	
serviceLevel	Um de Standard , Premium , ou Ultra	"" (aleatório)
location	Nome da localização do Azure onde os novos volumes serão criados. Opcional quando as identidades gerenciadas estão habilitadas em um cluster AKS.	

Parâmetro	Descrição	Padrão
resourceGroups	Lista de grupos de recursos para filtrar recursos descobertos	"" (sem filtro)
netappAccounts	Lista de contas NetApp para filtrar recursos descobertos	"" (sem filtro)
capacityPools	Lista de pools de capacidade para filtrar recursos descobertos	"" (sem filtro, aleatório)
virtualNetwork	Nome de uma rede virtual com uma sub-rede delegada	""
subnet	Nome de uma sub-rede delegada a Microsoft.Netapp/volumes	""
networkFeatures	Conjunto de recursos de VNet para um volume, pode ser Basic ou Standard. O recurso de Recursos de Rede não está disponível em todas as regiões e pode ser necessário ativá-lo por meio de uma assinatura. Especificando networkFeatures Quando a funcionalidade não está habilitada, o provisionamento de volumes falha.	""
nfsMountOptions	Controle preciso das opções de montagem NFS. Ignorado para volumes SMB. Para montar volumes usando NFS versão 4.1, inclua nfsvers=4 Na lista de opções de montagem separadas por vírgulas, escolha NFS v4.1. As opções de montagem definidas na definição de uma classe de armazenamento substituem as opções de montagem definidas na configuração do backend.	"nfsvers=3"
limitVolumeSize	O provisionamento falhará se o tamanho do volume solicitado for superior a este valor.	"" (não aplicado por padrão)
debugTraceFlags	Sinalizadores de depuração a serem usados na resolução de problemas. Exemplo, \{"api": false, "method": true, "discovery": true\}. Não utilize esta opção a menos que esteja solucionando problemas e precise de um despejo de logs detalhado.	nulo

Parâmetro	Descrição	Padrão
nasType	Configure a criação de volumes NFS ou SMB. As opções são <code>nfs</code> , <code>smb</code> ou nulo. Definir como nulo utiliza, por padrão, volumes NFS.	<code>nfs</code>
supportedTopologies	Representa uma lista de regiões e zonas que são suportadas por este sistema. Para obter mais informações, consulte "Utilizar a topologia CSI" .	
qosType	Representa o tipo de QoS: Automático ou Manual. Prévia técnica para Trident 25.06	Auto
maxThroughput	Define a taxa de transferência máxima permitida em MiB/s. Suportado apenas para pools de capacidade de QoS manual. Prévia técnica para Trident 25.06	4 MiB/sec



Para obter mais informações sobre os recursos de rede, consulte ["Configure os recursos de rede para um volume do Azure NetApp Files."](#)

Permissões e recursos necessários

Se você receber um erro "Nenhum pool de capacidade encontrado" ao criar um PVC, é provável que o registro do seu aplicativo não tenha as permissões e os recursos necessários (sub-rede, rede virtual, pool de capacidade) associados. Se o modo de depuração estiver ativado, o Trident registrará os recursos do Azure descobertos quando o backend for criado. Verifique se está sendo utilizada a função apropriada.

Os valores para `resourceGroups`, `netappAccounts`, `capacityPools`, `virtualNetwork`, e `subnet` Podem ser especificados usando nomes curtos ou nomes totalmente qualificados. Na maioria das situações, recomenda-se o uso de nomes completos, pois nomes curtos podem corresponder a vários recursos com o mesmo nome.

O `resourceGroups`, `netappAccounts`, e `capacityPools` Os valores são filtros que restringem o conjunto de recursos descobertos àqueles disponíveis para este backend de armazenamento e podem ser especificados em qualquer combinação. Os nomes completos seguem este formato:

Tipo	Formatar
Grupo de recursos	<grupo de recursos>
Conta NetApp	<grupo de recursos>/<conta NetApp>
piscina de capacidade	<grupo de recursos>/<conta NetApp>/<pool de capacidade>
Rede virtual	<grupo de recursos>/<rede virtual>
Sub-rede	<grupo de recursos>/<rede virtual>/<sub-rede>

Provisionamento de volume

Você pode controlar o provisionamento de volumes padrão especificando as seguintes opções em uma seção específica do arquivo de configuração. Consulte [Configurações de exemplo](#) para mais detalhes.

Parâmetro	Descrição	Padrão
<code>exportRule</code>	Regras de exportação para novos volumes. <code>exportRule</code> Deve ser uma lista separada por vírgulas de qualquer combinação de endereços IPv4 ou sub-redes IPv4 na notação CIDR. Ignorado para volumes SMB.	"0.0.0.0/0"
<code>snapshotDir</code>	Controla a visibilidade do diretório <code>.snapshot</code> .	"verdadeiro" para NFSv4 "falso" para NFSv3
<code>size</code>	O tamanho padrão de novos volumes	"100G"
<code>unixPermissions</code>	Permissões Unix de novos volumes (4 dígitos octais). Ignorado para volumes SMB.	"" (Recurso em pré-visualização, requer inclusão na lista de permissões na assinatura)

Configurações de exemplo

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

Configuração mínima

Esta é a configuração mínima absoluta do backend. Com essa configuração, o Trident descobre todas as suas contas NetApp, pools de capacidade e sub-redes delegadas ao Azure NetApp Files no local configurado e coloca novos volumes em um desses pools e sub-redes aleatoriamente. Porque `nasType` é omitido, o `nfs` A configuração padrão será aplicada e o servidor provisionará volumes NFS.

Essa configuração é ideal para quem está começando a usar o Azure NetApp Files e experimentando novos recursos, mas, na prática, você provavelmente desejará fornecer um escopo adicional para os volumes provisionados.

```
---
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-anf-1
  namespace: trident
spec:
  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
```

Identities gerenciadas para AKS

Esta configuração de backend omite `subscriptionID`, `tenantID`, `clientId`, e `clientSecret`, que são opcionais ao usar identities gerenciadas.

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-anf-1
  namespace: trident
spec:
  version: 1
  storageDriverName: azure-netapp-files
  capacityPools:
    - ultra-pool
  resourceGroups:
    - aks-ami-eastus-rg
  netappAccounts:
    - smb-na
  virtualNetwork: eastus-prod-vnet
  subnet: eastus-anf-subnet
```


Identidade na nuvem para AKS

Esta configuração de backend omite `tenantID`, `clientID`, e `clientSecret`, que são opcionais ao usar uma identidade na nuvem.

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-anf-1
  namespace: trident
spec:
  version: 1
  storageDriverName: azure-netapp-files
  capacityPools:
    - ultra-pool
  resourceGroups:
    - aks-ami-eastus-rg
  netappAccounts:
    - smb-na
  virtualNetwork: eastus-prod-vnet
  subnet: eastus-anf-subnet
  location: eastus
  subscriptionID: 9f87c765-4774-fake-ae98-a721add45451
```

Configuração específica de nível de serviço com filtros de capacidade

Essa configuração de backend coloca volumes no Azure. `eastus` localização em um `Ultra` pool de capacidade. O Trident descobre automaticamente todas as sub-redes delegadas ao Azure NetApp Files nesse local e coloca um novo volume em uma delas aleatoriamente.

```
---
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
```

Exemplo de backend com pools de capacidade de QoS manuais

Essa configuração de backend coloca volumes no Azure. eastus Localização com pools de capacidade QoS manuais. **Prévia técnica no NetApp Trident 25.06.**

```
---
version: 1
storageDriverName: azure-netapp-files
backendName: anfl
location: eastus
labels:
  clusterName: test-cluster-1
  cloud: anf
  nasType: nfs
defaults:
  qosType: Manual
storage:
  - serviceLevel: Ultra
    labels:
      performance: gold
    defaults:
      maxThroughput: 10
  - serviceLevel: Premium
    labels:
      performance: silver
    defaults:
      maxThroughput: 5
  - serviceLevel: Standard
    labels:
      performance: bronze
    defaults:
      maxThroughput: 3
```

Configuração avançada

Essa configuração de backend reduz ainda mais o escopo do posicionamento de volumes para uma única sub-rede e também modifica algumas configurações padrão de provisionamento de volumes.

```
---  
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"
```

Configuração de pool virtual

Essa configuração de backend define vários pools de armazenamento em um único arquivo. Isso é útil quando você tem vários pools de capacidade que suportam diferentes níveis de serviço e deseja criar classes de armazenamento no Kubernetes que os representem. Rótulos de piscinas virtuais foram usados para diferenciar as piscinas com base em 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
```

Configuração de topologias suportadas

O Trident facilita o provisionamento de volumes para cargas de trabalho com base em regiões e zonas de disponibilidade. O `supportedTopologies` O bloco nesta configuração de backend é usado para fornecer uma lista de regiões e zonas por backend. Os valores de região e zona especificados aqui devem corresponder aos valores de região e zona dos rótulos em cada nó do cluster Kubernetes. Essas regiões e zonas representam a lista de valores permitidos que podem ser fornecidos em uma classe de armazenamento. Para classes de armazenamento que contêm um subconjunto das regiões e zonas fornecidas em um backend, o Trident cria volumes na região e zona mencionadas. Para obter mais informações, consulte ["Utilizar a topologia CSI"](#).

```
---
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
supportedTopologies:
  - topology.kubernetes.io/region: eastus
    topology.kubernetes.io/zone: eastus-1
  - topology.kubernetes.io/region: eastus
    topology.kubernetes.io/zone: eastus-2
```

Definições de classe de armazenamento

A seguir `StorageClass` As definições referem-se aos conjuntos de armazenamento acima.

Definições de exemplo usando `parameter.selector` campo

Usando `parameter.selector` Você pode especificar para cada um `StorageClass` O pool virtual que é usado para hospedar um volume. O volume terá os aspectos definidos na piscina escolhida.

```

---
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

```

Exemplos de definições para volumes SMB

Usando `nasType`, `node-stage-secret-name`, e `node-stage-secret-namespace` Você pode especificar um volume SMB e fornecer as credenciais necessárias do Active Directory.

Configuração básica no namespace padrão

```
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"
```

Utilizando segredos diferentes por namespace

```
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}
```

Utilizando segredos diferentes em cada volume.

```
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`Filtros para pools que suportam volumes SMB. `nasType: nfs ou nasType: null Filtros para pools NFS.`

Crie o backend

Após criar o arquivo de configuração do backend, execute o seguinte comando:

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

Se a criação do backend falhar, há algo errado com a configuração do backend. Você pode visualizar os registros para determinar a causa executando o seguinte comando:

```
tridentctl logs
```

Após identificar e corrigir o problema com o arquivo de configuração, você poderá executar o comando de criação novamente.

Google Cloud NetApp Volumes

Configure um backend do Google Cloud NetApp Volumes

Agora você pode configurar o Google Cloud NetApp Volumes como backend para o Trident. Você pode conectar volumes NFS e SMB usando um backend do Google Cloud NetApp Volumes .

Detalhes do driver do Google Cloud NetApp Volumes

A Trident fornece o `google-cloud-netapp-volumes` O driver deve se comunicar com o cluster. Os modos de acesso suportados são: *ReadWriteOnce* (RWO), *ReadOnlyMany* (ROX), *ReadWriteMany* (RWX), *ReadWriteOncePod* (RWOP).

Motorista	Protocolo	modo de volume	Modos de acesso suportados	Sistemas de arquivos suportados
google-cloud-netapp-volumes	NFS SMB	Sistema de arquivos	RWO, ROX, RWX, RWOP	nfs, smb

Identidade na nuvem para GKE

A identidade na nuvem permite que os pods do Kubernetes acessem recursos do Google Cloud autenticando-se como uma identidade de carga de trabalho, em vez de fornecer credenciais explícitas do Google Cloud.

Para aproveitar as vantagens da identidade na nuvem do Google Cloud, você precisa ter:

- Um cluster Kubernetes implantado usando o GKE.
- Identidade de carga de trabalho configurada no cluster GKE e servidor de metadados GKE configurado nos pools de nós.

- Uma conta de serviço do GCP com a função de administrador de Google Cloud NetApp Volumes (roles/netapp.admin) ou uma função personalizada.
- O Trident instalado inclui o cloudProvider para especificar "GCP" e o cloudIdentity especificando a nova conta de serviço do GCP. Segue abaixo um exemplo.

Operador do Trident

Para instalar o Trident usando o operador Trident, edite `tridentorchestrator_cr.yaml` para definir `cloudProvider` para "GCP" e definir `cloudIdentity` para `iam.gke.io/gcp-service-account: cloudvolumes-admin-sa@mygcpproject.iam.gserviceaccount.com`.

Por exemplo:

```
apiVersion: trident.netapp.io/v1
kind: TridentOrchestrator
metadata:
  name: trident
spec:
  debug: true
  namespace: trident
  imagePullPolicy: IfNotPresent
  cloudProvider: "GCP"
  cloudIdentity: 'iam.gke.io/gcp-service-account: cloudvolumes-
admin-sa@mygcpproject.iam.gserviceaccount.com'
```

Leme

Defina os valores para os parâmetros **cloud-provider (CP)** e **cloud-identity (CI)** usando as seguintes variáveis de ambiente:

```
export CP="GCP"
export ANNOTATION="'iam.gke.io/gcp-service-account: cloudvolumes-admin-
sa@mygcpproject.iam.gserviceaccount.com'"
```

O exemplo a seguir instala o Trident e configura `cloudProvider` para o GCP usando a variável de ambiente `$CP` e define o `cloudIdentity` usando a variável de ambiente `$ANNOTATION`:

```
helm install trident trident-operator-100.6.0.tgz --set
cloudProvider=$CP --set cloudIdentity="$ANNOTATION"
```

<code>tridentctl</code>

Defina os valores para os parâmetros **provedor de nuvem** e **identidade de nuvem** usando as seguintes variáveis de ambiente:

```
export CP="GCP"
export ANNOTATION="'iam.gke.io/gcp-service-account: cloudvolumes-admin-
sa@mygcpproject.iam.gserviceaccount.com'"
```

O exemplo a seguir instala o Trident e configura o `cloud-provider` bandeira para `$CP`, e `cloud-identity` para `$ANNOTATION`:

```
tridentctl install --cloud-provider=$CP --cloud
-identity="$ANNOTATION" -n trident
```

Prepare-se para configurar um backend do Google Cloud NetApp Volumes.

Antes de configurar o backend do Google Cloud NetApp Volumes , você precisa garantir que os seguintes requisitos sejam atendidos.

Pré-requisitos para volumes NFS

Se você estiver usando o Google Cloud NetApp Volumes pela primeira vez ou em um novo local, será necessária alguma configuração inicial para configurar o Google Cloud NetApp Volumes e criar um volume NFS. Consulte ["Antes de começar"](#) .

Certifique-se de ter o seguinte antes de configurar o backend do Google Cloud NetApp Volumes :

- Uma conta do Google Cloud configurada com o serviço Google Cloud NetApp Volumes . Consulte ["Google Cloud NetApp Volumes"](#) .
- Número do projeto da sua conta do Google Cloud. Consulte ["Identificação de projetos"](#) .
- Uma conta de serviço do Google Cloud com o administrador de volumes da NetApp.(roles/netapp.admin) papel. Consulte ["Funções e permissões de Gestão de Identidade e Acesso"](#) .
- Arquivo de chave API para sua conta GCNV. Consulte ["Criar uma chave de conta de serviço"](#)
- Uma piscina de armazenamento. Consulte ["Visão geral dos pools de armazenamento"](#) .

Para obter mais informações sobre como configurar o acesso aos Google Cloud NetApp Volumes, consulte: ["Configure o acesso aos Google Cloud NetApp Volumes."](#) .

Opções e exemplos de configuração do backend do Google Cloud NetApp Volumes

Saiba mais sobre as opções de configuração de back-end para o Google Cloud NetApp Volumes e veja exemplos de configuração.

Opções de configuração do backend

Cada backend provisiona volumes em uma única região do Google Cloud. Para criar volumes em outras regiões, você pode definir backends adicionais.

Parâmetro	Descrição	Padrão
version		Sempre 1
storageDriverName	Nome do driver de armazenamento	O valor de storageDriverName deve ser especificado como "google-cloud-netapp-volumes".

Parâmetro	Descrição	Padrão
backendName	(Opcional) Nome personalizado do backend de armazenamento	Nome do driver + "_" + parte da chave da API
storagePools	Parâmetro opcional usado para especificar os conjuntos de armazenamento para a criação de volumes.	
projectNumber	Número do projeto da conta do Google Cloud. O valor pode ser encontrado na página inicial do portal do Google Cloud.	
location	Localização no Google Cloud onde o Trident cria volumes GCNV. Ao criar clusters Kubernetes entre regiões, os volumes criados em uma região são mantidos. location Pode ser usado em cargas de trabalho agendadas em nós em várias regiões do Google Cloud. O tráfego entre regiões diferentes acarreta um custo adicional.	
apiKey	Chave de API para a conta de serviço do Google Cloud com o netapp.admin papel. Inclui o conteúdo formatado em JSON do arquivo de chave privada de uma conta de serviço do Google Cloud (copiado integralmente para o arquivo de configuração do backend). O apiKey Deve incluir pares de chave-valor para as seguintes chaves: type , project_id , client_email , client_id , auth_uri , token_uri , auth_provider_x509_cert_url , e client_x509_cert_url .	
nfsMountOptions	Controle preciso das opções de montagem NFS.	"nfsvers=3"
limitVolumeSize	O provisionamento falhará se o tamanho do volume solicitado for superior a esse valor.	"" (não aplicado por padrão)
serviceLevel	O nível de serviço de um pool de armazenamento e seus volumes. Os valores são flex , standard , premium , ou extreme .	
labels	Conjunto de rótulos arbitrários formatados em JSON para aplicar aos volumes	""
network	A rede Google Cloud é usada para os volumes GCNV.	
debugTraceFlags	Sinalizadores de depuração a serem usados na resolução de problemas. Exemplo, {"api":false, "method":true} . Não utilize esta opção a menos que esteja solucionando problemas e precise de um despejo de logs detalhado.	nulo
nasType	Configure a criação de volumes NFS ou SMB. As opções são nfs , smb ou nulo. Definir como nulo utiliza, por padrão, volumes NFS.	nfs

Parâmetro	Descrição	Padrão
supportedTopologies	Representa uma lista de regiões e zonas que são suportadas por este sistema. Para obter mais informações, consulte "Utilizar a topologia CSI" . Por exemplo: supportedTopologies: - topology.kubernetes.io/region: asia-east1 topology.kubernetes.io/zone: asia-east1-a	

Opções de provisionamento de volume

Você pode controlar o provisionamento de volume padrão em `defaults` seção do arquivo de configuração.

Parâmetro	Descrição	Padrão
exportRule	Regras de exportação para novos volumes. Deve ser uma lista de endereços IPv4, separados por vírgulas, contendo qualquer combinação de endereços IPv4.	"0.0.0.0/0"
snapshotDir	Acesso ao <code>.snapshot</code> diretório	"verdadeiro" para NFSv4 "falso" para NFSv3
snapshotReserve	Percentagem do volume reservada para instantâneos	"" (aceitar o valor padrão de 0)
unixPermissions	Permissões Unix de novos volumes (4 dígitos octais).	""

Configurações de exemplo

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

Configuração mínima

Esta é a configuração mínima absoluta do backend. Com essa configuração, o Trident descobre todos os seus pools de armazenamento delegados ao Google Cloud NetApp Volumes no local configurado e coloca novos volumes em um desses pools aleatoriamente. Porque `nasType` é omitido, o `nfs` A configuração padrão será aplicada e o servidor provisionará volumes NFS.

Essa configuração é ideal para quem está começando a usar o Google Cloud NetApp Volumes e a experimentar, mas na prática, provavelmente será necessário definir um escopo adicional para os volumes provisionados.

```

---
apiVersion: v1
kind: Secret
metadata:
  name: backend-tbc-gcnv-secret
type: Opaque
stringData:
  private_key_id: f2cb6ed6d7cc10c453f7d3406fc700c5df0ab9ec
  private_key: |
    -----BEGIN PRIVATE KEY-----\n
    znHczZsrtrHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m\n
    znHczZsrtrHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m\n
    znHczZsrtrHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m\n
    XsYg6gyxy4zq7OlwWgLwGa==\n
    -----END PRIVATE KEY-----\n

---
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-gcnv
spec:
  version: 1
  storageDriverName: google-cloud-netapp-volumes
  projectNumber: "123455380079"
  location: europe-west6
  serviceLevel: premium
  apiKey:
    type: service_account
    project_id: my-gcnv-project
    client_email: myproject-prod@my-gcnv-
project.iam.gserviceaccount.com
    client_id: "103346282737811234567"
    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/myproject-prod%40my-
gcnv-project.iam.gserviceaccount.com
  credentials:
    name: backend-tbc-gcnv-secret

```

Configuração para volumes SMB

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-gcnv1
  namespace: trident
spec:
  version: 1
  storageDriverName: google-cloud-netapp-volumes
  projectNumber: "123456789"
  location: asia-east1
  serviceLevel: flex
  nasType: smb
  apiKey:
    type: service_account
    project_id: cloud-native-data
    client_email: trident-sample@cloud-native-
data.iam.gserviceaccount.com
    client_id: "123456789737813416734"
    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/trident-
sample%40cloud-native-data.iam.gserviceaccount.com
  credentials:
    name: backend-tbc-gcnv-secret
```




```

---
apiVersion: v1
kind: Secret
metadata:
  name: backend-tbc-gcnv-secret
type: Opaque
stringData:
  private_key_id: f2cb6ed6d7cc10c453f7d3406fc700c5df0ab9ec
  private_key: |
    -----BEGIN PRIVATE KEY-----
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m
    XsYg6gyxy4zq7OlwWgLwGa==
    -----END PRIVATE KEY-----

---
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-gcnv
spec:
  version: 1
  storageDriverName: google-cloud-netapp-volumes
  projectNumber: "123455380079"
  location: europe-west6
  serviceLevel: premium
  storagePools:
    - premium-pool1-europe-west6
    - premium-pool2-europe-west6
  apiKey:
    type: service_account
    project_id: my-gcnv-project
    client_email: myproject-prod@my-gcnv-
project.iam.gserviceaccount.com
    client_id: "103346282737811234567"
    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/myproject-prod%40my-
gcnv-project.iam.gserviceaccount.com
  credentials:
    name: backend-tbc-gcnv-secret

```

Configuração de pool virtual

Essa configuração de backend define vários pools virtuais em um único arquivo. Os pools virtuais são definidos em `storage` seção. São úteis quando você tem vários pools de armazenamento que suportam diferentes níveis de serviço e deseja criar classes de armazenamento no Kubernetes que os representem. Etiquetas de piscinas virtuais são usadas para diferenciar as piscinas. Por exemplo, no exemplo abaixo, `performance` rótulo e `serviceLevel` O tipo é usado para diferenciar pools virtuais.

Você também pode definir alguns valores padrão que serão aplicáveis a todos os pools virtuais e sobrescrever os valores padrão para pools virtuais individuais. No exemplo a seguir, `snapshotReserve` e `exportRule` servem como valores padrão para todos os pools virtuais.

Para obter mais informações, consulte ["Piscinas virtuais"](#) .

```
---
apiVersion: v1
kind: Secret
metadata:
  name: backend-tbc-gcnv-secret
type: Opaque
stringData:
  private_key_id: f2cb6ed6d7cc10c453f7d3406fc700c5df0ab9ec
  private_key: |
    -----BEGIN PRIVATE KEY-----
    znHczZsrtrHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrtrHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrtrHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrtrHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m
    XsYg6gyxy4zq7OlwWgLwGa==
    -----END PRIVATE KEY-----

---
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-gcnv
spec:
  version: 1
  storageDriverName: google-cloud-netapp-volumes
  projectNumber: "123455380079"
  location: europe-west6
  apiKey:
    type: service_account
    project_id: my-gcnv-project
    client_email: myproject-prod@my-gcnv-
project.iam.gserviceaccount.com
    client_id: "103346282737811234567"
```

```

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/myproject-prod%40my-
gcnv-project.iam.gserviceaccount.com
credentials:
  name: backend-tbc-gcnv-secret
defaults:
  snapshotReserve: "10"
  exportRule: 10.0.0.0/24
storage:
- labels:
  performance: extreme
  serviceLevel: extreme
  defaults:
    snapshotReserve: "5"
    exportRule: 0.0.0.0/0
- labels:
  performance: premium
  serviceLevel: premium
- labels:
  performance: standard
  serviceLevel: standard

```

Identidade na nuvem para GKE

```

apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-gcp-gcnv
spec:
  version: 1
  storageDriverName: google-cloud-netapp-volumes
  projectNumber: '012345678901'
  network: gcnv-network
  location: us-west2
  serviceLevel: Premium
  storagePool: pool-premium1

```

Configuração de topologias suportadas

O Trident facilita o provisionamento de volumes para cargas de trabalho com base em regiões e zonas de disponibilidade. O `supportedTopologies` O bloco nesta configuração de backend é usado para fornecer uma lista de regiões e zonas por backend. Os valores de região e zona especificados aqui devem corresponder aos valores de região e zona dos rótulos em cada nó do cluster Kubernetes. Essas regiões e zonas representam a lista de valores permitidos que podem ser fornecidos em uma classe de armazenamento. Para classes de armazenamento que contêm um subconjunto das regiões e zonas fornecidas em um backend, o Trident cria volumes na região e zona mencionadas. Para obter mais informações, consulte ["Utilizar a topologia CSI"](#).

```
---
version: 1
storageDriverName: google-cloud-netapp-volumes
subscriptionID: 9f87c765-4774-fake-ae98-a721add45451
tenantID: 68e4f836-edc1-fake-bff9-b2d865ee56cf
clientID: dd043f63-bf8e-fake-8076-8de91e5713aa
clientSecret: SECRET
location: asia-east1
serviceLevel: flex
supportedTopologies:
  - topology.kubernetes.io/region: asia-east1
    topology.kubernetes.io/zone: asia-east1-a
  - topology.kubernetes.io/region: asia-east1
    topology.kubernetes.io/zone: asia-east1-b
```

O que vem a seguir?

Após criar o arquivo de configuração do backend, execute o seguinte comando:

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

Para verificar se o backend foi criado com sucesso, execute o seguinte comando:

```
kubectl get tridentbackendconfig
```

NAME	BACKEND NAME	BACKEND UUID
backend-tbc-gcnv	backend-tbc-gcnv	b2fd1ff9-b234-477e-88fd-713913294f65
Bound	Success	

Se a criação do backend falhar, há algo errado com a configuração do backend. Você pode descrever o backend usando o `kubectl get tridentbackendconfig <backend-name>` Execute o comando ou visualize os registros para determinar a causa, executando o seguinte comando:

```
tridentctl logs
```

Após identificar e corrigir o problema com o arquivo de configuração, você pode excluir o backend e executar o comando de criação novamente.

Definições de classe de armazenamento

A seguir, apresentamos um básico. `StorageClass` definição que se refere ao backend acima.

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: gcnv-nfs-sc
provisioner: csi.trident.netapp.io
parameters:
  backendType: "google-cloud-netapp-volumes"
```

Exemplos de definições usando o `parameter.selector` campo:

Usando `parameter.selector` Você pode especificar para cada um `StorageClass` o "piscina virtual" que é usado para hospedar um volume. O volume terá os aspectos definidos na piscina escolhida.

```

apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: extreme-sc
provisioner: csi.trident.netapp.io
parameters:
  selector: performance=extreme
  backendType: google-cloud-netapp-volumes

---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: premium-sc
provisioner: csi.trident.netapp.io
parameters:
  selector: performance=premium
  backendType: google-cloud-netapp-volumes

---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: standard-sc
provisioner: csi.trident.netapp.io
parameters:
  selector: performance=standard
  backendType: google-cloud-netapp-volumes

```

Para obter mais detalhes sobre classes de armazenamento, consulte ["Criar uma classe de armazenamento"](#) .

Exemplos de definições para volumes SMB

Usando `nasType` , `node-stage-secret-name` , e `node-stage-secret-namespace` Você pode especificar um volume SMB e fornecer as credenciais necessárias do Active Directory. Qualquer usuário/senha do Active Directory, com quaisquer permissões ou sem permissões, pode ser usado como segredo de estágio do nó.

Configuração básica no namespace padrão

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

Utilizando segredos diferentes por namespace

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

Utilizando segredos diferentes em cada volume.

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: gcnv-sc-smb
provisioner: csi.trident.netapp.io
parameters:
  backendType: "google-cloud-netapp-volumes"
  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`Filtros para pools que suportam volumes SMB. `nasType: nfs ou nasType: null Filtros para pools NFS.

Exemplo de definição de PVC

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: gcnv-nfs-pvc
spec:
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 100Gi
  storageClassName: gcnv-nfs-sc
```

Para verificar se o PVC está vinculado, execute o seguinte comando:

```
kubectl get pvc gcnv-nfs-pvc
```

NAME	STATUS	VOLUME	CAPACITY
ACCESS MODES	STORAGECLASS	AGE	
gcnv-nfs-pvc	Bound	pvc-b00f2414-e229-40e6-9b16-ee03eb79a213	100Gi
RWX	gcnv-nfs-sc	1m	

Configure um Cloud Volumes Service para o backend do Google Cloud.

Aprenda como configurar o NetApp Cloud Volumes Service para Google Cloud como backend para sua instalação do Trident usando as configurações de exemplo fornecidas.

Detalhes do driver do Google Cloud

A Trident fornece o `gcp-cvs` O driver deve se comunicar com o cluster. Os modos de acesso suportados são: *ReadWriteOnce* (RWO), *ReadOnlyMany* (ROX), *ReadWriteMany* (RWX), *ReadWriteOncePod* (RWOP).

Motorista	Protocolo	modo de volume	Modos de acesso suportados	Sistemas de arquivos suportados
gcp-cvs	NFS	Sistema de arquivos	RWO, ROX, RWX, RWOP	nfs

Saiba mais sobre o suporte do Trident para o Cloud Volumes Service do Google Cloud.

O Trident pode criar volumes do Cloud Volumes Service de duas maneiras diferentes. ["tipos de serviço"](#) :

- **CVS-Performance:** O tipo de serviço Trident padrão. Este tipo de serviço otimizado para desempenho é mais adequado para cargas de trabalho de produção que valorizam o desempenho. O tipo de serviço CVS-Performance é uma opção de hardware que suporta volumes com tamanho mínimo de 100 GiB. Você pode escolher um dos seguintes: "três níveis de serviço" :

- standard
- premium
- extreme

- **CVS:** O tipo de serviço CVS oferece alta disponibilidade zonal com níveis de desempenho limitados a moderados. O tipo de serviço CVS é uma opção de software que utiliza pools de armazenamento para suportar volumes tão pequenos quanto 1 GiB. O pool de armazenamento pode conter até 50 volumes, onde todos os volumes compartilham a capacidade e o desempenho do pool. Você pode escolher um dos seguintes: "dois níveis de serviço" :

- standardsw
- zoneredundantstandardsw

O que você vai precisar

Para configurar e usar o "Cloud Volumes Service para Google Cloud" Para o backend, você precisa do seguinte:

- Uma conta do Google Cloud configurada com o serviço NetApp Cloud Volumes Service.
- Número do projeto da sua conta do Google Cloud
- conta de serviço do Google Cloud com o `netappcloudvolumes.admin` papel
- Arquivo de chave de API para sua conta do Cloud Volumes Service .

Opções de configuração do backend

Cada backend provisiona volumes em uma única região do Google Cloud. Para criar volumes em outras regiões, você pode definir backends adicionais.

Parâmetro	Descrição	Padrão
<code>version</code>		Sempre 1
<code>storageDriverName</code>	Nome do driver de armazenamento	"gcp-cvs"
<code>backendName</code>	Nome personalizado ou o backend de armazenamento	Nome do driver + "_" + parte da chave da API
<code>storageClass</code>	Parâmetro opcional usado para especificar o tipo de serviço CVS. Usar <code>software</code> Para selecionar o tipo de serviço CVS. Caso contrário, o Trident assume o tipo de serviço CVS-Performance.(<code>hardware</code>).	
<code>storagePools</code>	Somente o tipo de serviço CVS. Parâmetro opcional usado para especificar os conjuntos de armazenamento para a criação de volumes.	
<code>projectNumber</code>	Número do projeto da conta do Google Cloud. O valor pode ser encontrado na página inicial do portal do Google Cloud.	

Parâmetro	Descrição	Padrão
hostProjectNumber	Obrigatório se estiver usando uma rede VPC compartilhada. Neste cenário, <code>projectNumber</code> é o projeto de serviço, e <code>hostProjectNumber</code> é o projeto anfitrião.	
apiRegion	A região do Google Cloud onde o Trident cria volumes do Cloud Volumes Service . Ao criar clusters Kubernetes entre regiões, os volumes criados em uma região são mantidos. <code>apiRegion</code> Pode ser usado em cargas de trabalho agendadas em nós em várias regiões do Google Cloud. O tráfego entre regiões diferentes acarreta um custo adicional.	
apiKey	Chave de API para a conta de serviço do Google Cloud com o <code>netappcloudvolumes.admin</code> papel. Inclui o conteúdo formatado em JSON do arquivo de chave privada de uma conta de serviço do Google Cloud (copiado integralmente para o arquivo de configuração do backend).	
proxyURL	URL do proxy, caso seja necessário um servidor proxy para conectar-se à conta CVS. O servidor proxy pode ser um proxy HTTP ou um proxy HTTPS. Para um proxy HTTPS, a validação do certificado é ignorada para permitir o uso de certificados autoassinados no servidor proxy. Servidores proxy com autenticação ativada não são suportados.	
nfsMountOptions	Controle preciso das opções de montagem NFS.	"nfsvers=3"
limitVolumeSize	O provisionamento falhará se o tamanho do volume solicitado for superior a esse valor.	"" (não aplicado por padrão)
serviceLevel	O nível de desempenho ou serviço CVS para novos volumes. Os valores de desempenho do CVS são <code>standard</code> , <code>premium</code> , ou <code>extreme</code> . Os valores CVS são <code>standardsw</code> ou <code>zoneredundantstandardsw</code> .	A configuração padrão do CVS-Performance é "standard". O padrão do CVS é "standardsw".
network	A rede do Google Cloud é usada para os volumes do Cloud Volumes Service .	"padrão"
debugTraceFlags	Sinalizadores de depuração a serem usados na resolução de problemas. Exemplo, <code>\{"api":false,"method":true\}</code> . Não utilize esta opção a menos que esteja solucionando problemas e precise de um despejo de logs detalhado.	nulo
allowedTopologies	Para habilitar o acesso entre regiões, sua definição de <code>StorageClass</code> para <code>allowedTopologies</code> Deve incluir todas as regiões. Por exemplo: <ul style="list-style-type: none"> - <code>key: topology.kubernetes.io/region</code> <code>values:</code> - <code>us-east1</code> - <code>europa-west1</code> 	

Opções de provisionamento de volume

Você pode controlar o provisionamento de volume padrão em `defaults` seção do arquivo de configuração.

Parâmetro	Descrição	Padrão
<code>exportRule</code>	Regras de exportação para novos volumes. Deve ser uma lista separada por vírgulas de qualquer combinação de endereços IPv4 ou sub-redes IPv4 na notação CIDR.	"0.0.0.0/0"
<code>snapshotDir</code>	Acesso ao <code>.snapshot</code> diretório	"falso"
<code>snapshotReserve</code>	Porcentagem do volume reservada para instantâneos	"" (aceitar o valor padrão CVS de 0)
<code>size</code>	O tamanho dos novos volumes. O requisito mínimo de desempenho do CVS é de 100 GiB. O tamanho mínimo exigido pelo CVS é 1 GiB.	O tipo de serviço CVS-Performance tem como padrão "100GiB". O tipo de serviço CVS não define um valor padrão, mas requer um mínimo de 1 GiB.

Exemplos de tipos de serviço CVS-Performance

Os exemplos a seguir fornecem configurações de amostra para o tipo de serviço CVS-Performance.

Exemplo 1: Configuração mínima

Esta é a configuração mínima de backend usando o tipo de serviço CVS-Performance padrão com o nível de serviço "standard" padrão.

```
---
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-----
    <key_value>
    -----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
```

Exemplo 2: Configuração do nível de serviço

Este exemplo ilustra as opções de configuração do backend, incluindo o nível de serviço e os valores padrão de volume.

```
---
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-----
    <key_value>
    -----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
```

Exemplo 3: Configuração de pool virtual

Este exemplo usa `storage` para configurar pools virtuais e o `StorageClasses` que se referem a eles. Consulte [Definições de classe de armazenamento](#) para ver como as classes de armazenamento foram definidas.

Aqui, são definidos valores padrão específicos para todos os pools virtuais, que definem o `snapshotReserve` a 5% e o `exportRule` para 0.0.0.0/0. Os pools virtuais são definidos em `storage` seção. Cada piscina virtual individual define a sua própria. `serviceLevel` E algumas pools sobrescrevem os valores padrão. Rótulos de piscinas virtuais foram usados para diferenciar as piscinas com base em `performance` e `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-----
    <key_value>
    -----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

```

Definições de classe de armazenamento

As seguintes definições de StorageClass aplicam-se ao exemplo de configuração de pool virtual. Usando `parameters.selector` Você pode especificar para cada StorageClass o pool virtual usado para hospedar um volume. O volume terá os aspectos definidos na piscina escolhida.

Exemplo de classe de armazenamento

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

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

- A primeira StorageClass(cvs-extreme-extra-protection) mapeia para a primeira piscina virtual. Esta é a única piscina que oferece desempenho extremo com uma reserva instantânea de 10%.
- A última StorageClass(cvs-extra-protection) menciona qualquer pool de armazenamento que forneça uma reserva de snapshot de 10%. O Trident decide qual pool virtual será selecionado e garante que o requisito de reserva de snapshots seja atendido.

Exemplos de tipos de serviço CVS

Os exemplos a seguir fornecem configurações de amostra para o tipo de serviço CVS.

Exemplo 1: Configuração mínima

Esta é a configuração mínima de backend usando `storageClass` para especificar o tipo de serviço CVS e o padrão `standardsw` nível de serviço.

```
---
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-----
    <key_value>
    -----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
```

Exemplo 2: Configuração do pool de armazenamento

Esta configuração de backend de exemplo usa `storagePools` Para configurar um pool de armazenamento.

```
---
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-----
    <key_value>
    -----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
```

O que vem a seguir?

Após criar o arquivo de configuração do backend, execute o seguinte comando:

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

Se a criação do backend falhar, há algo errado com a configuração do backend. Você pode visualizar os registros para determinar a causa executando o seguinte comando:

```
tridentctl logs
```

Após identificar e corrigir o problema com o arquivo de configuração, você poderá executar o comando de criação novamente.

Configure um backend NetApp HCI ou SolidFire.

Aprenda como criar e usar um backend Element com sua instalação do Trident .

Detalhes do driver do elemento

A Trident fornece o `solidfire-san` Driver de armazenamento para comunicação com o cluster. Os modos de acesso suportados são: *ReadWriteOnce* (RWO), *ReadOnlyMany* (ROX), *ReadWriteMany* (RWX), *ReadWriteOncePod* (RWOP).

O `solidfire-san` O driver de armazenamento suporta os modos de volume *arquivo* e *bloco*. Para o `Filesystem` Com o comando `volumeMode`, o Trident cria um volume e um sistema de arquivos. O tipo de sistema de arquivos é especificado pela `StorageClass`.

Motorista	Protocolo	Modo de volume	Modos de acesso suportados	Sistemas de arquivos suportados
solidfire-san	iSCSI	Bloquear	RWO, ROX, RWX, RWOP	Sem sistema de arquivos. Dispositivo de bloco bruto.
solidfire-san	iSCSI	Sistema de arquivos	RWO, RWOP	xfs, ext3 , ext4

Antes de começar

Você precisará do seguinte antes de criar um backend Element.

- Um sistema de armazenamento compatível que execute o software Element.
- Credenciais de administrador de cluster NetApp HCI/ SolidFire ou de usuário locatário que possa gerenciar volumes.
- Todos os seus nós de trabalho do Kubernetes devem ter as ferramentas iSCSI apropriadas instaladas. Consulte ["Informações sobre a preparação do nó de trabalho"](#) .

Opções de configuração do backend

Consulte a tabela a seguir para obter as opções de configuração do backend:

Parâmetro	Descrição	Padrão
version		Sempre 1
storageDriverName	Nome do driver de armazenamento	Sempre "solidfire-san"

Parâmetro	Descrição	Padrão
backendName	Nome personalizado ou o backend de armazenamento	"solidfire_" + endereço IP de armazenamento (iSCSI)
Endpoint	MVIP para o cluster SolidFire com credenciais de locatário	
SVIP	Endereço IP e porta de armazenamento (iSCSI)	
labels	Conjunto de rótulos arbitrários formatados em JSON para aplicar aos volumes.	""
TenantName	Nome do locatário a ser usado (criado se não for encontrado)	
InitiatorIFace	Restringir o tráfego iSCSI a uma interface de host específica	"padrão"
UseCHAP	Use CHAP para autenticar iSCSI. Trident usa CHAP.	verdadeiro
AccessGroups	Lista de IDs de grupos de acesso a serem usados	Encontra o ID de um grupo de acesso chamado "trident".
Types	Especificações de QoS	
limitVolumeSize	O provisionamento falhará se o tamanho do volume solicitado for superior a este valor.	"" (não aplicado por padrão)
debugTraceFlags	Sinalizadores de depuração a serem usados na resolução de problemas. Exemplo: {"api":false, "method":true}	nulo



Não use `debugTraceFlags` a menos que você esteja solucionando problemas e precise de um despejo de logs detalhado.

Exemplo 1: Configuração de backend para `solidfire-san` driver com três tipos de volume

Este exemplo mostra um arquivo de backend que utiliza autenticação CHAP e modela três tipos de volume com garantias de QoS específicas. Muito provavelmente, você definiria classes de armazenamento para consumir cada um deles usando o `IOPS` parâmetro de classe de armazenamento.

```

---
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

```

Exemplo 2: Configuração de backend e classe de armazenamento para solidfire-san motorista com piscinas virtuais

Este exemplo mostra o arquivo de definição de backend configurado com pools virtuais, juntamente com StorageClasses que fazem referência a eles.

O Trident copia os rótulos presentes em um pool de armazenamento para o LUN de armazenamento de backend durante o provisionamento. Para maior conveniência, os administradores de armazenamento podem definir rótulos por pool virtual e agrupar volumes por rótulo.

No arquivo de definição de backend de exemplo mostrado abaixo, valores padrão específicos são definidos para todos os pools de armazenamento, que definem o `type` em Silver. Os pools virtuais são definidos em `storage` seção. Neste exemplo, alguns dos pools de armazenamento definem seu próprio tipo, e alguns pools substituem os valores padrão definidos acima.

```

---
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

```

As definições de StorageClass a seguir referem-se aos pools virtuais acima. Usando o

`parameters.selector` No campo `StorageClass`, cada `StorageClass` especifica qual(is) pool(s) virtual(is) pode(m) ser usado(s) para hospedar um volume. O volume terá os aspectos definidos na piscina virtual escolhida.

A primeira `StorageClass(solidfire-gold-four)` será mapeado para o primeiro pool virtual. Esta é a única piscina que oferece desempenho de ouro com um `Volume Type QoS` de ouro. A última `StorageClass(solidfire-silver)` menciona qualquer pool de armazenamento que ofereça um desempenho prata. A Trident decidirá qual pool virtual será selecionado e garantirá que o requisito de armazenamento seja atendido.

```
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
```

Encontre mais informações

- ["Grupos de acesso a volume"](#)

Motoristas ONTAP SAN

Visão geral do driver ONTAP SAN

Aprenda a configurar um backend ONTAP com os drivers ONTAP SAN do Cloud Volumes ONTAP .

Detalhes do driver ONTAP SAN

A Trident fornece os seguintes drivers de armazenamento SAN para comunicação com o cluster ONTAP . Os modos de acesso suportados são: *ReadWriteOnce* (RWO), *ReadOnlyMany* (ROX), *ReadWriteMany* (RWX), *ReadWriteOncePod* (RWOP).

Motorista	Protocolo	modo de volume	Modos de acesso suportados	Sistemas de arquivos suportados
ontap-san	iSCSI SCSI sobre FC	Bloquear	RWO, ROX, RWX, RWOP	Sem sistema de arquivos; dispositivo de bloco bruto
ontap-san	iSCSI SCSI sobre FC	Sistema de arquivos	RWO, RWOP ROX e RWX não estão disponíveis no modo de volume do sistema de arquivos.	xfs, ext3 , ext4

Motorista	Protocolo	modo de volume	Modos de acesso suportados	Sistemas de arquivos suportados
ontap-san	NVMe/TCP Consulte Considerações adicionais para NVMe/TCP .	Bloquear	RWO, ROX, RWX, RWOP	Sem sistema de arquivos; dispositivo de bloco bruto
ontap-san	NVMe/TCP Consulte Considerações adicionais para NVMe/TCP .	Sistema de arquivos	RWO, RWOP ROX e RWX não estão disponíveis no modo de volume do sistema de arquivos.	xfs, ext3 , ext4
ontap-san-economy	iSCSI	Bloquear	RWO, ROX, RWX, RWOP	Sem sistema de arquivos; dispositivo de bloco bruto
ontap-san-economy	iSCSI	Sistema de arquivos	RWO, RWOP ROX e RWX não estão disponíveis no modo de volume do sistema de arquivos.	xfs, ext3 , ext4



- Usar `ontap-san-economy` somente se a contagem de uso de volume persistente prevista for maior que "[limites de volume ONTAP suportados](#)" .
- Usar `ontap-nas-economy` somente se a contagem de uso de volume persistente prevista for maior que "[limites de volume ONTAP suportados](#)" e o `ontap-san-economy` O driver não pode ser usado.
- Não use `ontap-nas-economy` Se você prevê a necessidade de proteção de dados, recuperação de desastres ou mobilidade.
- A NetApp não recomenda o uso do Flexvol autogrow em todos os drivers ONTAP , exceto no `ontap-san`. Como solução alternativa, o Trident suporta o uso de reserva de snapshots e dimensiona os volumes Flexvol de acordo.

Permissões do usuário

O Trident espera ser executado como administrador 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 implementações do Amazon FSx for NetApp ONTAP , o Trident espera ser executado como administrador do ONTAP ou do SVM, usando o cluster. `fsxadmin` usuário ou um `vsadmin` Usuário SVM, ou um usuário com um nome diferente que tenha a mesma função. O `fsxadmin` O usuário é um substituto limitado para o usuário administrador do cluster.



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

Embora seja possível criar uma função mais restritiva dentro do ONTAP que um driver Trident possa usar, não recomendamos isso. A maioria das novas versões do Trident chamará APIs adicionais que precisarão ser consideradas, tornando as atualizações difíceis e propensas a erros.

Considerações adicionais para NVMe/TCP

O Trident suporta o protocolo de memória não volátil expressa (NVMe) usando o `ontap-san` motorista incluindo:

- IPv6
- Instantâneos e clones de volumes NVMe
- Redimensionar um volume NVMe
- Importar um volume NVMe criado fora do Trident para que seu ciclo de vida possa ser gerenciado pelo Trident.
- Multicaminhamento nativo NVMe
- Encerramento correto ou incorreto dos nós K8s (24.06)

O Trident não suporta:

- DH-HMAC-CHAP que é suportado nativamente por NVMe
- Mapeamento de dispositivos (DM) com múltiplos caminhos
- Criptografia LUKS



O NVMe é compatível apenas com APIs REST ONTAP e não com ONTAPI (ZAPI).

Prepare-se para configurar o backend com os drivers ONTAP SAN.

Compreenda os requisitos e as opções de autenticação para configurar um backend ONTAP com drivers ONTAP SAN.

Requisitos

Para todos os backends ONTAP , o Trident exige que pelo menos um agregado seja atribuído à SVM.



"Sistemas ASA r2" diferem de outros sistemas ONTAP (ASA, AFF e FAS) na implementação de sua camada de armazenamento. Nos sistemas ASA r2, as zonas de disponibilidade de armazenamento são usadas em vez de agregados. Consulte ["esse"](#) Artigo da Base de Conhecimento sobre como atribuir agregados a SVMs em sistemas ASA r2.

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

Todos os seus nós de trabalho do Kubernetes devem ter as ferramentas iSCSI apropriadas instaladas.

Consulte "[Prepare o nó de trabalho](#)" para mais detalhes.

Autenticar o backend ONTAP

O Trident oferece dois modos de autenticação de um backend ONTAP .

- Com base em credenciais: o nome de usuário e a senha de um usuário do ONTAP com as permissões necessárias. Recomenda-se o uso de uma função de login de segurança predefinida, como `admin` ou `vsadmin` Para garantir a máxima compatibilidade com as versões do ONTAP .
- Com base em certificado: o Trident também pode se comunicar com um cluster ONTAP usando um certificado instalado no backend. Aqui, a definição do backend deve conter os valores codificados em Base64 do certificado do cliente, da chave e do certificado da CA confiável, se utilizado (recomendado).

Você pode atualizar os sistemas de backend existentes para alternar entre métodos baseados em credenciais e métodos baseados em certificados. No entanto, apenas um método de autenticação é suportado por vez. Para mudar para um método de autenticação diferente, você deve remover o método existente da configuração do backend.



Se você tentar fornecer **tanto credenciais quanto certificados**, a criação do backend falhará com um erro informando que mais de um método de autenticação foi fornecido no arquivo de configuração.

Ativar autenticação baseada em credenciais

O Trident requer as credenciais de um administrador com escopo de SVM/cluster para se comunicar com o backend do ONTAP . Recomenda-se o uso de funções padrão predefinidas, como: `admin` ou `vsadmin` . Isso garante a compatibilidade futura com versões futuras do ONTAP que possam expor APIs de recursos a serem usadas por versões futuras do Trident . É possível criar e usar uma função de login de segurança personalizada com o Trident, mas isso não é recomendado.

Uma definição de backend de exemplo terá a seguinte aparência:

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"
}
```

Lembre-se de que a definição do backend é o único lugar onde as credenciais são armazenadas em texto simples. Após a criação do backend, os nomes de usuário/senhas são codificados em Base64 e armazenados como segredos do Kubernetes. A criação ou atualização de um backend é a única etapa que exige conhecimento das credenciais. Sendo assim, trata-se de uma operação exclusiva para administradores, a ser realizada pelo administrador do Kubernetes/armazenamento.

Habilitar autenticação baseada em certificado

Novos e existentes sistemas de backend podem usar um certificado e se comunicar com o backend ONTAP . São necessários três parâmetros na definição do backend.

- `clientCertificate`: Valor do certificado do cliente codificado em Base64.
- `clientPrivateKey`: Valor codificado em Base64 da chave privada associada.
- `trustedCACertificate`: Valor codificado em Base64 do certificado da Autoridade Certificadora (CA) confiável. Caso esteja utilizando uma Autoridade Certificadora (CA) confiável, este parâmetro deve ser fornecido. Isso pode ser ignorado se nenhuma Autoridade Certificadora (CA) confiável for utilizada.

Um fluxo de trabalho típico envolve as seguintes etapas.

Passos

1. Gere um certificado e uma chave de cliente. Ao gerar o código, defina o Nome Comum (CN) para o usuário ONTAP que será usado para autenticação.

```
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. Adicione um certificado CA confiável ao cluster ONTAP . Isso pode já estar sendo tratado pelo administrador de armazenamento. Ignore se nenhuma Autoridade Certificadora (CA) confiável for utilizada.

```
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 (do passo 1) no cluster 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 é compatível. cert 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 o certificado gerado. Substitua < ONTAP Management LIF> e <vserver name> pelo endereço IP do Management LIF e pelo nome do 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. Codifique o certificado, a chave e o certificado da CA confiável em Base64.

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

7. Crie o 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 |
+-----+-----+-----+-----+
+-----+-----+
```

Atualize os métodos de autenticação ou altere as credenciais.

Você pode atualizar um backend existente para usar um método de autenticação diferente ou para rotacionar suas credenciais. Isso funciona nos dois sentidos: os sistemas internos que utilizam nome de usuário/senha podem ser atualizados para usar certificados; os sistemas internos que utilizam certificados podem ser atualizados para usar nome de usuário/senha. Para fazer isso, você deve remover o método de autenticação existente e adicionar o novo método de autenticação. Em seguida, utilize o arquivo backend.json atualizado, que contém os parâmetros necessários, para executar o comando `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 rotacionar senhas, o administrador de armazenamento deve primeiro atualizar a senha do usuário no ONTAP. Em seguida, é realizada uma atualização do sistema interno. Ao rotacionar certificados, vários certificados podem ser adicionados ao usuário. Em seguida, o sistema de backend é atualizado para usar o novo certificado, após o que o certificado antigo pode ser excluído do cluster ONTAP .

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

Criar função ONTAP personalizada para Trident

Você pode criar uma função de cluster ONTAP com privilégios mínimos para que não precise usar a função de administrador do ONTAP para executar operações no Trident. Ao incluir o nome de usuário em uma configuração de backend do Trident , o Trident usa a função de cluster ONTAP que você criou para executar as operações.

Consulte "[Gerador de funções personalizadas Trident](#)" Para obter mais informações sobre como criar funções personalizadas do Trident .

Utilizando a CLI do ONTAP

1. Crie uma nova função usando o seguinte comando:

```
security login role create <role_name\> -cmddirname "command" -access all  
-vserver <svm_name\>
```

2. Crie um nome de usuário para o usuário do Trident :

```
security login create -username <user_name\> -application ontapi  
-authmethod <password\> -role <name_of_role_in_step_1\> -vserver  
<svm_name\> -comment "user_description"
```

3. Atribua a função ao usuário:

```
security login modify username <user_name\> -vserver <svm_name\> -role  
<role_name\> -application ontapi -application console -authmethod  
<password\>
```

Utilizando o Gerenciador de Sistemas

Execute as seguintes etapas no ONTAP System Manager:

1. **Criar uma função personalizada:**

- a. Para criar uma função personalizada no nível do cluster, selecione **Cluster > Configurações**.

(Ou) Para criar uma função personalizada no nível da SVM, selecione **Armazenamento > VMs de armazenamento > required svm > Configurações > Usuários e funções**.

- b. Selecione o ícone de seta (→) ao lado de **Usuários e Funções**.

- c. Selecione **+Adicionar** em **Funções**.

- d. Defina as regras para a função e clique em **Salvar**.

2. **Atribua a função ao usuário do Trident *: + Execute as seguintes etapas na página *Usuários e Funções:**

- a. Selecione o ícone Adicionar * em **Usuários**.

- b. Selecione o nome de usuário desejado e, em seguida, selecione uma função no menu suspenso **Função**.

- c. Clique em **Salvar**.

Consulte as páginas seguintes para obter mais informações:

- ["Funções personalizadas para administração do ONTAP"](#) ou ["Defina funções personalizadas"](#)
- ["Trabalhar com funções e usuários"](#)

Autenticar conexões com CHAP bidirecional

O Trident pode autenticar sessões iSCSI com CHAP bidirecional para o `ontap-san` e `ontap-san-economy` motoristas. Isso requer a ativação do `useCHAP` opção na sua definição de backend. Quando definido para `true` O Trident configura a segurança do iniciador padrão da SVM para CHAP bidirecional e define o nome de usuário e os segredos a partir do arquivo de backend. A NetApp recomenda o uso do protocolo 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` O parâmetro é uma opção booleana que pode ser configurada apenas uma vez. Por padrão, está definido como falso. Depois de definir como verdadeiro, você não poderá definir como falso.

Além de `useCHAP=true`, o `chapInitiatorSecret`, `chapTargetInitiatorSecret`, `chapTargetUsername`, e `chapUsername` Os campos devem ser incluídos na definição do backend. Os segredos podem ser alterados após a criação de um backend executando o seguinte comando: `tridentctl update`.

Como funciona

Ao configurar `useCHAP` Para confirmar, o administrador de armazenamento instrui o Trident a configurar o CHAP no backend de armazenamento. Isso inclui o seguinte:

- Configurando o CHAP no SVM:
 - Se o tipo de segurança do iniciador padrão da SVM for "nenhum" (definido por padrão) e não houver LUNs preexistentes no volume, o Trident definirá o tipo de segurança padrão como CHAP e prosseguirá para a configuração do iniciador CHAP e do nome de usuário e segredos de destino.
 - Se a SVM contiver LUNs, o Trident não habilitará o CHAP na SVM. Isso garante que o acesso aos LUNs já presentes na SVM não seja restringido.
- Configurar o nome de usuário e os segredos do iniciador e do alvo CHAP; essas opções devem ser especificadas na configuração do backend (como mostrado acima).

Após a criação do backend, o Trident cria um correspondente. `tridentbackend` O CRD armazena os segredos CHAP e os nomes de usuário como segredos do Kubernetes. Todos os PVs criados pelo Trident neste backend serão montados e conectados via CHAP.

Gire as credenciais e atualize os backends

Você pode atualizar as credenciais CHAP atualizando os parâmetros CHAP em `backend.json` arquivo. Isso exigirá a atualização dos segredos CHAP e o uso do `tridentctl update` comando para refletir essas mudanças.



Ao atualizar os segredos CHAP de um backend, você deve usar `tridentctl` para atualizar o backend. Não atualize as credenciais no cluster de armazenamento usando a CLI do ONTAP ou o ONTAP System Manager, pois o Trident não conseguirá detectar 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 permanecerão inalteradas; elas continuarão ativas se as credenciais forem atualizadas pelo Trident no SVM. Novas conexões utilizam as credenciais atualizadas e as conexões existentes permanecem ativas. Desconectar e reconectar os sistemas fotovoltaicos antigos fará com que eles passem a usar as credenciais atualizadas.

Opções e exemplos de configuração do ONTAP SAN

Aprenda como criar e usar drivers ONTAP SAN com sua instalação do Trident . Esta seção fornece exemplos de configuração de backend e detalhes para mapear backends para StorageClasses.

["Sistemas ASA r2"](#) Diferem de outros sistemas ONTAP (ASA, AFF e FAS) na implementação de sua camada de armazenamento. Essas variações afetam o uso de certos parâmetros, conforme indicado. ["Saiba mais sobre as diferenças entre os sistemas ASA r2 e outros sistemas ONTAP."](#)




Somente o `ontap-san` O driver (com protocolos iSCSI e NVMe/TCP) é compatível com sistemas ASA r2.


Na configuração do backend Trident , não é necessário especificar que seu sistema é um ASA r2. Ao selecionar `ontap-san` como o `storageDriverName` O Trident detecta automaticamente o ASA r2 ou o sistema ONTAP tradicional. Alguns parâmetros de configuração de backend não se aplicam aos sistemas ASA r2, conforme indicado na tabela abaixo.


Opções de configuração do backend

Consulte a tabela a seguir para obter as opções de configuração do backend:

Parâmetro	Descrição	Padrão
<code>version</code>		Sempre 1
<code>storageDriverName</code>	Nome do driver de armazenamento	<code>ontap-san</code> ou <code>ontap-san-economy</code>
<code>backendName</code>	Nome personalizado ou o backend de armazenamento	Nome do motorista + "_" + <code>dataLIF</code>
<code>managementLIF</code>	<p>Endereço IP de um cluster ou LIF de gerenciamento de SVM.</p> <p>É possível especificar um nome de domínio totalmente qualificado (FQDN).</p> <p>Pode ser configurado para usar endereços IPv6 se o Trident foi instalado usando a opção IPv6. Os endereços IPv6 devem ser definidos entre colchetes, como por exemplo:</p> <pre>[28e8:d9fb:a825:b7bf:69a8:d02f:9e7b:3555]</pre> <p>Para uma transição perfeita para o MetroCluster , consulte o Exemplo MetroCluster .</p> <div><p>Se você estiver usando as credenciais "vsadmin", <code>managementLIF</code> deve ser a do SVM; se estiver usando credenciais de "administrador", <code>managementLIF</code> deve ser o do cluster.</p></div>	"10.0.0.1", "[2001:1234:abcd::fefe]"

Parâmetro	Descrição	Padrão
dataLIF	Endereço IP do protocolo LIF. Pode ser configurado para usar endereços IPv6 se o Trident foi instalado usando a opção IPv6. Os endereços IPv6 devem ser definidos entre colchetes, como por exemplo: [28e8:d9fb:a825:b7bf:69a8:d02f:9e7b:3555] . Não especifique para iSCSI. Trident usa " Mapa LUN Seletivo ONTAP " para descobrir as LIFs iSCSI necessárias para estabelecer uma sessão de múltiplos caminhos. Um aviso é gerado se dataLIF está explicitamente definido. Omitir para Metrocluster. Veja o Exemplo MetroCluster .	Derivado pelo SVM
svm	Máquina virtual de armazenamento a ser usada Omitir para Metrocluster. Veja o Exemplo MetroCluster .	Derivado de uma SVM managementLIF é especificado
useCHAP	Usar CHAP para autenticar iSCSI para drivers ONTAP SAN [Booleano]. Definir para true Para que o Trident configure e utilize o CHAP bidirecional como autenticação padrão para a SVM fornecida no backend. Consulte " Prepare-se para configurar o backend com os drivers ONTAP SAN. " para mais detalhes. Não compatível com FCP ou NVMe/TCP.	false
chapInitiatorSecret	Segredo do iniciador CHAP. Obrigatório se useCHAP=true	""
labels	Conjunto de rótulos arbitrários formatados em JSON para aplicar aos volumes	""
chapTargetInitiatorSecret	Segredo iniciador do alvo CHAP. Obrigatório se useCHAP=true	""
chapUsername	Nome de usuário de entrada. Obrigatório se useCHAP=true	""
chapTargetUsername	Nome de usuário alvo. Obrigatório se useCHAP=true	""
clientCertificate	Valor do certificado do cliente codificado em Base64. Utilizado para autenticação baseada em certificado.	""
clientPrivateKey	Valor da chave privada do cliente codificado em Base64. Utilizado para autenticação baseada em certificado.	""
trustedCACertificate	Valor codificado em Base64 do certificado da Autoridade Certificadora (CA) confiável. Opcional. Utilizado para autenticação baseada em certificado.	""
username	Nome de usuário necessário para se comunicar com o cluster ONTAP . Usado para autenticação baseada em credenciais. Para autenticação do Active Directory, consulte " Autenticar o Trident em um SVM de backend usando credenciais do Active Directory ".	""

Parâmetro	Descrição	Padrão
password	Senha necessária para se comunicar com o cluster ONTAP . Usado para autenticação baseada em credenciais. Para autenticação do Active Directory, consulte "Autenticar o Trident em um SVM de backend usando credenciais do Active Directory" .	""
svm	Máquina virtual de armazenamento para usar	Derivado de uma SVM managementLIF é especificado
storagePrefix	Prefixo usado ao provisionar novos volumes no SVM. Não pode ser modificado posteriormente. Para atualizar esse parâmetro, você precisará criar um novo backend.	trident
aggregate	<p>Agregado para provisionamento (opcional; se definido, deve ser atribuído à SVM). Para o <code>ontap-nas-flexgroup</code> motorista, esta opção é ignorada. Caso não esteja atribuído, qualquer um dos agregados disponíveis pode ser usado para provisionar um volume FlexGroup .</p> <div>  <p>Quando o agregado é atualizado no SVM, ele é atualizado automaticamente no Trident por meio de polling no SVM, sem a necessidade de reiniciar o Controlador Trident . Quando você configura um agregado específico no Trident para provisionar volumes, se o agregado for renomeado ou movido para fora do SVM, o backend entrará em estado de falha no Trident durante a consulta ao agregado do SVM. Você deve alterar o agregado para um que esteja presente na SVM ou removê-lo completamente para que o backend volte a ficar online.</p> </div> <p>Não especificar para sistemas ASA r2.</p>	""
limitAggregateUsage	O provisionamento falhará se a utilização for superior a esta percentagem. Se você estiver usando um backend Amazon FSx for NetApp ONTAP , não especifique. <code>limitAggregateUsage</code> . O fornecido <code>fsxadmin</code> e <code>vsadmin</code> Não possuem as permissões necessárias para recuperar o uso agregado e limitá-lo usando o Trident. Não especificar para sistemas ASA r2.	"" (não aplicado por padrão)
limitVolumeSize	O provisionamento falhará se o tamanho do volume solicitado for superior a este valor. Também restringe o tamanho máximo dos volumes que gerencia para LUNs.	"" (não aplicado por padrão)

Parâmetro	Descrição	Padrão
lunsPerFlexvol	Número máximo de LUNs por Flexvol, deve estar no intervalo [50, 200]	100
debugTraceFlags	Sinalizadores de depuração a serem usados na resolução de problemas. Exemplo: {"api":false, "method":true} Não utilize a menos que esteja solucionando problemas e precise de um despejo de log detalhado.	null
useREST	<p>Parâmetro booleano para usar APIs REST do ONTAP.</p> <div> <p><code>`useREST`</code> Quando definido para <code>`true`</code> O Trident usa APIs REST do ONTAP para se comunicar com o backend; quando configurado para <code>`false`</code> O Trident utiliza chamadas ONTAPI (ZAPI) para se comunicar com o backend. Este recurso requer o ONTAP 9.11.1 e posterior. Além disso, a função de login do ONTAP utilizada deve ter acesso ao <code>`ontapi`</code> aplicativo. Isso é satisfeito pelo predefinido <code>`vsadmin`</code> e <code>`cluster-admin`</code> papéis. A partir da versão Trident 24.06 e do ONTAP 9.15.1 ou posterior, <code>`useREST`</code> está definido para <code>`true`</code> por padrão; alterar <code>`useREST`</code> para <code>`false`</code> para usar chamadas ONTAPI (ZAPI).</p> <p><code>`useREST`</code> Está totalmente qualificado para NVMe/TCP.</p> <div>  <p>O NVMe é compatível apenas com APIs REST ONTAP e não com ONTAPI (ZAPI).</p> </div> <p>Se especificado, sempre defina como <code>true</code> para sistemas ASA r2.</p> </div>	<code>true`</code> para ONTAP 9.15.1 ou posterior, caso contrário <code>`false`</code> .
sanType	Use para selecionar <code>iscsi</code> para iSCSI, <code>nvme</code> para NVMe/TCP ou <code>fcp</code> para SCSI sobre Fibre Channel (FC).	<code>`iscsi`</code> se estiver em branco

Parâmetro	Descrição	Padrão
formatOptions	<p>Usar formatOptions para especificar argumentos de linha de comando para o mkfs comando, que será aplicado sempre que um volume for formatado. Isso permite formatar o volume de acordo com suas preferências. Certifique-se de especificar as opções de formatação semelhantes às opções do comando mkfs, excluindo o caminho do dispositivo. Exemplo: "-E nodiscard"</p> <p>Compatível com ontap-san e ontap-san-economy drivers com protocolo iSCSI. Além disso, é compatível com sistemas ASA r2 ao usar os protocolos iSCSI e NVMe/TCP.</p>	
limitVolumePoolSize	Tamanho máximo de FlexVol solicitável ao usar LUNs no backend ontap-san-economy.	"" (não aplicado por padrão)
denyNewVolumePools	Restringe ontap-san-economy backends da criação de novos volumes FlexVol para conter seus LUNs. Apenas os Flexvols preexistentes são usados para provisionar novos PVs.	

Recomendações para usar formatOptions

A Trident recomenda a seguinte opção para agilizar o processo de formatação:

-E nodiscard:

- Mantenha os blocos salvos e não tente descartá-los durante a criação do sistema de arquivos (o descarte inicial de blocos é útil em dispositivos de estado sólido e em armazenamento com provisionamento esparsa/dinâmico). Esta opção substitui a opção obsoleta "-K" e é aplicável a todos os sistemas de arquivos (xfs, ext3 e ext4).

Autenticar o Trident em um SVM de backend usando credenciais do Active Directory

Você pode configurar o Trident para autenticar em um SVM de backend usando credenciais do Active Directory (AD). Antes que uma conta do AD possa acessar o SVM, você deve configurar o acesso do controlador de domínio do AD ao cluster ou SVM. Para administração de cluster com uma conta do AD, você deve criar um túnel de domínio. Consulte ["Configurar o acesso do controlador de domínio do Active Directory no ONTAP"](#) para mais detalhes.

passos

1. Configurar as definições do Sistema de Nomes de Domínio (DNS) para um SVM de backend:

```
vserver services dns create -vserver <svm_name> -dns-servers
<dns_server_ip1>,<dns_server_ip2>
```

2. Execute o seguinte comando para criar uma conta de computador para o SVM no Active Directory:

```
vserver active-directory create -vserver DataSVM -account-name ADSERVER1
-domain demo.netapp.com
```

3. Use este comando para criar um usuário ou grupo do AD para gerenciar o cluster ou SVM

```
security login create -vserver <svm_name> -user-or-group-name
<ad_user_or_group> -application <application> -authentication-method domain
-role vsadmin
```

4. No arquivo de configuração do backend do Trident, defina o `username` e `password` parâmetros para o nome do usuário ou grupo do AD e senha, respectivamente.

Opções de configuração de backend para provisionamento de volumes

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

Parâmetro	Descrição	Padrão
<code>spaceAllocation</code>	Alocação de espaço para LUNs	"verdadeiro" Se especificado, defina como <code>true</code> para sistemas ASA r2.
<code>spaceReserve</code>	Modo de reserva de espaço: "nenhum" (fino) ou "volume" (grosso). Definir para <code>none</code> para sistemas ASA r2.	"nenhum"
<code>snapshotPolicy</code>	Política de instantâneo a ser utilizada. Definir para <code>none</code> para sistemas ASA r2.	"nenhum"
<code>qosPolicy</code>	Grupo de políticas de QoS a ser atribuído aos volumes criados. Escolha uma das opções <code>qosPolicy</code> ou <code>adaptiveQosPolicy</code> para cada pool de armazenamento/backend. A utilização de grupos de políticas de QoS com o Trident requer o ONTAP 9.8 ou posterior. Você deve usar um grupo de políticas de QoS não compartilhado e garantir que o grupo de políticas seja aplicado a cada componente individualmente. Um grupo de políticas de QoS compartilhado impõe o limite máximo para a taxa de transferência total de todas as cargas de trabalho.	""
<code>adaptiveQosPolicy</code>	Grupo de políticas de QoS adaptativas a serem atribuídas aos volumes criados. Escolha <code>qosPolicy</code> ou <code>adaptiveQosPolicy</code> por pool de armazenamento/backend.	""
<code>snapshotReserve</code>	Percentagem do volume reservada para instantâneos. Não especificar para sistemas ASA r2.	"0" se <code>snapshotPolicy</code> é "nenhum", caso contrário ""
<code>splitOnClone</code>	Separar um clone de seu progenitor no momento da criação.	"falso"
<code>encryption</code>	Ative a Criptografia de Volume NetApp (NVE) no novo volume; o padrão é <code>false</code> . Para usar esta opção, o NVE precisa estar licenciado e habilitado no cluster. Se o NAE estiver habilitado no backend, qualquer volume provisionado no Trident terá o NAE habilitado. Para mais informações, consulte: " Como o Trident funciona com NVE e NAE ".	"falso" Se especificado, defina como <code>true</code> para sistemas ASA r2.

Parâmetro	Descrição	Padrão
luksEncryption	Ative a criptografia LUKS. Consulte "Use o Linux Unified Key Setup (LUKS)" .	"" Definido para <code>false</code> para sistemas ASA r2.
tieringPolicy	Política de escalonamento para usar "nenhum" Não especificar para sistemas ASA r2.	
nameTemplate	Modelo para criar nomes de volume personalizados.	""

Exemplos de provisionamento em volume

Aqui está um exemplo com valores padrão 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 usando o `ontap-san` O driver Trident adiciona 10% a mais de capacidade ao FlexVol para acomodar os metadados do LUN. O LUN será provisionado com o tamanho exato que o usuário solicitar no PVC. O Trident adiciona 10% ao FlexVol (mostrado como tamanho disponível no ONTAP). Os usuários agora receberão a quantidade de capacidade utilizável que solicitaram. Essa alteração também impede que as LUNs se tornem somente leitura, a menos que o espaço disponível esteja totalmente utilizado. Isso não se aplica a `ontap-san-economy`.

Para back-ends que definem `snapshotReserve` O Trident calcula o tamanho dos volumes da seguinte forma:

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

O 1.1 representa os 10% extras que a Trident adiciona ao FlexVol para acomodar os metadados do LUN. Para `snapshotReserve = 5%`, e solicitação de PVC = 5 GiB, o tamanho total do volume é 5,79 GiB e o tamanho disponível é 5,5 GiB. O `volume show` O comando deve exibir resultados semelhantes a este exemplo:

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.

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

Exemplos de configuração mínima

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



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

Exemplo de SAN ONTAP

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

```
---
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 MetroCluster

Você pode configurar o backend para evitar a necessidade de atualizar manualmente a definição do backend após a troca de modo (switchover) e o retorno ao modo anterior (switchback). "[Replicação e recuperação de SVM](#)".

Para uma transição perfeita e um retorno perfeito, especifique a SVM usando `managementLIF` e omitir o `svm` parâmetros. Por exemplo:

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

Exemplo de economia ONTAP SAN

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

Exemplo de autenticação baseada em certificado

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

```
---
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 bidirecionais

Esses exemplos criam um backend com useCHAP definido para true .

Exemplo 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 economia ONTAP SAN CHAP

```
---
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 NVMe/TCP

Você precisa ter uma SVM configurada com NVMe no seu backend ONTAP . Esta é uma configuração básica de backend 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
```

Exemplo de SCSI sobre FC (FCP)

Você precisa ter uma SVM configurada com FC em seu backend ONTAP . Esta é uma configuração básica de backend para FC.

```
---  
version: 1  
backendName: fcp-backend  
storageDriverName: ontap-san  
managementLIF: 10.0.0.1  
svm: svm_fc  
username: vsadmin  
password: password  
sanType: fcp  
useREST: true
```


Exemplo de configuração de backend com nameTemplate

```
---
version: 1
storageDriverName: ontap-san
backendName: ontap-san-backend
managementLIF: <ip address>
svm: svm0
username: <admin>
password: <password>
defaults:
  nameTemplate:
    "{{.volume.Name}}_{{.labels.cluster}}_{{.volume.Namespace}}_{{.volume.RequestName}}"
labels:
  cluster: ClusterA
PVC: "{{.volume.Namespace}}_{{.volume.RequestName}}"
```

Exemplo de formatOptions para o driver ontap-san-economy

```
---
version: 1
storageDriverName: ontap-san-economy
managementLIF: ""
svm: svm1
username: ""
password: "!"
storagePrefix: whelk_
debugTraceFlags:
  method: true
  api: true
defaults:
  formatOptions: -E nodiscard
```

Exemplos de backends com pools virtuais

Nesses arquivos de definição de backend de exemplo, valores padrão específicos são definidos para todos os pools de armazenamento, como: `spaceReserve` em `nenhum`, `spaceAllocation` em `falso`, e `encryption` `falso`. Os pools virtuais são definidos na seção de armazenamento.

O Trident define os rótulos de provisionamento no campo "Comentários". Os comentários são definidos no FlexVol volume. O Trident copia todos os rótulos presentes em um pool virtual para o volume de armazenamento durante o provisionamento. Para maior conveniência, os administradores de armazenamento podem definir rótulos por pool virtual e agrupar volumes por rótulo.

Nesses exemplos, alguns dos pools de armazenamento definem seus próprios limites. `spaceReserve` , `spaceAllocation` , e `encryption` valores, 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 ONTAP SAN

```
---
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 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"
```

Mapear backends para StorageClasses

As seguintes definições de StorageClass referem-se a: [Exemplos de backends com pools virtuais](#) . Usando o `parameters.selector` No campo `StorageClass`, cada `StorageClass` especifica quais pools virtuais podem ser usados para hospedar um volume. O volume terá os aspectos definidos na piscina virtual escolhida.

- O `protection-gold` A `StorageClass` será mapeada para o primeiro pool virtual no `ontap-san` backend. Esta é a única piscina que oferece proteção de nível ouro.

```

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 A StorageClass será mapeada para o segundo e terceiro pool virtual em ontap-san backend. Essas são as únicas pools 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 A StorageClass será mapeada para o terceiro pool virtual em ontap-san-economy backend. Este é o único pool que oferece configuração de pool de armazenamento para aplicativos do 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 A StorageClass será mapeada para o segundo pool virtual em ontap-san backend. Este é o único pool que oferece proteção de nível prata e 20.000 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 A StorageClass será mapeada para o terceiro pool virtual em ontap-san backend e o quarto pool virtual no ontap-san-economy backend. Essas são as únicas ofertas de piscina 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 A classe de armazenamento será mapeada para o testAPP piscina virtual no ontap-san motorista com sanType: nvme . 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"

```

A Trident decidirá qual pool virtual será selecionado e garantirá que o requisito de armazenamento seja atendido.

Drivers ONTAP NAS

Visão geral do driver ONTAP NAS

Aprenda a configurar um backend ONTAP com os drivers ONTAP NAS do ONTAP e do Cloud Volumes ONTAP .

Detalhes do driver ONTAP NAS

A Trident fornece os seguintes drivers de armazenamento NAS para comunicação com o cluster ONTAP . Os modos de acesso suportados são: *ReadWriteOnce* (RWO), *ReadOnlyMany* (ROX), *ReadWriteMany* (RWX), *ReadWriteOncePod* (RWOP).

Motorista	Protocolo	modo de volume	Modos de acesso suportados	Sistemas de arquivos suportados
ontap-nas	NFS SMB	Sistema de arquivos	RWO, ROX, RWX, RWOP	"," nfs , smb
ontap-nas-economy	NFS SMB	Sistema de arquivos	RWO, ROX, RWX, RWOP	"," nfs , smb
ontap-nas-flexgroup	NFS SMB	Sistema de arquivos	RWO, ROX, RWX, RWOP	"," nfs , smb



- Usar `ontap-san-economy` somente se a contagem de uso de volume persistente prevista for maior que "[limites de volume ONTAP suportados](#)" .
- Usar `ontap-nas-economy` somente se a contagem de uso de volume persistente prevista for maior que "[limites de volume ONTAP suportados](#)" e o `ontap-san-economy` O driver não pode ser usado.
- Não use `ontap-nas-economy` Se você prevê a necessidade de proteção de dados, recuperação de desastres ou mobilidade.
- A NetApp não recomenda o uso do Flexvol autogrow em todos os drivers ONTAP , exceto no `ontap-san`. Como solução alternativa, o Trident suporta o uso de reserva de snapshots e dimensiona os volumes Flexvol de acordo.

Permissões do usuário

O Trident espera ser executado como administrador 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 implementações do Amazon FSx for NetApp ONTAP , o Trident espera ser executado como administrador do ONTAP ou do SVM, usando o cluster. `fsxadmin` usuário ou um `vsadmin` Usuário SVM, ou um usuário com um nome diferente que tenha a mesma função. O `fsxadmin` O usuário é um substituto limitado para o usuário administrador do cluster.



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

Embora seja possível criar uma função mais restritiva dentro do ONTAP que um driver Trident possa usar, não recomendamos isso. A maioria das novas versões do Trident chamará APIs adicionais que precisarão ser consideradas, tornando as atualizações difíceis e propensas a erros.

Prepare-se para configurar um backend com drivers ONTAP NAS.

Compreenda os requisitos, as opções de autenticação e as políticas de exportação para configurar um backend ONTAP com drivers ONTAP NAS.

Requisitos

- Para todos os backends ONTAP , o Trident exige que pelo menos um agregado seja atribuído à SVM.
- Você pode executar mais de um driver e criar classes de armazenamento que apontem para um ou outro. Por exemplo, você poderia configurar uma classe Gold que usa o `ontap-nas-motorista` e uma classe Bronze que usa o `ontap-nas-economy` um.
- Todos os seus nós de trabalho do Kubernetes devem ter as ferramentas NFS apropriadas instaladas. Consulte ["aqui"](#) para mais detalhes.
- O Trident suporta volumes SMB montados em pods executados apenas em nós Windows. Consulte [Prepare-se para provisionar volumes SMB](#) para mais detalhes.

Autenticar o backend ONTAP

O Trident oferece dois modos de autenticação de um backend ONTAP .

- Baseado em credenciais: Este modo requer permissões suficientes no backend do ONTAP . Recomenda-se usar uma conta associada a uma função de login de segurança predefinida, como: `admin` ou `vsadmin` . Para garantir a máxima compatibilidade com as versões do ONTAP .
- Baseado em certificado: Este modo requer um certificado instalado no servidor para que o Trident se comunique com um cluster ONTAP . Aqui, a definição do backend deve conter os valores codificados em Base64 do certificado do cliente, da chave e do certificado da CA confiável, se utilizado (recomendado).

Você pode atualizar os sistemas de backend existentes para alternar entre métodos baseados em credenciais e métodos baseados em certificados. No entanto, apenas um método de autenticação é suportado por vez. Para mudar para um método de autenticação diferente, você deve remover o método existente da configuração do backend.



Se você tentar fornecer **tanto credenciais quanto certificados**, a criação do backend falhará com um erro informando que mais de um método de autenticação foi fornecido no arquivo de configuração.

Ativar autenticação baseada em credenciais

O Trident requer as credenciais de um administrador com escopo de SVM/cluster para se comunicar com o backend do ONTAP . Recomenda-se o uso de funções padrão predefinidas, como: `admin` ou `vsadmin` . Isso garante a compatibilidade futura com versões futuras do ONTAP que possam expor APIs de recursos a serem usadas por versões futuras do Trident . É possível criar e usar uma função de login de segurança personalizada com o Trident, mas isso não é recomendado.

Uma definição de backend de exemplo terá a seguinte aparência:

YAML

```
---
version: 1
backendName: ExampleBackend
storageDriverName: ontap-nas
managementLIF: 10.0.0.1
dataLIF: 10.0.0.2
svm: svm_nfs
credentials:
  name: secret-backend-creds
```

JSON

```
{
  "version": 1,
  "backendName": "ExampleBackend",
  "storageDriverName": "ontap-nas",
  "managementLIF": "10.0.0.1",
  "dataLIF": "10.0.0.2",
  "svm": "svm_nfs",
  "credentials": {
    "name": "secret-backend-creds"
  }
}
```

Lembre-se de que a definição do backend é o único lugar onde as credenciais são armazenadas em texto simples. Após a criação do backend, os nomes de usuário/senhas são codificados em Base64 e armazenados como segredos do Kubernetes. A criação/atualização de um backend é a única etapa que exige conhecimento das credenciais. Sendo assim, trata-se de uma operação exclusiva para administradores, a ser realizada pelo administrador do Kubernetes/armazenamento.

Habilitar autenticação baseada em certificado

Novos e existentes sistemas de backend podem usar um certificado e se comunicar com o backend ONTAP . São necessários três parâmetros na definição do backend.

- `clientCertificate`: Valor do certificado do cliente codificado em Base64.
- `clientPrivateKey`: Valor codificado em Base64 da chave privada associada.
- `trustedCACertificate`: Valor codificado em Base64 do certificado da Autoridade Certificadora (CA) confiável. Caso esteja utilizando uma Autoridade Certificadora (CA) confiável, este parâmetro deve ser fornecido. Isso pode ser ignorado se nenhuma Autoridade Certificadora (CA) confiável for utilizada.

Um fluxo de trabalho típico envolve as seguintes etapas.

Passos

1. Gere um certificado e uma chave de cliente. Ao gerar o código, defina o Nome Comum (CN) para o usuário ONTAP que será usado para autenticação.

```
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. Adicione um certificado CA confiável ao cluster ONTAP . Isso pode já estar sendo tratado pelo administrador de armazenamento. Ignore se nenhuma Autoridade Certificadora (CA) confiável for utilizada.

```
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 (do passo 1) no cluster 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 é compatível. cert método de autenticação.

```
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. Teste a autenticação usando o certificado gerado. Substitua < ONTAP Management LIF> e <vserver name> pelo endereço IP do Management LIF e pelo nome do SVM. Você deve garantir que a política de serviço do LIF esteja definida como 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. Codifique o certificado, a chave e o certificado da CA confiável em Base64.

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

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

```
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...Vaaalllluuuuueeee",
  "clientPrivateKey": "LS0tFaKE...0VaLuES0tLS0K",
  "storagePrefix": "myPrefix_"
}

#Update backend with tridentctl
tridentctl update backend NasBackend -f cert-backend-updated.json -n
trident
```

NAME	STORAGE DRIVER	UUID
NasBackend	ontap-nas	98e19b74-aec7-4a3d-8dcf-128e5033b214

Atualize os métodos de autenticação ou altere as credenciais.

Você pode atualizar um backend existente para usar um método de autenticação diferente ou para rotacionar suas credenciais. Isso funciona nos dois sentidos: os sistemas internos que utilizam nome de usuário/senha podem ser atualizados para usar certificados; os sistemas internos que utilizam certificados podem ser atualizados para usar nome de usuário/senha. Para isso, você deve remover o método de autenticação existente e adicionar o novo método de autenticação. Em seguida, utilize o arquivo backend.json atualizado, que contém os parâmetros necessários, para executar o comando `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
NasBackend	ontap-nas	98e19b74-aec7-4a3d-8dcf-128e5033b214



Ao rotacionar senhas, o administrador de armazenamento deve primeiro atualizar a senha do usuário no ONTAP. Em seguida, é realizada uma atualização do sistema interno. Ao rotacionar certificados, vários certificados podem ser adicionados ao usuário. Em seguida, o sistema de backend é atualizado para usar o novo certificado, após o que o certificado antigo pode ser excluído do cluster ONTAP .

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

Criar função ONTAP personalizada para Trident

Você pode criar uma função de cluster ONTAP com privilégios mínimos para que não precise usar a função de administrador do ONTAP para executar operações no Trident. Ao incluir o nome de usuário em uma configuração de backend do Trident , o Trident usa a função de cluster ONTAP que você criou para executar as operações.

Consulte "[Gerador de funções personalizadas Trident](#)" Para obter mais informações sobre como criar funções personalizadas do Trident .

Utilizando a CLI do ONTAP

1. Crie uma nova função usando o seguinte comando:

```
security login role create <role_name\> -cmddirname "command" -access all  
-vserver <svm_name\>
```

2. Crie um nome de usuário para o usuário do Trident :

```
security login create -username <user_name\> -application ontapi  
-authmethod <password\> -role <name_of_role_in_step_1\> -vserver  
<svm_name\> -comment "user_description"
```

3. Atribua a função ao usuário:

```
security login modify username <user_name\> -vserver <svm_name\> -role  
<role_name\> -application ontapi -application console -authmethod  
<password\>
```

Utilizando o Gerenciador de Sistemas

Execute as seguintes etapas no ONTAP System Manager:

1. **Criar uma função personalizada:**

- a. Para criar uma função personalizada no nível do cluster, selecione **Cluster > Configurações**.

(Ou) Para criar uma função personalizada no nível da SVM, selecione **Armazenamento > VMs de armazenamento > required svm > Configurações > Usuários e funções**.

- b. Selecione o ícone de seta (→) ao lado de **Usuários e Funções**.

- c. Selecione **+Adicionar** em **Funções**.

- d. Defina as regras para a função e clique em **Salvar**.

2. **Atribua a função ao usuário do Trident *: + Execute as seguintes etapas na página *Usuários e Funções:**

- a. Selecione o ícone Adicionar * em **Usuários**.

- b. Selecione o nome de usuário desejado e, em seguida, selecione uma função no menu suspenso **Função**.

- c. Clique em **Salvar**.

Consulte as páginas seguintes para obter mais informações:

- ["Funções personalizadas para administração do ONTAP"](#) ou ["Defina funções personalizadas"](#)
- ["Trabalhar com funções e usuários"](#)

Gerenciar políticas de exportação NFS

O Trident utiliza políticas de exportação NFS para controlar o acesso aos volumes que provisiona.

A Trident oferece duas opções para trabalhar com políticas de exportação:

- O Trident pode gerenciar dinamicamente a própria política de exportação; nesse modo de operação, o administrador de armazenamento especifica uma lista de blocos CIDR que representam endereços IP admissíveis. O Trident adiciona automaticamente à política de exportação, no momento da publicação, os endereços IP dos nós aplicáveis que se enquadram nesses intervalos. Alternativamente, quando nenhum CIDR for especificado, todos os IPs unicast de escopo global encontrados no nó para o qual o volume está sendo publicado serão adicionados à política de exportação.
- Os administradores de armazenamento podem criar uma política de exportação e adicionar regras manualmente. O Trident utiliza a política de exportação padrão, a menos que um nome de política de exportação diferente seja especificado na configuração.

Gerenciar políticas de exportação dinamicamente

O Trident oferece a capacidade de gerenciar dinamicamente as políticas de exportação para backends ONTAP . Isso permite ao administrador de armazenamento especificar um espaço de endereços permitido para os IPs dos nós de trabalho, em vez de definir regras explícitas manualmente. Isso simplifica bastante a gestão da política de exportação; as alterações na política de exportação não exigem mais intervenção manual no cluster de armazenamento. Além disso, isso ajuda a restringir o acesso ao cluster de armazenamento apenas aos nós de trabalho que estão montando volumes e possuem endereços IP no intervalo especificado, permitindo um gerenciamento preciso e automatizado.



Não utilize Network Address Translation (NAT) ao usar políticas de exportação dinâmicas. Com NAT, o controlador de armazenamento vê o endereço NAT de front-end e não o endereço IP real do host; portanto, o acesso será negado quando nenhuma correspondência for encontrada nas regras de exportação.

Exemplo

Existem duas opções de configuração que devem ser utilizadas. Aqui está um exemplo de definição de backend:

```
---
version: 1
storageDriverName: ontap-nas-economy
backendName: ontap_nas_auto_export
managementLIF: 192.168.0.135
svm: svm1
username: vsadmin
password: password
autoExportCIDRs:
  - 192.168.0.0/24
autoExportPolicy: true
```



Ao utilizar este recurso, você deve garantir que a junção raiz em sua SVM tenha uma política de exportação previamente criada com uma regra de exportação que permita o bloco CIDR do nó (como a política de exportação padrão). Siga sempre as melhores práticas recomendadas pela NetApp para dedicar uma SVM ao Trident.

Segue abaixo uma explicação de como essa funcionalidade opera, utilizando o exemplo acima:

- `autoExportPolicy` está definido para `true`. Isso indica que o Trident cria uma política de exportação para cada volume provisionado com esse backend para o `svm1` SVM e lidar com a adição e exclusão de regras usando `autoexportCIDRs` blocos de endereço. Até que um volume seja anexado a um nó, ele utiliza uma política de exportação vazia, sem regras para impedir o acesso indesejado a esse volume. Quando um volume é publicado em um nó, o Trident cria uma política de exportação com o mesmo nome da `qtree` subjacente, contendo o endereço IP do nó dentro do bloco CIDR especificado. Esses IPs também serão adicionados à política de exportação usada pelo FlexVol volume pai.
 - Por exemplo:
 - UUID do backend `403b5326-8482-40db-96d0-d83fb3f4daec`
 - `autoExportPolicy` definido para `true`
 - prefixo de armazenamento `trident`
 - UUID do PVC `a79bcf5f-7b6d-4a40-9876-e2551f159c1c`
 - `qtree` chamado `trident_pvc_a79bcf5f_7b6d_4a40_9876_e2551f159c1c` cria uma política de exportação para o FlexVol chamado `trident-403b5326-8482-40db96d0-d83fb3f4daec`, uma política de exportação para a `qtree` chamada `trident_pvc_a79bcf5f_7b6d_4a40_9876_e2551f159c1c` e uma política de exportação vazia chamada `trident_empty` na SVM. As regras para a política de exportação FlexVol serão um superconjunto de quaisquer regras contidas nas políticas de exportação `qtree`. A política de exportação vazia será reutilizada por quaisquer volumes que não estejam anexados.
- `autoExportCIDRs` Contém uma lista de blocos de endereços. Este campo é opcional e o valor padrão é `["0.0.0.0/0", "::/0"]`. Caso não esteja definido, o Trident adiciona todos os endereços unicast de escopo global encontrados nos nós de trabalho com publicações.

Neste exemplo, o `192.168.0.0/24` O espaço de endereçamento é fornecido. Isso indica que os endereços IP dos nós do Kubernetes que se enquadram nesse intervalo de endereços com publicações serão adicionados à política de exportação criada Trident. Quando o Trident registra um nó no qual está sendo executado, ele recupera os endereços IP do nó e os verifica em relação aos blocos de endereços fornecidos em `autoExportCIDRs`. No momento da publicação, após filtrar os IPs, o Trident cria as regras de política de exportação para os IPs do cliente para o nó no qual está publicando.

Você pode atualizar `autoExportPolicy` e `autoExportCIDRs` para os backends depois de criá-los. Você pode adicionar novos CIDRs para um backend que é gerenciado automaticamente ou excluir CIDRs existentes. Tenha cuidado ao excluir CIDRs para garantir que as conexões existentes não sejam interrompidas. Você também pode optar por desativar `autoExportPolicy` para um backend e recorrer a uma política de exportação criada manualmente como alternativa. Isso exigirá a configuração do `exportPolicy` parâmetro na sua configuração de backend.

Após o Trident criar ou atualizar um backend, você pode verificar o backend usando `tridentctl` ou o correspondente `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>
      fileType: ext4

```

Quando um nó é removido, o Trident verifica todas as políticas de exportação para remover as regras de acesso correspondentes ao nó. Ao remover o endereço IP deste nó das políticas de exportação dos backends gerenciados, o Trident impede montagens não autorizadas, a menos que este endereço IP seja reutilizado por um novo nó no cluster.

Para backends já existentes, atualize o backend com `tridentctl update backend`. Garante que o Trident gerencie as políticas de exportação automaticamente. Isso cria duas novas políticas de exportação com os nomes do UUID e da árvore de consulta (qtree) do backend, quando necessárias. Os volumes presentes no servidor usarão as políticas de exportação recém-criadas após serem desmontados e montados novamente.



Excluir um backend com políticas de exportação gerenciadas automaticamente excluirá a política de exportação criada dinamicamente. Se o backend for recriado, ele será tratado como um novo backend e resultará na criação de uma nova política de exportação.

Se o endereço IP de um nó ativo for atualizado, você deverá reiniciar o pod do Trident nesse nó. A Trident atualizará então a política de exportação dos servidores que gerencia para refletir essa alteração de IP.

Prepare-se para provisionar volumes SMB

Com um pouco de preparação adicional, você pode provisionar volumes SMB usando `ontap-nas` motoristas.



Você deve configurar os protocolos NFS e SMB/CIFS na SVM para criar um `ontap-nas-economy` Volume SMB para clusters ONTAP locais. A falha na configuração de qualquer um desses protocolos fará com que a criação do volume SMB falhe.



``autoExportPolicy`` Não é compatível com volumes SMB.

Antes de começar

Antes de poder provisionar volumes SMB, você precisa ter o seguinte.

- Um cluster Kubernetes com um nó controlador Linux e pelo menos um nó de trabalho Windows executando o Windows Server 2022. O Trident suporta volumes SMB montados em pods executados apenas em nós Windows.
- Pelo menos um segredo Trident contendo suas credenciais do Active Directory. Para gerar segredos `smbcreds` :

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

- Um proxy CSI configurado como um serviço do Windows. Para configurar um `csi-proxy` , consulte "[GitHub: Proxy CSI](#)" ou "[GitHub: CSI Proxy para Windows](#)" para nós do Kubernetes executados no Windows.

Passos

1. Para o ONTAP local, você pode opcionalmente criar um compartilhamento SMB ou a Trident pode criar um para você.



Os compartilhamentos SMB são necessários para o Amazon FSx para ONTAP.

Você pode criar os compartilhamentos administrativos SMB de duas maneiras: usando o "[Console de gerenciamento da Microsoft](#)" Acesse as Pastas Compartilhadas pelo snap-in ou usando a CLI do ONTAP . Para criar compartilhamentos SMB usando a CLI do ONTAP :

- a. Se necessário, crie a estrutura de diretórios para o compartilhamento.

O `vserver cifs share create` O comando verifica o caminho especificado na opção `-path` durante a criação do compartilhamento. Se o caminho especificado não existir, o comando falhará.

- b. Crie um compartilhamento SMB associado à SVM especificada:

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

- c. Verifique se o compartilhamento foi criado:

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



Consulte "[Criar um compartilhamento SMB](#)" Para obter detalhes completos.

2. Ao criar o backend, você deve configurar o seguinte para especificar os volumes SMB. Para todas as opções de configuração do backend FSx para ONTAP , consulte "[Opções e exemplos de configuração do FSx para ONTAP](#)" .

Parâmetro	Descrição	Exemplo
smbShare	Você pode especificar uma das seguintes opções: o nome de um compartilhamento SMB criado usando o Console de Gerenciamento da Microsoft ou a CLI do ONTAP ; um nome para permitir que o Trident crie o compartilhamento SMB; ou você pode deixar o parâmetro em branco para impedir o acesso comum aos volumes compartilhados. Este parâmetro é opcional para o ONTAP local. Este parâmetro é obrigatório para backends do Amazon FSx para ONTAP e não pode estar em branco.	smb-share
nasType	Deve ser configurado para smb . Se for nulo, o valor padrão é <code>nfs</code> .	smb
securityStyle	Estilo de segurança para novos volumes. Deve ser configurado para ntfs ou mixed para volumes SMB.	ntfs`ou `mixed para volumes SMB
unixPermissions	Modo para novos volumes. Deve ficar vazio para volumes SMB.	""

Ativar SMB seguro

A partir da versão 25.06, o NetApp Trident oferece suporte ao provisionamento seguro de volumes SMB criados usando `ontap-nas` e `ontap-nas-economy` back-ends. Quando o SMB seguro está habilitado, você pode fornecer acesso controlado aos compartilhamentos SMB para usuários e grupos de usuários do Active Directory (AD) usando Listas de Controle de Acesso (ACLs).

Pontos a serem lembrados

- Importando `ontap-nas-economy` O volume não é suportado.
- Somente clones somente leitura são suportados para `ontap-nas-economy` volumes.
- Se o SMB seguro estiver ativado, o Trident ignorará o compartilhamento SMB mencionado no backend.
- A atualização da anotação PVC, da anotação da classe de armazenamento e do campo de backend não atualiza a ACL de compartilhamento SMB.
- A ACL de compartilhamento SMB especificada na anotação do PVC clonado terá precedência sobre as do PVC de origem.
- Certifique-se de fornecer usuários válidos do Active Directory ao habilitar o SMB seguro. Usuários inválidos não serão adicionados à ACL.
- Se você fornecer o mesmo usuário do Active Directory no backend, na classe de armazenamento e no PVC com permissões diferentes, a prioridade de permissão será: PVC, classe de armazenamento e, por último, backend.
- O Secure SMB é compatível com `ontap-nas` Importações de volume gerenciadas e não aplicáveis a importações de volume não gerenciadas.

Passos

1. Especifique o usuário `adAdminUser` no `TridentBackendConfig` conforme mostrado no exemplo a seguir:

```

apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-ontap
  namespace: trident
spec:
  version: 1
  storageDriverName: ontap-nas
  managementLIF: 10.193.176.x
  svm: svm0
  useREST: true
  defaults:
    adAdminUser: tridentADtest
  credentials:
    name: backend-tbc-ontap-invest-secret

```

2. Adicione a anotação na classe de armazenamento.

Adicione o `trident.netapp.io/smbShareAdUser` Anotação na classe de armazenamento para habilitar o SMB seguro sem falhas. O valor do usuário especificado para a anotação `trident.netapp.io/smbShareAdUser` deve ser o mesmo que o nome de usuário especificado no `smbcreds` segredo. Você pode escolher uma das seguintes opções para `smbShareAdUserPermission`: `full_control`, `change`, ou `read`. A permissão padrão é `full_control`.

```

apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontap-smb-sc
  annotations:
    trident.netapp.io/smbShareAdUserPermission: change
    trident.netapp.io/smbShareAdUser: tridentADuser
parameters:
  backendType: ontap-nas
  csi.storage.k8s.io/node-stage-secret-name: smbcreds
  csi.storage.k8s.io/node-stage-secret-namespace: trident
  trident.netapp.io/nasType: smb
provisioner: csi.trident.netapp.io
reclaimPolicy: Delete
volumeBindingMode: Immediate

```

1. Criar um PVC.

O exemplo a seguir cria um PVC:

```

apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: my-pvc4
  namespace: trident
  annotations:
    trident.netapp.io/snapshotDirectory: "true"
    trident.netapp.io/smbShareAccessControl: |
      read:
        - tridentADtest
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: ontap-smb-sc

```

Opções e exemplos de configuração do ONTAP NAS



Aprenda a criar e usar drivers ONTAP NAS com sua instalação do Trident . Esta seção fornece exemplos de configuração de backend e detalhes para mapear backends para StorageClasses.


Opções de configuração do backend

Consulte a tabela a seguir para obter as opções de configuração do backend:

Parâmetro	Descrição	Padrão
version		Sempre 1
storageDriverName	Nome do driver de armazenamento	ontap-nas, ontap-nas-economy , ou ontap-nas-flexgroup
backendName	Nome personalizado ou o backend de armazenamento	Nome do motorista + "_" + 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 configurado para usar endereços IPv6 se o Trident foi instalado usando a opção IPv6. Os endereços IPv6 devem ser definidos entre colchetes, como por exemplo: [28e8:d9fb:a825:b7bf:69a8:d02f:9e7b:3555] . Para uma transição perfeita para o MetroCluster , consulte o Exemplo MetroCluster .	"10.0.0.1", "[2001:1234:abcd::fefe]"

Parâmetro	Descrição	Padrão
dataLIF	Endereço IP do protocolo LIF. A NetApp recomenda especificar <code>dataLIF</code> . Caso não sejam fornecidos, o Trident obtém os <code>dataLIFs</code> da SVM. Você pode especificar um nome de domínio totalmente qualificado (FQDN) para ser usado nas operações de montagem NFS, permitindo criar um DNS round-robin para balancear a carga entre várias <code>dataLIFs</code> . Pode ser alterado após a configuração inicial. Consulte . Pode ser configurado para usar endereços IPv6 se o Trident foi instalado usando a opção IPv6. Os endereços IPv6 devem ser definidos entre colchetes, como por exemplo: [28e8:d9fb:a825:b7bf:69a8:d02f:9e7b:3555] . Omitir para Metrocluster. Veja o Exemplo MetroCluster .	Endereço especificado ou derivado de SVM, caso não seja especificado (não recomendado).
svm	Máquina virtual de armazenamento a ser usada Omitir para Metrocluster. Veja o Exemplo MetroCluster .	Derivado de uma SVM <code>managementLIF</code> é especificado
autoExportPolicy	Ativar a criação e atualização automática da política de exportação [Booleano]. Usando o <code>autoExportPolicy</code> e <code>autoExportCIDRs</code> O Trident pode gerenciar políticas de exportação automaticamente, dependendo das opções disponíveis.	falso
autoExportCIDRs	Lista de CIDRs para filtrar os IPs dos nós do Kubernetes quando <code>autoExportPolicy</code> está habilitado. Usando o <code>autoExportPolicy</code> e <code>autoExportCIDRs</code> O Trident pode gerenciar políticas de exportação automaticamente, dependendo das opções disponíveis.	["0.0.0.0/0", ":::0"]
labels	Conjunto de rótulos arbitrários formatados em JSON para aplicar aos volumes	""
clientCertificate	Valor do certificado do cliente codificado em Base64. Utilizado para autenticação baseada em certificado.	""
clientPrivateKey	Valor da chave privada do cliente codificado em Base64. Utilizado para autenticação baseada em certificado.	""
trustedCACertificate	Valor codificado em Base64 do certificado da Autoridade Certificadora (CA) confiável. Opcional. Utilizado para autenticação baseada em certificado.	""
username	Nome de usuário para conectar ao cluster/SVM. Usado para autenticação baseada em credenciais. Para autenticação do Active Directory, consulte "Autenticar o Trident em um SVM de backend usando credenciais do Active Directory" .	

Parâmetro	Descrição	Padrão
password	Senha para conectar ao cluster/SVM. Usado para autenticação baseada em credenciais. Para autenticação do Active Directory, consulte "Autenticar o Trident em um SVM de backend usando credenciais do Active Directory" .	
storagePrefix	<p>Prefixo usado ao provisionar novos volumes no SVM. Não pode ser atualizado depois de configurado.</p> <div>  <p>Ao usar o ontap-nas-economy e um storagePrefix com 24 ou mais caracteres, as qtrees não terão o prefixo de armazenamento incorporado, embora ele esteja presente no nome do volume.</p> </div>	"tridente"
aggregate	<p>Agregado para provisionamento (opcional; se definido, deve ser atribuído à SVM). Para o ontap-nas-flexgroup motorista, esta opção é ignorada. Caso não esteja atribuído, qualquer um dos agregados disponíveis pode ser usado para provisionar um volume FlexGroup .</p> <div>  <p>Quando o agregado é atualizado no SVM, ele é atualizado automaticamente no Trident por meio de polling no SVM, sem a necessidade de reiniciar o Controlador Trident . Quando você configura um agregado específico no Trident para provisionar volumes, se o agregado for renomeado ou movido para fora do SVM, o backend entrará em estado de falha no Trident durante a consulta ao agregado do SVM. Você deve alterar o agregado para um que esteja presente na SVM ou removê-lo completamente para que o backend volte a ficar online.</p> </div>	""
limitAggregateUsage	O provisionamento falhará se a utilização for superior a esta percentagem. Não se aplica ao Amazon FSx para ONTAP.	"" (não aplicado por padrão)

Parâmetro	Descrição	Padrão
flexgroupAggregateList	<p>Lista de agregados para provisionamento (opcional; se definida, deve ser atribuída à SVM). Todos os agregados atribuídos à SVM são usados para provisionar um volume FlexGroup . Compatível com o driver de armazenamento ontap-nas-flexgroup.</p> <div>  <p>Quando a lista agregada é atualizada no SVM, a lista é atualizada automaticamente no Trident por meio de polling no SVM, sem a necessidade de reiniciar o Controlador Trident . Quando você configura uma lista de agregação específica no Trident para provisionar volumes, se a lista de agregação for renomeada ou movida para fora do SVM, o backend entrará em estado de falha no Trident durante a consulta da lista de agregação do SVM. Você deve alterar a lista agregada para uma que esteja presente na SVM ou removê-la completamente para que o backend volte a funcionar.</p> </div>	""
limitVolumeSize	O provisionamento falhará se o tamanho do volume solicitado for superior a este valor. Também restringe o tamanho máximo dos volumes que gerencia para qtrees, e o qtreesPerFlexvol Essa opção permite personalizar o número máximo de qtrees por FlexVol volume.	"" (não aplicado por padrão)
debugTraceFlags	Sinalizadores de depuração a serem usados na resolução de problemas. Exemplo: {"api":false, "method":true} Não use debugTraceFlags a menos que você esteja solucionando problemas e precise de um despejo de logs detalhado.	nulo
nasType	Configure a criação de volumes NFS ou SMB. As opções são nfs , smb ou nulo. Definir como nulo utiliza, por padrão, volumes NFS.	nfs

Parâmetro	Descrição	Padrão
nfsMountOptions	Lista de opções de montagem NFS separadas por vírgulas. As opções de montagem para volumes persistentes do Kubernetes são normalmente especificadas nas classes de armazenamento, mas se nenhuma opção de montagem for especificada em uma classe de armazenamento, o Trident usará as opções de montagem especificadas no arquivo de configuração do backend de armazenamento. Se nenhuma opção de montagem for especificada na classe de armazenamento ou no arquivo de configuração, o Trident não definirá nenhuma opção de montagem em um volume persistente associado.	""
qtreesPerFlexvol	Número máximo de Qtrees por FlexVol, deve estar no intervalo [50, 300]	"200"
smbShare	Você pode especificar uma das seguintes opções: o nome de um compartilhamento SMB criado usando o Console de Gerenciamento da Microsoft ou a CLI do ONTAP ; um nome para permitir que o Trident crie o compartilhamento SMB; ou você pode deixar o parâmetro em branco para impedir o acesso comum aos volumes compartilhados. Este parâmetro é opcional para o ONTAP local. Este parâmetro é obrigatório para backends do Amazon FSx para ONTAP e não pode estar em branco.	smb-share
useREST	Parâmetro booleano para usar APIs REST do ONTAP .useREST`Quando definido para `true O Trident usa APIs REST do ONTAP para se comunicar com o backend; quando configurado para false O Trident utiliza chamadas ONTAPI (ZAPI) para se comunicar com o backend. Este recurso requer o ONTAP 9.11.1 e posterior. Além disso, a função de login do ONTAP utilizada deve ter acesso ao <code>ontapi</code> aplicativo. Isso é satisfeito pelo predefinido <code>vsadmin</code> e <code>cluster-admin</code> papéis. A partir da versão Trident 24.06 e do ONTAP 9.15.1 ou posterior, <code>useREST</code> está definido para <code>true</code> por padrão; alterar <code>useREST</code> para <code>false</code> para usar chamadas ONTAPI (ZAPI).	<code>true</code> para ONTAP 9.15.1 ou posterior, caso contrário <code>false</code> .
limitVolumePoolSize	Tamanho máximo de FlexVol solicitável ao usar Qtrees no backend <code>ontap-nas-economy</code> .	"" (não aplicado por padrão)
denyNewVolumePools	Restringe <code>ontap-nas-economy</code> backends da criação de novos volumes FlexVol para conter suas Qtrees. Apenas os Flexvols preexistentes são usados para provisionar novos PVs.	
adAdminUser	Usuário ou grupo de usuários administradores do Active Directory com acesso total aos compartilhamentos SMB. Use este parâmetro para conceder direitos de administrador ao compartilhamento SMB com controle total.	

Opções de configuração de backend para provisionamento de volumes

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

Parâmetro	Descrição	Padrão
<code>spaceAllocation</code>	Alocação de espaço para Qtrees	"verdadeiro"
<code>spaceReserve</code>	Modo de reserva de espaço; "nenhum" (fino) ou "volume" (grosso)	"nenhum"
<code>snapshotPolicy</code>	Política de instantâneo a ser usada	"nenhum"
<code>qosPolicy</code>	Grupo de políticas de QoS a ser atribuído aos volumes criados. Escolha <code>qosPolicy</code> ou <code>adaptiveQosPolicy</code> por pool de armazenamento/backend.	""
<code>adaptiveQosPolicy</code>	Grupo de políticas de QoS adaptativas a serem atribuídas aos volumes criados. Escolha uma das opções <code>qosPolicy</code> ou <code>adaptiveQosPolicy</code> para cada pool de armazenamento/backend. Não suportado por <code>ontap-nas-economy</code> .	""
<code>snapshotReserve</code>	Percentagem do volume reservada para instantâneos	"0" se <code>snapshotPolicy</code> é "nenhum", caso contrário ""
<code>splitOnClone</code>	Separar um clone de seu progenitor no momento da criação.	"falso"
<code>encryption</code>	Ative a Criptografia de Volume NetApp (NVE) no novo volume; o padrão é <code>false</code> . Para usar esta opção, o NVE precisa estar licenciado e habilitado no cluster. Se o NAE estiver habilitado no backend, qualquer volume provisionado no Trident terá o NAE habilitado. Para mais informações, consulte: "Como o Trident funciona com NVE e NAE" .	"falso"
<code>tieringPolicy</code>	Política de níveis para usar "nenhum"	
<code>unixPermissions</code>	Modo para novos volumes	"777" para volumes NFS; vazio (não aplicável) para volumes SMB.
<code>snapshotDir</code>	Controla o acesso ao <code>.snapshot</code> diretório	"verdadeiro" para NFSv4 "falso" para NFSv3
<code>exportPolicy</code>	Política de exportação a ser utilizada	"padrão"
<code>securityStyle</code>	Estilo de segurança para novos volumes. Suporte a NFS <code>mixed</code> e <code>unix</code> estilos de segurança. Suporte para PMEs <code>mixed</code> e <code>ntfs</code> estilos de segurança.	O padrão do NFS é <code>unix</code> . O padrão SMB é <code>ntfs</code> .
<code>nameTemplate</code>	Modelo para criar nomes de volume personalizados.	""



A utilização de grupos de políticas de QoS com o Trident requer o ONTAP 9.8 ou posterior. Você deve usar um grupo de políticas de QoS não compartilhado e garantir que o grupo de políticas seja aplicado a cada componente individualmente. Um grupo de políticas de QoS compartilhado impõe o limite máximo para a taxa de transferência total de todas as cargas de trabalho.

Exemplos de provisionamento em volume

Aqui está um exemplo com valores padrão definidos:

```
---
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"
```

Para `ontap-nas` e `ontap-nas-flexgroups` O Trident agora usa um novo cálculo para garantir que o FlexVol seja dimensionado corretamente com a porcentagem `snapshotReserve` e o PVC. Quando o usuário solicita um PVC, o Trident cria o FlexVol original com mais espaço usando o novo cálculo. Esse cálculo garante que o usuário receba o espaço gravável solicitado no PVC, e não menos espaço do que o solicitado. Antes da versão 21.07, quando o usuário solicita um PVC (por exemplo, 5 GiB), com o `snapshotReserve` em 50%, ele obtém apenas 2,5 GiB de espaço gravável. Isso ocorre porque o que o usuário solicitou foi o volume completo e `snapshotReserve` é uma porcentagem disso. Com o Trident 21.07, o que o usuário solicita é o espaço gravável e o Trident define o `snapshotReserve` número como porcentagem do volume total. Isso não se aplica a `ontap-nas-economy`. Veja o exemplo a seguir para ver como isso funciona:

O cálculo é o seguinte:

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

Para `snapshotReserve = 50%` e solicitação de PVC = 5 GiB, o tamanho total do volume é $5/0.5 = 10$ GiB e o tamanho disponível é 5 GiB, que é o que o usuário solicitou na solicitação de PVC. O `volume show` O comando deve exibir resultados semelhantes a este exemplo:

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.

Os backends existentes de instalações anteriores provisionarão volumes conforme explicado acima ao atualizar o Trident. Para volumes criados antes da atualização, você deve redimensioná-los para que a alteração seja observada. Por exemplo, um PVC de 2 GiB com `snapshotReserve=50`. O resultado anterior era um volume que fornecia 1 GiB de espaço gravável. Redimensionar o volume para 3 GiB, por exemplo, fornece ao aplicativo 3 GiB de espaço gravável em um volume de 6 GiB.

Exemplos de configuração mínima

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



Se você estiver usando o Amazon FSx no NetApp ONTAP com Trident, a recomendação é especificar nomes DNS para LIFs em vez de endereços IP.

Exemplo de economia ONTAP NAS

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

Exemplo de grupo flexível ONTAP NAS

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

Exemplo MetroCluster

Você pode configurar o backend para evitar a necessidade de atualizar manualmente a definição do backend após a troca de modo (switchover) e o retorno ao modo anterior (switchback). ["Replicação e recuperação de SVM"](#).

Para uma transição perfeita e um retorno perfeito, especifique a SVM usando `managementLIF` e omitir o `dataLIF` e `svm` parâmetros. Por exemplo:

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

Exemplo de volumes SMB

```
---
version: 1
backendName: ExampleBackend
storageDriverName: ontap-nas
managementLIF: 10.0.0.1
nasType: smb
securityStyle: ntfs
unixPermissions: ""
dataLIF: 10.0.0.2
svm: svm_nfs
username: vsadmin
password: password
```

Exemplo de autenticação baseada em certificado

Este é um exemplo mínimo de configuração de backend. `clientCertificate`, `clientPrivateKey`, e `trustedCACertificate` (opcional, se estiver usando uma CA confiável) são preenchidos em `backend.json` e extraem os valores codificados em base64 do certificado do cliente, da chave privada e do certificado da CA confiável, respectivamente.

```
---
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_
```

Exemplo de política de exportação automática

Este exemplo mostra como você pode instruir o Trident a usar políticas de exportação dinâmicas para criar e gerenciar a política de exportação automaticamente. Isso funciona da mesma forma para o `ontap-nas-economy` e `ontap-nas-flexgroup` motoristas.

```
---
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
```

Exemplo de endereço IPv6

Este exemplo mostra managementLIF usando um endereço IPv6.

```
---
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
```

Exemplo de uso de volumes SMB no Amazon FSx para ONTAP

O smbShare Este parâmetro é necessário para o FSx para ONTAP usando volumes SMB.

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


Exemplo de configuração de backend com nameTemplate

```
---
version: 1
storageDriverName: ontap-nas
backendName: ontap-nas-backend
managementLIF: <ip address>
svm: svm0
username: <admin>
password: <password>
defaults:
  nameTemplate:
    "{{.volume.Name}}_{{.labels.cluster}}_{{.volume.Namespace}}_{{.vo\
      lume.RequestName}}"
labels:
  cluster: ClusterA
PVC: "{{.volume.Namespace}}_{{.volume.RequestName}}"
```

Exemplos de backends com pools virtuais

Nos arquivos de definição de backend de exemplo mostrados abaixo, valores padrão específicos são definidos para todos os pools de armazenamento, como: `spaceReserve` em `nenhum`, `spaceAllocation` em `falso`, e `encryption` `falso`. Os pools virtuais são definidos na seção de armazenamento.

O Trident define os rótulos de provisionamento no campo "Comentários". Os comentários estão definidos no `FlexVol` para `ontap-nas` ou `FlexGroup` para `ontap-nas-flexgroup`. O Trident copia todos os rótulos presentes em um pool virtual para o volume de armazenamento durante o provisionamento. Para maior conveniência, os administradores de armazenamento podem definir rótulos por pool virtual e agrupar volumes por rótulo.

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

Exemplo de ONTAP NAS

```
---
version: 1
storageDriverName: ontap-nas
managementLIF: 10.0.0.1
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:
      department: legal
      creditpoints: "5000"
      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: mysqlldb
    cost: "25"
    zone: us_east_1d
    defaults:
      spaceReserve: volume
      encryption: "false"
      unixPermissions: "0775"
```

Exemplo de FlexGroup NAS ONTAP

```
---
version: 1
storageDriverName: ontap-nas-flexgroup
managementLIF: 10.0.0.1
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"
```

Exemplo de economia ONTAP NAS

```
---
version: 1
storageDriverName: ontap-nas-economy
managementLIF: 10.0.0.1
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:
      protection: bronze
      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"
```

Mapear backends para StorageClasses

As seguintes definições de StorageClass referem-se a [Exemplos de backends com pools virtuais](#) . Usando o `parameters.selector` No campo StorageClass, cada StorageClass especifica quais pools virtuais podem ser usados para hospedar um volume. O volume terá os aspectos definidos na piscina virtual escolhida.

- O `protection-gold` A StorageClass será mapeada para o primeiro e o segundo pool virtual no `ontap-nas-flexgroup` backend. Essas são as únicas piscinas que oferecem proteção de nível ouro.

```
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` A StorageClass será mapeada para o terceiro e quarto pool virtual no `ontap-nas-flexgroup` backend. Essas são as únicas pools 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` A classe de armazenamento será mapeada para o quarto pool virtual no `ontap-nas` backend. Este é o único pool que oferece configuração de pool de armazenamento para aplicativos do 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 A StorageClass será mapeada para o terceiro pool virtual no ontap-nas-flexgroup backend. Este é o único pool que oferece proteção de nível prata e 20.000 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 A StorageClass será mapeada para o terceiro pool virtual no ontap-nas backend e o segundo pool virtual no ontap-nas-economy backend. Essas são as únicas ofertas de piscina 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"

```

A Trident decidirá qual pool virtual será selecionado e garantirá que o requisito de armazenamento seja atendido.

Atualizar dataLIF após a configuração inicial

Você pode alterar o dataLIF após a configuração inicial executando o seguinte comando para fornecer ao novo arquivo JSON de backend o dataLIF atualizado.


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



Se os PVCs estiverem conectados a um ou mais pods, você deve desligar todos os pods correspondentes e, em seguida, ligá-los novamente para que o novo dataLIF entre em vigor.

Exemplos de segurança SMB

Configuração de backend com o driver ontap-nas

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-ontap-nas
  namespace: trident
spec:
  version: 1
  storageDriverName: ontap-nas
  managementLIF: 10.0.0.1
  svm: svm2
  nasType: smb
  defaults:
    adAdminUser: tridentADtest
  credentials:
    name: backend-tbc-ontap-invest-secret
```

Configuração de backend com o driver ontap-nas-economy

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-ontap-nas
  namespace: trident
spec:
  version: 1
  storageDriverName: ontap-nas-economy
  managementLIF: 10.0.0.1
  svm: svm2
  nasType: smb
  defaults:
    adAdminUser: tridentADtest
  credentials:
    name: backend-tbc-ontap-invest-secret
```

Configuração de backend com pool de armazenamento

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-ontap-nas
  namespace: trident
spec:
  version: 1
  storageDriverName: ontap-nas
  managementLIF: 10.0.0.1
  svm: svm0
  useREST: false
  storage:
    - labels:
        app: msoffice
      defaults:
        adAdminUser: tridentADuser
  nasType: smb
  credentials:
    name: backend-tbc-ontap-invest-secret
```

Exemplo de classe de armazenamento com o driver ontap-nas

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontap-smb-sc
  annotations:
    trident.netapp.io/smbShareAdUserPermission: change
    trident.netapp.io/smbShareAdUser: tridentADtest
parameters:
  backendType: ontap-nas
  csi.storage.k8s.io/node-stage-secret-name: smbcreds
  csi.storage.k8s.io/node-stage-secret-namespace: trident
  trident.netapp.io/nasType: smb
provisioner: csi.trident.netapp.io
reclaimPolicy: Delete
volumeBindingMode: Immediate
```



Certifique-se de adicionar annotations Para habilitar o SMB seguro. O SMB seguro não funciona sem as anotações, independentemente das configurações definidas no Backend ou no PVC.

Exemplo de classe de armazenamento com o driver ontap-nas-economy

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontap-smb-sc
  annotations:
    trident.netapp.io/smbShareAdUserPermission: change
    trident.netapp.io/smbShareAdUser: tridentADuser3
parameters:
  backendType: ontap-nas-economy
  csi.storage.k8s.io/node-stage-secret-name: smbcreds
  csi.storage.k8s.io/node-stage-secret-namespace: trident
  trident.netapp.io/nasType: smb
provisioner: csi.trident.netapp.io
reclaimPolicy: Delete
volumeBindingMode: Immediate
```

Exemplo de PVC com um único usuário AD

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: my-pvc4
  namespace: trident
  annotations:
    trident.netapp.io/smbShareAccessControl: |
      change:
        - tridentADtest
      read:
        - tridentADuser
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: ontap-smb-sc
```

Exemplo de PVC com vários usuários de AD

```

apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: my-test-pvc
  annotations:
    trident.netapp.io/smbShareAccessControl: |
      full_control:
        - tridentTestuser
        - tridentuser
        - tridentTestuser1
        - tridentuser1
      change:
        - tridentADuser
        - tridentADuser1
        - tridentADuser4
        - tridentTestuser2
      read:
        - tridentTestuser2
        - tridentTestuser3
        - tridentADuser2
        - tridentADuser3
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi

```

Amazon FSx for NetApp ONTAP

Use o Trident com o Amazon FSx for NetApp ONTAP

"Amazon FSx for NetApp ONTAP"É um serviço AWS totalmente gerenciado que permite aos clientes iniciar e executar sistemas de arquivos com tecnologia do sistema operacional de armazenamento NetApp ONTAP . O FSx para ONTAP permite que você aproveite os recursos, o desempenho e as capacidades administrativas da NetApp com os quais você já está familiarizado, ao mesmo tempo que desfruta da simplicidade, agilidade, segurança e escalabilidade do armazenamento de dados na AWS. O FSx para ONTAP oferece suporte aos recursos do sistema de arquivos ONTAP e às APIs de administração.

Você pode integrar seu sistema de arquivos Amazon FSx for NetApp ONTAP com o Trident para garantir que os clusters Kubernetes em execução no Amazon Elastic Kubernetes Service (EKS) possam provisionar volumes persistentes de bloco e arquivo com suporte do ONTAP.

Um sistema de arquivos é o recurso principal no Amazon FSx, análogo a um cluster ONTAP em um ambiente

local. Dentro de cada SVM, você pode criar um ou vários volumes, que são contêineres de dados que armazenam os arquivos e pastas do seu sistema de arquivos. Com o Amazon FSx for NetApp ONTAP, este será fornecido como um sistema de arquivos gerenciado na nuvem. O novo tipo de sistema de arquivos é chamado * NetApp ONTAP*.

Ao usar o Trident com o Amazon FSx for NetApp ONTAP, você pode garantir que os clusters Kubernetes em execução no Amazon Elastic Kubernetes Service (EKS) possam provisionar volumes persistentes em bloco e em arquivo com suporte do ONTAP.

Requisitos

Além de ["Requisitos do Trident"](#) Para integrar o FSx para ONTAP com o Trident, você precisa de:

- Um cluster Amazon EKS existente ou um cluster Kubernetes autogerenciado com `kubectl` instalado.
- Um sistema de arquivos e uma máquina virtual de armazenamento (SVM) Amazon FSx for NetApp ONTAP existentes e acessíveis a partir dos nós de trabalho do seu cluster.
- Nós de trabalho que estão preparados para ["NFS ou iSCSI"](#) .



Certifique-se de seguir os passos de preparação do nó necessários para o Amazon Linux e o Ubuntu. ["Imagens de máquinas da Amazon"](#) (AMIs) dependendo do seu tipo de AMI EKS.

Considerações

- Volumes SMB:
 - Os volumes SMB são suportados usando o `ontap-nas` Somente o motorista.
 - Volumes SMB não são suportados com o complemento Trident EKS.
 - O Trident suporta volumes SMB montados em pods executados apenas em nós Windows. Consulte ["Prepare-se para provisionar volumes SMB"](#) para mais detalhes.
- Antes da versão 24.02 do Trident , os volumes criados em sistemas de arquivos Amazon FSx com backups automáticos ativados não podiam ser excluídos pelo Trident. Para evitar esse problema no Trident 24.02 ou posterior, especifique o `fsxFilesystemID` , `AWS apiRegion` , `AWS apikey` e `AWS secretKey` no arquivo de configuração de backend para AWS FSx para ONTAP.



Se você estiver especificando uma função do IAM para o Trident, poderá omitir a especificação do... `apiRegion` , `apiKey` , e `secretKey` campos para Trident explicitamente. Para obter mais informações, consulte ["Opções e exemplos de configuração do FSx para ONTAP"](#) .

Uso simultâneo do driver Trident SAN/iSCSI e EBS-CSI

Se você planeja usar drivers `ontap-san` (por exemplo, iSCSI) com AWS (EKS, ROSA, EC2 ou qualquer outra instância), a configuração multipath necessária nos nós pode entrar em conflito com o driver CSI do Amazon Elastic Block Store (EBS). Para garantir que o multipathing funcione sem interferir nos discos EBS no mesmo nó, você precisa excluir o EBS da sua configuração de multipathing. Este exemplo mostra um `multipath.conf` Arquivo que inclui as configurações Trident necessárias, excluindo os discos EBS do multipathing:

```
defaults {
    find_multipaths no
}
blacklist {
    device {
        vendor "NVME"
        product "Amazon Elastic Block Store"
    }
}
```

Autenticação

O Trident oferece dois modos de autenticação.

- Baseado em credenciais (recomendado): Armazena as credenciais com segurança no AWS Secrets Manager. Você pode usar o `fsxadmin` usuário para o seu sistema de arquivos ou o `vsadmin` Usuário configurou sua SVM.



A Trident espera ser administrada como uma `vsadmin` Usuário SVM ou como um usuário com um nome diferente que tenha a mesma função. O Amazon FSx for NetApp ONTAP possui um `fsxadmin` usuário que é um substituto limitado do ONTAP `admin` usuário do cluster. Recomendamos vivamente a utilização de `vsadmin` com Trident.

- Com base em certificado: o Trident se comunicará com a SVM no seu sistema de arquivos FSx usando um certificado instalado na sua SVM.

Para obter detalhes sobre como ativar a autenticação, consulte a documentação de autenticação para o seu tipo de driver:

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

Imagens de Máquina da Amazon (AMIs) testadas

O cluster EKS suporta diversos sistemas operacionais, mas a AWS otimizou determinadas Amazon Machine Images (AMIs) para contêineres e EKS. As seguintes AMIs foram testadas com o NetApp Trident 25.02.

AMI	NAS	NAS-economia	iSCSI	iSCSI-economia
AL2023_x86_64_STANDARD	Sim	Sim	Sim	Sim
AL2_x86_64	Sim	Sim	Sim*	Sim*
BOTTLEROCKET_x86_64	Sim**	Sim	N / D	N / D
AL2023_ARM_64_STANDARD	Sim	Sim	Sim	Sim
AL2_ARM_64	Sim	Sim	Sim*	Sim*

BOTTLEROCKET_ARM_64	Sim**	Sim	N / D	N / D
---------------------	-------	-----	-------	-------

- * Não é possível excluir o PV sem reiniciar o nó
- ** Não funciona com NFSv3 com Trident versão 25.02.



Se a AMI desejada não estiver listada aqui, isso não significa que ela não seja compatível; significa simplesmente que ela não foi testada. Esta lista serve como um guia para AMIs que comprovadamente funcionam.

Testes realizados com:

- Versão EKS: 1.32
- Método de instalação: Helm 25.06 e como um complemento AWS 25.06
- Para NAS, foram testados tanto o NFSv3 quanto o NFSv4.1.
- Para SAN, apenas o iSCSI foi testado, não o NVMe-oF.

Testes realizados:

- Criar: Classe de armazenamento, PVC, cápsula
- Excluir: pod, pvc (regular, qtree/lun – econômico, NAS com backup AWS)

Encontre mais informações

- ["Documentação do Amazon FSx for NetApp ONTAP"](#)
- ["Postagem no blog sobre Amazon FSx for NetApp ONTAP"](#)

Crie uma função do IAM e um segredo da AWS.

Você pode configurar pods do Kubernetes para acessar recursos da AWS autenticando-se como uma função do AWS IAM em vez de fornecer credenciais explícitas da AWS.



Para autenticar usando uma função do AWS IAM, você precisa ter um cluster Kubernetes implantado usando o EKS.

Criar segredo do AWS Secrets Manager

Como o Trident emitirá APIs contra um servidor virtual FSx para gerenciar o armazenamento para você, ele precisará de credenciais para fazer isso. A forma segura de transmitir essas credenciais é por meio de um segredo do AWS Secrets Manager. Portanto, se você ainda não possui uma, precisará criar um segredo do AWS Secrets Manager que contenha as credenciais da conta vsadmin.

Este exemplo cria um segredo do AWS Secrets Manager para armazenar credenciais do Trident CSI:

```
aws secretsmanager create-secret --name trident-secret --description
"Trident CSI credentials" \
  --secret-string
"{\"username\":\"vsadmin\", \"password\":\"<svmpassword>\"}"
```

Criar política de IAM

O Trident também precisa de permissões da AWS para funcionar corretamente. Portanto, você precisa criar uma política que conceda ao Trident as permissões necessárias.

Os exemplos a seguir criam uma política do IAM usando a AWS CLI:

```
aws iam create-policy --policy-name AmazonFSxNCSIDriverPolicy --policy-  
-document file://policy.json  
--description "This policy grants access to Trident CSI to FSxN and  
Secrets manager"
```

Exemplo de JSON de política:

```
{  
  "Statement": [  
    {  
      "Action": [  
        "fsx:DescribeFileSystems",  
        "fsx:DescribeVolumes",  
        "fsx:CreateVolume",  
        "fsx:RestoreVolumeFromSnapshot",  
        "fsx:DescribeStorageVirtualMachines",  
        "fsx:UntagResource",  
        "fsx:UpdateVolume",  
        "fsx:TagResource",  
        "fsx>DeleteVolume"  
      ],  
      "Effect": "Allow",  
      "Resource": "*"   
    },  
    {  
      "Action": "secretsmanager:GetSecretValue",  
      "Effect": "Allow",  
      "Resource": "arn:aws:secretsmanager:<aws-region>:<aws-account-  
id>:secret:<aws-secret-manager-name>*"   
    }  
  ],  
  "Version": "2012-10-17"  
}
```

Criar identidade de Pod ou função IAM para associação de conta de serviço (IRSA)

Você pode configurar uma conta de serviço do Kubernetes para assumir uma função do AWS Identity and Access Management (IAM) com a Identidade do Pod do EKS ou a função do IAM para associação de conta de serviço (IRSA). Qualquer Pod configurado para usar a conta de serviço poderá acessar qualquer serviço

da AWS para o qual a função tenha permissões de acesso.

Identidade do Pod

As associações de identidade de pods do Amazon EKS permitem gerenciar credenciais para seus aplicativos, de forma semelhante à maneira como os perfis de instância do Amazon EC2 fornecem credenciais para instâncias do Amazon EC2.

Instale o Pod Identity no seu cluster EKS:

Você pode criar uma identidade de Pod através do console da AWS ou usando o seguinte comando da AWS CLI:

```
aws eks create-addon --cluster-name <EKS_CLUSTER_NAME> --addon-name
eks-pod-identity-agent
```

Para obter mais informações, consulte ["Configure o agente de identidade do Amazon EKS Pod."](#) .

Criar trust-relationship.json:

Crie o arquivo trust-relationship.json para permitir que o Service Principal do EKS assuma essa função para a identidade do Pod. Em seguida, crie uma função com esta política de confiança:

```
aws iam create-role \
  --role-name fsxn-csi-role --assume-role-policy-document file://trust-
relationship.json \
  --description "fsxn csi pod identity role"
```

Arquivo trust-relationship.json:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "pods.eks.amazonaws.com"
      },
      "Action": [
        "sts:AssumeRole",
        "sts:TagSession"
      ]
    }
  ]
}
```

Anexe a política de função à função do IAM:

Anexe a política de função da etapa anterior à função IAM que foi criada:

```
aws iam attach-role-policy \  
  --policy-arn arn:aws:iam::aws:111122223333:policy/fsxn-csi-policy \  
  --role-name fsxn-csi-role
```

Criar uma associação de identidade de pod:

Crie uma associação de identidade de pod entre a função IAM e a conta de serviço Trident (trident-controller).

```
aws eks create-pod-identity-association \  
  --cluster-name <EKS_CLUSTER_NAME> \  
  --role-arn arn:aws:iam::111122223333:role/fsxn-csi-role \  
  --namespace trident --service-account trident-controller
```

Função IAM para associação de conta de serviço (IRSA)

Utilizando a AWS CLI:

```
aws iam create-role --role-name AmazonEKS_FSxN_CSI_DriverRole \  
  --assume-role-policy-document file://trust-relationship.json
```

Arquivo trust-relationship.json:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Federated": "arn:aws:iam::<account_id>:oidc-
provider/<oidc_provider>"
      },
      "Action": "sts:AssumeRoleWithWebIdentity",
      "Condition": {
        "StringEquals": {
          "<oidc_provider>:aud": "sts.amazonaws.com",
          "<oidc_provider>:sub":
"system:serviceaccount:trident:trident-controller"
        }
      }
    }
  ]
}
```

Atualize os seguintes valores em `trust-relationship.json` arquivo:

- **<account_id>** - Seu ID de conta da AWS
- **<oidc_provider>** - O OIDC do seu cluster EKS. Você pode obter o provedor oidc executando o seguinte comando:

```
aws eks describe-cluster --name my-cluster --query
"cluster.identity.oidc.issuer"\
--output text | sed -e "s/^https:\\/\\/\\/"
```

Associe a função IAM à política IAM:

Após a criação da função, associe a política (criada na etapa anterior) à função usando este comando:

```
aws iam attach-role-policy --role-name my-role --policy-arn <IAM policy
ARN>
```

Verifique se o provedor do OICD está associado:

Verifique se o seu provedor OIDC está associado ao seu cluster. Você pode verificar isso usando este comando:

```
aws iam list-open-id-connect-providers | grep $oidc_id | cut -d "/" -f4
```

Se a saída estiver vazia, use o seguinte comando para associar o IAM OIDC ao seu cluster:

```
eksctl utils associate-iam-oidc-provider --cluster $cluster_name  
--approve
```

Se você estiver usando o eksctl, utilize o exemplo a seguir para criar uma função IAM para a conta de serviço no EKS:

```
eksctl create iamserviceaccount --name trident-controller --namespace  
trident \  
  --cluster <my-cluster> --role-name AmazonEKS_FSxN_CSI_DriverRole  
--role-only \  
  --attach-policy-arn <IAM-Policy ARN> --approve
```

Instalar Trident

O Trident simplifica o gerenciamento de armazenamento do Amazon FSx for NetApp ONTAP no Kubernetes, permitindo que seus desenvolvedores e administradores se concentrem na implantação de aplicativos.

Você pode instalar o Trident usando um dos seguintes métodos:

- Leme
- Complemento EKS

Se você deseja utilizar a funcionalidade de instantâneo, instale o complemento CSI snapshot controller. Consulte ["Ative a funcionalidade de instantâneo para volumes CSI."](#) para mais informações.

Instale o Trident via Helm.

Identidade do Pod

1. Adicione o repositório Trident Helm:

```
helm repo add netapp-trident https://netapp.github.io/trident-helm-chart
```

2. Instale o Trident usando o seguinte exemplo:

```
helm install trident-operator netapp-trident/trident-operator  
--version 100.2502.1 --namespace trident --create-namespace
```

Você pode usar o `helm list` Comando para revisar detalhes da instalação, como nome, namespace, gráfico, status, versão do aplicativo e número da revisão.

```
helm list -n trident
```

NAME	NAMESPACE	REVISION	UPDATED
STATUS	CHART		APP VERSION
trident-operator	trident	1	2024-10-14
14:31:22.463122 +0300	IDT	deployed	trident-operator-
100.2502.0	25.02.0		

Associação de conta de serviço (IRSA)

1. Adicione o repositório Trident Helm:

```
helm repo add netapp-trident https://netapp.github.io/trident-helm-chart
```

2. Defina os valores para **provedor de nuvem e identidade de nuvem**:

```
helm install trident-operator netapp-trident/trident-operator  
--version 100.2502.1 \  
--set cloudProvider="AWS" \  
--set cloudIdentity="'eks.amazonaws.com/role-arn:  
arn:aws:iam::<accountID>:role/<AmazonEKS_FSxN_CSI_DriverRole>' " \  
--namespace trident \  
--create-namespace
```

Você pode usar o `helm list` Comando para revisar detalhes da instalação, como nome, namespace, gráfico, status, versão do aplicativo e número da revisão.

```
helm list -n trident
```

NAME	NAMESPACE	REVISION	UPDATED
STATUS	CHART		APP VERSION
trident-operator	trident	1	2024-10-14
14:31:22.463122 +0300	IDT	deployed	trident-operator-
100.2506.0	25.06.0		

Se você pretende usar iSCSI, certifique-se de que o iSCSI esteja habilitado em sua máquina cliente. Se você estiver usando o sistema operacional de nó de trabalho AL2023, poderá automatizar a instalação do cliente iSCSI adicionando o parâmetro `node prep` na instalação do Helm:



```
helm install trident-operator netapp-trident/trident-operator  
--version 100.2502.1 --namespace trident --create-namespace --  
set nodePrep={iscsi}
```

Instale o Trident através do complemento EKS.

O complemento Trident EKS inclui os patches de segurança e correções de bugs mais recentes, e é validado pela AWS para funcionar com o Amazon EKS. O complemento EKS permite garantir de forma consistente que seus clusters Amazon EKS estejam seguros e estáveis, além de reduzir o trabalho necessário para instalar, configurar e atualizar complementos.

Pré-requisitos

Certifique-se de ter o seguinte antes de configurar o complemento Trident para AWS EKS:

- Uma conta de cluster Amazon EKS com assinatura adicional.
- Permissões da AWS para o marketplace da AWS:
"aws-marketplace:ViewSubscriptions",
"aws-marketplace:Subscribe",
"aws-marketplace:Unsubscribe"
- Tipo de AMI: Amazon Linux 2 (AL2_x86_64) ou Amazon Linux 2 Arm (AL2_ARM_64)
- Tipo de nó: AMD ou ARM
- Um sistema de arquivos Amazon FSx for NetApp ONTAP

Habilite o complemento Trident para AWS.

Console de gerenciamento

1. Abra o console do Amazon EKS em <https://console.aws.amazon.com/eks/home#/clusters>.
2. No painel de navegação à esquerda, selecione **Clusters**.
3. Selecione o nome do cluster para o qual deseja configurar o complemento NetApp Trident CSI.
4. Selecione **Complementos** e depois selecione **Obter mais complementos**.
5. Siga estes passos para selecionar o complemento:
 - a. Desça a página até a seção **Complementos do AWS Marketplace** e digite **"Trident"** na caixa de pesquisa.
 - b. Selecione a caixa de seleção no canto superior direito da caixa Trident by NetApp.
 - c. Selecione **Avançar**.
6. Na página de configurações **Configurar complementos selecionados**, faça o seguinte:



Ignore estas etapas se estiver usando a associação de identidade do Pod.

- a. Selecione a **Versão** que deseja usar.
- b. Se você estiver usando a autenticação IRSA, certifique-se de definir os valores de configuração disponíveis nas Configurações opcionais:
 - Selecione a **Versão** que deseja usar.
 - Siga o **esquema de configuração do complemento** e defina o parâmetro **configurationValues** na seção **Valores de configuração** com o role-arn que você criou na etapa anterior (o valor deve estar no seguinte formato):

```
{  
  
  "cloudIdentity": "'eks.amazonaws.com/role-arn: <role ARN>'",  
  "cloudProvider": "AWS"  
  
}
```

+

Se você selecionar "Substituir" como método de resolução de conflitos, uma ou mais configurações do complemento existente poderão ser substituídas pelas configurações do complemento do Amazon EKS. Se você não ativar essa opção e houver um conflito com suas configurações existentes, a operação falhará. Você pode usar a mensagem de erro resultante para solucionar o conflito. Antes de selecionar esta opção, certifique-se de que o complemento Amazon EKS não gerencie configurações que você precise gerenciar manualmente.

7. Selecione **Próximo**.
8. Na página **Revisar e adicionar**, escolha **Criar**.

Após a conclusão da instalação do complemento, você verá o complemento instalado.

CLI da AWS

1. Crie o `add-on.json` arquivo:

Para a identidade do Pod, utilize o seguinte formato:

```
{
  "clusterName": "<eks-cluster>",
  "addonName": "netapp_trident-operator",
  "addonVersion": "v25.6.0-eksbuild.1",
}
```

Para autenticação IRSA, utilize o seguinte formato:

```
{
  "clusterName": "<eks-cluster>",
  "addonName": "netapp_trident-operator",
  "addonVersion": "v25.6.0-eksbuild.1",
  "serviceAccountRoleArn": "<role ARN>",
  "configurationValues": {
    "cloudIdentity": "'eks.amazonaws.com/role-arn: <role ARN>'",
    "cloudProvider": "AWS"
  }
}
```



Substituir <role ARN> com o ARN da função que foi criada na etapa anterior.

2. Instale o complemento Trident EKS.

```
aws eks create-addon --cli-input-json file://add-on.json
```

eksctl

O comando de exemplo a seguir instala o complemento Trident EKS:

```
eksctl create addon --name netapp_trident-operator --cluster
<cluster_name> --force
```

Atualize o complemento Trident EKS.

Console de gerenciamento

1. Abra o console do Amazon EKS <https://console.aws.amazon.com/eks/home#/clusters>.
2. No painel de navegação à esquerda, selecione **Clusters**.
3. Selecione o nome do cluster para o qual deseja atualizar o complemento NetApp Trident CSI.
4. Selecione a aba **Complementos**.
5. Selecione * Trident by NetApp* e depois selecione **Editar**.
6. Na página **Configurar Trident by NetApp**, faça o seguinte:
 - a. Selecione a **Versão** que deseja usar.
 - b. Expanda as **Configurações opcionais** e modifique conforme necessário.
 - c. Selecione **Salvar alterações**.

CLI da AWS

O exemplo a seguir atualiza o complemento EKS:

```
aws eks update-addon --cluster-name <eks_cluster_name> --addon-name
netapp_trident-operator --addon-version v25.6.0-eksbuild.1 \
  --service-account-role-arn <role-ARN> --resolve-conflict preserve \
  --configuration-values "{\"cloudIdentity\":
  \"'eks.amazonaws.com/role-arn: <role ARN>'\"}"
```

eksctl

- Verifique a versão atual do seu complemento FSxN Trident CSI. Substituir `my-cluster` com o nome do seu cluster.

```
eksctl get addon --name netapp_trident-operator --cluster my-cluster
```

Exemplo de saída:

NAME	VERSION	STATUS	ISSUES
IAMROLE	UPDATE AVAILABLE	CONFIGURATION VALUES	
netapp_trident-operator	v25.6.0-eksbuild.1	ACTIVE	0
{\"cloudIdentity\":\"'eks.amazonaws.com/role-arn: arn:aws:iam::139763910815:role/AmazonEKS_FSXN_CSI_DriverRole'\"}			

- Atualize o complemento para a versão retornada em ATUALIZAÇÃO DISPONÍVEL na saída da etapa anterior.

```
eksctl update addon --name netapp_trident-operator --version
v25.6.0-eksbuild.1 --cluster my-cluster --force
```

Se você remover o `--force` Se alguma das opções e configurações do complemento do Amazon EKS entrarem em conflito com suas configurações existentes, a atualização do complemento do Amazon EKS falhará e você receberá uma mensagem de erro para ajudá-lo a resolver o conflito. Antes de especificar esta opção, certifique-se de que o complemento Amazon EKS não gerencie configurações que você precisa gerenciar, pois essas configurações serão sobrescritas com esta opção. Para obter mais informações sobre outras opções para esta configuração, consulte "[Complementos](#)". Para obter mais informações sobre o gerenciamento de campos do Kubernetes no Amazon EKS, consulte "[gerenciamento de campos do Kubernetes](#)".

Desinstale/remova o complemento Trident EKS.

Você tem duas opções para remover um complemento do Amazon EKS:

- **Preservar software adicional no seu cluster** – Esta opção remove o gerenciamento de quaisquer configurações pelo Amazon EKS. Isso também remove a capacidade do Amazon EKS de notificá-lo sobre atualizações e de atualizar automaticamente o complemento do Amazon EKS após você iniciar uma atualização. No entanto, isso preserva o software adicional no seu cluster. Essa opção transforma o complemento em uma instalação autogerenciada, em vez de um complemento do Amazon EKS. Com essa opção, não há tempo de inatividade para o complemento. Mantenha o `--preserve` opção no comando para preservar o complemento.
- **Remova completamente o software complementar do seu cluster** – A NetApp recomenda que você remova o complemento Amazon EKS do seu cluster somente se não houver recursos no seu cluster que dependam dele. Remova o `--preserve` opção da `delete` comando para remover o complemento.



Se o complemento tiver uma conta IAM associada a ele, essa conta IAM não será removida.

Console de gerenciamento

1. Abra o console do Amazon EKS em <https://console.aws.amazon.com/eks/home#/clusters>.
2. No painel de navegação à esquerda, selecione **Clusters**.
3. Selecione o nome do cluster para o qual deseja remover o complemento NetApp Trident CSI.
4. Selecione a guia **Complementos** e, em seguida, selecione * Trident by NetApp*.
5. Selecione **Remover**.
6. Na caixa de diálogo **Remover confirmação do netapp_trident-operator**, faça o seguinte:
 - a. Se você deseja que o Amazon EKS pare de gerenciar as configurações do complemento, selecione **Preservar no cluster**. Faça isso se quiser manter o software complementar no seu cluster para poder gerenciar todas as configurações do complemento por conta própria.
 - b. Digite **netapp_trident-operator**.
 - c. Selecione **Remover**.

CLI da AWS

Substituir `my-cluster` com o nome do seu cluster e, em seguida, execute o seguinte comando.

```
aws eks delete-addon --cluster-name my-cluster --addon-name  
netapp_trident-operator --preserve
```

eksctl

O comando a seguir desinstala o complemento Trident EKS:

```
eksctl delete addon --cluster K8s-arm --name netapp_trident-operator
```

Configure o backend de armazenamento

Integração de drivers ONTAP SAN e NAS

Para criar um backend de armazenamento, você precisa criar um arquivo de configuração no formato JSON ou YAML. O arquivo precisa especificar o tipo de armazenamento desejado (NAS ou SAN), o sistema de arquivos, a SVM de onde os dados serão obtidos e como autenticar com ela. O exemplo a seguir mostra como definir o armazenamento baseado em NAS e usar um segredo da AWS para armazenar as credenciais da SVM que você deseja usar:

YAML

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-ontap-nas
  namespace: trident
spec:
  version: 1
  storageDriverName: ontap-nas
  backendName: tbc-ontap-nas
  svm: svm-name
  aws:
    fsxFilesystemID: fs-xxxxxxxxxx
  credentials:
    name: "arn:aws:secretsmanager:us-west-2:xxxxxxx:secret:secret-
name"
    type: awsarn
```

JSON

```
{
  "apiVersion": "trident.netapp.io/v1",
  "kind": "TridentBackendConfig",
  "metadata": {
    "name": "backend-tbc-ontap-nas"
    "namespace": "trident"
  },
  "spec": {
    "version": 1,
    "storageDriverName": "ontap-nas",
    "backendName": "tbc-ontap-nas",
    "svm": "svm-name",
    "aws": {
      "fsxFilesystemID": "fs-xxxxxxxxxx"
    },
    "managementLIF": null,
    "credentials": {
      "name": "arn:aws:secretsmanager:us-west-2:xxxxxxx:secret:secret-
name",
      "type": "awsarn"
    }
  }
}
```

Execute os seguintes comandos para criar e validar a Configuração de Backend do Trident (TBC):

- Crie a configuração de backend do Trident (TBC) a partir do arquivo YAML e execute o seguinte comando:

```
kubectl create -f backendconfig.yaml -n trident
```

```
tridentbackendconfig.trident.netapp.io/backend-tbc-ontap-nas created
```

- Verifique se a configuração do backend Trident (TBC) foi criada com sucesso:

```
Kubectl get tbc -n trident
```

NAME	BACKEND NAME	BACKEND UUID
PHASE	STATUS	
backend-tbc-ontap-nas	tbc-ontap-nas	933e0071-66ce-4324-
b9ff-f96d916ac5e9	Bound	Success

Detalhes do driver FSx para ONTAP

Você pode integrar o Trident com o Amazon FSx for NetApp ONTAP usando os seguintes drivers:

- `ontap-san` Cada PV provisionado é um LUN dentro de seu próprio volume Amazon FSx for NetApp ONTAP . Recomendado para armazenamento em bloco.
- `ontap-nas` Cada PV provisionado é um volume completo do Amazon FSx for NetApp ONTAP . Recomendado para NFS e SMB.
- `ontap-san-economy` Cada PV provisionado é um LUN com um número configurável de LUNs por volume do Amazon FSx for NetApp ONTAP .
- `ontap-nas-economy` Cada PV provisionado é uma qtree, com um número configurável de qtrees por volume do Amazon FSx for NetApp ONTAP .
- `ontap-nas-flexgroup` Cada PV provisionado é um volume FlexGroup completo do Amazon FSx for NetApp ONTAP .

Para obter detalhes sobre o motorista, consulte ["drivers NAS"](#) e ["Drivers SAN"](#) .

Após a criação do arquivo de configuração, execute este comando para criá-lo no seu EKS:

```
kubectl create -f configuration_file
```

Para verificar o status, execute este comando:

```
kubectl get tbc -n trident
```

NAME	BACKEND NAME	BACKEND UUID
PHASE STATUS		
backend-fsx-ontap-nas	backend-fsx-ontap-nas	7a551921-997c-4c37-a1d1-f2f4c87fa629
Bound	Success	

Configuração avançada do backend e exemplos

Consulte a tabela a seguir para obter as opções de configuração do backend:

Parâmetro	Descrição	Exemplo
version		Sempre 1
storageDriverName	Nome do driver de armazenamento	ontap-nas, ontap-nas-economy , ontap-nas-flexgroup , ontap-san , ontap-san-economy
backendName	Nome personalizado ou o backend de armazenamento	Nome do motorista + "_" + 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 configurado para usar endereços IPv6 se o Trident foi instalado usando a opção IPv6. Os endereços IPv6 devem ser definidos entre colchetes, como [28e8:d9fb:a825:b7bf:69a8:d02f:9e7b:3555]. Se você fornecer o fsxFilesystemID sob o aws campo, você não precisa fornecer o managementLIF porque o Trident recupera a SVM managementLIF Informações da AWS. Portanto, você deve fornecer as credenciais de um usuário no SVM (por exemplo: vsadmin) e o usuário deve ter a seguinte permissão: vsadmin papel.	"10.0.0.1", "[2001:1234:abcd::fefe]"

Parâmetro	Descrição	Exemplo
dataLIF	Endereço IP do protocolo LIF. * Drivers ONTAP NAS : A NetApp recomenda especificar dataLIF. Caso não sejam fornecidos, o Trident obtém os dataLIFs da SVM. Você pode especificar um nome de domínio totalmente qualificado (FQDN) para ser usado nas operações de montagem NFS, permitindo criar um DNS round-robin para balancear a carga entre várias dataLIFs. Pode ser alterado após a configuração inicial. Consulte . * Drivers ONTAP SAN: Não especifique para iSCSI. O Trident utiliza o ONTAP Selective LUN Map para descobrir os LIFs iSCSI necessários para estabelecer uma sessão de múltiplos caminhos. Um aviso é gerado se dataLIF for definido explicitamente. Pode ser configurado para usar endereços IPv6 se o Trident foi instalado usando a opção IPv6. Os endereços IPv6 devem ser definidos entre colchetes, como [28e8:d9fb:a825:b7bf:69a8:d02f:9e7b:3555].	
autoExportPolicy	Ativar a criação e atualização automática da política de exportação [Booleano]. Usando o autoExportPolicy e autoExportCIDRs O Trident pode gerenciar políticas de exportação automaticamente, dependendo das opções disponíveis.	false
autoExportCIDRs	Lista de CIDRs para filtrar os IPs dos nós do Kubernetes quando autoExportPolicy está habilitado. Usando o autoExportPolicy e autoExportCIDRs O Trident pode gerenciar políticas de exportação automaticamente, dependendo das opções disponíveis.	"["0.0.0.0/0", "::/0"]"

Parâmetro	Descrição	Exemplo
labels	Conjunto de rótulos arbitrários formatados em JSON para aplicar aos volumes	""
clientCertificate	Valor do certificado do cliente codificado em Base64. Utilizado para autenticação baseada em certificado.	""
clientPrivateKey	Valor da chave privada do cliente codificado em Base64. Utilizado para autenticação baseada em certificado.	""
trustedCACertificate	Valor codificado em Base64 do certificado da Autoridade Certificadora (CA) confiável. Opcional. Utilizado para autenticação baseada em certificado.	""
username	Nome de usuário para conectar-se ao cluster ou SVM. Usado para autenticação baseada em credenciais. Por exemplo, vsadmin.	
password	Senha para conectar-se ao cluster ou SVM. Usado para autenticação baseada em credenciais.	
svm	Máquina virtual de armazenamento para usar	Derivado se um managementLIF de SVM for especificado.
storagePrefix	Prefixo usado ao provisionar novos volumes no SVM. Não pode ser modificado após a criação. Para atualizar esse parâmetro, você precisará criar um novo backend.	trident
limitAggregateUsage	Não especifique para Amazon FSx for NetApp ONTAP. O fornecido fsxadmin e vsadmin Não possuem as permissões necessárias para recuperar o uso agregado e limitá-lo usando o Trident.	Não utilize.
limitVolumeSize	O provisionamento falhará se o tamanho do volume solicitado for superior a este valor. Também restringe o tamanho máximo dos volumes que gerencia para qtrees e LUNs, e o qtreesPerFlexvol Essa opção permite personalizar o número máximo de qtrees por FlexVol volume.	"" (não aplicado por padrão)

Parâmetro	Descrição	Exemplo
<code>lunsPerFlexvol</code>	O número máximo de LUNs por volume Flexvol deve estar no intervalo [50, 200]. Somente SAN.	"100"
<code>debugTraceFlags</code>	Sinalizadores de depuração a serem usados na resolução de problemas. Exemplo: {"api":false, "method":true} Não use <code>debugTraceFlags</code> a menos que você esteja solucionando problemas e precise de um despejo de logs detalhado.	nulo
<code>nfsMountOptions</code>	Lista de opções de montagem NFS separadas por vírgulas. As opções de montagem para volumes persistentes do Kubernetes são normalmente especificadas nas classes de armazenamento, mas se nenhuma opção de montagem for especificada em uma classe de armazenamento, o Trident usará as opções de montagem especificadas no arquivo de configuração do backend de armazenamento. Se nenhuma opção de montagem for especificada na classe de armazenamento ou no arquivo de configuração, o Trident não definirá nenhuma opção de montagem em um volume persistente associado.	""
<code>nasType</code>	Configure a criação de volumes NFS ou SMB. As opções são <code>nfs</code> , <code>smb</code> , ou nulo. Deve ser configurado para <code>smb</code> para volumes SMB. Definir como nulo utiliza, por padrão, volumes NFS.	<code>nfs</code>
<code>qtreesPerFlexvol</code>	Número máximo de Qtrees por FlexVol volume, deve estar no intervalo [50, 300]	"200"

Parâmetro	Descrição	Exemplo
smbShare	Você pode especificar uma das seguintes opções: o nome de um compartilhamento SMB criado usando o Console de Gerenciamento da Microsoft ou a CLI do ONTAP , ou um nome para permitir que o Trident crie o compartilhamento SMB. Este parâmetro é necessário para os backends do Amazon FSx para ONTAP .	smb-share
useREST	Parâmetro booleano para usar APIs REST do ONTAP . Quando definido para <code>true</code> A Trident utilizará APIs REST do ONTAP para se comunicar com o backend. Este recurso requer o ONTAP 9.11.1 e posterior. Além disso, a função de login do ONTAP utilizada deve ter acesso ao <code>ontap</code> aplicativo. Isso é satisfeito pelo predefinido <code>vsadmin</code> e <code>cluster-admin</code> papéis.	false
aws	Você pode especificar o seguinte no arquivo de configuração do AWS FSx para ONTAP: - <code>fsxFilesystemID</code> Especifique o ID do sistema de arquivos AWS FSx. - <code>apiRegion</code> : Nome da região da API da AWS. - <code>apiKey</code> Chave da API da AWS. - <code>secretKey</code> Chave secreta da AWS.	"" "" ""
credentials	Especifique as credenciais do FSx SVM a serem armazenadas no AWS Secrets Manager. - <code>name</code> : Nome de recurso da Amazon (ARN) do segredo, que contém as credenciais do SVM. - <code>type</code> : Definir para <code>awsarn</code> . Consulte "Criar um segredo do AWS Secrets Manager" para mais informações.	

Opções de configuração de backend para provisionamento de volumes

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

Parâmetro	Descrição	Padrão
spaceAllocation	Alocação de espaço para LUNs	true
spaceReserve	Modo de reserva de espaço; "nenhum" (fino) ou "volume" (grosso)	none
snapshotPolicy	Política de instantâneo a ser usada	none
qosPolicy	Grupo de políticas de QoS a ser atribuído aos volumes criados. Escolha uma das opções qosPolicy ou adaptiveQosPolicy por pool de armazenamento ou backend. A utilização de grupos de políticas de QoS com o Trident requer o ONTAP 9.8 ou posterior. Você deve usar um grupo de políticas de QoS não compartilhado e garantir que o grupo de políticas seja aplicado a cada componente individualmente. Um grupo de políticas de QoS compartilhado impõe o limite máximo para a taxa de transferência total de todas as cargas de trabalho.	""
adaptiveQosPolicy	Grupo de políticas de QoS adaptativas a serem atribuídas aos volumes criados. Escolha uma das opções qosPolicy ou adaptiveQosPolicy por pool de armazenamento ou backend. Não suportado por ontap-nas-economy.	""
snapshotReserve	Percentagem do volume reservada para instantâneos "0"	Se snapshotPolicy é none , else ""
splitOnClone	Separar um clone de seu progenitor no momento da criação.	false
encryption	Ative a Criptografia de Volume NetApp (NVE) no novo volume; o padrão é false . Para usar esta opção, o NVE precisa estar licenciado e habilitado no cluster. Se o NAE estiver habilitado no backend, qualquer volume provisionado no Trident terá o NAE habilitado. Para mais informações, consulte: "Como o Trident funciona com NVE e NAE" .	false
luksEncryption	Ative a criptografia LUKS. Consulte "Use o Linux Unified Key Setup (LUKS)" . Somente SAN.	""

Parâmetro	Descrição	Padrão
tieringPolicy	Política de escalonamento a ser usada <code>none</code>	
unixPermissions	Modo para novos volumes. Deixe em branco para volumes SMB.	""
securityStyle	Estilo de segurança para novos volumes. Suporte a NFS <code>mixed</code> e <code>unix</code> estilos de segurança. Suporte para PMEs <code>mixed</code> e <code>ntfs</code> estilos de segurança.	O padrão do NFS é <code>unix</code> . O padrão SMB é <code>ntfs</code> .

Prepare-se para provisionar volumes SMB

Você pode provisionar volumes SMB usando o `ontap-nas` motorista. Antes de concluir [Integração de drivers ONTAP SAN e NAS](#) Complete os seguintes passos.

Antes de começar

Antes de poder provisionar volumes SMB usando o `ontap-nas` Motorista, você deve ter o seguinte.

- Um cluster Kubernetes com um nó controlador Linux e pelo menos um nó de trabalho Windows executando o Windows Server 2019. O Trident suporta volumes SMB montados em pods executados apenas em nós Windows.
- Pelo menos um segredo Trident contendo suas credenciais do Active Directory. Para gerar segredos `smbcreds` :

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

- Um proxy CSI configurado como um serviço do Windows. Para configurar um `csi-proxy` , consulte "[GitHub: Proxy CSI](#)" ou "[GitHub: CSI Proxy para Windows](#)" para nós do Kubernetes executados no Windows.

Passos

1. Criar compartilhamentos SMB. Você pode criar os compartilhamentos administrativos SMB de duas maneiras: usando o "[Console de gerenciamento da Microsoft](#)" Acesse as Pastas Compartilhadas pelo snap-in ou usando a CLI do ONTAP . Para criar compartilhamentos SMB usando a CLI do ONTAP :
 - a. Se necessário, crie a estrutura de diretórios para o compartilhamento.

O `vserver cifs share create` O comando verifica o caminho especificado na opção `-path` durante a criação do compartilhamento. Se o caminho especificado não existir, o comando falhará.

- b. Crie um compartilhamento SMB associado à SVM especificada:

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

c. Verifique se o compartilhamento foi criado:

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



Consulte "[Criar um compartilhamento SMB](#)" Para obter detalhes completos.

2. Ao criar o backend, você deve configurar o seguinte para especificar os volumes SMB. Para todas as opções de configuração do backend FSx para ONTAP , consulte "[Opções e exemplos de configuração do FSx para ONTAP](#)" .

Parâmetro	Descrição	Exemplo
smbShare	Você pode especificar uma das seguintes opções: o nome de um compartilhamento SMB criado usando o Console de Gerenciamento da Microsoft ou a CLI do ONTAP , ou um nome para permitir que o Trident crie o compartilhamento SMB. Este parâmetro é necessário para os backends do Amazon FSx para ONTAP .	smb-share
nasType	Deve ser configurado para smb . Se for nulo, o valor padrão é nfs .	smb
securityStyle	Estilo de segurança para novos volumes. Deve ser configurado para ntfs ou mixed para volumes SMB.	ntfs` ou `mixed para volumes SMB
unixPermissions	Modo para novos volumes. Deve ficar vazio para volumes SMB.	""

Configure uma classe de armazenamento e um PVC.

Configure um objeto StorageClass do Kubernetes e crie a classe de armazenamento para instruir o Trident sobre como provisionar volumes. Crie um PersistentVolumeClaim (PVC) que utilize a StorageClass do Kubernetes configurada para solicitar acesso ao PV. Em seguida, você pode montar o painel fotovoltaico em um suporte.

Criar uma classe de armazenamento

Configure um objeto StorageClass do Kubernetes.

O "[Objeto StorageClass do Kubernetes](#)" O objeto identifica o Trident como o provisionador usado para essa classe e instrui o Trident sobre como provisionar um volume. Use este exemplo para configurar o Storageclass para volumes usando NFS (consulte a seção Atributo Trident abaixo para obter a lista completa de atributos):

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontap-gold
provisioner: csi.trident.netapp.io
parameters:
  backendType: "ontap-nas"
  provisioningType: "thin"
  snapshots: "true"
```

Use este exemplo para configurar o Storageclass para volumes usando iSCSI:

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontap-gold
provisioner: csi.trident.netapp.io
parameters:
  backendType: "ontap-san"
  provisioningType: "thin"
  snapshots: "true"
```

Para provisionar volumes NFSv3 no AWS Bottlerocket, adicione os seguintes itens necessários. mountOptions para a classe de armazenamento:

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontap-gold
provisioner: csi.trident.netapp.io
parameters:
  backendType: "ontap-nas"
  media: "ssd"
  provisioningType: "thin"
  snapshots: "true"
mountOptions:
  - nfsvers=3
  - nolock
```

Consulte ["Objetos Kubernetes e Trident"](#) Para obter detalhes sobre como as classes de armazenamento interagem com o PersistentVolumeClaim e parâmetros para controlar como o Trident provisiona volumes.

Criar uma classe de armazenamento

Passos

1. Este é um objeto do Kubernetes, então use `kubectl` para criá-lo no Kubernetes.

```
kubectl create -f storage-class-ontapas.yaml
```

2. Agora você deverá ver uma classe de armazenamento **basic-csi** tanto no Kubernetes quanto no Trident, e o Trident deverá ter descoberto os pools no backend.

```
kubectl get sc basic-csi
```

NAME	PROVISIONER	AGE
basic-csi	csi.trident.netapp.io	15h

Criar o PVC

Um "[*PersistentVolumeClaim*](#)" (PVC) é uma solicitação de acesso ao PersistentVolume no cluster.

O PVC pode ser configurado para solicitar armazenamento de um determinado tamanho ou modo de acesso. Utilizando a StorageClass associada, o administrador do cluster pode controlar mais do que apenas o tamanho e o modo de acesso do PersistentVolume, como o desempenho ou o nível de serviço.

Após criar o tubo de PVC, você pode montar o volume em um compartimento.

Exemplos de manifestos

Manifestações de exemplo de PersistentVolumeClaim

Estes exemplos mostram opções básicas de configuração de PVC.

PVC com acesso RWX

Este exemplo mostra um PVC básico com acesso RWX associado a uma StorageClass chamada `basic-csi`.

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: pvc-storage
spec:
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 1Gi
  storageClassName: ontap-gold
```

Exemplo de PVC usando iSCSI

Este exemplo mostra um PVC básico para iSCSI com acesso RWO associado a uma StorageClass chamada `protection-gold`.

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: pvc-san
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: protection-gold
```

Criar PVC

Passos

1. Crie o PVC.

```
kubectl create -f pvc.yaml
```

2. Verifique o status do PVC.

```
kubectl get pvc
```

NAME	STATUS	VOLUME	CAPACITY	ACCESS MODES	STORAGECLASS	AGE
pvc-storage	Bound	pv-name	2Gi	RWO		5m

Consulte "[Objetos Kubernetes e Trident](#)" Para obter detalhes sobre como as classes de armazenamento interagem com o `PersistentVolumeClaim` e parâmetros para controlar como o Trident provisiona volumes.

Atributos do Trident

Esses parâmetros determinam quais pools de armazenamento gerenciados pelo Trident devem ser utilizados para provisionar volumes de um determinado tipo.

Atributo	Tipo	Valores	Oferecer	Solicitar	Apoiado por
mídia ¹	corda	HDD, híbrido, SSD	A piscina contém mídias deste tipo; híbrido significa ambos	Tipo de mídia especificado	ontap-nas, ontap-nas-economy, ontap-nas-flexgroup, ontap-san, solidfire-san
tipo de provisionamento	corda	fino, grosso	O Pool suporta este método de provisionamento.	Método de provisionamento especificado	Espesso: tudo Ontap; fino: tudo Ontap e Solidfire-San
backendType	corda	ontap-nas, ontap-nas-economy, ontap-nas-flexgroup, ontap-san, solidfire-san, gcp-cvs, azure-netapp-files, ontap-san-economy	Pool pertence a este tipo de backend	Backend especificado	Todos os motoristas
instantâneos	bool	Verdadeiro, falso	O pool suporta volumes com snapshots.	Volume com snapshots ativados	ontap-nas, ontap-san, solidfire-san, gcp-cvs
clones	bool	Verdadeiro, falso	O Pool suporta a clonagem de volumes.	Volume com clones ativados	ontap-nas, ontap-san, solidfire-san, gcp-cvs

Atributo	Tipo	Valores	Oferecer	Solicitar	Apoiado por
criptografia	bool	Verdadeiro, falso	O Pool suporta volumes criptografados	Volume com criptografia ativada	ontap-nas, ontap-nas-economy, ontap-nas-flexgroups, ontap-san
IOPS	int	inteiro positivo	A Pool é capaz de garantir IOPS nessa faixa.	O volume garante esses IOPS.	solidfire-san

¹: Não suportado pelos sistemas ONTAP Select

Implantar aplicação de exemplo

Após a criação da classe de armazenamento e do PVC, você pode montar o PV em um pod. Esta seção lista os comandos e configurações de exemplo para anexar o PV a um pod.

Passos

1. Instale o volume em um pod.

```
kubectl create -f pv-pod.yaml
```

Estes exemplos mostram configurações básicas para conectar o PVC a um pod: **Configuração básica:**

```
kind: Pod
apiVersion: v1
metadata:
  name: pv-pod
spec:
  volumes:
    - name: pv-storage
      persistentVolumeClaim:
        claimName: basic
  containers:
    - name: pv-container
      image: nginx
      ports:
        - containerPort: 80
          name: "http-server"
      volumeMounts:
        - mountPath: "/my/mount/path"
          name: pv-storage
```



Você pode monitorar o progresso usando `kubectl get pod --watch`.

2. Verifique se o volume está montado em `/my/mount/path`.

```
kubectl exec -it pv-pod -- df -h /my/mount/path
```

Filesystem	Size
Used Avail Use% Mounted on	
192.168.188.78:/trident_pvc_ae45ed05_3ace_4e7c_9080_d2a83ae03d06	1.1G
320K 1.0G 1% /my/mount/path	

Agora você pode excluir o Pod. O aplicativo Pod deixará de existir, mas o volume permanecerá.

```
kubectl delete pod pv-pod
```

Configure o complemento Trident EKS em um cluster EKS.

O NetApp Trident simplifica o gerenciamento de armazenamento do Amazon FSx for NetApp ONTAP no Kubernetes, permitindo que seus desenvolvedores e administradores se concentrem na implantação de aplicativos. O complemento NetApp Trident EKS inclui os patches de segurança e correções de bugs mais recentes, e é validado pela AWS para funcionar com o Amazon EKS. O complemento EKS permite garantir de forma consistente que seus clusters Amazon EKS estejam seguros e estáveis, além de reduzir o trabalho necessário para instalar, configurar e atualizar complementos.

Pré-requisitos

Certifique-se de ter o seguinte antes de configurar o complemento Trident para AWS EKS:

- Uma conta de cluster Amazon EKS com permissões para trabalhar com complementos. Consulte ["Complementos do Amazon EKS"](#).
- Permissões da AWS para o marketplace da AWS:
"aws-marketplace:ViewSubscriptions",
"aws-marketplace:Subscribe",
"aws-marketplace:Unsubscribe"
- Tipo de AMI: Amazon Linux 2 (AL2_x86_64) ou Amazon Linux 2 Arm (AL2_ARM_64)
- Tipo de nó: AMD ou ARM
- Um sistema de arquivos Amazon FSx for NetApp ONTAP

Passos

1. Certifique-se de criar uma função do IAM e um segredo da AWS para permitir que os pods do EKS acessem os recursos da AWS. Para obter instruções, consulte ["Crie uma função do IAM e um segredo da AWS."](#)

2. No seu cluster Kubernetes EKS, navegue até a guia **Complementos**.

The screenshot shows the AWS EKS console interface for a cluster named 'tri-env-eks'. At the top, there are buttons for 'Delete cluster', 'Upgrade version', and 'View dashboard'. A notification banner at the top states: 'End of standard support for Kubernetes version 1.30 is July 28, 2025. On that date, your cluster will enter the extended support period with additional fees. For more information, see the [pricing page](#).' Below this, the 'Cluster info' section displays: Status (Active), Kubernetes version (1.30), Support period (Standard support until July 28, 2025), and Provider (EKS). It also shows 'Cluster health issues' and 'Upgrade insights', both with a green checkmark and '0'. A navigation bar includes tabs for Overview, Resources, Compute, Networking, Add-ons (selected), Access, Observability, Update history, and Tags. A message states: 'New versions are available for 1 add-on.' The 'Add-ons (3)' section includes a search bar, filters for 'Any category', 'Any status', and '3 matches', along with buttons for 'View details', 'Edit', 'Remove', and 'Get more add-ons'.

3. Acesse os complementos do **AWS Marketplace** e escolha a categoria *armazenamento*.

The screenshot displays the 'AWS Marketplace add-ons (1)' page. It includes a search bar with the placeholder 'Find add-on' and filtering options for 'Any category', 'NetApp, Inc.', 'Any pricing model', and a 'Clear filters' button. A breadcrumb shows 'NetApp, Inc.' with a close button. The main content area features the 'NetApp Trident' add-on. It includes the NetApp logo, a description: 'NetApp Trident streamlines Amazon FSx for NetApp ONTAP storage management in Kubernetes to let your developers and administrators focus on application deployment. FSx for ONTAP flexibility, scalability, and integration capabilities make it the ideal choice for organizations seeking efficient containerized storage workflows. [Product details](#)', a 'Standard Contract' badge, and a table of details: Category (storage), Listed by (NetApp, Inc.), Supported versions (1.31, 1.30, 1.29, 1.28, 1.27, 1.26, 1.25, 1.24, 1.23), and Pricing starting at (View pricing details). At the bottom right, there are 'Cancel' and 'Next' buttons.

4. Localize * NetApp Trident*, selecione a caixa de seleção do complemento Trident e clique em **Avançar**.

5. Escolha a versão desejada do complemento.

Configure selected add-ons settings


Configure the add-ons for your cluster by selecting settings.

NetApp TridentRemove add-on

Listed by

Category

Status



storage

Ready to install

You're subscribed to this software

You can view the terms and pricing details for this product or choose another offer if one is available.

View subscription

Version

Select the version for this add-on.

v25.6.0-eksbuild.1

Optional configuration settings

Cancel

Previous

Next

6. Configure as configurações necessárias do complemento.

Review and add

Step 1: Select add-ons

[Edit](#)

Selected add-ons (1)

Find add-on

< 1 >

Add-on name	Type	Status
netapp_trident-operator	storage	Ready to install

Step 2: Configure selected add-ons settings

[Edit](#)

Selected add-ons version (1)

< 1 >

Add-on name	Version	IAM role for service account (IRSA)
netapp_trident-operator	v24.10.0-eksbuild.1	Not set

EKS Pod Identity (0)

< 1 >

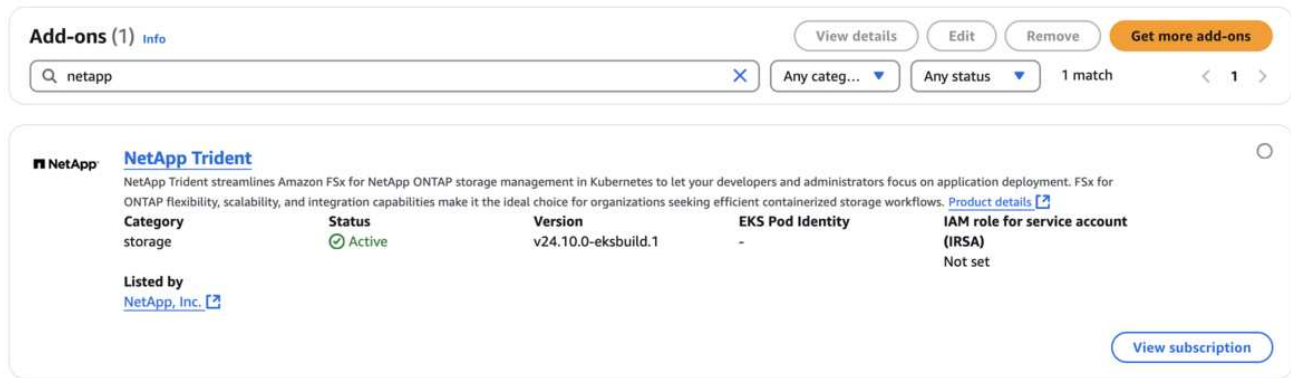
Add-on name	IAM role	Service account
No Pod Identity associations None of the selected add-on(s) have Pod Identity associations.		

[Cancel](#)[Previous](#)[Create](#)

7. Se você estiver usando o IRSA (funções do IAM para conta de serviço), consulte as etapas de configuração adicionais. ["aqui"](#).

8. Selecione **Criar**.

9. Verifique se o status do complemento é *Ativo*.



The screenshot shows the AWS EKS console 'Add-ons' page. At the top, there's a search bar with 'netapp' and a '1 match' indicator. Below the search bar, a table lists the installed add-ons. The first entry is 'NetApp Trident', which is in the 'Active' status. The table also shows the category as 'storage', the version as 'v24.10.0-eksbuild.1', and the IAM role for the service account as 'Not set'. There are buttons for 'View details', 'Edit', 'Remove', and 'Get more add-ons' at the top right. A 'View subscription' button is at the bottom right of the add-on entry.

Category	Status	Version	EKS Pod Identity	IAM role for service account (IRSA)
storage	Active	v24.10.0-eksbuild.1	-	Not set

10. Execute o seguinte comando para verificar se o Trident está instalado corretamente no cluster:

```
kubectl get pods -n trident
```

11. Continue a configuração e configure o backend de armazenamento. Para obter informações, consulte "[Configure o backend de armazenamento](#)".

Instale/desinstale o complemento Trident EKS usando a CLI.

Instale o complemento NetApp Trident EKS usando a CLI:

O comando de exemplo a seguir instala o complemento Trident EKS:

```
eksctl create addon --cluster clusterName --name netapp_trident-operator  
--version v25.6.0-eksbuild.1 (com uma versão dedicada)
```

Desinstale o complemento NetApp Trident EKS usando a CLI:

O comando a seguir desinstala o complemento Trident EKS:

```
eksctl delete addon --cluster K8s-arm --name netapp_trident-operator
```

Crie backends com kubectl

Um backend define a relação entre o Trident e um sistema de armazenamento. Ele informa ao Trident como se comunicar com esse sistema de armazenamento e como o Trident deve provisionar volumes a partir dele. Após a instalação do Trident, o próximo passo é criar um backend. O `TridentBackendConfig` A Definição de Recurso Personalizado (CRD) permite criar e gerenciar backends do Trident diretamente através da interface do Kubernetes. Você pode fazer isso usando `kubectl` ou a ferramenta de linha de comando equivalente para sua distribuição Kubernetes.

`TridentBackendConfig`

`TridentBackendConfig` (`tbc`, `tbconfig`, `tbackendconfig`) é um CRD de front-end com namespace que permite gerenciar back-ends do Trident usando `kubectl`. Administradores de Kubernetes e de armazenamento agora podem criar e gerenciar backends diretamente por meio da CLI do Kubernetes, sem a

necessidade de um utilitário de linha de comando dedicado.(`tridentctl`).

Ao criar um `TridentBackendConfig` objeto, o seguinte acontece:

- Um ambiente de backend é criado automaticamente pelo Trident com base na configuração fornecida. Isso é representado internamente como um `TridentBackend` (`tbe` , `tridentbackend`) CR.
- O `TridentBackendConfig` está exclusivamente ligado a um `TridentBackend` que foi criado pela Trident.

Cada `TridentBackendConfig` mantém um mapeamento um-para-um com um `TridentBackend` A primeira é a interface fornecida ao usuário para projetar e configurar backends; a segunda é como o Trident representa o objeto backend propriamente dito.



`TridentBackend``Os CRs são criados automaticamente pelo Trident. Você **não deve** modificá-los. Se você deseja fazer atualizações nos backends, faça isso modificando o `TridentBackendConfig` objeto.

Veja o exemplo a seguir para o formato do `TridentBackendConfig` CR:

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-ontap-san
spec:
  version: 1
  backendName: ontap-san-backend
  storageDriverName: ontap-san
  managementLIF: 10.0.0.1
  dataLIF: 10.0.0.2
  svm: trident_svm
  credentials:
    name: backend-tbc-ontap-san-secret
```

Você também pode conferir os exemplos em "[instalador de tridente](#)" Diretório com exemplos de configurações para a plataforma/serviço de armazenamento desejado.

O `spec` Utiliza parâmetros de configuração específicos do backend. Neste exemplo, o backend usa o `ontap-san` O driver de armazenamento utiliza os parâmetros de configuração que estão tabelados aqui. Para obter a lista de opções de configuração para o driver de armazenamento desejado, consulte o "[Informações de configuração do backend para o seu driver de armazenamento](#)".

O `spec` esta seção também inclui `credentials` e `deletionPolicy` campos, que são recentemente introduzidos no `TridentBackendConfig` CR:

- `credentials``Este parâmetro é um campo obrigatório e contém as credenciais usadas para autenticar com o sistema/serviço de armazenamento. Isso é definido como um segredo do Kubernetes criado pelo usuário. As credenciais não podem ser transmitidas em texto sem formatação e resultarão em um erro.
- `deletionPolicy``Este campo define o que deve acontecer quando o

``TridentBackendConfig` foi apagado. Pode assumir um de dois valores possíveis:

- `delete` Isso resulta na exclusão de ambos. ``TridentBackendConfig` CR e o backend associado. Este é o valor padrão.
- `retain`: Quando um `TridentBackendConfig` O CR é excluído, a definição do backend ainda estará presente e poderá ser gerenciada com `tridentctl`. Definir a política de exclusão para `retain` Permite que os usuários façam o downgrade para uma versão anterior (anterior à 21.04) e mantenham os backends criados. O valor deste campo pode ser atualizado após um `TridentBackendConfig` é criado.



O nome de um backend é definido usando `spec.backendName`. Se não for especificado, o nome do backend será definido como o nome do `TridentBackendConfig` objeto (nome.metadados). É recomendável definir explicitamente os nomes de backend usando `spec.backendName`.



Backends que foram criados com `tridentctl` não possuem um associado `TridentBackendConfig` objeto. Você pode optar por gerenciar esses back-ends com `kubectl` criando um `TridentBackendConfig` CR. É preciso ter cuidado para especificar parâmetros de configuração idênticos (como, por exemplo, `spec.backendName`, `spec.storagePrefix`, `spec.storageDriverName`, e assim por diante). O Trident irá vincular automaticamente o dispositivo recém-criado. `TridentBackendConfig` com o backend pré-existente.

Visão geral das etapas

Para criar um novo backend usando `kubectl` Você deve fazer o seguinte:

1. Criar um **"Segredo do Kubernetes"** O segredo contém as credenciais que o Trident precisa para se comunicar com o cluster/serviço de armazenamento.
2. Criar um `TridentBackendConfig` objeto. Esta seção contém detalhes específicos sobre o cluster/serviço de armazenamento e faz referência ao segredo criado na etapa anterior.

Após criar um backend, você pode observar seu status usando `kubectl get tbc <tbc-name> -n <trident-namespace>` e coletar detalhes adicionais.

Passo 1: Crie um segredo do Kubernetes

Crie um segredo que contenha as credenciais de acesso ao backend. Isso é específico para cada serviço/plataforma de armazenamento. Aqui está um exemplo:

```
kubectl -n trident create -f backend-tbc-ontap-san-secret.yaml
```

```

apiVersion: v1
kind: Secret
metadata:
  name: backend-tbc-ontap-san-secret
type: Opaque
stringData:
  username: cluster-admin
  password: password

```

Esta tabela resume os campos que devem ser incluídos no Segredo para cada plataforma de armazenamento:

Descrição dos campos secretos da plataforma de armazenamento	Segredo	Descrição dos campos
Azure NetApp Files	ID do cliente	O ID do cliente obtido durante o cadastro no aplicativo.
Cloud Volumes Service para GCP	id_da_chave_privada	ID da chave privada. Parte da chave de API para conta de serviço do GCP com função de administrador do CVS
Cloud Volumes Service para GCP	chave_privada	Chave privada. Parte da chave de API para conta de serviço do GCP com função de administrador do CVS
Elemento (NetApp HCI/ SolidFire)	Ponto final	MVIP para o cluster SolidFire com credenciais de locatário
ONTAP	nome de usuário	Nome de usuário para conectar ao cluster/SVM. Utilizado para autenticação baseada em credenciais.
ONTAP	senha	Senha para conectar ao cluster/SVM. Utilizado para autenticação baseada em credenciais.
ONTAP	chavePrivadaDoCliente	Valor da chave privada do cliente codificado em Base64. Utilizado para autenticação baseada em certificado.

Descrição dos campos secretos da plataforma de armazenamento	Segredo	Descrição dos campos
ONTAP	chapUsername	Nome de usuário de entrada. Obrigatório se useCHAP=true. Para ontap-san e ontap-san-economy
ONTAP	chapIniciadorSecreto	Segredo do iniciador CHAP. Obrigatório se useCHAP=true. Para ontap-san e ontap-san-economy
ONTAP	chapTargetUsername	Nome de usuário alvo. Obrigatório se useCHAP=true. Para ontap-san e ontap-san-economy
ONTAP	chapTargetInitiatorSecret	Segredo iniciador do alvo CHAP. Obrigatório se useCHAP=true. Para ontap-san e ontap-san-economy

O segredo criado nesta etapa será referenciado em `spec.credentials` campo do `TridentBackendConfig` objeto que será criado na próxima etapa.

Passo 2: Crie o `TridentBackendConfig` CR

Agora você está pronto para criar o seu `TridentBackendConfig` CR. Neste exemplo, um backend que usa o `ontap-san` O driver é criado usando o `TridentBackendConfig` Objeto mostrado abaixo:

```
kubectl -n trident create -f backend-tbc-ontap-san.yaml
```

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-ontap-san
spec:
  version: 1
  backendName: ontap-san-backend
  storageDriverName: ontap-san
  managementLIF: 10.0.0.1
  dataLIF: 10.0.0.2
  svm: trident_svm
  credentials:
    name: backend-tbc-ontap-san-secret
```

Etapa 3: Verifique o status do TridentBackendConfig CR

Agora que você criou o TridentBackendConfig CR, você pode verificar o status. Veja o exemplo a seguir:

```
kubectl -n trident get tbc backend-tbc-ontap-san
NAME                                BACKEND NAME                                BACKEND UUID
PHASE    STATUS
backend-tbc-ontap-san    ontap-san-backend    8d24fce7-6f60-4d4a-8ef6-
bab2699e6ab8    Bound    Success
```

Um backend foi criado e vinculado com sucesso ao TridentBackendConfig CR.

A fase pode assumir um dos seguintes valores:

- **Bound:** O TridentBackendConfig O CR está associado a um backend, e esse backend contém configRef definido para o TridentBackendConfig UID do CR.
- **Unbound** Representado usando `""`. O TridentBackendConfig O objeto não está vinculado a um backend. Tudo recém-criado TridentBackendConfig Os CRs estão nessa fase por padrão. Após a mudança de fase, não é possível reverter para o estado Não Vinculado novamente.
- **Deleting:** O TridentBackendConfig CR's deletionPolicy estava configurado para excluir. Quando o TridentBackendConfig Quando o CR é excluído, ele entra no estado de exclusão.
 - Se não existirem reivindicações de volume persistentes (PVCs) no backend, exclua o TridentBackendConfig resultará na exclusão do backend e também do Trident . TridentBackendConfig CR.
 - Se um ou mais PVCs estiverem presentes no backend, ele entra em estado de exclusão. O TridentBackendConfig Posteriormente, o CR também entra na fase de deleção. O backend e TridentBackendConfig São apagados somente depois que todos os PVCs forem apagados.
- **Lost** O backend associado ao `TridentBackendConfig CR foi apagado acidentalmente ou deliberadamente e o TridentBackendConfig O CR ainda possui uma referência ao backend excluído. O TridentBackendConfig O CR ainda pode ser excluído independentemente do deletionPolicy valor.
- **Unknown** O Trident não consegue determinar o estado ou a existência do backend associado ao `TridentBackendConfig CR. Por exemplo, se o servidor da API não estiver respondendo ou se o tridentbackends.trident.netapp.io O CRD está faltando. Isso pode exigir intervenção.

Nesta etapa, o backend foi criado com sucesso! Existem diversas operações adicionais que podem ser realizadas, tais como: [atualizações e exclusões de backend](#) .

(Opcional) Passo 4: Obtenha mais detalhes

Você pode executar o seguinte comando para obter mais informações sobre seu backend:

```
kubectl -n trident get tbc backend-tbc-ontap-san -o wide
```

NAME	BACKEND NAME	BACKEND UUID
PHASE STATUS STORAGE DRIVER DELETION POLICY		
backend-tbc-ontap-san	ontap-san-backend	8d24fce7-6f60-4d4a-8ef6-bab2699e6ab8
Bound Success	ontap-san	delete

Além disso, você também pode obter um dump YAML/JSON de `TridentBackendConfig`.

```
kubectl -n trident get tbc backend-tbc-ontap-san -o yaml
```

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  creationTimestamp: 2021-04-21T20:45:11Z
  finalizers:
    - trident.netapp.io
  generation: 1
  name: backend-tbc-ontap-san
  namespace: trident
  resourceVersion: "947143"
  uid: 35b9d777-109f-43d5-8077-c74a4559d09c
spec:
  backendName: ontap-san-backend
  credentials:
    name: backend-tbc-ontap-san-secret
  managementLIF: 10.0.0.1
  dataLIF: 10.0.0.2
  storageDriverName: ontap-san
  svm: trident_svm
  version: 1
status:
  backendInfo:
    backendName: ontap-san-backend
    backendUUID: 8d24fce7-6f60-4d4a-8ef6-bab2699e6ab8
  deletionPolicy: delete
  lastOperationStatus: Success
  message: Backend 'ontap-san-backend' created
  phase: Bound
```

`backendInfo` contém o `backendName` e o `backendUUID` do backend que foi criado em resposta ao `TridentBackendConfig` CR. O `lastOperationStatus` O campo representa o status da última operação do `TridentBackendConfig` CR, que pode ser acionada pelo usuário (por exemplo, o usuário alterou algo em `spec`) ou acionado pelo Trident (por exemplo, durante reinicializações do Trident). Pode ser sucesso ou fracasso. `phase` representa o estado da relação entre o `TridentBackendConfig` CR e o backend. No

exemplo acima, `phase` tem o valor `Bound`, o que significa que o `TridentBackendConfig` CR está associado ao backend.

Você pode executar o `kubectl -n trident describe tbc <tbc-cr-name>` comando para obter detalhes dos registros de eventos.



Não é possível atualizar ou excluir um backend que contenha um associado. `TridentBackendConfig` objeto usando `tridentctl`. Para entender os passos envolvidos na transição entre `tridentctl` e `TridentBackendConfig`, [veja aqui](#).

Gerenciar back-ends

Realize o gerenciamento de backend com kubectl

Aprenda como executar operações de gerenciamento de back-end usando `kubectl`.

Excluir um backend

Ao excluir um `TridentBackendConfig`, você instrui o Trident a excluir/reter backends (com base em `deletionPolicy`). Para excluir um backend, certifique-se de que `deletionPolicy` está configurado para excluir. Para excluir apenas o `TridentBackendConfig`, assegure-se de que `deletionPolicy` está definido para manter. Isso garante que o backend ainda esteja presente e possa ser gerenciado usando `tridentctl`.

Execute o seguinte comando:

```
kubectl delete tbc <tbc-name> -n trident
```

O Trident não exclui os segredos do Kubernetes que estavam em uso pelo `TridentBackendConfig`. O usuário do Kubernetes é responsável por limpar os segredos. É preciso ter cuidado ao apagar segredos. Você deve excluir segredos somente se eles não estiverem sendo usados pelos servidores de backend.

Veja os backends existentes

Execute o seguinte comando:

```
kubectl get tbc -n trident
```

Você também pode executar `tridentctl get backend -n trident` ou `tridentctl get backend -o yaml -n trident` Para obter uma lista de todos os backends existentes. Esta lista também incluirá backends que foram criados com `tridentctl`.

Atualizar um backend

Existem diversos motivos para atualizar um backend:

- As credenciais de acesso ao sistema de armazenamento foram alteradas. Para atualizar as credenciais, o segredo do Kubernetes que é usado no `TridentBackendConfig` O objeto precisa ser atualizado. O Trident atualizará automaticamente o backend com as credenciais mais recentes fornecidas. Execute o

seguinte comando para atualizar o segredo do Kubernetes:

```
kubectl apply -f <updated-secret-file.yaml> -n trident
```

- É necessário atualizar os parâmetros (como o nome da SVM do ONTAP que está sendo usada).
 - Você pode atualizar `TridentBackendConfig` objetos diretamente através do Kubernetes usando o seguinte comando:

```
kubectl apply -f <updated-backend-file.yaml>
```

- Alternativamente, você pode fazer alterações no existente. `TridentBackendConfig` CR usando o seguinte comando:

```
kubectl edit tbc <tbc-name> -n trident
```



- Se uma atualização do backend falhar, o backend permanecerá em sua última configuração conhecida. Você pode visualizar os registros para determinar a causa executando o seguinte comando: `kubectl get tbc <tbc-name> -o yaml -n trident` ou `kubectl describe tbc <tbc-name> -n trident`.
- Após identificar e corrigir o problema com o arquivo de configuração, você pode executar o comando de atualização novamente.

Realize a gestão de backend com o `tridentctl`

Aprenda como executar operações de gerenciamento de back-end usando `tridentctl`.

Crie um backend

Depois de criar um "arquivo de configuração do backend" Execute o seguinte comando:

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

Se a criação do backend falhar, algo estava errado com a configuração do backend. Você pode visualizar os registros para determinar a causa executando o seguinte comando:

```
tridentctl logs -n trident
```

Após identificar e corrigir o problema com o arquivo de configuração, você pode simplesmente executar o `create` Comande novamente.

Excluir um backend

Para excluir um backend do Trident, faça o seguinte:

1. Recuperar o nome do backend:

```
tridentctl get backend -n trident
```

2. Exclua o backend:

```
tridentctl delete backend <backend-name> -n trident
```



Se o Trident tiver provisionado volumes e snapshots desse backend que ainda existam, a exclusão do backend impede que novos volumes sejam provisionados por ele. O sistema de backend continuará existindo em um estado de "Exclusão".

Veja os backends existentes

Para visualizar os backends que o Trident conhece, faça o seguinte:

- Para obter um resumo, execute o seguinte comando:

```
tridentctl get backend -n trident
```

- Para obter todos os detalhes, execute o seguinte comando:

```
tridentctl get backend -o json -n trident
```

Atualizar um backend

Após criar um novo arquivo de configuração de backend, execute o seguinte comando:

```
tridentctl update backend <backend-name> -f <backend-file> -n trident
```

Se a atualização do servidor falhar, algo estava errado com a configuração do servidor ou você tentou uma atualização inválida. Você pode visualizar os registros para determinar a causa executando o seguinte comando:

```
tridentctl logs -n trident
```

Após identificar e corrigir o problema com o arquivo de configuração, você pode simplesmente executar o `update` Comande novamente.

Identifique as classes de armazenamento que utilizam um backend.

Este é um exemplo do tipo de pergunta que você pode responder com o JSON que `tridentctl` Saídas para objetos de backend. Isso usa o `jq` utilitário, que você precisa instalar.

```
tridentctl get backend -o json | jq '[.items[] | {backend: .name,
storageClasses: [.storage[].storageClasses]|unique}]'
```

Isso também se aplica a backends que foram criados usando `TridentBackendConfig`.

Alternar entre opções de gerenciamento de back-end

Aprenda sobre as diferentes maneiras de gerenciar back-ends no Trident.

Opções para gerenciar back-ends

Com a introdução de `TridentBackendConfig`, os administradores agora têm duas maneiras exclusivas de gerenciar os sistemas de back-end. Isso levanta as seguintes questões:

- É possível criar back-ends usando `tridentctl` ser gerenciado com `TridentBackendConfig`?
- É possível criar back-ends usando `TridentBackendConfig` ser gerenciado usando `tridentctl`?

Gerenciar `tridentctl` backends usando `TridentBackendConfig`

Esta seção aborda os passos necessários para gerenciar backends criados usando `tridentctl` diretamente através da interface do Kubernetes, criando `TridentBackendConfig` objetos.

Isso se aplica aos seguintes cenários:

- Backends pré-existentes que não possuem um `TridentBackendConfig` porque foram criados com `tridentctl`.
- Novos backends que foram criados com `tridentctl`, enquanto outros `TridentBackendConfig` Os objetos existem.

Em ambos os cenários, os servidores de backend continuarão presentes, com o Trident agendando volumes e operando sobre eles. Os administradores têm duas opções aqui:

- Continuar usando `tridentctl` para gerenciar os backends que foram criados usando-o.
- Vincule os backends criados usando `tridentctl` para um novo `TridentBackendConfig` objeto. Fazer isso significaria que os back-ends seriam gerenciados usando `kubectl` e não `tridentctl`.

Para gerenciar um backend pré-existente usando `kubectl`, você precisará criar um `TridentBackendConfig` que se integra ao backend existente. Segue abaixo uma visão geral de como isso funciona:

1. Criar um segredo do Kubernetes. O segredo contém as credenciais que o Trident precisa para se comunicar com o cluster/serviço de armazenamento.
2. Criar um `TridentBackendConfig` objeto. Esta seção contém detalhes específicos sobre o cluster/serviço de armazenamento e faz referência ao segredo criado na etapa anterior. É preciso ter

cuidado para especificar parâmetros de configuração idênticos (como, por exemplo, `spec.backendName`, `spec.storagePrefix`, `spec.storageDriverName`, e assim por diante). `spec.backendName` Deve ser definido com o nome do backend existente.

Etapa 0: Identificar o backend

Para criar um `TridentBackendConfig` Para se conectar a um backend existente, você precisará obter a configuração do backend. Neste exemplo, vamos supor que um backend foi criado usando a seguinte definição JSON:

```
tridentctl get backend ontap-nas-backend -n trident
+-----+-----+
+-----+-----+-----+-----+
|          NAME          | STORAGE DRIVER |          UUID          |
| STATE  | VOLUMES |          |
+-----+-----+-----+-----+
+-----+-----+-----+-----+
| ontap-nas-backend      | ontap-nas      | 52f2eb10-e4c6-4160-99fc-
96b3be5ab5d7 | online |          25 |
+-----+-----+-----+-----+
+-----+-----+-----+-----+
```

```
cat ontap-nas-backend.json
```

```

{
  "version": 1,
  "storageDriverName": "ontap-nas",
  "managementLIF": "10.10.10.1",
  "dataLIF": "10.10.10.2",
  "backendName": "ontap-nas-backend",
  "svm": "trident_svm",
  "username": "cluster-admin",
  "password": "admin-password",
  "defaults": {
    "spaceReserve": "none",
    "encryption": "false"
  },
  "labels": {
    "store": "nas_store"
  },
  "region": "us_east_1",
  "storage": [
    {
      "labels": {
        "app": "msoffice",
        "cost": "100"
      },
      "zone": "us_east_1a",
      "defaults": {
        "spaceReserve": "volume",
        "encryption": "true",
        "unixPermissions": "0755"
      }
    },
    {
      "labels": {
        "app": "mysqldb",
        "cost": "25"
      },
      "zone": "us_east_1d",
      "defaults": {
        "spaceReserve": "volume",
        "encryption": "false",
        "unixPermissions": "0775"
      }
    }
  ]
}

```

Passo 1: Crie um segredo do Kubernetes

Crie um segredo que contenha as credenciais para o backend, conforme mostrado neste exemplo:

```
cat tbc-ontap-nas-backend-secret.yaml
```

```
apiVersion: v1
kind: Secret
metadata:
  name: ontap-nas-backend-secret
type: Opaque
stringData:
  username: cluster-admin
  password: admin-password
```

```
kubectl create -f tbc-ontap-nas-backend-secret.yaml -n trident
secret/backend-tbc-ontap-san-secret created
```

Passo 2: Crie um TridentBackendConfig CR

O próximo passo é criar um `TridentBackendConfig` CR que se vinculará automaticamente ao preexistente `ontap-nas-backend` (como neste exemplo). Assegure-se de que os seguintes requisitos sejam atendidos:

- O mesmo nome de backend está definido em `spec.backendName`.
- Os parâmetros de configuração são idênticos aos do backend original.
- Os pools virtuais (se presentes) devem manter a mesma ordem que no backend original.
- As credenciais são fornecidas por meio de um segredo do Kubernetes e não em texto simples.

Neste caso, o `TridentBackendConfig` Vai ficar assim:

```
cat backend-tbc-ontap-nas.yaml
```

```

apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: tbc-ontap-nas-backend
spec:
  version: 1
  storageDriverName: ontap-nas
  managementLIF: 10.10.10.1
  dataLIF: 10.10.10.2
  backendName: ontap-nas-backend
  svm: trident_svm
  credentials:
    name: mysecret
  defaults:
    spaceReserve: none
    encryption: 'false'
  labels:
    store: nas_store
    region: us_east_1
  storage:
  - labels:
      app: msoffice
      cost: '100'
      zone: us_east_1a
      defaults:
        spaceReserve: volume
        encryption: 'true'
        unixPermissions: '0755'
  - labels:
      app: mysqlldb
      cost: '25'
      zone: us_east_1d
      defaults:
        spaceReserve: volume
        encryption: 'false'
        unixPermissions: '0775'

```

```

kubectl create -f backend-tbc-ontap-nas.yaml -n trident
tridentbackendconfig.trident.netapp.io/tbc-ontap-nas-backend created

```

Etapas 3: Verifique o status do TridentBackendConfig CR

Depois do TridentBackendConfig foi criada, sua fase deve ser Bound . Deve também refletir o mesmo nome de backend e UUID do backend existente.

```
kubectl get tbc tbc-ontap-nas-backend -n trident
```

NAME	BACKEND NAME	BACKEND UUID
tbc-ontap-nas-backend	ontap-nas-backend	52f2eb10-e4c6-4160-99fc-96b3be5ab5d7

```
tridentctl get backend -n trident
```

NAME	STORAGE DRIVER	UUID
ontap-nas-backend	ontap-nas	52f2eb10-e4c6-4160-99fc-96b3be5ab5d7

O backend agora será totalmente gerenciado usando o tbc-ontap-nas-backend TridentBackendConfig objeto.

Gerenciar TridentBackendConfig **backends usando** tridentctl

`tridentctl` pode ser usado para listar backends que foram criados usando `TridentBackendConfig`. Além disso, os administradores também podem optar por gerenciar completamente esses back-ends por meio de `tridentctl` ao excluir `TridentBackendConfig` e garantindo `spec.deletionPolicy` está definido para `retain`.

Etapa 0: Identificar o backend

Por exemplo, vamos supor que o seguinte backend foi criado usando TridentBackendConfig:

```
kubectl get tbc backend-tbc-ontap-san -n trident -o wide
```

NAME	BACKEND NAME	BACKEND UUID
backend-tbc-ontap-san	ontap-san-backend	81abcb27-ea63-49bb-b606-0a5315ac5f82

```
tridentctl get backend ontap-san-backend -n trident
```

NAME	STORAGE DRIVER	UUID
ontap-san-backend	ontap-san	81abcb27-ea63-49bb-b606-0a5315ac5f82

A partir dos resultados, observa-se que TridentBackendConfig Foi criado com sucesso e está vinculado a um backend [observe o UUID do backend].

Passo 1: Confirmar deletionPolicy está definido para retain

Vamos analisar o valor de deletionPolicy . Isso precisa ser configurado para retain . Isso garante que quando um TridentBackendConfig O CR é excluído, mas a definição do backend ainda estará presente e poderá ser gerenciada com tridentctl .

```
kubectl get tbc backend-tbc-ontap-san -n trident -o wide
```

NAME	BACKEND NAME	BACKEND UUID
backend-tbc-ontap-san	ontap-san-backend	81abcb27-ea63-49bb-b606-0a5315ac5f82

```
# Patch value of deletionPolicy to retain
kubectl patch tbc backend-tbc-ontap-san --type=merge -p
'{"spec":{"deletionPolicy":"retain"}}' -n trident
tridentbackendconfig.trident.netapp.io/backend-tbc-ontap-san patched

#Confirm the value of deletionPolicy
kubectl get tbc backend-tbc-ontap-san -n trident -o wide
```

NAME	BACKEND NAME	BACKEND UUID
backend-tbc-ontap-san	ontap-san-backend	81abcb27-ea63-49bb-b606-0a5315ac5f82



Não prossiga para a próxima etapa a menos que deletionPolicy está definido para retain

Passo 2: Exclua o TridentBackendConfig CR

O passo final é excluir o TridentBackendConfig CR. Após confirmar o deletionPolicy está definido para retain Você pode prosseguir com a exclusão:

```
kubectl delete tbc backend-tbc-ontap-san -n trident
tridentbackendconfig.trident.netapp.io "backend-tbc-ontap-san" deleted

tridentctl get backend ontap-san-backend -n trident
+-----+-----+
+-----+-----+-----+-----+
|          NAME          | STORAGE DRIVER |                               UUID                               |
| STATE | VOLUMES |                               |
+-----+-----+-----+-----+
+-----+-----+-----+-----+
| ontap-san-backend | ontap-san      | 81abcb27-ea63-49bb-b606-0a5315ac5f82 |
| online |      33 |                               |
+-----+-----+-----+-----+
+-----+-----+-----+-----+
```

Após a exclusão do TridentBackendConfig O objeto, o Trident simplesmente o remove sem realmente excluir o próprio backend.

Criar e gerenciar classes de armazenamento

Criar uma classe de armazenamento

Configure um objeto StorageClass do Kubernetes e crie a classe de armazenamento para instruir o Trident sobre como provisionar volumes.

Configure um objeto StorageClass do Kubernetes.

O "[Objeto StorageClass do Kubernetes](#)" Identifica o Trident como o provisionador usado para essa classe e instrui o Trident sobre como provisionar um volume. Por exemplo:

```

apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontap-gold
provisioner: csi.trident.netapp.io
mountOptions:
  - nfsvers=3
  - nolock
parameters:
  backendType: "ontap-nas"
  media: "ssd"
allowVolumeExpansion: true
volumeBindingMode: Immediate

```

Consulte ["Objetos Kubernetes e Trident"](#) Para obter detalhes sobre como as classes de armazenamento interagem com o PersistentVolumeClaim e parâmetros para controlar como o Trident provisiona volumes.

Criar uma classe de armazenamento

Após criar o objeto StorageClass, você pode criar a classe de armazenamento. [Exemplos de classe de armazenamento](#) Fornece alguns exemplos básicos que você pode usar ou modificar.

Passos

1. Este é um objeto do Kubernetes, então use `kubectl` para criá-lo no Kubernetes.

```
kubectl create -f sample-input/storage-class-basic-csi.yaml
```

2. Agora você deverá ver uma classe de armazenamento **basic-csi** tanto no Kubernetes quanto no Trident, e o Trident deverá ter descoberto os pools no backend.

```
kubectl get sc basic-csi
```

NAME	PROVISIONER	AGE
basic-csi	csi.trident.netapp.io	15h

```
./tridentctl -n trident get storageclass basic-csi -o json
```

```

{
  "items": [
    {
      "Config": {
        "version": "1",
        "name": "basic-csi",
        "attributes": {
          "backendType": "ontap-nas"
        },
        "storagePools": null,
        "additionalStoragePools": null
      },
      "storage": {
        "ontapnas_10.0.0.1": [
          "aggr1",
          "aggr2",
          "aggr3",
          "aggr4"
        ]
      }
    }
  ]
}

```

Exemplos de classe de armazenamento

A Trident fornece ["Definições simples de classes de armazenamento para backends específicos"](#) .

Alternativamente, você pode editar `sample-input/storage-class-csi.yaml.templ` arquivo que acompanha o instalador e substituir `BACKEND_TYPE` com o nome do driver de armazenamento.

```
./tridentctl -n trident get backend
+-----+-----+-----+
+-----+-----+
|   NAME   | STORAGE DRIVER |          UUID          |
STATE | VOLUMES |
+-----+-----+-----+
+-----+-----+
| nas-backend | ontap-nas      | 98e19b74-aec7-4a3d-8dcf-128e5033b214 |
online |         0 |
+-----+-----+-----+
+-----+-----+

cp sample-input/storage-class-csi.yaml.templ sample-input/storage-class-
basic-csi.yaml

# Modify __BACKEND_TYPE__ with the storage driver field above (e.g.,
ontap-nas)
vi sample-input/storage-class-basic-csi.yaml
```

Gerenciar classes de armazenamento

Você pode visualizar as classes de armazenamento existentes, definir uma classe de armazenamento padrão, identificar o backend da classe de armazenamento e excluir classes de armazenamento.

Veja as classes de armazenamento existentes.

- Para visualizar as classes de armazenamento do Kubernetes existentes, execute o seguinte comando:

```
kubectl get storageclass
```

- Para visualizar os detalhes da classe de armazenamento do Kubernetes, execute o seguinte comando:

```
kubectl get storageclass <storage-class> -o json
```

- Para visualizar as classes de armazenamento sincronizadas do Trident, execute o seguinte comando:

```
tridentctl get storageclass
```

- Para visualizar os detalhes da classe de armazenamento sincronizado do Trident, execute o seguinte comando:

```
tridentctl get storageclass <storage-class> -o json
```

Defina uma classe de armazenamento padrão.

O Kubernetes 1.6 adicionou a capacidade de definir uma classe de armazenamento padrão. Esta é a classe de armazenamento que será usada para provisionar um Volume Persistente caso o usuário não especifique uma em uma Solicitação de Volume Persistente (PVC).

- Defina uma classe de armazenamento padrão configurando a anotação.
`storageclass.kubernetes.io/is-default-class` para verdadeiro na definição da classe de armazenamento. De acordo com a especificação, qualquer outro valor ou a ausência da anotação é interpretada como falsa.
- Você pode configurar uma classe de armazenamento existente para ser a classe de armazenamento padrão usando o seguinte comando:

```
kubectl patch storageclass <storage-class-name> -p '{"metadata":  
{"annotations":{"storageclass.kubernetes.io/is-default-class":"true"}}}'
```

- Da mesma forma, você pode remover a anotação da classe de armazenamento padrão usando o seguinte comando:

```
kubectl patch storageclass <storage-class-name> -p '{"metadata":  
{"annotations":{"storageclass.kubernetes.io/is-default-class":"false"}}}'
```

Existem também exemplos no pacote de instalação do Trident que incluem essa anotação.



Deve haver apenas uma classe de armazenamento padrão em seu cluster por vez. Tecnicamente, o Kubernetes não impede que você tenha mais de uma classe de armazenamento, mas ele se comportará como se não houvesse nenhuma classe de armazenamento padrão.

Identifique o backend para uma classe de armazenamento.

Este é um exemplo do tipo de pergunta que você pode responder com o JSON que `tridentctl` Saídas para objetos do backend Trident . Isso usa o `jq` utilitário, que você talvez precise instalar primeiro.

```
tridentctl get storageclass -o json | jq '[.items[] | {storageClass:  
.Config.name, backends: [.storage]|unique}]'
```

Excluir uma classe de armazenamento

Para excluir uma classe de armazenamento do Kubernetes, execute o seguinte comando:

```
kubectl delete storageclass <storage-class>
```

`<storage-class>` deve ser substituído pela sua classe de armazenamento.

Quaisquer volumes persistentes criados por meio dessa classe de armazenamento permanecerão intactos e o Trident continuará a gerenciá-los.



Trident impõe um vazio `fsType` pelos volumes que gera. Para backends iSCSI, recomenda-se impor `parameters.fsType` na classe de armazenamento. Você deve excluir as `StorageClasses` existentes e recriá-las com `parameters.fsType` especificado.

Provisionar e gerenciar volumes

Forneça um volume

Crie um `PersistentVolumeClaim` (PVC) que utilize a `StorageClass` do Kubernetes configurada para solicitar acesso ao PV. Em seguida, você pode montar o painel fotovoltaico em um suporte.

Visão geral

Um "*PersistentVolumeClaim*" (PVC) é uma solicitação de acesso ao `PersistentVolume` no cluster.

O PVC pode ser configurado para solicitar armazenamento de um determinado tamanho ou modo de acesso. Utilizando a `StorageClass` associada, o administrador do cluster pode controlar mais do que apenas o tamanho e o modo de acesso do `PersistentVolume`, como o desempenho ou o nível de serviço.

Após criar o tubo de PVC, você pode montar o volume em um compartimento.

Criar o PVC

Passos

1. Crie o PVC.

```
kubectl create -f pvc.yaml
```

2. Verifique o status do PVC.

```
kubectl get pvc
```

NAME	STATUS	VOLUME	CAPACITY	ACCESS MODES	STORAGECLASS	AGE
pvc-storage	Bound	pv-name	1Gi	RWO		5m

1. Instale o volume em um pod.

```
kubectl create -f pv-pod.yaml
```



Você pode monitorar o progresso usando `kubectl get pod --watch`.

2. Verifique se o volume está montado em `/my/mount/path`.

```
kubectl exec -it task-pv-pod -- df -h /my/mount/path
```

3. Agora você pode excluir o Pod. O aplicativo Pod deixará de existir, mas o volume permanecerá.

```
kubectl delete pod pv-pod
```

Exemplos de manifestos

Manifestações de exemplo de PersistentVolumeClaim

Estes exemplos mostram opções básicas de configuração de PVC.

PVC com acesso RWO

Este exemplo mostra um PVC básico com acesso RWO associado a uma StorageClass chamada `basic-csi`.

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: pvc-storage
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: basic-csi
```

PVC com NVMe/TCP

Este exemplo mostra um PVC básico para NVMe/TCP com acesso RWO associado a uma StorageClass chamada `protection-gold`.

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: pvc-san-nvme
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 300Mi
  storageClassName: protection-gold
```


Exemplos de manifesto de pod

Estes exemplos mostram configurações básicas para fixar o PVC a um pod.

Configuração básica

```
kind: Pod
apiVersion: v1
metadata:
  name: pv-pod
spec:
  volumes:
    - name: storage
      persistentVolumeClaim:
        claimName: pvc-storage
  containers:
    - name: pv-container
      image: nginx
      ports:
        - containerPort: 80
          name: "http-server"
      volumeMounts:
        - mountPath: "/my/mount/path"
          name: storage
```

Configuração básica de NVMe/TCP

```
apiVersion: v1
kind: Pod
metadata:
  name: pod-nginx
spec:
  volumes:
    - name: basic-pvc
      persistentVolumeClaim:
        claimName: pvc-san-nvme
  containers:
    - name: task-pv-container
      image: nginx
      volumeMounts:
        - mountPath: "/my/mount/path"
          name: basic-pvc
```

Consulte ["Objetos Kubernetes e Trident"](#) Para obter detalhes sobre como as classes de armazenamento interagem com o PersistentVolumeClaim e parâmetros para controlar como o Trident provisiona volumes.

Expandir volumes

O Trident oferece aos usuários do Kubernetes a capacidade de expandir seus volumes depois que eles são criados. Encontre informações sobre as configurações necessárias para expandir volumes iSCSI, NFS, SMB, NVMe/TCP e FC.

Expandir um volume iSCSI

Você pode expandir um Volume Persistente (PV) iSCSI usando o provisionador CSI.



A expansão de volume iSCSI é suportada pelo `ontap-san`, `ontap-san-economy`, `solidfire-san` requer drivers e Kubernetes 1.16 ou posterior.

Etapa 1: Configure a classe de armazenamento para suportar a expansão de volume.

Edite a definição da classe de armazenamento para definir o `allowVolumeExpansion` campo para `true`.

```
cat storageclass-ontapsan.yaml
```

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontap-san
provisioner: csi.trident.netapp.io
parameters:
  backendType: "ontap-san"
allowVolumeExpansion: True
```

Para uma StorageClass já existente, edite-a para incluir o `allowVolumeExpansion` parâmetro.

Passo 2: Crie um PVC com a StorageClass que você criou.

Edite a definição de PVC e atualize o `spec.resources.requests.storage` para refletir o novo tamanho desejado, que deve ser maior que o tamanho original.

```
cat pvc-ontapsan.yaml
```

```

kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: san-pvc
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: ontap-san

```

A Trident cria um Volume Persistente (PV) e o associa a esta Reivindicação de Volume Persistente (PVC).

```

kubectl get pvc
NAME          STATUS    VOLUME                                     CAPACITY
ACCESS MODES  STORAGECLASS  AGE
san-pvc      Bound       pvc-8a814d62-bd58-4253-b0d1-82f2885db671  1Gi
RWO          ontap-san    8s

kubectl get pv
NAME          CAPACITY  ACCESS MODES  RECLAIM POLICY   STATUS    CLAIM                                STORAGECLASS  REASON    AGE
pvc-8a814d62-bd58-4253-b0d1-82f2885db671  1Gi      RWO          Delete           Bound     default/san-pvc  ontap-san                                10s

```

Etapa 3: Defina um compartimento que conecte o PVC.

Conecte o PV a um pod para que ele possa ser redimensionado. Existem dois cenários ao redimensionar um PV iSCSI:

- Se o PV estiver conectado a um pod, o Trident expande o volume no backend de armazenamento, verifica novamente o dispositivo e redimensiona o sistema de arquivos.
- Ao tentar redimensionar um PV não anexado, o Trident expande o volume no servidor de armazenamento. Após o PVC ser vinculado a um pod, o Trident verifica novamente o dispositivo e redimensiona o sistema de arquivos. Em seguida, o Kubernetes atualiza o tamanho do PVC após a conclusão bem-sucedida da operação de expansão.

Neste exemplo, é criado um pod que utiliza o `san-pvc`.

```
kubectl get pod
NAME          READY   STATUS    RESTARTS   AGE
ubuntu-pod    1/1     Running   0           65s

kubectl describe pvc san-pvc
Name:          san-pvc
Namespace:     default
StorageClass:  ontap-san
Status:        Bound
Volume:        pvc-8a814d62-bd58-4253-b0d1-82f2885db671
Labels:        <none>
Annotations:   pv.kubernetes.io/bind-completed: yes
               pv.kubernetes.io/bound-by-controller: yes
               volume.beta.kubernetes.io/storage-provisioner:
               csi.trident.netapp.io
Finalizers:    [kubernetes.io/pvc-protection]
Capacity:      1Gi
Access Modes:  RWO
VolumeMode:    Filesystem
Mounted By:    ubuntu-pod
```

Etapa 4: Expandir o PV

Para redimensionar o PV criado de 1Gi para 2Gi, edite a definição do PVC e atualize o `spec.resources.requests.storage` para 2Gi.

```
kubectl edit pvc san-pvc
```

```

# Please edit the object below. Lines beginning with a '#' will be
ignored,
# and an empty file will abort the edit. If an error occurs while saving
this file will be
# reopened with the relevant failures.
#
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  annotations:
    pv.kubernetes.io/bind-completed: "yes"
    pv.kubernetes.io/bound-by-controller: "yes"
    volume.beta.kubernetes.io/storage-provisioner: csi.trident.netapp.io
  creationTimestamp: "2019-10-10T17:32:29Z"
  finalizers:
  - kubernetes.io/pvc-protection
  name: san-pvc
  namespace: default
  resourceVersion: "16609"
  selfLink: /api/v1/namespaces/default/persistentvolumeclaims/san-pvc
  uid: 8a814d62-bd58-4253-b0d1-82f2885db671
spec:
  accessModes:
  - ReadWriteOnce
  resources:
    requests:
      storage: 2Gi
# ...

```

Etapa 5: Valide a expansão

Você pode verificar se a expansão funcionou corretamente conferindo o tamanho do PVC, do PV e o volume do Trident :

```
kubectl get pvc san-pvc
NAME          STATUS    VOLUME                                     CAPACITY
ACCESS MODES  STORAGECLASS  AGE
san-pvc      Bound       pvc-8a814d62-bd58-4253-b0d1-82f2885db671  2Gi
RWO           ontap-san    11m

kubectl get pv
NAME          CAPACITY  ACCESS MODES
RECLAIM POLICY STATUS    CLAIM          STORAGECLASS  REASON    AGE
pvc-8a814d62-bd58-4253-b0d1-82f2885db671  2Gi        RWO
Delete              Bound      default/san-pvc  ontap-san    12m

tridentctl get volumes -n trident
+-----+-----+-----+
+-----+-----+-----+-----+-----+
|          NAME          | SIZE | STORAGE CLASS |
+-----+-----+-----+-----+-----+
|          BACKEND UUID  | STATE | MANAGED |
+-----+-----+-----+-----+-----+
| pvc-8a814d62-bd58-4253-b0d1-82f2885db671 | 2.0 GiB | ontap-san |
| block      | a9b7bfff-0505-4e31-b6c5-59f492e02d33 | online | true      |
+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+
```

Expandir um volume FC

Você pode expandir um Volume Persistente (PV) FC usando o provisionador CSI.



A expansão do volume FC é suportada pelo `ontap-san` O driver requer Kubernetes 1.16 ou posterior.

Etapa 1: Configure a classe de armazenamento para suportar a expansão de volume.

Edite a definição da classe de armazenamento para definir o `allowVolumeExpansion` campo para `true`.

```
cat storageclass-ontapsan.yaml
```

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontap-san
provisioner: csi.trident.netapp.io
parameters:
  backendType: "ontap-san"
allowVolumeExpansion: True
```

Para uma StorageClass já existente, edite-a para incluir o `allowVolumeExpansion` parâmetro.

Passo 2: Crie um PVC com a StorageClass que você criou.

Edite a definição de PVC e atualize o `spec.resources.requests.storage` para refletir o novo tamanho desejado, que deve ser maior que o tamanho original.

```
cat pvc-ontapsan.yaml
```

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: san-pvc
spec:
  accessModes:
  - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: ontap-san
```

A Trident cria um Volume Persistente (PV) e o associa a esta Reivindicação de Volume Persistente (PVC).

```
kubectl get pvc
NAME          STATUS    VOLUME                                     CAPACITY
ACCESS MODES  STORAGECLASS  AGE
san-pvc      Bound       pvc-8a814d62-bd58-4253-b0d1-82f2885db671  1Gi
RWO           ontap-san    8s

kubectl get pv
NAME          CAPACITY  ACCESS MODES  RECLAIM POLICY   STATUS    CLAIM                                STORAGECLASS  REASON    AGE
pvc-8a814d62-bd58-4253-b0d1-82f2885db671  1Gi      RWO           Delete           Bound     default/san-pvc  ontap-san                10s
```

Etapa 3: Defina um compartimento que conecte o PVC.

Conecte o PV a um pod para que ele possa ser redimensionado. Existem dois cenários ao redimensionar um painel fotovoltaico de célula de combustível:

- Se o PV estiver conectado a um pod, o Trident expande o volume no backend de armazenamento, verifica novamente o dispositivo e redimensiona o sistema de arquivos.
- Ao tentar redimensionar um PV não anexado, o Trident expande o volume no servidor de armazenamento. Após o PVC ser vinculado a um pod, o Trident verifica novamente o dispositivo e redimensiona o sistema de arquivos. Em seguida, o Kubernetes atualiza o tamanho do PVC após a conclusão bem-sucedida da

operação de expansão.

Neste exemplo, é criado um pod que utiliza o `san-pvc`.

```
kubectl get pod
NAME          READY   STATUS    RESTARTS   AGE
ubuntu-pod    1/1     Running   0           65s

kubectl describe pvc san-pvc
Name:          san-pvc
Namespace:     default
StorageClass:  ontap-san
Status:        Bound
Volume:        pvc-8a814d62-bd58-4253-b0d1-82f2885db671
Labels:        <none>
Annotations:   pv.kubernetes.io/bind-completed: yes
               pv.kubernetes.io/bound-by-controller: yes
               volume.beta.kubernetes.io/storage-provisioner:
               csi.trident.netapp.io
Finalizers:    [kubernetes.io/pvc-protection]
Capacity:      1Gi
Access Modes:  RWO
VolumeMode:    Filesystem
Mounted By:    ubuntu-pod
```

Etapa 4: Expandir o PV

Para redimensionar o PV criado de 1Gi para 2Gi, edite a definição do PVC e atualize o `spec.resources.requests.storage` para 2Gi.

```
kubectl edit pvc san-pvc
```



```

# Please edit the object below. Lines beginning with a '#' will be
ignored,
# and an empty file will abort the edit. If an error occurs while saving
this file will be
# reopened with the relevant failures.
#
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  annotations:
    pv.kubernetes.io/bind-completed: "yes"
    pv.kubernetes.io/bound-by-controller: "yes"
    volume.beta.kubernetes.io/storage-provisioner: csi.trident.netapp.io
  creationTimestamp: "2019-10-10T17:32:29Z"
  finalizers:
  - kubernetes.io/pvc-protection
  name: san-pvc
  namespace: default
  resourceVersion: "16609"
  selfLink: /api/v1/namespaces/default/persistentvolumeclaims/san-pvc
  uid: 8a814d62-bd58-4253-b0d1-82f2885db671
spec:
  accessModes:
  - ReadWriteOnce
  resources:
    requests:
      storage: 2Gi
# ...

```

Etapas 5: Valide a expansão

Você pode verificar se a expansão funcionou corretamente conferindo o tamanho do PVC, do PV e o volume do Trident :

```
kubectl get pvc san-pvc
NAME          STATUS    VOLUME                                     CAPACITY
ACCESS MODES  STORAGECLASS  AGE
san-pvc      Bound      pvc-8a814d62-bd58-4253-b0d1-82f2885db671  2Gi
RWO           ontap-san    11m

kubectl get pv
NAME          CAPACITY  ACCESS MODES
RECLAIM POLICY STATUS    CLAIM          STORAGECLASS  REASON    AGE
pvc-8a814d62-bd58-4253-b0d1-82f2885db671  2Gi        RWO
Delete              Bound      default/san-pvc  ontap-san    12m

tridentctl get volumes -n trident
+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+
|          NAME          | SIZE | STORAGE CLASS |
+-----+-----+-----+-----+-----+-----+
|          BACKEND UUID          | STATE | MANAGED |
+-----+-----+-----+-----+-----+-----+
| pvc-8a814d62-bd58-4253-b0d1-82f2885db671 | 2.0 GiB | ontap-san |
+-----+-----+-----+-----+-----+-----+
| block | a9b7bfff-0505-4e31-b6c5-59f492e02d33 | online | true |
+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+
```

Expandir um volume NFS

O Trident suporta a expansão de volume para PVs NFS provisionados em `ontap-nas`, `ontap-nas-economy`, `ontap-nas-flexgroup`, `gcp-cvs`, e `azure-netapp-files` back-ends.

Etapa 1: Configure a classe de armazenamento para suportar a expansão de volume.

Para redimensionar um PV NFS, o administrador primeiro precisa configurar a classe de armazenamento para permitir a expansão do volume, definindo o seguinte: `allowVolumeExpansion` campo para `true`:

```
cat storageclass-ontapnas.yaml
```

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontapnas
provisioner: csi.trident.netapp.io
parameters:
  backendType: ontap-nas
allowVolumeExpansion: true
```

Se você já criou uma classe de armazenamento sem essa opção, basta editar a classe de armazenamento

existente usando `kubectl edit storageclass` para permitir a expansão do volume.

Passo 2: Crie um PVC com a StorageClass que você criou.

```
cat pvc-ontapnas.yaml
```

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: ontapnas20mb
spec:
  accessModes:
  - ReadWriteOnce
  resources:
    requests:
      storage: 20Mi
  storageClassName: ontapnas
```

A Trident deve criar um PV NFS de 20 MiB para este PVC:

```
kubectl get pvc
NAME                STATUS    VOLUME
CAPACITY            ACCESS MODES  STORAGECLASS  AGE
ontapnas20mb        Bound      pvc-08f3d561-b199-11e9-8d9f-5254004dfdb7  20Mi
RW0                  ontapnas      9s

kubectl get pv pvc-08f3d561-b199-11e9-8d9f-5254004dfdb7
NAME                CAPACITY  ACCESS MODES
RECLAIM POLICY      STATUS    CLAIM                STORAGECLASS  REASON
AGE
pvc-08f3d561-b199-11e9-8d9f-5254004dfdb7  20Mi      RW0
Delete              Bound      default/ontapnas20mb  ontapnas
2m42s
```

Etapa 3: Expandir o PV

Para redimensionar o PV recém-criado de 20 MiB para 1 GiB, edite o PVC e defina `spec.resources.requests.storage` para 1 GiB:

```
kubectl edit pvc ontapnas20mb
```

```

# Please edit the object below. Lines beginning with a '#' will be
ignored,
# and an empty file will abort the edit. If an error occurs while saving
this file will be
# reopened with the relevant failures.
#
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  annotations:
    pv.kubernetes.io/bind-completed: "yes"
    pv.kubernetes.io/bound-by-controller: "yes"
    volume.beta.kubernetes.io/storage-provisioner: csi.trident.netapp.io
  creationTimestamp: 2018-08-21T18:26:44Z
  finalizers:
  - kubernetes.io/pvc-protection
  name: ontapnas20mb
  namespace: default
  resourceVersion: "1958015"
  selfLink: /api/v1/namespaces/default/persistentvolumeclaims/ontapnas20mb
  uid: c1bd7fa5-a56f-11e8-b8d7-fa163e59eaab
spec:
  accessModes:
  - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
# ...

```

Etapa 4: Valide a expansão

Você pode verificar se o redimensionamento funcionou corretamente conferindo o tamanho do PVC, do PV e do volume Trident :

```
kubectl get pvc ontapnas20mb
NAME          STATUS    VOLUME
CAPACITY     ACCESS MODES  STORAGECLASS  AGE
ontapnas20mb  Bound      pvc-08f3d561-b199-11e9-8d9f-5254004dfdb7  1Gi
RWO          ontapnas      4m44s

kubectl get pv pvc-08f3d561-b199-11e9-8d9f-5254004dfdb7
NAME          CAPACITY  ACCESS MODES
RECLAIM POLICY STATUS    CLAIM          STORAGECLASS  REASON
AGE
pvc-08f3d561-b199-11e9-8d9f-5254004dfdb7  1Gi      RWO
Delete          Bound      default/ontapnas20mb  ontapnas
5m35s

tridentctl get volume pvc-08f3d561-b199-11e9-8d9f-5254004dfdb7 -n trident
+-----+-----+-----+
+-----+-----+-----+-----+
|          NAME          | SIZE  | STORAGE CLASS |
+-----+-----+-----+-----+
| PROTOCOL | BACKEND UUID          | STATE | MANAGED |
+-----+-----+-----+-----+
+-----+-----+-----+-----+
| pvc-08f3d561-b199-11e9-8d9f-5254004dfdb7 | 1.0 GiB | ontapnas      |
file      | c5a6f6a4-b052-423b-80d4-8fb491a14a22 | online | true     |
+-----+-----+-----+-----+
+-----+-----+-----+-----+
```

Volumes de importação

Você pode importar volumes de armazenamento existentes como um PV do Kubernetes usando `tridentctl import`.

Visão geral e considerações

Você pode importar um volume para o Trident para:

- Containerize uma aplicação e reutilize seu conjunto de dados existente.
- Utilize um clone de um conjunto de dados para uma aplicação efêmera.
- Reconstruir um cluster Kubernetes com falha
- Migrar dados de aplicativos durante a recuperação de desastres

Considerações

Antes de importar um volume, revise as seguintes considerações.

- O Trident só pode importar volumes ONTAP do tipo RW (leitura e gravação). Os volumes do tipo DP (proteção de dados) são volumes de destino do SnapMirror. Você deve quebrar a relação de espelhamento antes de importar o volume para o Trident.

- Sugerimos importar volumes sem conexões ativas. Para importar um volume que esteja sendo usado ativamente, clone o volume e, em seguida, execute a importação.



Isso é especialmente importante para volumes de bloco, pois o Kubernetes não teria conhecimento da conexão anterior e poderia facilmente anexar um volume ativo a um pod. Isso pode resultar em corrupção de dados.

- No entanto `StorageClass` Deve ser especificado em um PVC; o Trident não usa esse parâmetro durante a importação. As classes de armazenamento são usadas durante a criação de volumes para selecionar entre os pools disponíveis com base nas características de armazenamento. Como o volume já existe, não é necessário selecionar um pool durante a importação. Portanto, a importação não falhará mesmo que o volume exista em um backend ou pool que não corresponda à classe de armazenamento especificada no PVC.
- O volume existente é determinado e definido no PVC. Após o volume ser importado pelo driver de armazenamento, o PV é criado com uma referência `ClaimRef` para o PVC.
 - A política de recuperação é inicialmente definida como `retain` no PV. Após o Kubernetes vincular com sucesso o PVC e o PV, a política de recuperação é atualizada para corresponder à política de recuperação da classe de armazenamento.
 - Se a política de recuperação da Classe de Armazenamento for `delete` O volume de armazenamento será excluído quando o PV for excluído.
- Por padrão, o Trident gerencia o PVC e renomeia o FlexVol volume e o LUN no backend. Você pode passar o `--no-manage` sinalizador para importar um volume não gerenciado. Se você usar `--no-manage` A Trident não realiza nenhuma operação adicional no PVC ou PV durante o ciclo de vida dos objetos. O volume de armazenamento não é excluído quando o PV é excluído, e outras operações como clonagem e redimensionamento de volume também são ignoradas.



Essa opção é útil se você deseja usar o Kubernetes para cargas de trabalho em contêineres, mas, fora isso, prefere gerenciar o ciclo de vida do volume de armazenamento fora do Kubernetes.

- Uma anotação é adicionada ao PVC e ao PV com a dupla função de indicar que o volume foi importado e se o PVC e o PV são gerenciados. Esta anotação não deve ser modificada nem removida.

Importar um volume

Você pode usar `tridentctl import` Importar um volume.

Passos

1. Crie o arquivo Persistent Volume Claim (PVC) (por exemplo, `pvc.yaml`) que será usado para criar o PVC. O arquivo PVC deve incluir `name` , `namespace` , `accessModes` , e `storageClassName` . Opcionalmente, você pode especificar `unixPermissions` na sua definição de PVC.

Segue abaixo um exemplo de especificação mínima:

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: my_claim
  namespace: my_namespace
spec:
  accessModes:
    - ReadWriteOnce
  storageClassName: my_storage_class
```



Não inclua parâmetros adicionais como nome do PV ou tamanho do volume. Isso pode fazer com que o comando de importação falhe.

2. Use o `tridentctl import volume` Comando para especificar o nome do backend Trident que contém o volume e o nome que identifica exclusivamente o volume no armazenamento (por exemplo: ONTAP FlexVol, Element Volume, caminho do Cloud Volumes Service). O `-f` É necessário fornecer o argumento para especificar o caminho para o arquivo PVC.

```
tridentctl import volume <backendName> <volumeName> -f <path-to-pvc-
file>
```

Exemplos

Consulte os exemplos de importação de volume a seguir para obter informações sobre os drivers compatíveis.

ONTAP NAS e ONTAP NAS FlexGroup

O Trident suporta a importação de volumes usando o `ontap-nas` e `ontap-nas-flexgroup` motoristas.



- O Trident não suporta importação de volume usando o `ontap-nas-economy` motorista.
- O `ontap-nas` e `ontap-nas-flexgroup` Os drivers não permitem nomes de volume duplicados.

Cada volume criado com o `ontap-nas` O driver é um FlexVol volume no cluster ONTAP . Importando volumes FlexVol com o `ontap-nas` O driver funciona da mesma forma. Um volume FlexVol que já existe em um cluster ONTAP pode ser importado como um `ontap-nas` PVC. Da mesma forma, os volumes do FlexGroup podem ser importados como `ontap-nas-flexgroup` PVCs.

Exemplos de ONTAP NAS

A seguir, apresentamos um exemplo de importação de um volume gerenciado e de um volume não gerenciado.

Volume gerenciado

O exemplo a seguir importa um volume chamado `managed_volume` em um backend chamado `ontap_nas`:

```
tridentctl import volume ontap_nas managed_volume -f <path-to-pvc-file>
```

NAME	SIZE	STORAGE CLASS
pvc-bf5ad463-afbb-11e9-8d9f-5254004dfdb7	1.0 GiB	standard
file	online	true

Volume não gerenciado

Ao usar o `--no-manage` argumento: Trident não renomeia o volume.

O exemplo a seguir importa `unmanaged_volume` no `ontap_nas` backend:

```
tridentctl import volume nas_blog unmanaged_volume -f <path-to-pvc-file> --no-manage
```

NAME	SIZE	STORAGE CLASS
pvc-df07d542-afbc-11e9-8d9f-5254004dfdb7	1.0 GiB	standard
file	online	false

ONTAP SAN

O Trident suporta importação de volume usando o `ontap-san` (iSCSI, NVMe/TCP e FC) e `ontap-san-economy` motoristas.

O Trident pode importar volumes ONTAP SAN FlexVol que contêm um único LUN. Isto é consistente com o `ontap-san` driver, que cria um FlexVol volume para cada PVC e um LUN dentro do FlexVol volume. O Trident importa o FlexVol volume e o associa à definição de PVC. Trident pode importar `ontap-san-economy`

volumes que contêm vários LUNs.

Exemplos de SAN ONTAP

A seguir, apresentamos um exemplo de importação de um volume gerenciado e de um volume não gerenciado.

Volume gerenciado

Para volumes gerenciados, o Trident renomeia o FlexVol volume para `pvc-<uuid>` formato e o LUN dentro do FlexVol volume para `lun0`.

O exemplo a seguir importa o `ontap-san-managed` FlexVol volume que está presente no `ontap_san_default` backend:

```
tridentctl import volume ontapsan_san_default ontap-san-managed -f pvc-basic-import.yaml -n trident -d
```

	NAME		SIZE		STORAGE CLASS
PROTOCOL	BACKEND UUID			STATE	
	pvc-d6ee4f54-4e40-4454-92fd-d00fc228d74a		20 MiB		basic
block		cd394786-ddd5-4470-adc3-10c5ce4ca757		online	

Volume não gerenciado

O exemplo a seguir importa `unmanaged_example_volume` no `ontap_san` backend:

```
tridentctl import volume -n trident san_blog unmanaged_example_volume -f pvc-import.yaml --no-manage
```

	NAME		SIZE		STORAGE CLASS
PROTOCOL	BACKEND UUID			STATE	
	pvc-1fc999c9-ce8c-459c-82e4-ed4380a4b228		1.0 GiB		san-blog
block		e3275890-7d80-4af6-90cc-c7a0759f555a		online	

Se você tiver LUNs mapeadas para igroups que compartilham um IQN com o IQN de um nó do Kubernetes,

como mostrado no exemplo a seguir, você receberá o seguinte erro: LUN already mapped to initiator(s) in this group. Você precisará remover o iniciador ou desvincular o LUN para importar o volume.

Vserver	Igroup	Protocol	OS Type	Initiators
svm0	k8s-nodename.example.com-fe5d36f2-cded-4f38-9eb0-c7719fc2f9f3	iscsi	linux	iqn.1994-05.com.redhat:4c2e1cf35e0
svm0	unmanaged-example-igroup	mixed	linux	iqn.1994-05.com.redhat:4c2e1cf35e0

Elemento

O Trident oferece suporte ao software NetApp Element e à importação de volumes NetApp HCI usando o solidfire-san motorista.



O driver Element suporta nomes de volume duplicados. No entanto, o Trident retorna um erro se houver nomes de volume duplicados. Como solução alternativa, clone o volume, forneça um nome exclusivo para o volume e importe o volume clonado.

Exemplo de elemento

O exemplo a seguir importa um element-managed volume no backend element_default.

```
tridentctl import volume element_default element-managed -f pvc-basic-import.yaml -n trident -d
```

PROTOCOL	NAME	BACKEND UUID	SIZE	STORAGE CLASS	STATE	MANAGED
block	pvc-970ce1ca-2096-4ecd-8545-ac7edc24a8fe	d3ba047a-ea0b-43f9-9c42-e38e58301c49	10 GiB	basic-element	online	true

Plataforma Google Cloud

O Trident suporta a importação de volumes usando o gcp-cvs motorista.



Para importar um volume com suporte do NetApp Cloud Volumes Service no Google Cloud Platform, identifique o volume pelo seu caminho. O caminho do volume é a porção do caminho de exportação do volume após o `:/`. Por exemplo, se o caminho de exportação for `10.0.0.1:/adroit-jolly-swift`, o caminho do volume é `adroit-jolly-swift`.

Exemplo do Google Cloud Platform

O exemplo a seguir importa um `gcp-cvs` volume no backend `gcpcvs_YEppr` com o caminho de volume de `adroit-jolly-swift`.

```
tridentctl import volume gcpcvs_YEppr adroit-jolly-swift -f <path-to-pvc-file> -n trident
```

NAME	SIZE	STORAGE CLASS
pvc-a46ccab7-44aa-4433-94b1-e47fc8c0fa55	93 GiB	gcp-storage
e1a6e65b-299e-4568-ad05-4f0a105c888f	online	true

Azure NetApp Files

O Trident suporta a importação de volumes usando o `azure-netapp-files` motorista.



Para importar um volume do Azure NetApp Files, identifique o volume pelo seu caminho. O caminho do volume é a porção do caminho de exportação do volume após o `:/`. Por exemplo, se o caminho de montagem for `10.0.0.2:/importvol1`, o caminho do volume é `importvol1`.

Exemplo de Azure NetApp Files

O exemplo a seguir importa um `azure-netapp-files` volume no backend `azurenetafiles_40517` com o caminho do volume `importvol1`.

```
tridentctl import volume azurenetappfiles_40517 importvol1 -f <path-to-pvc-file> -n trident
```

```
+-----+-----+-----+
+-----+-----+-----+-----+
|          NAME          | SIZE | STORAGE CLASS |
| PROTOCOL | BACKEND UUID          | STATE | MANAGED |
+-----+-----+-----+
+-----+-----+-----+-----+
| pvc-0ee95d60-fd5c-448d-b505-b72901b3a4ab | 100 GiB | anf-storage |
file      | 1c01274f-d94b-44a3-98a3-04c953c9a51e | online | true      |
+-----+-----+-----+
+-----+-----+-----+-----+
```

Google Cloud NetApp Volumes

O Trident suporta a importação de volumes usando o `google-cloud-netapp-volumes` motorista.

Exemplo de Google Cloud NetApp Volumes

O exemplo a seguir importa um `google-cloud-netapp-volumes` volume no backend `backend-tbc-gcnv1` com o volume `testvoleasiaeast1`.

```
tridentctl import volume backend-tbc-gcnv1 "testvoleasiaeast1" -f < path-to-pvc> -n trident
```

```
+-----+-----+-----+
+-----+-----+-----+-----+
+-----+-----+
|          NAME          | SIZE | STORAGE CLASS |
| PROTOCOL | BACKEND UUID          | STATE | MANAGED |
+-----+-----+-----+
+-----+-----+-----+-----+
+-----+-----+
| pvc-a69cda19-218c-4ca9-a941-aea05dd13dc0 | 10 GiB | gcnv-nfs-sc-
identity | file      | 8c18cdf1-0770-4bc0-bcc5-c6295fe6d837 | online | true
|
+-----+-----+-----+
+-----+-----+-----+-----+
+-----+-----+-----+-----+
```

O exemplo a seguir importa um `google-cloud-netapp-volumes` Volume quando dois volumes estão presentes na mesma região:

```
tridentctl import volume backend-tbc-gcnv1
"projects/123456789100/locations/asia-east1-a/volumes/testvoleasiaeast1"
-f <path-to-pvc> -n trident
```

```
+-----+-----+-----+-----+
+-----+-----+-----+-----+
+-----+-----+
|          NAME          |  SIZE  | STORAGE CLASS |
| PROTOCOL |  BACKEND UUID  | STATE | MANAGED |
+-----+-----+-----+-----+
+-----+-----+-----+-----+
+-----+-----+
| pvc-a69cda19-218c-4ca9-a941-aea05dd13dc0 | 10 GiB | gcnv-nfs-sc-
identity | file      | 8c18cdf1-0770-4bc0-bcc5-c6295fe6d837 | online | true
|
+-----+-----+-----+-----+
+-----+-----+-----+-----+
+-----+-----+-----+-----+
```

Personalize os nomes e rótulos dos volumes.

Com o Trident, você pode atribuir nomes e rótulos significativos aos volumes que criar. Isso ajuda você a identificar e mapear facilmente os volumes aos seus respectivos recursos do Kubernetes (PVCs). Você também pode definir modelos no nível do backend para criar nomes de volume e rótulos personalizados; quaisquer volumes que você criar, importar ou clonar seguirão os modelos.

Antes de começar

Suporte para nomes e rótulos de volume personalizáveis:

1. Operações de criação, importação e clonagem de volumes.
2. No caso do driver ontap-nas-economy, apenas o nome do volume Qtree está em conformidade com o modelo de nome.
3. No caso do driver ontap-san-economy, apenas o nome LUN está em conformidade com o modelo de nome.

Limitações

1. Os nomes de volume personalizáveis são compatíveis apenas com drivers ONTAP locais.
2. Os nomes de volume personalizáveis não se aplicam a volumes existentes.

Principais comportamentos dos nomes de volume personalizáveis

1. Se ocorrer uma falha devido a uma sintaxe inválida em um modelo de nome, a criação do backend falhará. No entanto, se a aplicação do modelo falhar, o volume será nomeado de acordo com a convenção de nomenclatura existente.

2. O prefixo de armazenamento não se aplica quando um volume é nomeado usando um modelo de nome da configuração do backend. Qualquer valor de prefixo desejado pode ser adicionado diretamente ao modelo.

Exemplos de configuração de backend com modelo de nome e rótulos

É possível definir modelos de nomes personalizados no nível raiz e/ou no nível do pool.

Exemplo de nível raiz

```
{
  "version": 1,
  "storageDriverName": "ontap-nas",
  "backendName": "ontap-nfs-backend",
  "managementLIF": "<ip address>",
  "svm": "svm0",
  "username": "<admin>",
  "password": "<password>",
  "defaults": {
    "nameTemplate":
      "{{.volume.Name}}_{{.labels.cluster}}_{{.volume.Namespace}}_{{.volume.RequestName}}"
  },
  "labels": {
    "cluster": "ClusterA",
    "PVC": "{{.volume.Namespace}}_{{.volume.RequestName}}"
  }
}
```

Exemplo de nível de piscina

```
{
  "version": 1,
  "storageDriverName": "ontap-nas",
  "backendName": "ontap-nfs-backend",
  "managementLIF": "<ip address>",
  "svm": "svm0",
  "username": "<admin>",
  "password": "<password>",
  "useREST": true,
  "storage": [
    {
      "labels": {
        "labelname": "label1",
        "name": "{{ .volume.Name }}"
      },
      "defaults": {
        "nameTemplate": "pool01_{{ .volume.Name }}_{{ .labels.cluster }}_{{ .volume.Namespace }}_{{ .volume.RequestName }}"
      }
    },
    {
      "labels": {
        "cluster": "label2",
        "name": "{{ .volume.Name }}"
      },
      "defaults": {
        "nameTemplate": "pool02_{{ .volume.Name }}_{{ .labels.cluster }}_{{ .volume.Namespace }}_{{ .volume.RequestName }}"
      }
    }
  ]
}
```

Exemplos de modelos de nome

Exemplo 1:

```
"nameTemplate": "{{ .config.StoragePrefix }}_{{ .volume.Name }}_{{ .config.BackendName }}"
```

Exemplo 2:

```
"nameTemplate": "pool_{{ .config.StoragePrefix }}_{{ .volume.Name }}_{{ slice .volume.RequestName 1 5 }}"
```

Pontos a considerar

1. No caso de importações em volume, os rótulos são atualizados somente se o volume existente possuir rótulos em um formato específico. Por exemplo: {"provisioning":{"Cluster":"ClusterA", "PVC": "pvcname"}}.
2. No caso de importações de volumes gerenciados, o nome do volume segue o modelo de nome definido no nível raiz na definição do backend.
3. O Trident não suporta o uso de um operador de fatiamento com o prefixo de armazenamento.
4. Se os modelos não resultarem em nomes de volume exclusivos, o Trident acrescentará alguns caracteres aleatórios para criar nomes de volume exclusivos.
5. Se o nome personalizado para um volume econômico do NAS exceder 64 caracteres, o Trident nomeará os volumes de acordo com a convenção de nomenclatura existente. Para todos os outros drivers ONTAP, se o nome do volume exceder o limite de nomes, o processo de criação do volume falhará.

Compartilhar um volume NFS entre namespaces

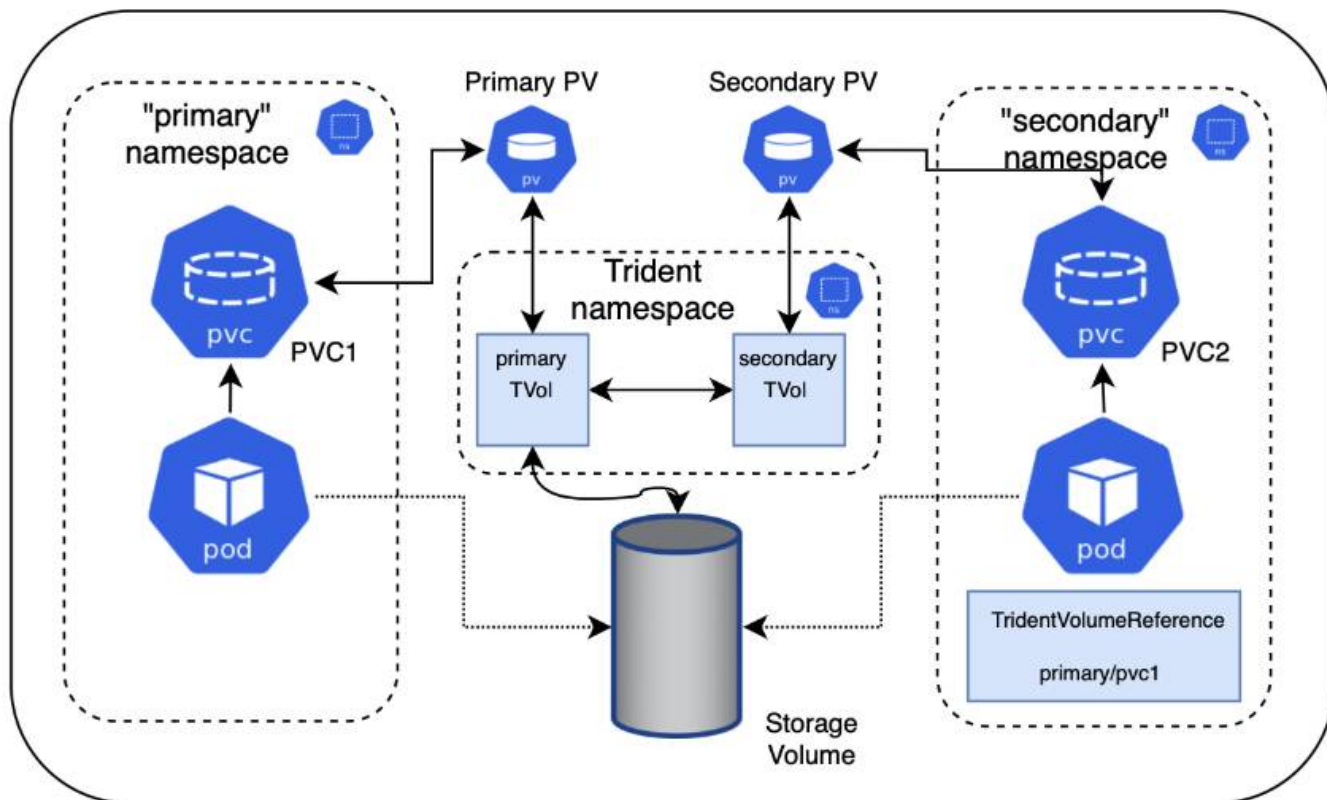
Usando o Trident, você pode criar um volume em um namespace primário e compartilhá-lo em um ou mais namespaces secundários.

Características

O CR `TridentVolumeReference` permite compartilhar com segurança volumes NFS `ReadWriteMany` (RWX) em um ou mais namespaces do Kubernetes. Essa solução nativa do Kubernetes oferece os seguintes benefícios:

- Múltiplos níveis de controle de acesso para garantir a segurança.
- Compatível com todos os drivers de volume Trident NFS.
- Sem dependência do `tridentctl` ou de qualquer outro recurso não nativo do Kubernetes.

Este diagrama ilustra o compartilhamento de volumes NFS entre dois namespaces do Kubernetes.



Início rápido

Você pode configurar o compartilhamento de volumes NFS em apenas alguns passos.

1

Configure o PVC de origem para compartilhar o volume.

O proprietário do namespace de origem concede permissão para acessar os dados no PVC de origem.

2

Conceder permissão para criar uma solicitação de configuração (CR) no espaço de nomes de destino.

O administrador do cluster concede permissão ao proprietário do namespace de destino para criar o CR `TridentVolumeReference`.

3

Crie uma `TridentVolumeReference` no namespace de destino.

O proprietário do namespace de destino cria o CR `TridentVolumeReference` para se referir ao PVC de origem.

4

Crie o PVC subordinado no namespace de destino.

O proprietário do namespace de destino cria o PVC subordinado para usar a fonte de dados do PVC de origem.

Configure os namespaces de origem e destino.

Para garantir a segurança, o compartilhamento entre namespaces exige colaboração e ação do proprietário do namespace de origem, do administrador do cluster e do proprietário do namespace de destino. A função do usuário é definida em cada etapa.

Passos

1. **Proprietário do namespace de origem:** Criar o PVC(`pvc1`) no namespace de origem que concede permissão para compartilhar com o namespace de destino(`namespace2`) usando o `shareToNamespace` anotação.

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: pvc1
  namespace: namespace1
  annotations:
    trident.netapp.io/shareToNamespace: namespace2
spec:
  accessModes:
    - ReadWriteMany
  storageClassName: trident-csi
  resources:
    requests:
      storage: 100Gi
```

O Trident cria o PV e seu volume de armazenamento NFS de backend.



- Você pode compartilhar o PVC com vários namespaces usando uma lista separada por vírgulas. Por exemplo, `trident.netapp.io/shareToNamespace: namespace2, namespace3, namespace4`.
- Você pode compartilhar com todos os namespaces usando `*`. Por exemplo, `trident.netapp.io/shareToNamespace: *`
- Você pode atualizar o PVC para incluir o `shareToNamespace` Anotações a qualquer momento.

2. **Administrador do cluster:** Certifique-se de que o RBAC adequado esteja em vigor para conceder permissão ao proprietário do namespace de destino para criar o `TridentVolumeReference` CR no namespace de destino.
3. **Proprietário do namespace de destino:** Crie uma solicitação de configuração (CR) `TridentVolumeReference` no namespace de destino que faça referência ao namespace de origem. `pvc1`.

```

apiVersion: trident.netapp.io/v1
kind: TridentVolumeReference
metadata:
  name: my-first-tvr
  namespace: namespace2
spec:
  pvcName: pvc1
  pvcNamespace: namespace1

```

4. **Proprietário do namespace de destino:** Criar um PVC(`pvc2`) no espaço de nomes de destino(`namespace2`) usando o `shareFromPVC` Anotação para designar o PVC de origem.

```

kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  annotations:
    trident.netapp.io/shareFromPVC: namespace1/pvc1
  name: pvc2
  namespace: namespace2
spec:
  accessModes:
    - ReadWriteMany
  storageClassName: trident-csi
  resources:
    requests:
      storage: 100Gi

```



O diâmetro do tubo de PVC de destino deve ser menor ou igual ao do tubo de PVC de origem.

Resultados

Trident lê o `shareFromPVC` A anotação no PVC de destino cria o PV de destino como um volume subordinado sem recurso de armazenamento próprio, que aponta para o PV de origem e compartilha o recurso de armazenamento do PV de origem. O PVC e o PV de destino parecem estar conectados normalmente.

Excluir um volume compartilhado

Você pode excluir um volume que é compartilhado entre vários namespaces. O Trident removerá o acesso ao volume no namespace de origem e manterá o acesso para outros namespaces que compartilham o volume. Quando todos os namespaces que fazem referência ao volume são removidos, o Trident exclui o volume.

Usar `tridentctl get` para consultar volumes subordinados

Usando o `tridentctl` utilitário, você pode executar o `get` comando para obter volumes subordinados. Para

obter mais informações, consulte o `tridentctl` [comandos e opções](#).

```
Usage:
  tridentctl get [option]
```

Bandeiras:

- `--h, --help`` Ajuda para volumes.
- `--parentOfSubordinate string`: Limitar a consulta ao volume de origem subordinado.
- `--subordinateOf string`: Limitar a consulta aos subordinados do volume.

Limitações

- O Trident não pode impedir que os namespaces de destino gravem no volume compartilhado. Você deve usar bloqueio de arquivos ou outros processos para evitar a sobrescrita de dados em volumes compartilhados.
- Não é possível revogar o acesso ao PVC de origem removendo o `shareToNamespace` ou `shareFromNamespace` anotações ou exclusão das `TridentVolumeReference` CR. Para revogar o acesso, você deve excluir o PVC subordinado.
- Não é possível criar snapshots, clones ou espelhar volumes subordinados.

Para maiores informações

Para saber mais sobre acesso a volumes entre namespaces:

- Visite ["Compartilhamento de volumes entre namespaces: Dê as boas-vindas ao acesso a volumes entre namespaces."](#) .
- Veja a demonstração em ["NetAppTV"](#) .

Clonar volumes entre namespaces

Usando o Trident, você pode criar novos volumes utilizando volumes existentes ou snapshots de volumes de um namespace diferente dentro do mesmo cluster Kubernetes.

Pré-requisitos

Antes de clonar volumes, certifique-se de que os backends de origem e destino sejam do mesmo tipo e tenham a mesma classe de armazenamento.



A clonagem entre namespaces é suportada apenas para o `ontap-san` e `ontap-nas` drivers de armazenamento. Clones somente leitura não são suportados.

Início rápido

Você pode configurar a clonagem de volumes em apenas alguns passos.

1

Configure o PVC de origem para clonar o volume.

O proprietário do namespace de origem concede permissão para acessar os dados no PVC de origem.

2

Conceder permissão para criar uma solicitação de configuração (CR) no espaço de nomes de destino.

O administrador do cluster concede permissão ao proprietário do namespace de destino para criar o CR `TridentVolumeReference`.

3

Crie uma `TridentVolumeReference` no namespace de destino.

O proprietário do namespace de destino cria o CR `TridentVolumeReference` para se referir ao PVC de origem.

4

Crie o PVC clonado no namespace de destino.

O proprietário do namespace de destino cria um PVC para clonar o PVC do namespace de origem.

Configure os namespaces de origem e destino.

Para garantir a segurança, a clonagem de volumes entre namespaces requer a colaboração e a ação do proprietário do namespace de origem, do administrador do cluster e do proprietário do namespace de destino. A função do usuário é definida em cada etapa.

Passos

1. **Proprietário do namespace de origem:** Criar o PVC(`pvc1`) no espaço de nomes de origem(`namespace1`) que concede permissão para compartilhar com o espaço de nomes de destino(`namespace2`) usando o `cloneToNamespace` anotação.

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: pvc1
  namespace: namespace1
  annotations:
    trident.netapp.io/cloneToNamespace: namespace2
spec:
  accessModes:
    - ReadWriteMany
  storageClassName: trident-csi
  resources:
    requests:
      storage: 100Gi
```

A Trident cria o PV e seu volume de armazenamento de backend.



- Você pode compartilhar o PVC com vários namespaces usando uma lista separada por vírgulas. Por exemplo, `trident.netapp.io/cloneToNamespace: namespace2, namespace3, namespace4`.
- Você pode compartilhar com todos os namespaces usando `*`. Por exemplo, `trident.netapp.io/cloneToNamespace: *`
- Você pode atualizar o PVC para incluir o `cloneToNamespace` Anotações a qualquer momento.

2. **Administrador do cluster:** Certifique-se de que o RBAC adequado esteja implementado para conceder permissão ao proprietário do namespace de destino para criar o CR `TridentVolumeReference` no namespace de destino.(`namespace2`).

3. **Proprietário do namespace de destino:** Crie uma solicitação de configuração (CR) `TridentVolumeReference` no namespace de destino que faça referência ao namespace de origem. `pvc1`.

```
apiVersion: trident.netapp.io/v1
kind: TridentVolumeReference
metadata:
  name: my-first-tvr
  namespace: namespace2
spec:
  pvcName: pvc1
  pvcNamespace: namespace1
```

4. **Proprietário do namespace de destino:** Criar um PVC(`pvc2`) no espaço de nomes de destino(`namespace2`) usando o `cloneFromPVC` ou `cloneFromSnapshot`, e `cloneFromNamespace` Anotações para designar o PVC de origem.

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  annotations:
    trident.netapp.io/cloneFromPVC: pvc1
    trident.netapp.io/cloneFromNamespace: namespace1
  name: pvc2
  namespace: namespace2
spec:
  accessModes:
    - ReadWriteMany
  storageClassName: trident-csi
  resources:
    requests:
      storage: 100Gi
```

Limitações

- Para PVCs provisionadas usando drivers ontap-nas-economy, clones somente leitura não são suportados.

Replicar volumes usando o SnapMirror

O Trident suporta relações de espelhamento entre um volume de origem em um cluster e o volume de destino no cluster emparelhado para replicação de dados em casos de recuperação de desastres. Você pode usar uma Definição de Recurso Personalizado (CRD) com namespace, chamada Trident Mirror Relationship (TMR), para executar as seguintes operações:

- Criar relações de espelhamento entre volumes (PVCs)
- Remover relações de espelhamento entre volumes
- Quebre os relacionamentos espelhados
- Promover o volume secundário em situações de desastre (falhas).
- Realizar a transição sem perda de dados de aplicações de um cluster para outro (durante failovers ou migrações planejadas).

Pré-requisitos para replicação

Certifique-se de que os seguintes pré-requisitos sejam atendidos antes de começar:

Clusters ONTAP

- *** Trident***: A versão 22.10 ou posterior do Trident deve estar presente nos clusters Kubernetes de origem e destino que utilizam o ONTAP como backend.
- **Licenças**: As licenças assíncronas do ONTAP SnapMirror que utilizam o pacote Data Protection devem estar habilitadas nos clusters ONTAP de origem e destino. Consulte ["Visão geral do licenciamento do SnapMirror no ONTAP"](#) para mais informações.

A partir do ONTAP 9.10.1, todas as licenças são fornecidas como um arquivo de licença NetApp (NLF), que é um único arquivo que habilita vários recursos. Consulte ["Licenças incluídas no ONTAP One"](#) para mais informações.



Somente a proteção assíncrona SnapMirror é suportada.

Espionagem

- **Cluster e SVM**: Os backends de armazenamento ONTAP devem estar interligados. Consulte ["Visão geral do peering de clusters e SVMs"](#) para mais informações.



Certifique-se de que os nomes SVM usados na relação de replicação entre dois clusters ONTAP sejam únicos.

- *** Trident e SVM***: Os SVMs remotos emparelhados devem estar disponíveis para o Trident no cluster de destino.

Motoristas com suporte

O NetApp Trident oferece suporte à replicação de volumes com a tecnologia NetApp SnapMirror, utilizando classes de armazenamento com suporte dos seguintes drivers: **ontap-nas : NFS** **ontap-san : iSCSI**

ontap-san : FC ontap-san : NVMe/TCP (requer versão mínima do ONTAP 9.15.1)



A replicação de volumes usando o SnapMirror não é compatível com sistemas ASA r2. Para obter informações sobre sistemas ASA r2, consulte ["Saiba mais sobre os sistemas de armazenamento ASA r2"](#).

Crie um PVC espelhado

Siga estes passos e utilize os exemplos de CRD para criar uma relação de espelhamento entre os volumes primário e secundário.

Passos

1. Execute os seguintes passos no cluster Kubernetes primário:
 - a. Crie um objeto StorageClass com o `trident.netapp.io/replication: true` parâmetro.

Exemplo

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: csi-nas
provisioner: csi.trident.netapp.io
parameters:
  backendType: "ontap-nas"
  fsType: "nfs"
  trident.netapp.io/replication: "true"
```

- b. Crie um PVC com uma StorageClass criada anteriormente.

Exemplo

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: csi-nas
spec:
  accessModes:
  - ReadWriteMany
  resources:
    requests:
      storage: 1Gi
  storageClassName: csi-nas
```

- c. Crie uma relação de espelhamento (MirrorRelationship) com informações locais.

Exemplo

```
kind: TridentMirrorRelationship
apiVersion: trident.netapp.io/v1
metadata:
  name: csi-nas
spec:
  state: promoted
  volumeMappings:
  - localPVCName: csi-nas
```

O Trident obtém as informações internas do volume e o estado atual de proteção de dados (DP) do volume e, em seguida, preenche o campo de status do MirrorRelationship.

- d. Obtenha o CR TridentMirrorRelationship para obter o nome interno e o SVM do PVC.

```
kubectl get tmr csi-nas
```

```
kind: TridentMirrorRelationship
apiVersion: trident.netapp.io/v1
metadata:
  name: csi-nas
  generation: 1
spec:
  state: promoted
  volumeMappings:
  - localPVCName: csi-nas
status:
  conditions:
  - state: promoted
  localVolumeHandle:
"datavserver:trident_pvc_3bedd23c_46a8_4384_b12b_3c38b313c1e1"
  localPVCName: csi-nas
  observedGeneration: 1
```

2. Execute os seguintes passos no cluster Kubernetes secundário:
- Crie uma StorageClass com o parâmetro `trident.netapp.io/replication: true`.

Exemplo

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: csi-nas
provisioner: csi.trident.netapp.io
parameters:
  trident.netapp.io/replication: true
```

- b. Crie um relacionamento de espelhamento (MirrorRelationship) com informações de destino e origem.

Exemplo

```
kind: TridentMirrorRelationship
apiVersion: trident.netapp.io/v1
metadata:
  name: csi-nas
spec:
  state: established
  volumeMappings:
    - localPVCName: csi-nas
      remoteVolumeHandle:
        "datavserver:trident_pvc_3bedd23c_46a8_4384_b12b_3c38b313c1e1"
```

O Trident criará um relacionamento SnapMirror com o nome da política de relacionamento configurada (ou padrão para ONTAP) e o inicializará.

- c. Crie um PVC com uma StorageClass previamente criada para atuar como destino secundário (SnapMirror).

Exemplo

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: csi-nas
  annotations:
    trident.netapp.io/mirrorRelationship: csi-nas
spec:
  accessModes:
    - ReadWriteMany
resources:
  requests:
    storage: 1Gi
storageClassName: csi-nas
```

O Trident verificará a existência do CRD `TridentMirrorRelationship` e não conseguirá criar o volume caso o relacionamento não exista. Caso exista a relação, o Trident garantirá que o novo FlexVol volume seja colocado em uma SVM que esteja emparelhada com a SVM remota definida no `MirrorRelationship`.

Estados de replicação de volume

Uma relação de espelhamento Trident (TMR, do inglês `Trident Mirror Relationship`) é um CRD (Componente de Referência Completa) que representa uma das extremidades de uma relação de replicação entre PVCs (Componentes Variáveis Parciais). O TMR de destino possui um estado, que informa ao Trident qual é o estado desejado. O TMR de destino possui os seguintes estados:

- **Estabelecido:** o PVC local é o volume de destino de uma relação de espelhamento, e esta é uma nova relação.
- **Em destaque:** o PVC local é `ReadWrite` e pode ser montado, sem nenhuma relação de espelhamento em vigor no momento.
- **Reestabelecido:** o PVC local é o volume de destino de uma relação de espelhamento e também estava anteriormente nessa relação de espelhamento.
 - O estado restabelecido deve ser usado se o volume de destino já teve alguma relação com o volume de origem, pois ele sobrescreve o conteúdo do volume de destino.
 - O estado restabelecido falhará se o volume não estiver previamente relacionado com a fonte.

Promover o PVC secundário durante uma falha não planejada.

Execute a seguinte etapa no cluster Kubernetes secundário:

- Atualize o campo `spec.state` de `TridentMirrorRelationship` para `promoted`.

Promover o PVC secundário durante uma falha planejada

Durante uma migração (failover) planejada, execute os seguintes passos para promover o PVC secundário:

Passos

1. No cluster Kubernetes primário, crie um snapshot do PVC e aguarde até que o snapshot seja criado.
2. No cluster Kubernetes primário, crie o CR `SnapshotInfo` para obter detalhes internos.

Exemplo

```
kind: SnapshotInfo
apiVersion: trident.netapp.io/v1
metadata:
  name: csi-nas
spec:
  snapshot-name: csi-nas-snapshot
```

3. No cluster Kubernetes secundário, atualize o campo `spec.state` do CR `TridentMirrorRelationship` para `promoted` e o campo `spec.promotedSnapshotHandle` para o nome interno do snapshot.
4. No cluster Kubernetes secundário, confirme o status (campo `status.state`) do `TridentMirrorRelationship` como promovido.

Restaurar uma relação de espelhamento após uma falha.

Antes de restaurar a relação de espelhamento, escolha o lado que deseja definir como o novo lado principal.

Passos

1. No cluster Kubernetes secundário, verifique se os valores do campo *spec.remoteVolumeHandle* no *TridentMirrorRelationship* estão atualizados.
2. No cluster Kubernetes secundário, atualize o campo *spec.mirror* do *TridentMirrorRelationship* para *reestablished*.

Operações adicionais

O Trident suporta as seguintes operações nos volumes primário e secundário:

Replicar o PVC primário em um novo PVC secundário.

Certifique-se de já possuir um tubo de PVC primário e um tubo de PVC secundário.

Passos

1. Exclua os CRDs *PersistentVolumeClaim* e *TridentMirrorRelationship* do cluster secundário (de destino) estabelecido.
2. Exclua o CRD *TridentMirrorRelationship* do cluster primário (de origem).
3. Crie um novo CRD *TridentMirrorRelationship* no cluster primário (de origem) para o novo PVC secundário (de destino) que você deseja estabelecer.

Redimensionar um PVC espelhado, primário ou secundário

O PVC pode ser redimensionado normalmente; o ONTAP expandirá automaticamente qualquer flexbox de destino se a quantidade de dados exceder o tamanho atual.

Remover a replicação de um PVC

Para remover a replicação, execute uma das seguintes operações no volume secundário atual:

- Exclua o *MirrorRelationship* no PVC secundário. Isso rompe a relação de replicação.
- Ou, atualize o campo *spec.state* para *promoted*.

Excluir um PVC (que foi previamente espelhado)

O Trident verifica se há PVCs replicados e libera a relação de replicação antes de tentar excluir o volume.

Excluir um TMR

A exclusão de um TMR em um dos lados de um relacionamento espelhado faz com que o TMR restante passe para o estado *promovido* antes que o Trident conclua a exclusão. Se o TMR selecionado para exclusão já estiver no estado *promovido*, não haverá relação de espelhamento existente e o TMR será removido, e o Trident promoverá o PVC local para *Leitura/Gravação*. Essa exclusão libera os metadados do SnapMirror para o volume local no ONTAP. Caso este volume seja utilizado em uma relação de espelhamento no futuro, será necessário usar um novo TMR com um estado de replicação de volume *estabelecido* ao criar a nova relação de espelhamento.

Atualize os relacionamentos de espelhamento quando o ONTAP estiver online.

As relações de espelhamento podem ser atualizadas a qualquer momento após serem estabelecidas. Você pode usar o `state: promoted` ou `state: reestablished` campos para atualizar os relacionamentos. Ao promover um volume de destino para um volume ReadWrite regular, você pode usar `promotedSnapshotHandle` para especificar um snapshot específico para restaurar o volume atual.

Atualizar relações de espelhamento quando o ONTAP estiver offline

Você pode usar um CRD para executar uma atualização do SnapMirror sem que o Trident tenha conectividade direta com o cluster ONTAP . Consulte o seguinte exemplo de formato do TridentActionMirrorUpdate:

Exemplo

```
apiVersion: trident.netapp.io/v1
kind: TridentActionMirrorUpdate
metadata:
  name: update-mirror-b
spec:
  snapshotHandle: "pvc-1234/snapshot-1234"
  tridentMirrorRelationshipName: mirror-b
```

``status.state`` Reflete o estado do CRD TridentActionMirrorUpdate. Pode assumir um valor entre *Succeeded*, *In Progress* ou *Failed*.

Utilizar a topologia CSI

O Trident pode criar e anexar volumes seletivamente aos nós presentes em um cluster Kubernetes, utilizando o ["Recurso de topologia CSI"](#) .

Visão geral

Usando o recurso Topologia CSI, o acesso aos volumes pode ser limitado a um subconjunto de nós, com base em regiões e zonas de disponibilidade. Atualmente, os provedores de nuvem permitem que os administradores do Kubernetes criem nós baseados em zonas. Os nós podem estar localizados em diferentes zonas de disponibilidade dentro de uma região ou em várias regiões. Para facilitar o provisionamento de volumes para cargas de trabalho em uma arquitetura multizona, o Trident usa a topologia CSI.



Saiba mais sobre o recurso de Topologia CSI ["aqui"](#) .

O Kubernetes oferece dois modos exclusivos de vinculação de volumes:

- Com `VolumeBindingMode` definido para `Immediate` O Trident cria o volume sem qualquer conhecimento de topologia. A vinculação de volumes e o provisionamento dinâmico são tratados quando o PVC é criado. Este é o padrão `VolumeBindingMode` e é adequado para clusters que não impõem restrições de topologia. Volumes persistentes são criados sem qualquer dependência dos requisitos de agendamento do pod solicitante.
- Com `VolumeBindingMode` definido para `WaitForFirstConsumer` A criação e vinculação de um Volume Persistente para um PVC é adiada até que um pod que utilize o PVC seja agendado e criado. Dessa forma, os volumes são criados para atender às restrições de agendamento impostas pelos requisitos de topologia.



O `WaitForFirstConsumer` O modo de vinculação não requer rótulos de topologia. Isso pode ser usado independentemente do recurso de Topologia CSI.

O que você vai precisar

Para utilizar a topologia CSI, você precisa do seguinte:

- Um cluster Kubernetes executando um ["versão do Kubernetes suportada"](#)

```
kubectl version
Client Version: version.Info{Major:"1", Minor:"19",
GitVersion:"v1.19.3",
GitCommit:"1e11e4a2108024935ecfcb2912226cedeaafd99df",
GitTreeState:"clean", BuildDate:"2020-10-14T12:50:19Z",
GoVersion:"go1.15.2", Compiler:"gc", Platform:"linux/amd64"}
Server Version: version.Info{Major:"1", Minor:"19",
GitVersion:"v1.19.3",
GitCommit:"1e11e4a2108024935ecfcb2912226cedeaafd99df",
GitTreeState:"clean", BuildDate:"2020-10-14T12:41:49Z",
GoVersion:"go1.15.2", Compiler:"gc", Platform:"linux/amd64"}
```

- Os nós do cluster devem ter rótulos que introduzam informações sobre a topologia (`topology.kubernetes.io/region` e `topology.kubernetes.io/zone`). Essas etiquetas **devem estar presentes nos nós do cluster** antes da instalação do Trident para que ele reconheça a topologia.

```
kubectl get nodes -o=jsonpath='{range .items[*]}[{.metadata.name},
{.metadata.labels}]{ "\n"}{end}' | grep --color "topology.kubernetes.io"
[node1,
{"beta.kubernetes.io/arch":"amd64","beta.kubernetes.io/os":"linux","kubernetes.io/arch":"amd64","kubernetes.io/hostname":"node1","kubernetes.io/os":"linux","node-role.kubernetes.io/master":"","topology.kubernetes.io/region":"us-east1","topology.kubernetes.io/zone":"us-east1-a"}]
[node2,
{"beta.kubernetes.io/arch":"amd64","beta.kubernetes.io/os":"linux","kubernetes.io/arch":"amd64","kubernetes.io/hostname":"node2","kubernetes.io/os":"linux","node-role.kubernetes.io/worker":"","topology.kubernetes.io/region":"us-east1","topology.kubernetes.io/zone":"us-east1-b"}]
[node3,
{"beta.kubernetes.io/arch":"amd64","beta.kubernetes.io/os":"linux","kubernetes.io/arch":"amd64","kubernetes.io/hostname":"node3","kubernetes.io/os":"linux","node-role.kubernetes.io/worker":"","topology.kubernetes.io/region":"us-east1","topology.kubernetes.io/zone":"us-east1-c"}]
```

Passo 1: Crie um backend com reconhecimento de topologia

Os sistemas de armazenamento Trident podem ser configurados para provisionar volumes seletivamente com base em zonas de disponibilidade. Cada backend pode conter um opcional `supportedTopologies` Bloco que representa uma lista de zonas e regiões suportadas. Para `StorageClasses` que fazem uso desse backend, um volume só seria criado se solicitado por um aplicativo agendado em uma região/zona suportada.

Aqui está um exemplo de definição de backend:

YAML

```
---
version: 1
storageDriverName: ontap-san
backendName: san-backend-us-east1
managementLIF: 192.168.27.5
svm: iscsi_svm
username: admin
password: password
supportedTopologies:
  - topology.kubernetes.io/region: us-east1
    topology.kubernetes.io/zone: us-east1-a
  - topology.kubernetes.io/region: us-east1
    topology.kubernetes.io/zone: us-east1-b
```

JSON

```
{
  "version": 1,
  "storageDriverName": "ontap-san",
  "backendName": "san-backend-us-east1",
  "managementLIF": "192.168.27.5",
  "svm": "iscsi_svm",
  "username": "admin",
  "password": "password",
  "supportedTopologies": [
    {
      "topology.kubernetes.io/region": "us-east1",
      "topology.kubernetes.io/zone": "us-east1-a"
    },
    {
      "topology.kubernetes.io/region": "us-east1",
      "topology.kubernetes.io/zone": "us-east1-b"
    }
  ]
}
```



`supportedTopologies` É utilizado para fornecer uma lista de regiões e zonas por backend. Essas regiões e zonas representam a lista de valores permitidos que podem ser fornecidos em uma StorageClass. Para StorageClasses que contêm um subconjunto das regiões e zonas fornecidas em um backend, o Trident cria um volume no backend.

Você pode definir `supportedTopologies` por pool de armazenamento também. Veja o exemplo a seguir:


```

---
version: 1
storageDriverName: ontap-nas
backendName: nas-backend-us-centrall
managementLIF: 172.16.238.5
svm: nfs_svm
username: admin
password: password
supportedTopologies:
  - topology.kubernetes.io/region: us-centrall
    topology.kubernetes.io/zone: us-centrall-a
  - topology.kubernetes.io/region: us-centrall
    topology.kubernetes.io/zone: us-centrall-b
storage:
  - labels:
      workload: production
    supportedTopologies:
      - topology.kubernetes.io/region: us-centrall
        topology.kubernetes.io/zone: us-centrall-a
  - labels:
      workload: dev
    supportedTopologies:
      - topology.kubernetes.io/region: us-centrall
        topology.kubernetes.io/zone: us-centrall-b

```

Neste exemplo, o region e zone As etiquetas indicam a localização da piscina de armazenamento. topology.kubernetes.io/region e topology.kubernetes.io/zone ditam de onde os pools de armazenamento podem ser consumidos.

Etapa 2: Defina StorageClasses que levem em consideração a topologia.

Com base nos rótulos de topologia fornecidos aos nós do cluster, as StorageClasses podem ser definidas para conter informações de topologia. Isso determinará os pools de armazenamento que servem como candidatos para solicitações de PVC feitas e o subconjunto de nós que podem usar os volumes provisionados pelo Trident.

Veja o exemplo a seguir:

```

apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata: null
name: netapp-san-us-east1
provisioner: csi.trident.netapp.io
volumeBindingMode: WaitForFirstConsumer
allowedTopologies:
  - matchLabelExpressions: null
  - key: topology.kubernetes.io/zone
    values:
      - us-east1-a
      - us-east1-b
  - key: topology.kubernetes.io/region
    values:
      - us-east1
parameters:
  fsType: ext4

```

Na definição de StorageClass fornecida acima, volumeBindingMode está definido para WaitForFirstConsumer. Os PVCs solicitados com esta StorageClass não serão processados até que sejam referenciados em um pod. E, allowedTopologies Fornece as zonas e a região a serem utilizadas. O netapp-san-us-east1 O StorageClass cria PVCs no san-backend-us-east1 backend definido acima.

Etapa 3: Criar e usar um PVC

Com a StorageClass criada e mapeada para um backend, agora você pode criar PVCs.

Veja o exemplo spec abaixo:

```

---
kind: PersistentVolumeClaim
apiVersion: v1
metadata: null
name: pvc-san
spec: null
accessModes:
  - ReadWriteOnce
resources:
  requests:
    storage: 300Mi
storageClassName: netapp-san-us-east1

```

A criação de um PVC usando este manifesto resultaria no seguinte:

```

kubect1 create -f pvc.yaml
persistentvolumeclaim/pvc-san created
kubect1 get pvc
NAME          STATUS      VOLUME      CAPACITY    ACCESS MODES    STORAGECLASS
AGE
pvc-san      Pending                                netapp-san-us-east1
2s
kubect1 describe pvc
Name:          pvc-san
Namespace:     default
StorageClass:  netapp-san-us-east1
Status:        Pending
Volume:
Labels:        <none>
Annotations:   <none>
Finalizers:    [kubernetes.io/pvc-protection]
Capacity:
Access Modes:
VolumeMode:    Filesystem
Mounted By:    <none>
Events:
  Type      Reason              Age    From                                Message
  ----      -
  Normal    WaitForFirstConsumer  6s     persistentvolume-controller        waiting
for first consumer to be created before binding

```

Para que o Trident crie um volume e o fixe ao PVC, utilize o PVC em um invólucro. Veja o exemplo a seguir:

```

apiVersion: v1
kind: Pod
metadata:
  name: app-pod-1
spec:
  affinity:
    nodeAffinity:
      requiredDuringSchedulingIgnoredDuringExecution:
        nodeSelectorTerms:
          - matchExpressions:
              - key: topology.kubernetes.io/region
                operator: In
                values:
                  - us-east1
      preferredDuringSchedulingIgnoredDuringExecution:
        - weight: 1
          preference:
            matchExpressions:
              - key: topology.kubernetes.io/zone
                operator: In
                values:
                  - us-east1-a
                  - us-east1-b
  securityContext:
    runAsUser: 1000
    runAsGroup: 3000
    fsGroup: 2000
  volumes:
    - name: voll
      persistentVolumeClaim:
        claimName: pvc-san
  containers:
    - name: sec-ctx-demo
      image: busybox
      command: [ "sh", "-c", "sleep 1h" ]
      volumeMounts:
        - name: voll
          mountPath: /data/demo
      securityContext:
        allowPrivilegeEscalation: false

```

Este podSpec instrui o Kubernetes a agendar o pod em nós que estejam presentes no us-east1 região, e escolha qualquer nó que esteja presente na us-east1-a ou us-east1-b zonas.

Veja a seguinte saída:

```
kubectl get pods -o wide
```

NAME	READY	STATUS	RESTARTS	AGE	IP	NODE
NOMINATED	NODE	READINESS	GATES			
app-pod-1	1/1	Running	0	19s	192.168.25.131	node2
<none>		<none>				

```
kubectl get pvc -o wide
```

NAME	STATUS	VOLUME	CAPACITY
ACCESS MODES	STORAGECLASS	AGE	VOLUMEMODE
pvc-san	Bound	pvc-ecb1e1a0-840c-463b-8b65-b3d033e2e62b	300Mi
RWO		netapp-san-us-east1	48s
		Filesystem	

Atualizar os backends para incluir supportedTopologies

Os backends preexistentes podem ser atualizados para incluir uma lista de `supportedTopologies` usando `tridentctl backend update`. Isso não afetará os volumes que já foram provisionados e será usado apenas para PVCs subsequentes.

Encontre mais informações

- ["Gerenciar recursos para contêineres"](#)
- ["seletor de nó"](#)
- ["Afinidade e antiafinidade"](#)
- ["Manchas e Tolerâncias"](#)

Trabalhar com instantâneos

Os snapshots de volumes persistentes (PVs) do Kubernetes permitem a criação de cópias pontuais dos volumes. Você pode criar um snapshot de um volume criado usando o Trident, importar um snapshot criado fora do Trident, criar um novo volume a partir de um snapshot existente e recuperar dados de volumes a partir de snapshots.

Visão geral

O snapshot de volume é suportado por `ontap-nas`, `ontap-nas-flexgroup`, `ontap-san`, `ontap-san-economy`, `solidfire-san`, `gcp-cvs`, `azure-netapp-files`, e `google-cloud-netapp-volumes` motoristas.

Antes de começar

Para trabalhar com snapshots, você precisa de um controlador de snapshots externo e definições de recursos personalizados (CRDs). Essa é a responsabilidade do orquestrador do Kubernetes (por exemplo: Kubeadm, GKE, OpenShift).

Se a sua distribuição Kubernetes não incluir o controlador de snapshots e os CRDs, consulte [Implante um controlador de instantâneo de volume](#).



Não crie um controlador de snapshots se estiver criando snapshots de volume sob demanda em um ambiente GKE. O GKE utiliza um controlador de snapshots integrado e oculto.

Criar um instantâneo de volume

Passos

1. Criar um `VolumeSnapshotClass` Para obter mais informações, consulte "[VolumeSnapshotClass](#)".
 - O driver Aponta para o motorista do Trident CSI.
 - `deletionPolicy` pode ser `Delete` ou `Retain`. Quando definido para `Retain`, o snapshot físico subjacente no cluster de armazenamento é mantido mesmo quando o `VolumeSnapshot` O objeto foi excluído.

Exemplo

```
cat snap-sc.yaml
```

```
apiVersion: snapshot.storage.k8s.io/v1
kind: VolumeSnapshotClass
metadata:
  name: csi-snapclass
driver: csi.trident.netapp.io
deletionPolicy: Delete
```

2. Crie uma imagem instantânea de um tubo de PVC existente.

Exemplos

- Este exemplo cria um instantâneo de um PVC existente.

```
cat snap.yaml
```

```
apiVersion: snapshot.storage.k8s.io/v1
kind: VolumeSnapshot
metadata:
  name: pvc1-snap
spec:
  volumeSnapshotClassName: csi-snapclass
  source:
    persistentVolumeClaimName: pvc1
```

- Este exemplo cria um objeto de instantâneo de volume para um PVC chamado `pvc1` e o nome do instantâneo está definido como `pvc1-snap`. Um `VolumeSnapshot` é análogo a um PVC e está associado a um `VolumeSnapshotContent` objeto que representa a captura instantânea real.

```
kubectl create -f snap.yaml
volumesnapshot.snapshot.storage.k8s.io/pvc1-snap created

kubectl get volumesnapshots
NAME                                AGE
pvc1-snap                          50s
```

- Você pode identificar o `VolumeSnapshotContent` objeto para o `pvc1-snap`. O `VolumeSnapshot` é descrito dessa forma. O `Snapshot Content Name` identifica o objeto `VolumeSnapshotContent` que fornece este snapshot. O `Ready To Use` O parâmetro indica que o instantâneo pode ser usado para criar um novo PVC.

```
kubectl describe volumesnapshots pvc1-snap
Name:          pvc1-snap
Namespace:     default
...
Spec:
  Snapshot Class Name:    pvc1-snap
  Snapshot Content Name:  snapcontent-e8d8a0ca-9826-11e9-9807-
525400f3f660
  Source:
    API Group:
    Kind:      PersistentVolumeClaim
    Name:      pvc1
Status:
  Creation Time:  2019-06-26T15:27:29Z
  Ready To Use:   true
  Restore Size:   3Gi
...
```

Criar um PVC a partir de um instantâneo de volume

Você pode usar `dataSource` para criar um PVC usando um `VolumeSnapshot` chamado `<pvc-name>` como fonte dos dados. Após a fabricação do PVC, ele pode ser fixado a um invólucro e utilizado como qualquer outro PVC.



O PVC será criado no mesmo backend que o volume de origem. Consulte ["KB: A criação de um PVC a partir de um Snapshot de PVC do Trident não pode ser feita em um backend alternativo."](#)

O exemplo a seguir cria o PVC usando `pvc1-snap` como fonte de dados.

```
cat pvc-from-snap.yaml
```

```

apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: pvc-from-snap
spec:
  accessModes:
    - ReadWriteOnce
  storageClassName: golden
  resources:
    requests:
      storage: 3Gi
  dataSource:
    name: pvcl-snap
    kind: VolumeSnapshot
    apiGroup: snapshot.storage.k8s.io

```

Importar um instantâneo de volume

A Trident apoia o "[Processo de snapshot pré-provisionado do Kubernetes](#)" para permitir que o administrador do cluster crie um `VolumeSnapshotContent` snapshots de objetos e importações criados fora do Trident.

Antes de começar

O Trident deve ter criado ou importado o volume pai do snapshot.

Passos

1. **Administrador do cluster:** Criar um `VolumeSnapshotContent` objeto que faz referência ao snapshot do backend. Isso inicia o fluxo de trabalho de captura de instantâneos no Trident.
 - Especifique o nome do snapshot do backend em annotations como `trident.netapp.io/internalSnapshotName: <"backend-snapshot-name">`.
 - Especificar `<name-of-parent-volume-in-trident>/<volume-snapshot-content-name>` em `snapshotHandle` Esta é a única informação fornecida ao Trident pelo snapshotter externo. `ListSnapshots` chamar.



O `<volumeSnapshotContentName>` Nem sempre é possível que o nome do snapshot do backend corresponda ao nome original devido a restrições de nomenclatura do CR.

Exemplo

O exemplo a seguir cria um `VolumeSnapshotContent` objeto que faz referência a um snapshot do backend `snap-01`.


```

apiVersion: snapshot.storage.k8s.io/v1
kind: VolumeSnapshotContent
metadata:
  name: import-snap-content
  annotations:
    trident.netapp.io/internalSnapshotName: "snap-01" # This is the
name of the snapshot on the backend
spec:
  deletionPolicy: Retain
  driver: csi.trident.netapp.io
  source:
    snapshotHandle: pvc-f71223b5-23b9-4235-bbfe-e269ac7b84b0/import-
snap-content # <import PV name or source PV name>/<volume-snapshot-
content-name>
  volumeSnapshotRef:
    name: import-snap
    namespace: default

```

2. **Administrador do cluster:** Crie o VolumeSnapshot CR que faz referência ao VolumeSnapshotContent objeto. Este pedido permite o acesso para usar o VolumeSnapshot em um determinado espaço de nomes.

Exemplo

O exemplo a seguir cria um VolumeSnapshot CR nomeado import-snap que faz referência ao VolumeSnapshotContent nomeado import-snap-content.

```

apiVersion: snapshot.storage.k8s.io/v1
kind: VolumeSnapshot
metadata:
  name: import-snap
spec:
  # volumeSnapshotClassName: csi-snapclass (not required for pre-
provisioned or imported snapshots)
  source:
    volumeSnapshotContentName: import-snap-content

```

3. **Processamento interno (nenhuma ação necessária):** O snapshotter externo reconhece o recém-criado VolumeSnapshotContent e executa o ListSnapshots chamar. Trident cria o TridentSnapshot.
 - O capturador de instantâneos externo define o VolumeSnapshotContent para readyToUse e o VolumeSnapshot para true.
 - Trident retorna readyToUse=true.
4. **Qualquer usuário:** Criar um PersistentVolumeClaim para fazer referência ao novo VolumeSnapshot, onde o spec.dataSource (ou spec.dataSourceRef) o nome é o VolumeSnapshot nome.

Exemplo

O exemplo a seguir cria um PVC que faz referência a VolumeSnapshot nomeado `import-snap`.

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: pvc-from-snap
spec:
  accessModes:
    - ReadWriteOnce
  storageClassName: simple-sc
  resources:
    requests:
      storage: 1Gi
  dataSource:
    name: import-snap
    kind: VolumeSnapshot
    apiGroup: snapshot.storage.k8s.io
```

Recupere dados de volume usando snapshots

O diretório de snapshots fica oculto por padrão para facilitar a máxima compatibilidade dos volumes provisionados usando o `ontap-nas` e `ontap-nas-economy` motoristas. Ative o `.snapshot` diretório para recuperar dados diretamente de snapshots.

Utilize o comando `volume snapshot restore` da CLI do ONTAP para restaurar um volume a um estado registrado em um snapshot anterior.

```
cluster1::*> volume snapshot restore -vserver vs0 -volume vol3 -snapshot
vol3_snap_archive
```



Ao restaurar uma cópia de snapshot, a configuração de volume existente é sobrescrita. As alterações feitas nos dados do volume após a criação da cópia instantânea serão perdidas.

Restauração de volume in-place a partir de um snapshot

O Trident proporciona restauração de volume rápida e in situ a partir de uma imagem instantânea usando o `TridentActionSnapshotRestore` (TASR) CR. Essa solicitação de configuração (CR) funciona como uma ação imperativa do Kubernetes e não persiste após a conclusão da operação.

O Trident suporta a restauração de snapshots no `ontap-san`, `ontap-san-economy`, `ontap-nas`, `ontap-nas-flexgroup`, `azure-netapp-files`, `gcp-cvs`, `google-cloud-netapp-volumes`, e `solidfire-san` motoristas.

Antes de começar

Você precisa ter um PVC encadernado e um snapshot de volume disponível.

- Verifique se o status do PVC está vinculado.

```
kubectl get pvc
```

- Verifique se o snapshot do volume está pronto para uso.

```
kubectl get vs
```

Passos

1. Crie o TASR CR. Este exemplo cria um CR para PVC `pvc1` e instantâneo de volume `pvc1-snapshot`.



O TASR CR deve estar em um namespace onde o PVC e o VS existam.

```
cat tasr-pvc1-snapshot.yaml
```

```
apiVersion: trident.netapp.io/v1
kind: TridentActionSnapshotRestore
metadata:
  name: trident-snap
  namespace: trident
spec:
  pvcName: pvc1
  volumeSnapshotName: pvc1-snapshot
```

2. Aplique a solicitação de restauração (CR) para restaurar a partir do snapshot. Este exemplo restaura a partir de um snapshot. `pvc1`.

```
kubectl create -f tasr-pvc1-snapshot.yaml
```

```
tridentactionsnapshotrestore.trident.netapp.io/trident-snap created
```

Resultados

O Trident restaura os dados a partir do snapshot. Você pode verificar o status da restauração do snapshot:

```
kubectl get tasr -o yaml
```

```

apiVersion: trident.netapp.io/v1
items:
- apiVersion: trident.netapp.io/v1
  kind: TridentActionSnapshotRestore
  metadata:
    creationTimestamp: "2023-04-14T00:20:33Z"
    generation: 3
    name: trident-snap
    namespace: trident
    resourceVersion: "3453847"
    uid: <uid>
  spec:
    pvcName: pvc1
    volumeSnapshotName: pvc1-snapshot
  status:
    startTime: "2023-04-14T00:20:34Z"
    completionTime: "2023-04-14T00:20:37Z"
    state: Succeeded
kind: List
metadata:
  resourceVersion: ""

```



- Na maioria dos casos, o Trident não tentará novamente a operação automaticamente em caso de falha. Você precisará repetir a operação.
- Usuários do Kubernetes sem acesso de administrador podem precisar de permissão do administrador para criar um CR do TASR em seu namespace de aplicação.

Excluir um PV com snapshots associados

Ao excluir um Volume Persistente com snapshots associados, o volume Trident correspondente é atualizado para o estado "Excluindo". Remova os snapshots de volume para excluir o volume Trident .

Implante um controlador de instantâneo de volume

Se a sua distribuição Kubernetes não incluir o controlador de snapshots e os CRDs, você pode implantá-los da seguinte forma.

Passos

1. Criar CRDs de snapshot de volume.

```
cat snapshot-setup.sh
```

```
#!/bin/bash
# Create volume snapshot CRDs
kubectl apply -f https://raw.githubusercontent.com/kubernetes-csi/external-snapshotter/release-6.1/client/config/crd/snapshot.storage.k8s.io_volumesnapshotclasses.yaml
kubectl apply -f https://raw.githubusercontent.com/kubernetes-csi/external-snapshotter/release-6.1/client/config/crd/snapshot.storage.k8s.io_volumesnapshotcontents.yaml
kubectl apply -f https://raw.githubusercontent.com/kubernetes-csi/external-snapshotter/release-6.1/client/config/crd/snapshot.storage.k8s.io_volumesnapshots.yaml
```

2. Crie o controlador de instantâneo.

```
kubectl apply -f https://raw.githubusercontent.com/kubernetes-csi/external-snapshotter/release-6.1/deploy/kubernetes/snapshot-controller/rbac-snapshot-controller.yaml
```

```
kubectl apply -f https://raw.githubusercontent.com/kubernetes-csi/external-snapshotter/release-6.1/deploy/kubernetes/snapshot-controller/setup-snapshot-controller.yaml
```



Se necessário, abra `deploy/kubernetes/snapshot-controller/rbac-snapshot-controller.yaml` e atualizar namespace para o seu espaço de nomes.

Links relacionados

- ["Instantâneos de volume"](#)
- ["VolumeSnapshotClass"](#)

Trabalhar com snapshots de grupos de volumes

O NetApp Trident permite criar snapshots de vários volumes (um grupo de snapshots de volumes) em grupos de volumes do Kubernetes. Este instantâneo do grupo de volumes representa cópias de vários volumes feitas no mesmo momento.



VolumeGroupSnapshot é um recurso beta do Kubernetes com APIs beta. O Kubernetes 1.32 é a versão mínima necessária para o VolumeGroupSnapshot.

Criar snapshots de grupos de volumes

O snapshot do grupo de volumes é compatível com o `ontap-san` Driver compatível apenas com o protocolo

iSCSI, ainda não suportado com Fibre Channel (FCP) nem NVMe/TCP. Antes de começar

- Certifique-se de que sua versão do Kubernetes seja K8s 1.32 ou superior.
- Para trabalhar com snapshots, você precisa de um controlador de snapshots externo e definições de recursos personalizados (CRDs). Essa é a responsabilidade do orquestrador do Kubernetes (por exemplo: Kubeadm, GKE, OpenShift).

Se a sua distribuição Kubernetes não incluir o controlador de snapshots externo e os CRDs, consulte [Implante um controlador de instantâneo de volume](#).



Não crie um controlador de snapshots se estiver criando snapshots de grupos de volumes sob demanda em um ambiente GKE. O GKE utiliza um controlador de snapshots integrado e oculto.

- No arquivo YAML do controlador de snapshots, defina o `CSIVolumeGroupSnapshot`. Defina o recurso como 'true' para garantir que o snapshot do grupo de volumes esteja habilitado.
- Crie as classes de snapshot de grupo de volumes necessárias antes de criar um snapshot de grupo de volumes.
- Certifique-se de que todos os PVCs/volumes estejam no mesmo SVM para poder criar o `VolumeGroupSnapshot`.

Passos

- Crie uma classe `VolumeGroupSnapshotClass` antes de criar um `VolumeGroupSnapshot`. Para obter mais informações, consulte ["VolumeGroupSnapshotClass"](#).

```
apiVersion: groupsnapshot.storage.k8s.io/v1beta1
kind: VolumeGroupSnapshotClass
metadata:
  name: csi-group-snap-class
  annotations:
    kubernetes.io/description: "Trident group snapshot class"
driver: csi.trident.netapp.io
deletionPolicy: Delete
```

- Crie PVCs com os rótulos necessários usando as classes de armazenamento existentes ou adicione esses rótulos aos PVCs existentes.

O exemplo a seguir cria o PVC usando `pvc1-group-snap` como fonte de dados e rótulo `consistentGroupSnapshot: groupA`. Defina a chave e o valor do rótulo com base em suas necessidades.

```

kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: pvcl-group-snap
  labels:
    consistentGroupSnapshot: groupA
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 100Mi
  storageClassName: sc1-1

```

- Crie um VolumeGroupSnapshot com o mesmo rótulo.(consistentGroupSnapshot: groupA) especificado no PVC.

Este exemplo cria um instantâneo de um grupo de volumes:

```

apiVersion: groupsnapshot.storage.k8s.io/v1beta1
kind: VolumeGroupSnapshot
metadata:
  name: "vgs1"
  namespace: trident
spec:
  volumeGroupSnapshotClassName: csi-group-snap-class
  source:
    selector:
      matchLabels:
        consistentGroupSnapshot: groupA

```

Recupere dados de volume usando um instantâneo de grupo.

Você pode restaurar Volumes Persistentes individuais usando os snapshots individuais que foram criados como parte do Snapshot do Grupo de Volumes. Não é possível recuperar o Snapshot do Grupo de Volumes como uma unidade.

Utilize o comando `volume snapshot restore` da CLI do ONTAP para restaurar um volume a um estado registrado em um snapshot anterior.

```

cluster1::*> volume snapshot restore -vserver vs0 -volume vol3 -snapshot
vol3_snap_archive

```



Ao restaurar uma cópia de snapshot, a configuração de volume existente é sobrescrita. As alterações feitas nos dados do volume após a criação da cópia instantânea serão perdidas.

Restauração de volume in-place a partir de um snapshot

O Trident proporciona restauração de volume rápida e in situ a partir de uma imagem instantânea usando o `TridentActionSnapshotRestore` (TASR) CR. Essa solicitação de configuração (CR) funciona como uma ação imperativa do Kubernetes e não persiste após a conclusão da operação.

Para mais informações, consulte ["Restauração de volume in-place a partir de um snapshot"](#).

Excluir um PV com snapshots de grupo associados

Ao excluir um instantâneo de volume de grupo:

- Você pode excluir o grupo de snapshots `VolumeGroupSnapshots` como um todo, e não snapshots individuais dentro do grupo.
- Se os `PersistentVolumes` forem excluídos enquanto existir um snapshot para esse `PersistentVolume`, o Trident moverá esse volume para um estado de "exclusão", pois o snapshot deve ser removido antes que o volume possa ser removido com segurança.
- Se um clone tiver sido criado usando um snapshot agrupado e, em seguida, o grupo for excluído, uma operação de divisão no clone será iniciada e o grupo não poderá ser excluído até que a divisão seja concluída.

Implante um controlador de instantâneo de volume

Se a sua distribuição Kubernetes não incluir o controlador de snapshots e os CRDs, você pode implantá-los da seguinte forma.

Passos

1. Criar CRDs de snapshot de volume.

```
cat snapshot-setup.sh
```



```
#!/bin/bash
# Create volume snapshot CRDs
kubectl apply -f https://raw.githubusercontent.com/kubernetes-csi/external-snapshotter/release-8.2/client/config/crd/groupsnapshot.storage.k8s.io_volumegroupsnapshotclasses.yaml
kubectl apply -f https://raw.githubusercontent.com/kubernetes-csi/external-snapshotter/release-8.2/client/config/crd/groupsnapshot.storage.k8s.io_volumegroupsnapshotcontents.yaml
kubectl apply -f https://raw.githubusercontent.com/kubernetes-csi/external-snapshotter/release-8.2/client/config/crd/groupsnapshot.storage.k8s.io_volumegroupsnapshots.yaml
```

2. Crie o controlador de instantâneo.

```
kubectl apply -f https://raw.githubusercontent.com/kubernetes-csi/external-snapshotter/release-8.2/deploy/kubernetes/snapshot-controller/rbac-snapshot-controller.yaml
```

```
kubectl apply -f https://raw.githubusercontent.com/kubernetes-csi/external-snapshotter/release-8.2/deploy/kubernetes/snapshot-controller/setup-snapshot-controller.yaml
```



Se necessário, abra `deploy/kubernetes/snapshot-controller/rbac-snapshot-controller.yaml` e atualizar namespace para o seu espaço de nomes.

Links relacionados

- ["VolumeGroupSnapshotClass"](#)
- ["Instantâneos de volume"](#)

Informações sobre direitos autorais

Copyright © 2026 NetApp, Inc. Todos os direitos reservados. Impresso nos EUA. Nenhuma parte deste documento protegida por direitos autorais pode ser reproduzida de qualquer forma ou por qualquer meio — gráfico, eletrônico ou mecânico, incluindo fotocópia, gravação, gravação em fita ou storage em um sistema de recuperação eletrônica — sem permissão prévia, por escrito, do proprietário dos direitos autorais.

O software derivado do material da NetApp protegido por direitos autorais está sujeito à seguinte licença e isenção de responsabilidade:

ESTE SOFTWARE É FORNECIDO PELA NETAPP "NO PRESENTE ESTADO" E SEM QUAISQUER GARANTIAS EXPRESSAS OU IMPLÍCITAS, INCLUINDO, SEM LIMITAÇÕES, GARANTIAS IMPLÍCITAS DE COMERCIALIZAÇÃO E ADEQUAÇÃO A UM DETERMINADO PROPÓSITO, CONFORME A ISENÇÃO DE RESPONSABILIDADE DESTES DOCUMENTOS. EM HIPÓTESE ALGUMA A NETAPP SERÁ RESPONSÁVEL POR QUALQUER DANO DIRETO, INDIRETO, INCIDENTAL, ESPECIAL, EXEMPLAR OU CONSEQUENCIAL (INCLUINDO, SEM LIMITAÇÕES, AQUISIÇÃO DE PRODUTOS OU SERVIÇOS SOBRESSALIENTES; PERDA DE USO, DADOS OU LUCROS; OU INTERRUPÇÃO DOS NEGÓCIOS), INDEPENDENTEMENTE DA CAUSA E DO PRINCÍPIO DE RESPONSABILIDADE, SEJA EM CONTRATO, POR RESPONSABILIDADE OBJETIVA OU PREJUÍZO (INCLUINDO NEGLIGÊNCIA OU DE OUTRO MODO), RESULTANTE DO USO DESTES DOCUMENTOS, MESMO SE ADVERTIDA DA RESPONSABILIDADE DE TAL DANO.

A NetApp reserva-se o direito de alterar quaisquer produtos descritos neste documento, a qualquer momento e sem aviso. A NetApp não assume nenhuma responsabilidade nem obrigação decorrentes do uso dos produtos descritos neste documento, exceto conforme expressamente acordado por escrito pela NetApp. O uso ou a compra deste produto não representam uma licença sob quaisquer direitos de patente, direitos de marca comercial ou quaisquer outros direitos de propriedade intelectual da NetApp.

O produto descrito neste manual pode estar protegido por uma ou mais patentes dos EUA, patentes estrangeiras ou pedidos pendentes.

LEGENDA DE DIREITOS LIMITADOS: o uso, a duplicação ou a divulgação pelo governo estão sujeitos a restrições conforme estabelecido no subparágrafo (b)(3) dos Direitos em Dados Técnicos - Itens Não Comerciais no DFARS 252.227-7013 (fevereiro de 2014) e no FAR 52.227- 19 (dezembro de 2007).

Os dados aqui contidos pertencem a um produto comercial e/ou serviço comercial (conforme definido no FAR 2.101) e são de propriedade da NetApp, Inc. Todos os dados técnicos e software de computador da NetApp fornecidos sob este Contrato são de natureza comercial e desenvolvidos exclusivamente com despesas privadas. O Governo dos EUA tem uma licença mundial limitada, irrevogável, não exclusiva, intransferível e não sublicenciável para usar os Dados que estão relacionados apenas com o suporte e para cumprir os contratos governamentais desse país que determinam o fornecimento de tais Dados. Salvo disposição em contrário no presente documento, não é permitido usar, divulgar, reproduzir, modificar, executar ou exibir os dados sem a aprovação prévia por escrito da NetApp, Inc. Os direitos de licença pertencentes ao governo dos Estados Unidos para o Departamento de Defesa estão limitados aos direitos identificados na cláusula 252.227-7015(b) (fevereiro de 2014) do DFARS.

Informações sobre marcas comerciais

NETAPP, o logotipo NETAPP e as marcas listadas em <http://www.netapp.com/TM> são marcas comerciais da NetApp, Inc. Outros nomes de produtos e empresas podem ser marcas comerciais de seus respectivos proprietários.