



# Utilice Astra Trident

## Astra Trident

NetApp  
September 04, 2024

# Tabla de contenidos

- Utilice Astra Trident ..... 1
  - Prepare el nodo de trabajo ..... 1
  - Configurar y gestionar back-ends ..... 6
  - Crear y gestionar clases de almacenamiento ..... 128
  - Aprovisione y gestione volúmenes ..... 133

# Utilice Astra Trident

## Prepare el nodo de trabajo

Todos los nodos de trabajadores del clúster de Kubernetes deben poder montar los volúmenes que haya aprovisionado para los pods. Para preparar los nodos de trabajo, debe instalar las herramientas NFS, iSCSI o NVMe/TCP según haya seleccionado los controladores.

### Seleccionar las herramientas adecuadas

Si está utilizando una combinación de controladores, debe instalar todas las herramientas necesarias para sus controladores. Las versiones recientes de RedHat CoreOS tienen las herramientas instaladas de forma predeterminada.

#### Herramientas de NFS

Instale las herramientas NFS si utiliza: `ontap-nas`, `ontap-nas-economy`, `ontap-nas-flexgroup`, `azure-netapp-files`, `gcp-cvs`.

#### Herramientas iSCSI

Instale las herramientas iSCSI si utiliza: `ontap-san`, `ontap-san-economy`, `solidfire-san`.

#### Herramientas de NVMe

Instale las herramientas de NVMe si está utilizando `ontap-san` Para el protocolo de memoria no volátil rápida (NVMe) sobre TCP (NVMe/TCP).



Recomendamos ONTAP 9,12 o posterior para NVMe/TCP.

## Detección del servicio de nodos

Astra Trident intenta detectar automáticamente si el nodo puede ejecutar servicios iSCSI o NFS.



La detección de servicios de nodo identifica los servicios detectados, pero no garantiza que los servicios se configuren correctamente. Por el contrario, la ausencia de un servicio detectado no garantiza que se produzca un error en el montaje del volumen.

### Revisar los eventos

Astra Trident crea eventos para que el nodo identifique los servicios detectados. Para revisar estos eventos, ejecute:

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

### Revisar los servicios detectados

Astra Trident identifica los servicios habilitados para cada nodo en el CR del nodo de Trident. Para ver los servicios detectados, ejecute:

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

## Volúmenes de NFS

Instale las herramientas de NFS mediante los comandos del sistema operativo. Asegúrese de que el servicio NFS se haya iniciado durante el arranque.

### RHEL 8 O POSTERIOR

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

### Ubuntu

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



Reinicie los nodos de trabajo después de instalar las herramientas NFS para evitar que se produzcan fallos cuando conecte volúmenes a los contenedores.

## Volúmenes iSCSI

Astra Trident puede establecer automáticamente una sesión iSCSI, analizar LUN y detectar dispositivos multivía, darles formato y montarlos en un pod.

### Funcionalidades de reparación automática de iSCSI

En el caso de los sistemas ONTAP, Astra Trident ejecuta la reparación automática de iSCSI cada cinco minutos para:

1. **Identifique** el estado de sesión iSCSI deseado y el estado actual de la sesión iSCSI.
2. **Compare** el estado deseado al estado actual para identificar las reparaciones necesarias. Astra Trident determina las prioridades de reparación y cuándo deben anticiparse a las reparaciones.
3. **Realice las reparaciones** necesarias para devolver el estado actual de la sesión iSCSI al estado deseado de la sesión iSCSI.



Los registros de la actividad de reparación automática se encuentran en la `trident-main` Contenedor en el dosis de Demonset correspondiente. Para ver los registros, debe haber configurado `debug A "verdadero"` durante la instalación de Astra Trident.

Las funcionalidades de reparación automática de iSCSI de Astra Trident pueden ayudar a prevenir:

- Sesiones iSCSI obsoletas o poco saludables que podrían producirse después de un problema de conectividad de red. En caso de una sesión obsoleta, Astra Trident espera siete minutos antes de cerrar la sesión para restablecer la conexión con un portal.



Por ejemplo, si los secretos CHAP se rotaban en la controladora de almacenamiento y la red pierde la conectividad, podrían persistir los secretos CHAP antiguos (*obsoleta*). La reparación automática puede reconocer esto y restablecer automáticamente la sesión para aplicar los secretos CHAP actualizados.

- Faltan sesiones iSCSI
- Faltan LUN

## Instale las herramientas iSCSI

Instale las herramientas iSCSI mediante los comandos del sistema operativo.

### Antes de empezar

- Cada nodo del clúster de Kubernetes debe tener un IQN único. **Este es un requisito previo necesario.**
- Si utiliza RHCOS versión 4.5 o posterior, u otra distribución Linux compatible con RHEL, con `solidfire-san` Controlador y Element OS 12.5 o anterior, asegúrese de que el algoritmo de autenticación CHAP esté establecido en MD5 in `/etc/iscsi/iscsid.conf`. Los algoritmos CHAP SHA1, SHA-256 y SHA3-256 compatibles con FIPS están disponibles con Element 12.7.

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

- Cuando utilice nodos de trabajo que ejecutan RHEL/RedHat CoreOS con VP iSCSI, especifique el `discard MountOption` en `StorageClass` para realizar un reclamación de espacio en línea. Consulte "[Documentación de redhat](#)".

## RHEL 8 O POSTERIOR

1. Instale los siguientes paquetes del sistema:

```
sudo yum install -y lsscsi iscsi-initiator-utils sg3_utils device-  
mapper-multipath
```

2. Compruebe que la versión de iscsi-initiator-utils sea 6.2.0.874-2.el7 o posterior:

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

3. Configure el escaneo en manual:

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

4. Activar accesos múltiples:

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



Asegúrese etc/multipath.conf contiene find\_multipaths no inferior defaults.

5. Asegúrese de que así sea iscsid y.. multipathd están en ejecución:

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

6. Activar e iniciar iscsi:

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

## Ubuntu

1. Instale los siguientes paquetes del sistema:

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

2. Compruebe que la versión Open-iscsi sea 2.0.874-5ubuntu2.10 o posterior (para bionic) o 2.0.874-7.1ubuntu6.1 o posterior (para focal):

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

### 3. Configure el escaneo en manual:

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

### 4. Activar accesos múltiples:

```
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
```



Asegúrese etc/multipath.conf contiene find\_multipaths no inferior defaults.

### 5. Asegúrese de que así sea open-iscsi y.. multipath-tools están habilitadas y en ejecución:

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



Para Ubuntu 18.04, debe descubrir los puertos de destino con `iscsiadm` antes de comenzar `open-iscsi` Para que se inicie el daemon iSCSI. También puede modificar el `iscsi` servicio para empezar `iscsid` automáticamente.



Reinicie los nodos de trabajo después de instalar las herramientas iSCSI para evitar que se produzcan fallos cuando conecte volúmenes a contenedores.

## Volúmenes NVMe/TCP

Instale las herramientas NVMe mediante los comandos de su sistema operativo.



- NVMe requiere RHEL 9 o posterior.
- Si la versión del kernel de su nodo de Kubernetes es demasiado antigua o si el paquete NVMe no está disponible para la versión de kernel, es posible que deba actualizar la versión del kernel del nodo a una con el paquete 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 la instalación

Después de la instalación, compruebe que cada nodo del clúster de Kubernetes tenga un NQN único mediante el comando:

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



Astra Trident modifica el `ctrl_device_tmo` Valor para garantizar que NVMe no se rinda en el camino si deja de funcionar. No cambie esta configuración.

# Configurar y gestionar back-ends

## Configurar los back-ends

Un back-end define la relación entre Astra Trident y un sistema de almacenamiento. Le indica a Astra Trident cómo se comunica con ese sistema de almacenamiento y cómo debe aprovisionar volúmenes a partir de él.

Astra Trident ofrece automáticamente pools de almacenamiento a partir de los back-ends que cumplan los requisitos definidos por una clase de almacenamiento. Aprenda a configurar el back-end para el sistema de almacenamiento.

- ["Configure un back-end de Azure NetApp Files"](#)
- ["Configure un back-end de Cloud Volumes Service para Google Cloud Platform"](#)
- ["Configure un back-end de NetApp HCI o SolidFire"](#)
- ["Configure un back-end con controladores NAS ONTAP o Cloud Volumes ONTAP"](#)



- ["Configurar un back-end con controladores SAN ONTAP o Cloud Volumes ONTAP"](#)
- ["Utilice Astra Trident con Amazon FSX para ONTAP de NetApp"](#)

## Azure NetApp Files

### Configure un back-end de Azure NetApp Files

Puede configurar Azure NetApp Files como back-end de Astra Trident. Puede asociar volúmenes NFS y SMB con un back-end de Azure NetApp Files. Astra Trident también es compatible con la gestión de credenciales mediante identidades gestionadas para clústeres de Azure Kubernetes Services (AKS).

#### Información del controlador de Azure NetApp Files

Astra Trident proporciona los controladores de almacenamiento de Azure NetApp Files siguientes para comunicarse con el clúster. Los modos de acceso admitidos son: *ReadWriteOnce* (RWO), *ReadOnlyMany* (ROX), *ReadWriteMany* (RWX), *ReadWriteOncePod* (RWOP).

Controlador	Protocolo	VolumeMo de	Modos de acceso compatibles	Sistemas de archivos compatibles
azure-netapp-files	NFS SMB	Sistema de archivos	RWO, ROX, RWX, RWOP	nfs, smb

### Consideraciones

- El servicio Azure NetApp Files no admite volúmenes de menos de 100 GB. Astra Trident crea automáticamente volúmenes de 100 GiB si se solicita un volumen más pequeño.
- Astra Trident admite volúmenes de SMB montados en pods que se ejecutan solo en nodos de Windows.

#### Identidades administradas para AKS

Astra Trident es compatible ["identidades administradas"](#) Para clústeres de Azure Kubernetes Services. Para aprovechar la gestión de credenciales optimizada que ofrecen las identidades gestionadas, debe tener:

- Un clúster de Kubernetes puesto en marcha mediante AKS
- Identidades gestionadas configuradas en el clúster de kubernetes de AKS
- Astra Trident instalado que incluye el `cloudProvider` para especificar "Azure".

## Operador de Trident

Para instalar Astra Trident con el operador Trident, edite `tridentorchestrator_cr.yaml` para ajustar `cloudProvider` para "Azure". Por ejemplo:

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

## Timón

En el siguiente ejemplo se instalan conjuntos Astra Trident `cloudProvider` A Azure mediante la variable de entorno `$CP`:

```
helm install trident trident-operator-23.10.0-custom.tgz --create
--namespace --namespace <trident-namespace> --set cloudProvider=$CP
```

## `tridentctl`

En el siguiente ejemplo se instala Astra Trident sets `cloudProvider` marcar a. Azure:

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

## Prepárese para configurar un back-end de Azure NetApp Files

Antes de configurar el back-end de Azure NetApp Files, debe asegurarse de que se cumplan los siguientes requisitos.

### Requisitos previos para volúmenes NFS y SMB

Si utiliza Azure NetApp Files por primera vez o en una ubicación nueva, es necesario realizar alguna configuración inicial para configurar Azure NetApp Files y crear un volumen NFS. Consulte ["Azure: Configure Azure NetApp Files y cree un volumen NFS"](#).

Para configurar y utilizar un ["Azure NetApp Files"](#) back-end, necesita lo siguiente:



`subscriptionID`, `tenantID`, `clientID`, `location`, y `clientSecret` Son opcionales cuando se utilizan identidades administradas en un clúster AKS.

- Un pool de capacidad. Consulte ["Microsoft: Cree un pool de capacidad para Azure NetApp Files"](#).

- Una subred delegada en Azure NetApp Files. Consulte ["Microsoft: Delege una subred en Azure NetApp Files"](#).
- `subscriptionID` Desde una suscripción de Azure con Azure NetApp Files habilitado.
- `tenantID`, `clientID`, y `clientSecret` desde una ["Registro de aplicaciones"](#) En Azure Active Directory con permisos suficientes para el servicio Azure NetApp Files. El registro de aplicaciones debe usar:
  - El rol propietario o Colaborador ["Predefinidos por Azure"](#).
  - A. ["Rol Colaborador personalizado"](#) en el nivel de suscripción (`assignableScopes`) Con los siguientes permisos que están limitados únicamente a lo que Astra Trident necesita. Después de crear el rol personalizado, ["Asigne el rol mediante el portal de Azure"](#).

## Rol de contribuyente 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/MountTarge
ts/read",
          "Microsoft.Network/virtualNetworks/read",
          "Microsoft.Network/virtualNetworks/subnets/read",
          "Microsoft.Features/featureProviders/subscriptionFeatureRegistrat
ions/read",
          "Microsoft.Features/featureProviders/subscriptionFeatureRegistrat
```

```

ions/write",

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

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

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

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

```

- Azure location que contiene al menos uno "subred delegada". A partir de Trident 22.01, la location parámetro es un campo obligatorio en el nivel superior del archivo de configuración del back-end. Los valores de ubicación especificados en los pools virtuales se ignoran.

### Requisitos adicionales para volúmenes SMB

Para crear un volumen de SMB, debe tener lo siguiente:

- Active Directory configurado y conectado a Azure NetApp Files. Consulte ["Microsoft: Cree y gestione conexiones de Active Directory para Azure NetApp Files"](#).
- Un clúster de Kubernetes con un nodo de controladora Linux y al menos un nodo de trabajo de Windows que ejecuta Windows Server 2019. Astra Trident admite volúmenes de SMB montados en pods que se ejecutan solo en nodos de Windows.
- Al menos un secreto de Astra Trident que contiene sus credenciales de Active Directory para que Azure NetApp Files pueda autenticarse en Active Directory. Generar secreto smbcreds:

```

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

```

- Proxy CSI configurado como servicio de Windows. Para configurar un csi-proxy, consulte ["GitHub: Proxy CSI"](#) o ["GitHub: Proxy CSI para Windows"](#) Para nodos Kubernetes que se ejecutan en Windows.

## Opciones y ejemplos de configuración del back-end de Azure NetApp Files

Obtenga más información sobre las opciones de configuración de back-end NFS y SMB para Azure NetApp Files y revise los ejemplos de configuración.

### Opciones de configuración del back-end

Astra Trident utiliza la configuración de back-end (subred, red virtual, nivel de servicio y ubicación) para crear volúmenes de Azure NetApp Files en los pools de capacidad que están disponibles en la ubicación solicitada y que coincidan con el nivel de servicio y la subred solicitados.



Astra Trident no admite pools de capacidad de calidad de servicio manual.

Los back-ends de Azure NetApp Files proporcionan estas opciones de configuración.

Parámetro	Descripción	Predeterminado
version		Siempre 1
storageDriverName	Nombre del controlador de almacenamiento	"azure-netapp-files"
backendName	Nombre personalizado o el back-end de almacenamiento	Nombre del controlador + "_" + caracteres aleatorios
subscriptionID	El ID de suscripción de su suscripción de Azure  Opcional cuando se activan identidades gestionadas en un clúster de AKS.	
tenantID	El ID de inquilino de un registro de aplicación  Opcional cuando se activan identidades gestionadas en un clúster de AKS.	
clientID	El ID de cliente de un registro de aplicación  Opcional cuando se activan identidades gestionadas en un clúster de AKS.	
clientSecret	El secreto de cliente de un registro de aplicaciones  Opcional cuando se activan identidades gestionadas en un clúster de AKS.	
serviceLevel	Uno de Standard, Premium, o. Ultra	"" (aleatorio)

Parámetro	Descripción	Predeterminado
<code>location</code>	<p>Nombre de la ubicación de Azure donde se crearán los nuevos volúmenes</p> <p>Opcional cuando se activan identidades gestionadas en un clúster de AKS.</p>	
<code>resourceGroups</code>	Lista de grupos de recursos para filtrar los recursos detectados	"[]" (sin filtro)
<code>netappAccounts</code>	Lista de cuentas de NetApp para filtrar los recursos detectados	"[]" (sin filtro)
<code>capacityPools</code>	Lista de pools de capacidad para filtrar los recursos detectados	"[]" (sin filtro, aleatorio)
<code>virtualNetwork</code>	Nombre de una red virtual con una subred delegada	""
<code>subnet</code>	Nombre de una subred delegada a. <code>Microsoft.Netapp/volumes</code>	""
<code>networkFeatures</code>	<p>Puede que el conjunto de funciones de vnet para un volumen sea <code>Basic</code> o <code>Standard</code>.</p> <p>Las funciones de red no están disponibles en todas las regiones y es posible que tengan que activarse en una suscripción. Especificando <code>networkFeatures</code> cuando la funcionalidad no está habilitada, hace que no se pueda realizar el aprovisionamiento del volumen.</p>	""
<code>nfsMountOptions</code>	<p>Control preciso de las opciones de montaje NFS.</p> <p>Ignorada para volúmenes de SMB.</p> <p>Para montar volúmenes con NFS versión 4.1, incluya <code>nfsvers=4</code> En la lista de opciones de montaje delimitadas por comas para elegir NFS v4.1.</p> <p>Las opciones de montaje establecidas en una definición de clase de almacenamiento anulan las opciones de montaje establecidas en la configuración de back-end.</p>	"nfsvers=3"

Parámetro	Descripción	Predeterminado
limitVolumeSize	No se puede aprovisionar si el tamaño del volumen solicitado es superior a este valor	"" (no se aplica de forma predeterminada)
debugTraceFlags	Indicadores de depuración que se deben usar para la solución de problemas. Ejemplo: <code>\{"api": false, "method": true, "discovery": true\}</code> . No lo utilice a menos que esté solucionando problemas y necesite un volcado de registro detallado.	nulo
nasType	Configure la creación de volúmenes NFS o SMB.  Las opciones son <code>nfs</code> , <code>smb</code> o <code>nulo</code> . El valor predeterminado es <code>nulo</code> en volúmenes de NFS.	<code>nfs</code>



Para obtener más información sobre las funciones de red, consulte ["Configure las funciones de red para un volumen de Azure NetApp Files"](#).

### Permisos y recursos necesarios

Si recibes un error “No se han encontrado pools de capacidad” al crear una RVP, es probable que el registro de tu aplicación no tenga los permisos y recursos necesarios (subred, red virtual, pool de capacidad) asociados. Si la depuración está habilitada, Astra Trident registrará los recursos de Azure detectados cuando se cree el back-end. Compruebe que se está utilizando un rol adecuado.

Los valores para `resourceGroups`, `netappAccounts`, `capacityPools`, `virtualNetwork`, y `subnet` puede especificarse utilizando nombres cortos o completos. En la mayoría de las situaciones, se recomiendan nombres completos, ya que los nombres cortos pueden coincidir con varios recursos con el mismo nombre.

La `resourceGroups`, `netappAccounts`, y `capacityPools` los valores son filtros que restringen el conjunto de recursos detectados a los disponibles en este back-end de almacenamiento y pueden especificarse en cualquier combinación de estos. Los nombres completos siguen este formato:

Tipo	Formato
Grupo de recursos	<code>&lt;resource group&gt;</code>
Cuenta de NetApp	<code>&lt;resource group&gt;/&lt;netapp account&gt;</code>
Pool de capacidad	<code>&lt;resource group&gt;/&lt;netapp account&gt;/&lt;capacity pool&gt;</code>
Red virtual	<code>&lt;resource group&gt;/&lt;virtual network&gt;</code>
Subred	<code>&lt;resource group&gt;/&lt;virtual network&gt;/&lt;subnet&gt;</code>

### Aprovisionamiento de volúmenes

Puede controlar el aprovisionamiento de volúmenes predeterminado especificando las siguientes opciones en una sección especial del archivo de configuración. Consulte [Configuraciones de ejemplo](#) para obtener más



detalles.

Parámetro	Descripción	Predeterminado
exportRule	Reglas de exportación de volúmenes nuevos.  exportRule Debe ser una lista separada por comas con cualquier combinación de direcciones IPv4 o subredes IPv4 en notación CIDR.  Ignorada para volúmenes de SMB.	"0.0.0.0/0"
snapshotDir	Controla la visibilidad del directorio .snapshot	"falso"
size	El tamaño predeterminado de los volúmenes nuevos	"100 G"
unixPermissions	Los permisos unix de nuevos volúmenes (4 dígitos octal).  Ignorada para volúmenes de SMB.	"" (función de vista previa, requiere incluir en la lista blanca de suscripciones)

### Configuraciones de ejemplo

Los ejemplos siguientes muestran configuraciones básicas que dejan la mayoría de los parámetros en los valores predeterminados. Esta es la forma más sencilla de definir un back-end.

### Configuración mínima

Ésta es la configuración mínima absoluta del back-end. Con esta configuración, Astra Trident detecta todas sus cuentas de NetApp, pools de capacidad y subredes delegadas en Azure NetApp Files en la ubicación configurada, y coloca volúmenes nuevos en uno de esos pools y subredes de forma aleatoria. Porque `nasType` se omite, la `nfs` El valor predeterminado es aplicable, y el back-end aprovisionará para volúmenes NFS.

Esta configuración es ideal cuando solo se está empezando a usar Azure NetApp Files y probando cosas, pero en la práctica va a querer proporcionar un ámbito adicional para los volúmenes que aprovisiona.

```
---
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
```

## Identidades administradas para AKS

Esta configuración de backend omite `subscriptionID`, `tenantID`, `clientID`, y `clientSecret`, que son opcionales cuando se utilizan identidades gestionadas.

```
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
```

## Configuración de niveles de servicio específica con filtros de pools de capacidad

Esta configuración de back-end coloca volúmenes en las de Azure `eastus` ubicación en una `Ultra` pool de capacidad. Astra Trident detecta automáticamente todas las subredes delegadas en Azure NetApp Files en esa ubicación y coloca un volumen nuevo en una de ellas de forma aleatoria.

```
---
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
```

## Configuración avanzada

Esta configuración de back-end reduce aún más el alcance de la ubicación de volúmenes en una única subred y también modifica algunos valores predeterminados de aprovisionamiento de volúmenes.

```
---
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'
```

## Configuración de pool virtual

Esta configuración back-end define varios pools de almacenamiento en un único archivo. Esto resulta útil cuando hay varios pools de capacidad que admiten diferentes niveles de servicio y desea crear clases de almacenamiento en Kubernetes que representan estos. Se utilizaron etiquetas de pools virtuales para diferenciar los pools según *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
```

## Definiciones de clase de almacenamiento

Lo siguiente `StorageClass` las definiciones hacen referencia a los pools de almacenamiento anteriores.

## Definiciones de ejemplo mediante `parameter.selector` campo

Uso `parameter.selector` puede especificar para cada una de ellas `StorageClass` el pool virtual que se utiliza para alojar un volumen. Los aspectos definidos en el pool elegido serán el volumen.

```
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
```

## Definiciones de ejemplo de volúmenes SMB

Uso `nasType`, `node-stage-secret-name`, y `node-stage-secret-namespace`, Puede especificar un volumen SMB y proporcionar las credenciales necesarias de Active Directory.

## Configuración básica en el espacio de nombres predeterminado

```
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"
```

## Uso de diferentes secretos por espacio de nombres

```
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}
```

## Uso de diferentes secretos por volumen

```
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` Filtra los pools que admiten volúmenes SMB. `nasType: nfs` o `nasType: null` Filtros para pools NFS.

### Cree el back-end

Después de crear el archivo de configuración del back-end, ejecute el siguiente comando:

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

Si la creación del back-end falla, algo está mal con la configuración del back-end. Puede ver los registros para determinar la causa ejecutando el siguiente comando:

```
tridentctl logs
```

Después de identificar y corregir el problema con el archivo de configuración, puede ejecutar de nuevo el comando `create`.

## Configure un back-end de Cloud Volumes Service para Google Cloud

Descubra cómo configurar Cloud Volumes Service de NetApp para Google Cloud como back-end para su instalación de Astra Trident con las configuraciones de ejemplo proporcionadas.

### Detalles del controlador de Google Cloud

Astra Trident proporciona la `gcp-cvs` controlador para comunicarse con el clúster. Los modos de acceso admitidos son: *ReadWriteOnce* (RWO), *ReadOnlyMany* (ROX), *ReadWriteMany* (RWX), *ReadWriteOncePod* (RWOP).

Controlador	Protocolo	VolumeMode	Modos de acceso compatibles	Sistemas de archivos compatibles
<code>gcp-cvs</code>	NFS	Sistema de archivos	RWO, ROX, RWX, RWOP	<code>nfs</code>

### Obtenga más información sobre la compatibilidad de Astra Trident con Cloud Volumes Service para Google Cloud

Astra Trident puede crear volúmenes de Cloud Volumes Service en uno de dos "tipos de servicio":

- **CVS-Performance:** El tipo de servicio predeterminado Astra Trident. Este tipo de servicio optimizado para el rendimiento es más adecuado para cargas de trabajo de producción que valoran el rendimiento. El tipo de servicio CVS-Performance es una opción de hardware que admite volúmenes con un tamaño mínimo de 100 GiB. Puede elegir uno de "tres niveles de servicio":

- `standard`
- `premium`

- extreme

- **CVS:** El tipo de servicio CVS proporciona una alta disponibilidad zonal con niveles de rendimiento limitados a moderados. El tipo de servicio CVS es una opción de software que usa pools de almacenamiento para admitir volúmenes de solo 1 GiB. El pool de almacenamiento puede contener hasta 50 volúmenes en los que todos los volúmenes comparten la capacidad y el rendimiento del pool. Puede elegir uno de "dos niveles de servicio":

- standardsw

- zoneredundantstandardsw

## Lo que necesitará

Para configurar y usar el "Cloud Volumes Service para Google Cloud" back-end, necesita lo siguiente:

- Una cuenta de Google Cloud configurada con Cloud Volumes Service de NetApp
- Número de proyecto de su cuenta de Google Cloud
- Cuenta de servicio de Google Cloud con el `netappcloudvolumes.admin` función
- Archivo de claves API para la cuenta de Cloud Volumes Service

## Opciones de configuración del back-end

Cada back-end aprovisiona volúmenes en una única región de Google Cloud. Para crear volúmenes en otras regiones, se pueden definir back-ends adicionales.

Parámetro	Descripción	Predeterminado
<code>version</code>		Siempre 1
<code>storageDriverName</code>	Nombre del controlador de almacenamiento	"gcp-cvs"
<code>backendName</code>	Nombre personalizado o el back-end de almacenamiento	Nombre de controlador + "_ " + parte de la clave de API
<code>storageClass</code>	Parámetro opcional utilizado para especificar el tipo de servicio CVS.  Uso <code>software</code> Para seleccionar el tipo de servicio CVS. De lo contrario, Astra Trident asume el tipo de servicio CVS-Performance ( <code>hardware</code> ).	
<code>storagePools</code>	Solo tipo de servicio CVS. Parámetro opcional que se utiliza para especificar pools de almacenamiento para la creación del volumen.	
<code>projectNumber</code>	Número de proyecto de cuenta de Google Cloud. El valor está disponible en la página de inicio del portal de Google Cloud.	
<code>hostProjectNumber</code>	Se requiere si se utiliza una red VPC compartida. En este escenario, <code>projectNumber</code> es el proyecto de servicio, y <code>hostProjectNumber</code> es el proyecto anfitrión.	



Parámetro	Descripción	Predeterminado
apiRegion	<p>Región de Google Cloud en la que Astra Trident crea volúmenes de Cloud Volumes Service. Cuando se crean clústeres de Kubernetes de diversas regiones, se crean volúmenes en un apiRegion Se puede utilizar en cargas de trabajo programadas en nodos en varias regiones de Google Cloud.</p> <p>El tráfico entre regiones conlleva un coste adicional.</p>	
apiKey	<p>Clave de API para la cuenta de servicio de Google Cloud con el netappcloudvolumes.admin función.</p> <p>Incluye el contenido en formato JSON del archivo de clave privada de una cuenta de servicio de Google Cloud (copiado literal en el archivo de configuración de back-end).</p>	
proxyURL	<p>URL de proxy si se requiere servidor proxy para conectarse a la cuenta CVS. El servidor proxy puede ser un proxy HTTP o HTTPS.</p> <p>En el caso de un proxy HTTPS, se omite la validación de certificados para permitir el uso de certificados autofirmados en el servidor proxy.</p> <p>No se admiten los servidores proxy con autenticación habilitada.</p>	
nfsMountOptions	Control preciso de las opciones de montaje NFS.	"nfsvers=3"
limitVolumeSize	No se puede aprovisionar si el tamaño del volumen solicitado es superior a este valor.	"" (no se aplica de forma predeterminada)
serviceLevel	<p>El nivel de servicio CVS-Performance o CVS para nuevos volúmenes.</p> <p>Los valores de CVS-Performance son standard, premium, o. extreme.</p> <p>Los valores CVS son standardsw o. zoneredundantstandardsw.</p>	<p>El valor predeterminado de CVS-Performance es "estándar".</p> <p>El valor predeterminado de CVS es "standardsw".</p>
network	Se utiliza la red de Google Cloud para Cloud Volumes Service Volumes.	"predeterminado"
debugTraceFlags	<p>Indicadores de depuración que se deben usar para la solución de problemas. Ejemplo: <code>\{"api":false, "method":true\}</code>.</p> <p>No lo utilice a menos que esté solucionando problemas y necesite un volcado de registro detallado.</p>	nulo

Parámetro	Descripción	Predeterminado
allowedTopologies	<p>Para habilitar el acceso a varias regiones, se debe definir StorageClass para allowedTopologies debe incluir todas las regiones.</p> <p>Por ejemplo:</p> <ul style="list-style-type: none"> <li>- key: topology.kubernetes.io/region</li> </ul> <p>values:</p> <ul style="list-style-type: none"> <li>- us-east1</li> <li>- europe-west1</li> </ul>	

## Opciones de aprovisionamiento de volúmenes

Es posible controlar el aprovisionamiento de volúmenes predeterminado en la `defaults` sección del archivo de configuración.

Parámetro	Descripción	Predeterminado
exportRule	Las reglas de exportación de nuevos volúmenes. Debe ser una lista separada por comas con cualquier combinación de direcciones IPv4 o subredes IPv4 en notación CIDR.	"0.0.0.0/0"
snapshotDir	Acceso a la <code>.snapshot</code> directorio	"falso"
snapshotReserve	Porcentaje de volumen reservado para las Snapshot	"" (Aceptar CVS por defecto de 0)
size	<p>El tamaño de los volúmenes nuevos.</p> <p>CVS-Performance mínimo es 100 GIB.</p> <p>El mínimo de CVS es 1 GIB.</p>	<p>El tipo de servicio CVS-Performance se establece de manera predeterminada en "100GIB".</p> <p>El tipo de servicio CVS no establece un valor predeterminado, pero requiere un mínimo de 1 GIB.</p>

## Ejemplos de tipo de servicio CVS-Performance

Los siguientes ejemplos proporcionan ejemplos de configuraciones para el tipo de servicio CVS-Performance.



```
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
```



```
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
```



```

znHczZsrtrtHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m
XsYg6gyxy4zq7OlwWgLwGa==
-----END PRIVATE KEY-----
client_email: cloudvolumes-admin-sa@my-gcp-
project.iam.gserviceaccount.com
client_id: '123456789012345678901'
auth_uri: https://accounts.google.com/o/oauth2/auth
token_uri: https://oauth2.googleapis.com/token
auth_provider_x509_cert_url:
https://www.googleapis.com/oauth2/v1/certs
client_x509_cert_url:
https://www.googleapis.com/robot/v1/metadata/x509/cloudvolumes-admin-
sa%40my-gcp-project.iam.gserviceaccount.com
nfsMountOptions: vers=3,proto=tcp,timeo=600
defaults:
  snapshotReserve: '5'
  exportRule: 0.0.0.0/0
labels:
  cloud: gcp
  region: us-west2
storage:
- labels:
  performance: extreme
  protection: extra
  serviceLevel: extreme
  defaults:
  snapshotDir: 'true'
  snapshotReserve: '10'
  exportRule: 10.0.0.0/24
- labels:
  performance: extreme
  protection: standard
  serviceLevel: extreme
- labels:
  performance: premium
  protection: extra
  serviceLevel: premium
  defaults:
  snapshotDir: 'true'
  snapshotReserve: '10'
- labels:
  performance: premium
  protection: standard
  serviceLevel: premium
- labels:
  performance: standard

```



```
serviceLevel: standard
```

### **Definiciones de clases de almacenamiento**

Las siguientes definiciones de StorageClass se aplican al ejemplo de configuración de pool virtual. Uso `parameters.selector`, Puede especificar para cada clase de almacenamiento el pool virtual utilizado para alojar un volumen. Los aspectos definidos en el pool elegido serán el volumen.

## Ejemplo de clase de almacenamiento

```
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
```

- El primer tipo de almacenamiento (`cvs-extreme-extra-protection`) se asigna al primer grupo virtual. Se trata del único pool que ofrece un rendimiento extremo con una reserva Snapshot del 10%.
- El último tipo de almacenamiento (`cvs-extra-protection`) llama a cualquier agrupación de almacenamiento que ofrezca una reserva de instantáneas del 10%. Astra Trident decide qué pool virtual se selecciona y garantiza que se cumpla el requisito de reserva de Snapshot.

### Ejemplos de tipo de servicio CVS

Los siguientes ejemplos proporcionan configuraciones de ejemplo para el tipo de servicio CVS.

### Ejemplo 1: Configuración mínima

Esta es la configuración mínima del back-end mediante `storageClass` Para especificar el tipo de servicio CVS y el valor predeterminado `standardsw` nivel de servicio.

```
---
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-----
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzllzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzllzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzllzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzllzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzllzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzllzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzllzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzllzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzllzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzllzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzllzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzllzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzllzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzllzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzllzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzllzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzllzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzllzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzllzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzllzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzllzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzllzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzllzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzllzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzllzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzllzZE4jK3bl/qp8B4Kws8zX5ojY9m
    XsYg6gyxy4zq70lwWgLwGa==
    -----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
```

## Ejemplo 2: Configuración del pool de almacenamiento

Esta configuración de entorno de administración de ejemplo utiliza `storagePools` para configurar un pool de almacenamiento.

```
---
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-----
    MIIIEvAIBADANBgkqhkiG9w0BAQEFAASCbKwggSiAgEAAoIBAQDaT+Oui9FBAw19
    L1AGEkrYU5xd9K5NlO5jMkIFND5wCD+Nv+jd1GvtFRLaLK5RvXyF5wzvztmODNS+
    qtScpQ+5cFpQkuGtv9U9+N6qtuVYYO3b504Kp5CtqVPJCgMJaK2j8pZTIqUiMum/
    5/Y9oTbZrjAHSMsgJm2nHzFq2X0rQvMaHghI6ATm4DOuWx8XGWKGTGIPlc0qPqJlqS
    LLaWOH4VIZQZCAyW5IU99CAMwqHgdG0uhFNfCgMmED6PBUvVLsLvcq86X+QSWR9k
    ETqElj/sGCenPF7ti1DhGBFafd9hPnxg9PZY29ArEZwY9G/ZjZQX7WPgs0VvxiNR
    DxZRC3GXAgMBAECggEACn5c59bG/qnVEVI1CwMAalM5M2z09JFh1L1ljKwntNPj
    Vilw2eTW2+UE7HbJru/S7KQgA5Dnn9kvCraEahPRuddUMrD0vG4kTl/IODV6uFuk
    Y0sZfbqd4jMUQ21smvGsqFzwloYWS5qzO1W83ivXH/HW/iqkmY2eW+EPRS/hwSSu
    SscR+Soji7PB0BWSJhlV4yqYf3vcd/D95el2CVHfRCkL85DKumeZ+yHENpiXGZAE
    t8xSs4a500Pm6NHhevCw2a/UQ95/foXNUR450HtbjieJo5o+FF6EYZQGfU2ZHZO8
    37FBKuaJkdGW5xqaI9TL7aqkGkFMF4F2qvOZM+vy8QKBgQD4oVuOkJDlhkTHP86W
    esFlw1kpWyJR9ZA7LI0g/rVpslnX+XdDq0WQf4umdlNau5hYEH9LU6ZSGs1Xk3/B
    NHwR6OXFuqEKNiu83d0zSlHhTy7PZpOZdj5a/vVvQfPDMz7OvsqLRd7YCAbdzuQ0
    +Ahq0Ztwvg0HQ64hdW0ukpYRRwKBgQDgyHj98oqswoYuIa+pP1yS0pPwLmjwKyNm
    /HayzCp+Qjiiyy7Tzg8AUqlH1Ou83XbV428jvg7kDhO7PCCKFq+mMmfqHmTpb0Maq
    KpKnZg4ipsqPlyHNNEOrmcailXbwIhCLewMqMrggUiLOmCw4PscL5nK+4GKu2XE1
    jLqjWAZFMQKBgFHkQ9XXRAJ1kR3XpGHoGN890pZOkCVSrqju6aUef/5KY1FCt8ew
    F/+aIxM2iQsvmWQYOvVCnhuY/F2GfAQ7d0om3decuwI0CX/xy7PjHmLXa2uaZs4
    WR17sLduj62RqXRLX0c0QkwBiNFyHbRcpdkZJQujbyMhBa+7j7SxT4BtAoGAWMWT
    UucocRXZm/pdvz9wteNH3YDwnJLMxm1KC06qMXbBoYrliY4sm3ywJWMC+iCd/H8A
    Gecxd/xVu5mA2L2N3KMq18Zhz8Th0G5DwKyDRJgOQ0Q46yuNXOoYEjlo4Wjyk8Me
    +tlQ8iK98E0UmZnhTgfSpSNElbz2AqnzQ3MN9uECgYAqdvvdVPnKGfvdTz2DjyMoJ
    E89UIC41WjjJGmHsd8W65+3X0RwMzKMT6aZc5tK9J5dHvmWIETnbM+1TImdBbFga
    NWOC6f3r2xbGXHhaWSl+nobpTuvlo56ZRJVvV71FMsiddzMuHH8pxfgNjemwA4P
    ThDHcejv035NNV6Kyo00tA==
    -----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
```

## El futuro

Después de crear el archivo de configuración del back-end, ejecute el siguiente comando:

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

Si la creación del back-end falla, algo está mal con la configuración del back-end. Puede ver los registros para determinar la causa ejecutando el siguiente comando:

```
tridentctl logs
```

Después de identificar y corregir el problema con el archivo de configuración, puede ejecutar de nuevo el comando create.

## Configure un back-end de NetApp HCI o SolidFire

Descubre cómo crear y utilizar un back-end de Element con tu instalación de Astra Trident.

### Detalles del controlador de elementos

Astra Trident proporciona la `solidfire-san` el controlador de almacenamiento para comunicarse con el clúster. Los modos de acceso admitidos son: *ReadWriteOnce* (RWO), *ReadOnlyMany* (ROX), *ReadWriteMany* (RWX), *ReadWriteOncePod* (RWOP).

La `solidfire-san` el controlador de almacenamiento admite los modos de volumen *file* y *block*. Para la `Filesystem VolumeMode`, Astra Trident crea un volumen y crea un sistema de archivos. El tipo de sistema de archivos se especifica mediante `StorageClass`.

Controlador	Protocolo	Modo VolumeMode	Modos de acceso compatibles	Sistemas de archivos compatibles
solidfire-san	ISCSI	Bloque	RWO, ROX, RWX, RWOP	No hay sistema de archivos. Dispositivo de bloque RAW.
solidfire-san	ISCSI	Sistema de archivos	RWO, RWOP	xfs, ext3, ext4

## Antes de empezar

Necesitarás lo siguiente antes de crear un backend de elemento.

- Es un sistema de almacenamiento compatible que ejecuta el software Element.
- Credenciales a un usuario administrador del clúster o inquilino de HCI de NetApp/SolidFire que puede gestionar volúmenes.
- Todos sus nodos de trabajo de Kubernetes deben tener instaladas las herramientas iSCSI adecuadas. Consulte ["información de preparación del nodo de trabajo"](#).

## Opciones de configuración del back-end

Consulte la siguiente tabla para ver las opciones de configuración del back-end:

Parámetro	Descripción	Predeterminado
version		Siempre 1
storageDriverName	Nombre del controlador de almacenamiento	Siempre "solidfire-san"
backendName	Nombre personalizado o el back-end de almacenamiento	Dirección IP "SolidFire_" + almacenamiento (iSCSI)
Endpoint	MVIP para el clúster de SolidFire con credenciales de inquilino	
SVIP	La dirección IP y el puerto de almacenamiento (iSCSI)	
labels	Conjunto de etiquetas con formato JSON arbitrario que se aplica en los volúmenes.	""
TenantName	Nombre de inquilino que se va a usar (creado si no se encuentra)	
InitiatorIFace	Restringir el tráfico de iSCSI a una interfaz de host específica	"predeterminado"
UseCHAP	Utilice CHAP para autenticar iSCSI. Astra Trident utiliza CHAP.	verdadero
AccessGroups	Lista de ID de grupos de acceso que se van a usar	Busca el código de un grupo de acceso denominado "trident".



Parámetro	Descripción	Predeterminado
Types	Especificaciones de calidad de servicio	
limitVolumeSize	Error en el aprovisionamiento si el tamaño del volumen solicitado es superior a este valor	"" (no se aplica de forma predeterminada)
debugTraceFlags	Indicadores de depuración que se deben usar para la solución de problemas. Ejemplo, {"api":false, "method":true}	nulo



No utilizar `debugTraceFlags` a menos que esté solucionando problemas y necesite un volcado de registro detallado.

### Ejemplo 1: Configuración de back-end para `solidfire-san` controlador con tres tipos de volumen

Este ejemplo muestra un archivo de back-end mediante autenticación CHAP y modelado de tres tipos de volúmenes con garantías de calidad de servicio específicas. Lo más probable es que, a continuación, defina clases de almacenamiento para consumir cada una de ellas mediante el `IOPS` parámetro de clase de almacenamiento.

```

---
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

```

## Ejemplo 2: Configuración de clase de almacenamiento y de entorno de administración para solidfire-san controlador con pools virtuales

En este ejemplo, se muestra el archivo de definición del back-end configurado con pools virtuales junto con StorageClasses que les devuelve referencia.

Astra Trident copia las etiquetas presentes en un pool de almacenamiento a la LUN de almacenamiento del entorno de administración al aprovisionar. Para mayor comodidad, los administradores de almacenamiento pueden definir etiquetas por pool virtual y agrupar volúmenes por etiqueta.

En el archivo de definición de backend de ejemplo que se muestra a continuación, se establecen valores predeterminados específicos para todos los grupos de almacenamiento, que establecen el `type` En Silver. Los pools virtuales se definen en la `storage` sección. En este ejemplo, algunos pools de almacenamiento establecen su propio tipo, y algunos pools anulan los valores predeterminados definidos anteriormente.

```

---
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

```

Las siguientes definiciones de StorageClass se refieren a los pools virtuales anteriores. Con el

`parameters.selector` Field, cada clase de almacenamiento llama a qué pools virtuales se pueden utilizar para alojar un volumen. El volumen tendrá los aspectos definidos en el pool virtual elegido.

El primer tipo de almacenamiento (`solidfire-gold-four`) se asignará al primer grupo virtual. Este es el único pool que ofrece rendimiento de oro con un `Volume Type QoS` De oro. El último tipo de almacenamiento (`solidfire-silver`) llama a cualquier pool de almacenamiento que ofrezca un rendimiento elevado. Astra Trident decidirá qué pool virtual se selecciona y garantizará que se cumplan los requisitos de almacenamiento.

```
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"
```

## Obtenga más información

- ["Los grupos de acceso de volúmenes"](#)

## Controladores para SAN de ONTAP

### Información general del controlador de SAN de ONTAP

Obtenga información sobre la configuración de un back-end de ONTAP con controladores SAN de ONTAP y Cloud Volumes ONTAP.

### Información sobre el controlador de SAN de ONTAP

Astra Trident proporciona los siguientes controladores de almacenamiento SAN para comunicarse con el clúster de ONTAP. Los modos de acceso admitidos son: *ReadWriteOnce* (RWO), *ReadOnlyMany* (ROX), *ReadWriteMany* (RWX), *ReadWriteOncePod* (RWOP).



Si utiliza Astra Control para protección, recuperación y movilidad, lea [Compatibilidad de controladores Astra Control](#).

Controlador	Protocolo	VolumeMo de	Modos de acceso compatibles	Sistemas de archivos compatibles
ontap-san	ISCSI	Bloque	RWO, ROX, RWX, RWOP	Sin sistema de archivos; dispositivo de bloque sin procesar
ontap-san	ISCSI	Sistema de archivos	RWO, RWOP  ROX y RWX no están disponibles en el modo de volumen del sistema de archivos.	xf s, ext3, ext4
ontap-san	NVMe/TCP  Consulte <a href="#">Consideraciones adicionales para NVMe/TCP</a> .	Bloque	RWO, ROX, RWX, RWOP	Sin sistema de archivos; dispositivo de bloque sin procesar
ontap-san	NVMe/TCP  Consulte <a href="#">Consideraciones adicionales para NVMe/TCP</a> .	Sistema de archivos	RWO, RWOP  ROX y RWX no están disponibles en el modo de volumen del sistema de archivos.	xf s, ext3, ext4

Controlador	Protocolo	VolumenMo de	Modos de acceso compatibles	Sistemas de archivos compatibles
ontap-san-economy	ISCSI	Bloque	RWO, ROX, RWX, RWOP	Sin sistema de archivos; dispositivo de bloque sin procesar
ontap-san-economy	ISCSI	Sistema de archivos	RWO, RWOP  ROX y RWX no están disponibles en el modo de volumen del sistema de archivos.	xfst, ext3, ext4

### Compatibilidad de controladores Astra Control

Astra Control proporciona una protección fluida, recuperación ante desastres y movilidad (mover volúmenes entre clústeres de Kubernetes) para los volúmenes creados con el `ontap-nas`, `ontap-nas-flexgroup`, y `ontap-san` de windows Consulte "[Requisitos previos de replicación de Astra Control](#)" para obtener más detalles.



- Uso `ontap-san-economy` solo si se espera que el número de uso de volúmenes persistentes sea superior a "[Límites de volumen ONTAP compatibles](#)".
- Uso `ontap-nas-economy` solo si se espera que el número de uso de volúmenes persistentes sea superior a "[Límites de volumen ONTAP compatibles](#)" y la `ontap-san-economy` no se puede utilizar el conductor.
- No utilizar `ontap-nas-economy` si prevé la necesidad de protección de datos, recuperación ante desastres o movilidad.

### Permisos de usuario

Astra Trident espera que se ejecute como administrador de ONTAP o SVM, normalmente mediante el `admin` usuario del clúster o un `vsadmin` Usuario de SVM o un usuario con un nombre diferente que tenga el mismo rol. Para puestas en marcha de Amazon FSX para ONTAP de NetApp, Astra Trident espera que se ejecute como administrador de ONTAP o SVM, mediante el clúster `fsxadmin` usuario o un `vsadmin` Usuario de SVM o un usuario con un nombre diferente que tenga el mismo rol. La `fsxadmin` el usuario es un reemplazo limitado para el usuario administrador del clúster.



Si utiliza la `limitAggregateUsage` parámetro, se necesitan permisos de administrador de clúster. Cuando se utiliza Amazon FSX para ONTAP de NetApp con Astra Trident, el `limitAggregateUsage` el parámetro no funciona con el `vsadmin` y `fsxadmin` cuentas de usuario. La operación de configuración generará un error si se especifica este parámetro.

Si bien es posible crear un rol más restrictivo dentro de ONTAP que puede utilizar un controlador Trident, no lo recomendamos. La mayoría de las nuevas versiones de Trident denominan API adicionales que se tendrían que tener en cuenta, por lo que las actualizaciones son complejas y propensas a errores.

### Consideraciones adicionales para NVMe/TCP

Astra Trident admite el protocolo de memoria no volátil rápida (NVMe) mediante el `ontap-san` controlador

incluyendo:

- IPv6
- Snapshots y clones de volúmenes NVMe
- Cambiar el tamaño de un volumen NVMe
- Se importa un volumen NVMe que se creó fuera de Astra Trident para que Astra Trident gestione su ciclo de vida
- Multivía nativa de NVMe
- Cierre correcto o sin complicaciones de los K8s nodos (23,10)

Astra Trident no es compatible:

- DH-HMAC-CHAP que es compatible con NVMe de forma nativa
- Rutas múltiples del asignador de dispositivos (DM)
- Cifrado LUKS

## Prepárese para configurar el back-end con los controladores SAN de ONTAP

Conozca los requisitos y las opciones de autenticación para configurar un back-end de ONTAP con controladores SAN de ONTAP.

### Requisitos

Para todos los back-ends de ONTAP, Astra Trident requiere al menos un agregado asignado a la SVM.

Recuerde que también puede ejecutar más de un controlador y crear clases de almacenamiento que señalen a uno o a otro. Por ejemplo, puede configurar un `san-dev` clase que utiliza `ontap-san` controlador y a `san-default` clase que utiliza `ontap-san-economy` uno.

Todos sus nodos de trabajo de Kubernetes deben tener instaladas las herramientas iSCSI adecuadas. Consulte "[Prepare el nodo de trabajo](#)" para obtener más detalles.

### Autentique el backend de ONTAP

Astra Trident ofrece dos modos de autenticación de un back-end de ONTAP.

- Basado en credenciales: El nombre de usuario y la contraseña de un usuario ONTAP con los permisos requeridos. Se recomienda utilizar un rol de inicio de sesión de seguridad predefinido, como `admin` o `vsadmin`. Garantizar la máxima compatibilidad con versiones de ONTAP.
- Basado en certificados: Astra Trident también puede comunicarse con un clúster de ONTAP mediante un certificado instalado en el back-end. Aquí, la definición de backend debe contener valores codificados en Base64 del certificado de cliente, la clave y el certificado de CA de confianza si se utiliza (recomendado).

Puede actualizar los back-ends existentes para moverse entre métodos basados en credenciales y basados en certificados. Sin embargo, solo se admite un método de autenticación a la vez. Para cambiar a un método de autenticación diferente, debe eliminar el método existente de la configuración del back-end.



Si intenta proporcionar **tanto credenciales como certificados**, la creación de backend fallará y se producirá un error en el que se haya proporcionado más de un método de autenticación en el archivo de configuración.



## Habilite la autenticación basada en credenciales

Astra Trident requiere las credenciales a un administrador con ámbito de SVM o clúster para comunicarse con el back-end de ONTAP. Se recomienda utilizar funciones estándar predefinidas como `admin` o `vsadmin`. De este modo se garantiza la compatibilidad con futuras versiones de ONTAP que puedan dar a conocer API de funciones que podrán utilizarse en futuras versiones de Astra Trident. Se puede crear y utilizar una función de inicio de sesión de seguridad personalizada con Astra Trident, pero no es recomendable.

Una definición de backend de ejemplo tendrá este aspecto:

### 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"
}
```

Tenga en cuenta que la definición de backend es el único lugar en el que las credenciales se almacenan en texto sin formato. Una vez creado el back-end, los nombres de usuario y las contraseñas se codifican con Base64 y se almacenan como secretos de Kubernetes. La creación o actualización de un backend es el único paso que requiere conocimiento de las credenciales. Por tanto, es una operación de solo administración que deberá realizar el administrador de Kubernetes o almacenamiento.

## Habilite la autenticación basada en certificados

Los back-ends nuevos y existentes pueden utilizar un certificado y comunicarse con el back-end de ONTAP. Se necesitan tres parámetros en la definición de backend.

- `ClientCertificate`: Valor codificado en base64 del certificado de cliente.
- `ClientPrivateKey`: Valor codificado en base64 de la clave privada asociada.
- `TrustedCACertificate`: Valor codificado en base64 del certificado de CA de confianza. Si se utiliza una CA

de confianza, se debe proporcionar este parámetro. Esto se puede ignorar si no se utiliza ninguna CA de confianza.

Un flujo de trabajo típico implica los pasos siguientes.

### Pasos

1. Genere una clave y un certificado de cliente. Al generar, establezca el nombre común (CN) en el usuario de ONTAP para autenticarse como.

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

2. Añada un certificado de CA de confianza al clúster ONTAP. Es posible que ya sea gestionado por el administrador de almacenamiento. Ignore si no se utiliza ninguna CA de confianza.

```
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 el certificado y la clave de cliente (desde el paso 1) en el clúster 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 los compatibilidad con el rol de inicio de sesión de seguridad ONTAP `cert` método de autenticación.

```
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. Probar la autenticación mediante un certificado generado. Reemplace `<LIF de gestión de ONTAP>` y `<vserver name>` por la IP de LIF de gestión y el nombre de 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 certificados, claves y certificados de CA de confianza con Base64.

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

## 7. Cree un backend utilizando los valores obtenidos del paso anterior.

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

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

### Actualice los métodos de autenticación o gire las credenciales

Puede actualizar un back-end existente para utilizar un método de autenticación diferente o para rotar sus credenciales. Esto funciona de las dos maneras: Los back-ends que utilizan nombre de usuario/contraseña se pueden actualizar para usar certificados. Los back-ends que utilizan certificados pueden actualizarse a nombre de usuario/contraseña. Para ello, debe eliminar el método de autenticación existente y agregar el nuevo método de autenticación. A continuación, utilice el archivo backend.json actualizado que contiene los parámetros necesarios para ejecutarse `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 |
+-----+-----+-----+
+-----+-----+

```



Quando gira contraseñas, el administrador de almacenamiento debe actualizar primero la contraseña del usuario en ONTAP. A esto le sigue una actualización de back-end. Al rotar certificados, se pueden agregar varios certificados al usuario. A continuación, el back-end se actualiza para usar el nuevo certificado, siguiendo el cual se puede eliminar el certificado antiguo del clúster de ONTAP.

La actualización de un back-end no interrumpe el acceso a los volúmenes que se han creado ni afecta a las conexiones de volúmenes realizadas después. Una actualización de back-end correcta indica que Astra Trident puede comunicarse con el back-end de ONTAP y gestionar futuras operaciones de volúmenes.

#### **Autentica conexiones con CHAP bidireccional**

Astra Trident puede autenticar sesiones iSCSI con CHAP bidireccional para `ontap-san` y `ontap-san-economy` de windows. Esto requiere habilitar el `useCHAP` opción en su definición de backend. Cuando se establece en `true`, Astra Trident configura la seguridad del iniciador predeterminado de la SVM en CHAP bidireccional y establece el nombre de usuario y los secretos del archivo backend. NetApp recomienda utilizar CHAP bidireccional para autenticar las conexiones. Consulte la siguiente configuración de ejemplo:

```
---
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
```



La `useCHAP` Parameter es una opción booleana que solo se puede configurar una vez. De forma predeterminada, se establece en `FALSE`. Después de configurarlo en `true`, no puede establecerlo en `false`.

Además de `useCHAP=true`, la `chapInitiatorSecret`, `chapTargetInitiatorSecret`, `chapTargetUsername`, y `chapUsername` los campos deben incluirse en la definición del backend. Los secretos se pueden cambiar después de crear un back-end ejecutando `tridentctl update`.

### Cómo funciona

Mediante ajuste `useCHAP` Para `true`, el administrador de almacenamiento ordena a Astra Trident que configure CHAP en el back-end de almacenamiento. Esto incluye lo siguiente:

- Configuración de CHAP en la SVM:
  - Si el tipo de seguridad de iniciador predeterminado de la SVM es `none` (establecido de forma predeterminada) **y** no hay LUN preexistentes ya presentes en el volumen, Astra Trident establecerá el tipo de seguridad predeterminado en `CHAP` Y continúe configurando el iniciador de CHAP, el nombre de usuario y los secretos de destino.
  - Si la SVM contiene LUN, Astra Trident no habilitará CHAP en la SVM. De este modo se garantiza que no se restrinja el acceso a las LUN que ya están presentes en la SVM.
- Configurar el iniciador de CHAP, el nombre de usuario y los secretos de destino; estas opciones deben especificarse en la configuración del back-end (como se muestra más arriba).

Una vez creado el back-end, Astra Trident crea una correspondiente `tridentbackend` CRD y almacena los secretos y nombres de usuario de CHAP como secretos de Kubernetes. Todos los VP creados por Astra Trident en este back-end se montarán y se conectan mediante CHAP.

### Rotar las credenciales y actualizar los back-ends

Para actualizar las credenciales de CHAP, se deben actualizar los parámetros de CHAP en `backend.json` archivo. Para ello, será necesario actualizar los secretos CHAP y utilizar el `tridentctl update` comando para reflejar estos cambios.



Al actualizar los secretos CHAP para un back-end, debe utilizar `tridentctl` para actualizar el back-end. No actualice las credenciales en el clúster de almacenamiento a través de la interfaz de usuario de CLI/ONTAP, ya que Astra Trident no podrá recoger estos cambios.

```
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 |
+-----+-----+-----+-----+
+-----+-----+
```

Las conexiones existentes no se verán afectadas; seguirán activas si Astra Trident actualiza las credenciales en la SVM. Las nuevas conexiones utilizarán las credenciales actualizadas y las conexiones existentes seguirán activas. Al desconectar y volver a conectar los VP antiguos, se utilizarán las credenciales actualizadas.

## Opciones y ejemplos de configuración DE SAN ONTAP

Descubre cómo crear y utilizar controladores SAN de ONTAP con tu instalación de Astra Trident. Esta sección proporciona ejemplos de configuración de backend y detalles para la asignación de back-ends a StorageClasses.

### Opciones de configuración del back-end

Consulte la siguiente tabla para ver las opciones de configuración del back-end:

Parámetro	Descripción	Predeterminado
version		Siempre 1
storageDrive rName	Nombre del controlador de almacenamiento	ontap-nas, ontap-nas- economy, ontap-nas- flexgroup, ontap-san, ontap- san-economy
backendName	Nombre personalizado o el back-end de almacenamiento	Nombre de controlador + «_» + LIF de datos
managementLIF	<p>La dirección IP de un clúster o una LIF de gestión de SVM.</p> <p>Se puede especificar un nombre de dominio completo (FQDN).</p> <p>Puede configurarse para que utilice direcciones IPv6 si Astra Trident se instaló mediante la marca IPv6. Las direcciones IPv6 deben definirse entre corchetes, como [28e8:d9fb:a825:b7bf:69a8:d02f:9e7b:3555].</p> <p>Para un cambio de MetroCluster fluido, consulte <a href="#">Ejemplo de MetroCluster</a>.</p>	“10.0.0.1”, “[2001:1234:abcd::fefe]”
dataLIF	<p>Dirección IP de LIF de protocolo.</p> <p><b>No especifique para iSCSI.</b> Astra Trident utiliza <a href="#">"Asignación de LUN selectiva de ONTAP"</a> Para descubrir los LIF iSCSI necesarios para establecer una sesión de ruta múltiple. Se genera una advertencia if <code>dataLIF</code> se define explícitamente.</p> <p><b>Omitir para MetroCluster.</b> Ver <a href="#">Ejemplo de MetroCluster</a>.</p>	Derivado del SVM
svm	<p>Máquina virtual de almacenamiento que usar</p> <p><b>Omitir para MetroCluster.</b> Ver <a href="#">Ejemplo de MetroCluster</a>.</p>	Derivado si una SVM <code>managementLIF</code> está especificado
useCHAP	<p>Use CHAP para autenticar iSCSI para los controladores SAN de ONTAP [Boolean].</p> <p>Establezca en <code>true</code> Para Astra Trident, configure y utilice CHAP bidireccional como autenticación predeterminada para la SVM proporcionada en el back-end. Consulte <a href="#">"Prepárese para configurar el back-end con los controladores SAN de ONTAP"</a> para obtener más detalles.</p>	false
chapInitiatorSecret	Secreto CHAP del iniciador. Obligatorio si <code>useCHAP=true</code>	""

Parámetro	Descripción	Predeterminado
labels	Conjunto de etiquetas con formato JSON arbitrario que se aplica en los volúmenes	""
chapTargetInitiatorSecret	Secreto CHAP del iniciador de destino. Obligatorio si useCHAP=true	""
chapUsername	Nombre de usuario entrante. Obligatorio si useCHAP=true	""
chapTargetUsername	Nombre de usuario de destino. Obligatorio si useCHAP=true	""
clientCertificate	Valor codificado en base64 del certificado de cliente. Se utiliza para autenticación basada en certificados	""
clientPrivateKey	Valor codificado en base64 de la clave privada de cliente. Se utiliza para autenticación basada en certificados	""
trustedCACertificate	Valor codificado en base64 del certificado de CA de confianza. Opcional. Se utiliza para autenticación basada en certificados.	""
username	El nombre de usuario necesario para comunicarse con el clúster de ONTAP. Se utiliza para autenticación basada en credenciales.	""
password	La contraseña necesaria para comunicarse con el clúster de ONTAP. Se utiliza para autenticación basada en credenciales.	""
svm	Máquina virtual de almacenamiento que usar	Derivado si una SVM managementLIF está especificado
storagePrefix	El prefijo que se utiliza cuando se aprovisionan volúmenes nuevos en la SVM.  No se puede modificar más adelante. Para actualizar este parámetro, deberá crear un nuevo backend.	trident
limitAggregateUsage	Error al aprovisionar si el uso supera este porcentaje.  Si utiliza un entorno de administración de Amazon FSX para ONTAP de NetApp, no especifique limitAggregateUsage. El proporcionado fsxadmin y. vsadmin No incluya los permisos necesarios para recuperar el uso de agregados y limitarlo mediante Astra Trident.	"" (no se aplica de forma predeterminada)
limitVolumeSize	Error en el aprovisionamiento si el tamaño del volumen solicitado es superior a este valor.  También restringe el tamaño máximo de los volúmenes que gestiona para qtrees y LUN.	" (no se aplica por defecto)



Parámetro	Descripción	Predeterminado
lunsPerFlexvol	El número máximo de LUN por FlexVol debe estar comprendido entre [50 y 200]	100
debugTraceFlags	Indicadores de depuración que se deben usar para la solución de problemas. Ejemplo, {«api»:false, «method»:true}  No lo utilice a menos que esté solucionando problemas y necesite un volcado de log detallado.	null
useREST	Parámetro booleano para usar las API DE REST de ONTAP. <b>Vista previa técnica</b>  useREST se proporciona como <b>avance técnico</b> que se recomienda para entornos de prueba y no para cargas de trabajo de producción. Cuando se establece en <code>true</code> , Astra Trident utilizará las API DE REST de ONTAP para comunicarse con el back-end. Esta función requiere ONTAP 9.11.1 o posterior. Además, el rol de inicio de sesión de ONTAP utilizado debe tener acceso a <code>ontap</code> cliente más. Esto está satisfecho por el predefinido <code>vsadmin</code> y <code>cluster-admin</code> funciones.  useREST No es compatible con MetroCluster.  useREST Está totalmente cualificado para NVMe/TCP.	false
sanType	Utilice para seleccionar <code>iscsi</code> Para iSCSI o <code>nvme</code> Para NVMe/TCP.	<code>iscsi</code> si está en blanco

#### Opciones de configuración de back-end para el aprovisionamiento de volúmenes

Puede controlar el aprovisionamiento predeterminado utilizando estas opciones en la `defaults` sección de la configuración. Para ver un ejemplo, vea los ejemplos de configuración siguientes.

Parámetro	Descripción	Predeterminado
spaceAllocation	Asignación de espacio para las LUN	verdadero
spaceReserve	Modo de reserva de espacio; «ninguno» (fino) o «volumen» (grueso)	ninguno
snapshotPolicy	Política de Snapshot que se debe usar	ninguno

Parámetro	Descripción	Predeterminado
qosPolicy	<p>Grupo de políticas de calidad de servicio que se asignará a los volúmenes creados. Elija uno de qosPolicy o adaptiveQosPolicy por pool/back-end de almacenamiento.</p> <p>El uso de grupos de políticas de calidad de servicio con Astra Trident requiere ONTAP 9.8 o posterior. Recomendamos utilizar un grupo de políticas QoS no compartido y garantizar que el grupo de políticas se aplique a cada componente por separado. Un grupo de políticas de calidad de servicio compartido hará que se aplique el techo para el rendimiento total de todas las cargas de trabajo.</p>	""
adaptiveQosPolicy	Grupo de políticas de calidad de servicio adaptativo que permite asignar los volúmenes creados. Elija uno de qosPolicy o adaptiveQosPolicy por pool/back-end de almacenamiento	""
snapshotReserve	Porcentaje de volumen reservado para las Snapshot	«0» si snapshotPolicy no es "ninguno", de lo contrario
splitOnClone	Divida un clon de su elemento principal al crearlo	"falso"
encryption	<p>Habilite el cifrado de volúmenes de NetApp (NVE) en el volumen nuevo; el valor predeterminado es false. Para usar esta opción, debe tener una licencia para NVE y habilitarse en el clúster.</p> <p>Si NAE está habilitado en el back-end, cualquier volumen aprovisionado en Astra Trident estará habilitado para NAE.</p> <p>Para obtener más información, consulte: <a href="#">"Cómo funciona Astra Trident con NVE y NAE"</a>.</p>	"falso"
luksEncryption	<p>Active el cifrado LUKS. Consulte <a href="#">"Usar la configuración de clave unificada de Linux (LUKS)"</a>.</p> <p>El cifrado LUKS no es compatible con NVMe/TCP.</p>	""
securityStyle	Estilo de seguridad para nuevos volúmenes	unix
tieringPolicy	Política de organización en niveles para utilizar ninguna	«Solo Snapshot» para la configuración SVM-DR anterior a ONTAP 9,5

## Ejemplos de aprovisionamiento de volúmenes

Aquí hay un ejemplo con los valores predeterminados 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 los volúmenes creados mediante la `ontap-san` Controlador, Astra Trident añade un 10 % adicional de capacidad a FlexVol para acomodar los metadatos de las LUN. La LUN se aprovisionará con el tamaño exacto que el usuario solicite en la RVP. Astra Trident añade el 10 % a FlexVol (se muestra como tamaño disponible en ONTAP). Los usuarios obtienen ahora la cantidad de capacidad utilizable que soliciten. Este cambio también impide que las LUN se conviertan en de solo lectura a menos que se utilice completamente el espacio disponible. Esto no se aplica a `ontap-san-economy`.

Para los back-ends que definen `snapshotReserve`, Astra Trident calcula el tamaño de los volúmenes de la siguiente manera:

$$\text{Total volume size} = [(\text{PVC requested size}) / (1 - (\text{snapshotReserve percentage} / 100))] * 1.1$$

El 1.1 es el 10 % adicional que Astra Trident añade a FlexVol para acomodar los metadatos de las LUN. Para `snapshotReserve = 5 %` y la solicitud de PVC = 5GIB, el tamaño total del volumen es de 5.79GIB y el tamaño disponible es de 5.5GIB. La `volume show` el comando debería mostrar resultados similares a los de este ejemplo:

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.

En la actualidad, el cambio de tamaño es la única manera de utilizar el nuevo cálculo para un volumen existente.

### Ejemplos de configuración mínima

Los ejemplos siguientes muestran configuraciones básicas que dejan la mayoría de los parámetros en los valores predeterminados. Esta es la forma más sencilla de definir un back-end.



Si utiliza Amazon FSx en NetApp ONTAP con Astra Trident, le recomendamos que especifique nombres de DNS para las LIF en lugar de las direcciones IP.

### Ejemplo de SAN ONTAP

Se trata de una configuración básica que utiliza el `ontap-san` controlador.

```
---
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>
```

### Ejemplo de economía de SAN ONTAP

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

## Ejemplo de MetroCluster

Puede configurar el backend para evitar tener que actualizar manualmente la definición de backend después del switchover y el switchover durante "[Replicación y recuperación de SVM](#)".

Para obtener una conmutación de sitios y una conmutación de estado sin problemas, especifique la SVM con `managementLIF` y omita la `dataLIF` y.. `svm` parámetros. Por ejemplo:

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

## Ejemplo de autenticación basada en certificados

En este ejemplo de configuración básica `clientCertificate`, `clientPrivateKey`, y `trustedCACertificate` (Opcional, si se utiliza una CA de confianza) se completan en `backend.json`. Y tome los valores codificados base64 del certificado de cliente, la clave privada y el certificado de CA de confianza, respectivamente.

```
---
version: 1
storageDriverName: ontap-san
backendName: DefaultSANBackend
managementLIF: 10.0.0.1
svm: svm_iscsi
useCHAP: true
chapInitiatorSecret: c19qxIm36DKyawxy
chapTargetInitiatorSecret: rqxigXgkesIpwxyz
chapTargetUsername: iJF4heBRT0TCwxyz
chapUsername: uh2aNCLSD6cNwxyz
clientCertificate: ZXR0ZXJwYXB...ICMgJ3BhcGVyc2
clientPrivateKey: vciwKIyAgZG...0cnksIGRlc2NyaX
trustedCACertificate: zcyBbaG...b3Igb3duIGNsYXNz
```

## Ejemplos de CHAP bidireccional

Estos ejemplos crean un backend con `useCHAP` establezca en `true`.

### Ejemplo de CHAP de SAN de ONTAP

```
---
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>
```

### Ejemplo de CHAP de economía de SAN ONTAP

```
---
version: 1
storageDriverName: ontap-san-economy
managementLIF: 10.0.0.1
svm: svm_iscsi_eco
useCHAP: true
chapInitiatorSecret: cl9qxIm36DKyawxy
chapTargetInitiatorSecret: rqxigXgkesIpwxyz
chapTargetUsername: iJF4heBRT0TCwxyz
chapUsername: uh2aNCLSD6cNwxyz
username: vsadmin
password: <password>
```

## Ejemplo de NVMe/TCP

Debe tener una SVM configurada con NVMe en el back-end de ONTAP. Esta es una configuración de back-end básica 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
```

## Ejemplos de back-ends con pools virtuales

En estos archivos de definición de backend de ejemplo, se establecen valores predeterminados específicos para todos los pools de almacenamiento, como `spaceReserve` en ninguno, `spaceAllocation` en falso, y `encryption` en falso. Los pools virtuales se definen en la sección de almacenamiento.

Astra Trident establece etiquetas de aprovisionamiento en el campo «Comentarios». Los comentarios se establecen en la FlexVol. Astra Trident copia todas las etiquetas presentes en un pool virtual al volumen de almacenamiento al aprovisionar. Para mayor comodidad, los administradores de almacenamiento pueden definir etiquetas por pool virtual y agrupar volúmenes por etiqueta.

En estos ejemplos, algunos de los pools de almacenamiento establecen sus propios `spaceReserve`, `spaceAllocation`, y `encryption` y algunos pools sustituyen los valores predeterminados.

**Ejemplo 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'
```

## Ejemplo de economía de SAN ONTAP

```
---
version: 1
storageDriverName: ontap-san-economy
managementLIF: 10.0.0.1
svm: svm_iscsi_eco
useCHAP: true
chapInitiatorSecret: cl9qxIm36DKyawxy
chapTargetInitiatorSecret: rqxigXgkesIpwxyz
chapTargetUsername: iJF4heBRT0TCwxyz
chapUsername: uh2aNCLSD6cNwxyz
username: vsadmin
password: <password>
defaults:
  spaceAllocation: 'false'
  encryption: 'false'
labels:
  store: san_economy_store
region: us_east_1
storage:
- labels:
  app: oracledb
  cost: '30'
  zone: us_east_1a
  defaults:
    spaceAllocation: 'true'
    encryption: 'true'
- labels:
  app: postgresdb
  cost: '20'
  zone: us_east_1b
  defaults:
    spaceAllocation: 'false'
    encryption: 'true'
- labels:
  app: mysqldb
  cost: '10'
  zone: us_east_1c
  defaults:
    spaceAllocation: 'true'
    encryption: 'false'
- labels:
  department: legal
  creditpoints: '5000'
  zone: us_east_1c
```

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

## Ejemplo de NVMe/TCP

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

## Asigne los back-ends a StorageClass

Las siguientes definiciones de StorageClass hacen referencia a la [Ejemplos de back-ends con pools virtuales](#). Con el `parameters.selector` Cada StorageClass llama la atención sobre qué pools virtuales pueden usarse para alojar un volumen. El volumen tendrá los aspectos definidos en el pool virtual elegido.

- La `protection-gold` StorageClass se asignará al primer pool virtual del `ontap-san` back-end. Este es el único pool que ofrece protección de nivel Gold.

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

- La `protection-not-gold` StorageClass se asignará al segundo y tercer pool virtual en `ontap-san` back-end. Estos son los únicos pools que ofrecen un nivel de protección distinto del oro.

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

- La `app-mysqldb` StorageClass se asignará al tercer pool virtual en `ontap-san-economy` back-end. Este es el único pool que ofrece configuración de pool de almacenamiento para la aplicación de tipo `mysqldb`.

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

- La `protection-silver-creditpoints-20k` StorageClass se asignará al segundo pool virtual de `ontap-san` back-end. Este es el único pool que ofrece protección de nivel plata y 20000 puntos 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"
```

- La `creditpoints-5k` StorageClass se asignará al tercer pool virtual en `ontap-san` backend y cuarto pool virtual en `ontap-san-economy` back-end. Estas son las únicas ofertas de grupo con 5000 puntos de crédito.

```

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

```

- La `my-test-app-sc` StorageClass se asignará al `testAPP` pool virtual en el `ontap-san` conductor con `sanType: nvme`. Esta es la única oferta de pool `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"

```

Astra Trident decidirá qué pool virtual se selecciona y garantizará que se cumplan los requisitos de almacenamiento.

## Unidades NAS de ONTAP

### Información general del controlador NAS de ONTAP

Obtenga más información sobre la configuración de un entorno de administración de ONTAP con controladores NAS de ONTAP y Cloud Volumes ONTAP.

### Información del controlador NAS de ONTAP

Astra Trident proporciona los siguientes controladores de almacenamiento NAS para comunicarse con el clúster de ONTAP. Los modos de acceso admitidos son: *ReadWriteOnce* (RWO), *ReadOnlyMany* (ROX), *ReadWriteMany* (RWX), *ReadWriteOncePod* (RWOP).



Si utiliza Astra Control para protección, recuperación y movilidad, lea [Compatibilidad de controladores Astra Control](#).

Controlador	Protocolo	VolumeMo de	Modos de acceso compatibles	Sistemas de archivos compatibles
ontap-nas	NFS SMB	Sistema de archivos	RWO, ROX, RWX, RWOP	« », nfs, smb
ontap-nas-economy	NFS SMB	Sistema de archivos	RWO, ROX, RWX, RWOP	« », nfs, smb
ontap-nas-flexgroup	NFS SMB	Sistema de archivos	RWO, ROX, RWX, RWOP	« », nfs, smb

## Compatibilidad de controladores Astra Control

Astra Control proporciona una protección fluida, recuperación ante desastres y movilidad (mover volúmenes entre clústeres de Kubernetes) para los volúmenes creados con el `ontap-nas`, `ontap-nas-flexgroup`, y `ontap-san` de windows Consulte ["Requisitos previos de replicación de Astra Control"](#) para obtener más detalles.



- Uso `ontap-san-economy` solo si se espera que el número de uso de volúmenes persistentes sea superior a ["Límites de volumen ONTAP compatibles"](#).
- Uso `ontap-nas-economy` solo si se espera que el número de uso de volúmenes persistentes sea superior a ["Límites de volumen ONTAP compatibles"](#) y la `ontap-san-economy` no se puede utilizar el conductor.
- No utilizar `ontap-nas-economy` si prevé la necesidad de protección de datos, recuperación ante desastres o movilidad.

## Permisos de usuario

Astra Trident espera que se ejecute como administrador de ONTAP o SVM, normalmente mediante el `admin` usuario del clúster o un `vsadmin` Usuario de SVM o un usuario con un nombre diferente que tenga el mismo rol.

Para puestas en marcha de Amazon FSX para ONTAP de NetApp, Astra Trident espera que se ejecute como administrador de ONTAP o SVM, mediante el clúster `fsxadmin` usuario o un `vsadmin` Usuario de SVM o un usuario con un nombre diferente que tenga el mismo rol. La `fsxadmin` el usuario es un reemplazo limitado para el usuario administrador del clúster.



Si utiliza la `limitAggregateUsage` parámetro, se necesitan permisos de administrador de clúster. Cuando se utiliza Amazon FSX para ONTAP de NetApp con Astra Trident, el `limitAggregateUsage` el parámetro no funciona con el `vsadmin` y `fsxadmin` cuentas de usuario. La operación de configuración generará un error si se especifica este parámetro.

Si bien es posible crear un rol más restrictivo dentro de ONTAP que puede utilizar un controlador Trident, no lo recomendamos. La mayoría de las nuevas versiones de Trident denominan API adicionales que se tendrían que tener en cuenta, por lo que las actualizaciones son complejas y propensas a errores.

## Prepárese para configurar un back-end con controladores NAS de ONTAP

Conozca los requisitos, las opciones de autenticación y las políticas de exportación para configurar un backend de ONTAP con controladores NAS de ONTAP.

### Requisitos

- Para todos los back-ends de ONTAP, Astra Trident requiere al menos un agregado asignado a la SVM.
- Puede ejecutar más de un controlador y crear clases de almacenamiento que apunten a uno u otro. Por ejemplo, puede configurar una clase Gold que utilice `ontap-nas` Controlador y clase Bronze que utiliza `ontap-nas-economy` uno.
- Todos sus nodos de trabajo de Kubernetes deben tener instaladas las herramientas NFS adecuadas. Consulte ["aquí"](#) para obtener más detalles.
- Astra Trident admite volúmenes de SMB montados en pods que se ejecutan solo en nodos de Windows. Consulte [Prepárese para aprovisionar los volúmenes de SMB](#) para obtener más detalles.

### Autentique el backend de ONTAP

Astra Trident ofrece dos modos de autenticación de un back-end de ONTAP.

- Basado en Credenciales: Este modo requiere permisos suficientes para el backend de ONTAP. Se recomienda utilizar una cuenta asociada con un rol de inicio de sesión de seguridad predefinido, como `admin` o `vsadmin`. Garantizar la máxima compatibilidad con versiones de ONTAP.
- Basado en certificado: Este modo requiere un certificado instalado en el back-end para que Astra Trident se comunique con un clúster de ONTAP. Aquí, la definición de backend debe contener valores codificados en Base64 del certificado de cliente, la clave y el certificado de CA de confianza si se utiliza (recomendado).

Puede actualizar los back-ends existentes para moverse entre métodos basados en credenciales y basados en certificados. Sin embargo, solo se admite un método de autenticación a la vez. Para cambiar a un método de autenticación diferente, debe eliminar el método existente de la configuración del back-end.



Si intenta proporcionar **tanto credenciales como certificados**, la creación de backend fallará y se producirá un error en el que se haya proporcionado más de un método de autenticación en el archivo de configuración.

### Habilite la autenticación basada en credenciales

Astra Trident requiere las credenciales a un administrador con ámbito de SVM o clúster para comunicarse con el back-end de ONTAP. Se recomienda utilizar funciones estándar predefinidas como `admin` o `vsadmin`. De este modo se garantiza la compatibilidad con futuras versiones de ONTAP que puedan dar a conocer API de funciones que podrán utilizarse en futuras versiones de Astra Trident. Se puede crear y utilizar una función de inicio de sesión de seguridad personalizada con Astra Trident, pero no es recomendable.

Una definición de backend de ejemplo tendrá este aspecto:

## YAML

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

## JSON

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

Tenga en cuenta que la definición de backend es el único lugar en el que las credenciales se almacenan en texto sin formato. Una vez creado el back-end, los nombres de usuario y las contraseñas se codifican con Base64 y se almacenan como secretos de Kubernetes. La creación/mejora de un backend es el único paso que requiere conocimiento de las credenciales. Por tanto, es una operación de solo administración que deberá realizar el administrador de Kubernetes o almacenamiento.

### Habilite la autenticación basada en certificados

Los back-ends nuevos y existentes pueden utilizar un certificado y comunicarse con el back-end de ONTAP. Se necesitan tres parámetros en la definición de backend.

- ClientCertificate: Valor codificado en base64 del certificado de cliente.
- ClientPrivateKey: Valor codificado en base64 de la clave privada asociada.
- TrustedCACertificate: Valor codificado en base64 del certificado de CA de confianza. Si se utiliza una CA de confianza, se debe proporcionar este parámetro. Esto se puede ignorar si no se utiliza ninguna CA de confianza.

Un flujo de trabajo típico implica los pasos siguientes.

### Pasos

1. Genere una clave y un certificado de cliente. Al generar, establezca el nombre común (CN) en el usuario



de ONTAP para autenticarse como.

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

2. Añada un certificado de CA de confianza al clúster ONTAP. Es posible que ya sea gestionado por el administrador de almacenamiento. Ignore si no se utiliza ninguna CA de confianza.

```
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 el certificado y la clave de cliente (desde el paso 1) en el clúster 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 los compatibilidad con el rol de inicio de sesión de seguridad ONTAP `cert` método de autenticación.

```
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. Probar la autenticación mediante un certificado generado. Reemplace <LIF de gestión de ONTAP> y <vserver name> por la IP de LIF de gestión y el nombre de SVM. Debe asegurarse de que la LIF tiene su política de servicio establecida en `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 certificados, claves y certificados de CA de confianza con Base64.

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

## 7. Cree un backend utilizando los valores obtenidos del paso 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...Vaaalllluuueeee",
  "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
online	9	

### Actualice los métodos de autenticación o gire las credenciales

Puede actualizar un back-end existente para utilizar un método de autenticación diferente o para rotar sus credenciales. Esto funciona de las dos maneras: Los back-ends que utilizan nombre de usuario/contraseña se pueden actualizar para usar certificados. Los back-ends que utilizan certificados pueden actualizarse a nombre de usuario/contraseña. Para ello, debe eliminar el método de autenticación existente y agregar el nuevo método de autenticación. A continuación, utilice el archivo backend.json actualizado que contiene los parámetros necesarios para ejecutarse `tridentctl update backend`.

```

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

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

```



Quando gira contraseñas, el administrador de almacenamiento debe actualizar primero la contraseña del usuario en ONTAP. A esto le sigue una actualización de back-end. Al rotar certificados, se pueden agregar varios certificados al usuario. A continuación, el back-end se actualiza para usar el nuevo certificado, siguiendo el cual se puede eliminar el certificado antiguo del clúster de ONTAP.

La actualización de un back-end no interrumpe el acceso a los volúmenes que se han creado ni afecta a las conexiones de volúmenes realizadas después. Una actualización de back-end correcta indica que Astra Trident puede comunicarse con el back-end de ONTAP y gestionar futuras operaciones de volúmenes.

### Gestione las políticas de exportación de NFS

Astra Trident utiliza las políticas de exportación de NFS para controlar el acceso a los volúmenes que aprovisiona.

Astra Trident ofrece dos opciones al trabajar con directivas de exportación:

- Astra Trident puede gestionar dinámicamente la propia política de exportación; en este modo de funcionamiento, el administrador de almacenamiento especifica una lista de bloques CIDR que representan direcciones IP admisibles. Astra Trident agrega automáticamente las IP de nodo que se incluyen en estos rangos a la directiva de exportación. Como alternativa, cuando no se especifican CIDR,

toda IP de unidifusión de ámbito global encontrada en los nodos se agregará a la política de exportación.

- Los administradores de almacenamiento pueden crear una normativa de exportación y añadir reglas manualmente. Astra Trident utiliza la directiva de exportación predeterminada a menos que se especifique un nombre de directiva de exportación diferente en la configuración.

## Gestione de forma dinámica políticas de exportación

Astra Trident proporciona la capacidad de gestionar dinámicamente las políticas de exportación para los back-ends de ONTAP. De este modo, el administrador de almacenamiento puede especificar un espacio de direcciones permitido para las IP de nodos de trabajo, en lugar de definir reglas explícitas de forma manual. Simplifica en gran medida la gestión de políticas de exportación; las modificaciones de la política de exportación ya no requieren intervención manual en el clúster de almacenamiento. Además, esto ayuda a restringir el acceso al clúster de almacenamiento solo a nodos de trabajo con IP en el rango especificado, lo que permite una gestión automatizada y de gran granularidad.



No utilice la traducción de direcciones de red (NAT) cuando utilice políticas de exportación dinámicas. Con NAT, el controlador de almacenamiento ve la dirección NAT de frontend y no la dirección de host IP real, por lo que el acceso se denegará cuando no se encuentre ninguna coincidencia en las reglas de exportación.

## Ejemplo

Hay dos opciones de configuración que deben utilizarse. He aquí un ejemplo de definición de backend:

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



Al usar esta función, debe asegurarse de que la unión raíz de la SVM tenga una política de exportación creada previamente con una regla de exportación que permite el bloque CIDR de nodo (como la política de exportación predeterminada). Siga siempre las prácticas recomendadas de NetApp para dedicar una SVM para Astra Trident.

A continuación se ofrece una explicación del funcionamiento de esta función utilizando el ejemplo anterior:

- `autoExportPolicy` se establece en `true`. Esto indica que Astra Trident creará una directiva de exportación para `svm1` SVM y gestionan la adición y eliminación de reglas mediante `autoExportCIDRs` bloques de direcciones. Por ejemplo, un back-end con UUID `403b5326-8482-40db-96d0-d83fb3f4daec` y `autoExportPolicy` establezca en `true` crea una política de exportación llamada `trident-403b5326-8482-40db-96d0-d83fb3f4daec` En la SVM.

- `autoExportCIDRs` contiene una lista de bloques de direcciones. Este campo es opcional y se establece de forma predeterminada en `["0.0.0.0/0", "*/0"]`. Si no se define, Astra Trident agrega todas las direcciones de unidifusión de ámbito global que se encuentran en los nodos de trabajo.

En este ejemplo, la `192.168.0.0/24` se proporciona espacio de dirección. Esto indica que las IP de nodo de Kubernetes que entran dentro de este rango de direcciones se añadirán a la política de exportación que crea Astra Trident. Cuando Astra Trident registra un nodo en el que se ejecuta, recupera las direcciones IP del nodo y las comprueba con respecto a los bloques de direcciones proporcionados en `autoExportCIDRs`. Después de filtrar las IP, Astra Trident crea reglas de política de exportación para las IP de cliente que detecta, con una regla para cada nodo que identifica.

Puede actualizar `autoExportPolicy` y `autoExportCIDRs` para los back-ends después de crearlos. Puede añadir CIDR nuevos para un back-end que se gestiona o elimina automáticamente CIDR existentes. Tenga cuidado al eliminar CIDR para asegurarse de que las conexiones existentes no se hayan caído. También puede optar por desactivar `autoExportPolicy` para un back-end y caer en una política de exportación creada manualmente. Esto requerirá establecer la `exportPolicy` parámetro en la configuración del back-end.

Una vez que Astra Trident crea o actualiza un back-end, puede comprobar el backend mediante `tridentctl` o el correspondiente `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
```

A medida que se añaden nodos a un clúster de Kubernetes y se registran con la controladora Astra Trident, se actualizan las políticas de exportación de los back-ends existentes (siempre que entren en el rango de direcciones especificado en la `autoExportCIDRs` para el back-end).

Cuando se quita un nodo, Astra Trident comprueba todos los back-ends que están en línea para quitar la regla de acceso del nodo. Al eliminar esta IP de nodo de las políticas de exportación de los back-ends gestionados, Astra Trident evita los montajes no autorizados, a menos que se vuelva a utilizar esta IP con un nodo nuevo

del clúster.

Para los back-ends anteriores, actualizando el back-end con `tridentctl update backend` Se asegurará de que Astra Trident gestiona las políticas de exportación de forma automática. Esto creará una nueva política de exportación llamada después del UUID del back-end y los volúmenes que están presentes en el back-end utilizarán la política de exportación recién creada cuando se vuelvan a montar.



Si se elimina un back-end con políticas de exportación gestionadas automáticamente, se eliminará la política de exportación creada de forma dinámica. Si se vuelve a crear el back-end, se trata como un nuevo back-end y dará lugar a la creación de una nueva política de exportación.

Si se actualiza la dirección IP de un nodo activo, debe reiniciar el pod Astra Trident en el nodo. A continuación, Astra Trident actualizará la política de exportación para los back-ends que gestiona para reflejar este cambio de IP.

### Prepárese para aprovisionar los volúmenes de SMB

Con un poco de preparación adicional, puede aprovisionar volúmenes SMB con `ontap-nas` de windows



Debe configurar tanto los protocolos NFS como SMB/CIFS en la SVM para crear un `ontap-nas-economy` Volumen SMB para ONTAP en las instalaciones. Si no se configura ninguno de estos protocolos, se producirá un error en la creación del volumen de SMB.

### Antes de empezar

Para poder aprovisionar volúmenes de SMB, debe tener lo siguiente.

- Un clúster de Kubernetes con un nodo de controladora Linux y al menos un nodo de trabajo de Windows que ejecuta Windows Server 2019. Astra Trident admite volúmenes de SMB montados en pods que se ejecutan solo en nodos de Windows.
- Al menos un secreto Astra Trident que contiene sus credenciales de Active Directory. Generar secreto `smbcreds`:

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

- Proxy CSI configurado como servicio de Windows. Para configurar un `csi-proxy`, consulte "[GitHub: Proxy CSI](#)" o "[GitHub: Proxy CSI para Windows](#)" Para nodos Kubernetes que se ejecutan en Windows.

### Pasos

1. Para la ONTAP en las instalaciones, puede crear opcionalmente un recurso compartido de SMB, o bien Astra Trident puede crearlo para usted.



Los recursos compartidos de SMB se requieren para Amazon FSx para ONTAP.

Puede crear recursos compartidos de administrador de SMB de una de dos formas mediante el "[Consola de administración de Microsoft](#)" Complemento carpetas compartidas o uso de la CLI de ONTAP. Para crear los recursos compartidos de SMB mediante la CLI de ONTAP:

- a. Si es necesario, cree la estructura de ruta de acceso de directorio para el recurso compartido.

La `vserver cifs share create` comando comprueba la ruta especificada en la opción `-path` durante la creación del recurso compartido. Si la ruta especificada no existe, el comando falla.

b. Cree un recurso compartido de SMB asociado con la SVM especificada:

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

c. Compruebe que se ha creado el recurso compartido:

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



Consulte "[Cree un recurso compartido de SMB](#)" para obtener todos los detalles.

2. Al crear el back-end, debe configurar lo siguiente para especificar volúmenes de SMB. Para obtener información sobre todas las opciones de configuración del entorno de administración de ONTAP, consulte "[Opciones y ejemplos de configuración de FSX para ONTAP](#)".

Parámetro	Descripción	Ejemplo
<code>smbShare</code>	<p>Puede especificar una de las siguientes opciones: El nombre de un recurso compartido de SMB creado mediante la consola de administración de Microsoft o la interfaz de línea de comandos de ONTAP; un nombre para permitir que Astra Trident cree el recurso compartido de SMB; o bien puede dejar el parámetro en blanco para evitar el acceso de recurso compartido común a los volúmenes.</p> <p>Este parámetro es opcional para ONTAP en las instalaciones.</p> <p>Este parámetro es necesario para los back-ends de Amazon FSx para ONTAP y no puede estar en blanco.</p>	<code>smb-share</code>
<code>nasType</code>	<b>Debe establecer en <code>smb</code>.</b> Si es nulo, el valor predeterminado es <code>nfs</code> .	<code>smb</code>
<code>securityStyle</code>	<p>Estilo de seguridad para nuevos volúmenes.</p> <p><b>Debe estar configurado en <code>ntfs</code> o <code>mixed</code> Para volúmenes SMB.</b></p>	<code>ntfs</code> o <code>mixed</code> Para volúmenes de SMB
<code>unixPermissions</code>	Modo para volúmenes nuevos. <b>Se debe dejar vacío para volúmenes SMB.</b>	<code>""</code>

## Opciones y ejemplos de configuración NAS de ONTAP

Descubre cómo crear y utilizar controladores NAS de ONTAP con tu instalación de Astra Trident. Esta sección proporciona ejemplos de configuración de backend y detalles para la asignación de back-ends a StorageClasses.

### Opciones de configuración del back-end

Consulte la siguiente tabla para ver las opciones de configuración del back-end:

Parámetro	Descripción	Predeterminado
version		Siempre 1
storageDriveName	Nombre del controlador de almacenamiento	«ontap-nas», «ontap-nas-economy», «ontap-nas-flexgroup», «ontap-san», «ontap-san-economy»
backendName	Nombre personalizado o el back-end de almacenamiento	Nombre de controlador + «_» + LIF de datos
managementLIF	<p>La dirección IP de una LIF de gestión de clústeres o SVM</p> <p>Se puede especificar un nombre de dominio completo (FQDN).</p> <p>Puede configurarse para que utilice direcciones IPv6 si Astra Trident se instaló mediante la marca IPv6. Las direcciones IPv6 deben definirse entre corchetes, como [28e8:d9fb:a825:b7bf:69a8:d02f:9e7b:3555].</p> <p>Para un cambio de MetroCluster fluido, consulte <a href="#">Ejemplo de MetroCluster</a>.</p>	“10.0.0.1”, “[2001:1234:abcd::fefe]”



Parámetro	Descripción	Predeterminado
dataLIF	<p>Dirección IP de LIF de protocolo.</p> <p>Recomendamos especificar dataLIF. En caso de no proporcionar esta información, Astra Trident busca las LIF de datos desde la SVM. Puede especificar un nombre de dominio completo (FQDN) para las operaciones de montaje de NFS, lo que permite crear un DNS round-robin para lograr el equilibrio de carga entre varios LIF de datos.</p> <p>Se puede cambiar después del ajuste inicial. Consulte .</p> <p>Puede configurarse para que utilice direcciones IPv6 si Astra Trident se instaló mediante la marca IPv6. Las direcciones IPv6 deben definirse entre corchetes, como [28e8:d9fb:a825:b7bf:69a8:d02f:9e7b:3555].</p> <p><b>Omitir para MetroCluster.</b> Ver <a href="#">Ejemplo de MetroCluster</a>.</p>	Dirección especificada o derivada de la SVM, si no se especifica (no recomendada)
svm	<p>Máquina virtual de almacenamiento que usar</p> <p><b>Omitir para MetroCluster.</b> Ver <a href="#">Ejemplo de MetroCluster</a>.</p>	Derivado si una SVM managementLIF está especificado
autoExportPolicy	<p>Habilite la creación y actualización automática de la política de exportación [Boolean].</p> <p>Con el autoExportPolicy y.. autoExportCIDRs Astra Trident puede gestionar automáticamente las políticas de exportación.</p>	falso
autoExportCIDRs	<p>Lista de CIDRs para filtrar las IP del nodo de Kubernetes contra cuando autoExportPolicy está habilitado.</p> <p>Con el autoExportPolicy y.. autoExportCIDRs Astra Trident puede gestionar automáticamente las políticas de exportación.</p>	[«0,0.0,0/0», «:/0»]
labels	Conjunto de etiquetas con formato JSON arbitrario que se aplica en los volúmenes	""
clientCertificate	Valor codificado en base64 del certificado de cliente. Se utiliza para autenticación basada en certificados	""
clientPrivateKey	Valor codificado en base64 de la clave privada de cliente. Se utiliza para autenticación basada en certificados	""

Parámetro	Descripción	Predeterminado
trustedCACertificate	Valor codificado en base64 del certificado de CA de confianza. Opcional. Se utiliza para autenticación basada en certificados	""
username	Nombre de usuario para conectarse al clúster/SVM. Se utiliza para autenticación basada en credenciales	
password	Contraseña para conectarse al clúster/SVM. Se utiliza para autenticación basada en credenciales	
storagePrefix	El prefijo que se utiliza cuando se aprovisionan volúmenes nuevos en la SVM. No se puede actualizar después de configurarlo	«trident»
limitAggregateUsage	Error al aprovisionar si el uso supera este porcentaje. <b>No se aplica a Amazon FSX para ONTAP</b>	"" (no se aplica de forma predeterminada)
limitVolumeSize	Error en el aprovisionamiento si el tamaño del volumen solicitado es superior a este valor.  También restringe el tamaño máximo de los volúmenes que gestiona para qtrees y LUN, y la qtreesPerFlexvol Permite personalizar el número máximo de qtrees por FlexVol.	" (no se aplica por defecto)
lunsPerFlexvol	El número máximo de LUN por FlexVol debe estar comprendido entre [50 y 200]	«100»
debugTraceFlags	Indicadores de depuración que se deben usar para la solución de problemas. Ejemplo, {«api»:false, «method»:true}  No utilizar debugTraceFlags a menos que esté solucionando problemas y necesite un volcado de registro detallado.	nulo
nasType	Configure la creación de volúmenes NFS o SMB.  Las opciones son nfs, smb o nulo. El valor predeterminado es nulo en volúmenes de NFS.	nfs

Parámetro	Descripción	Predeterminado
nfsMountOptions	<p>Lista de opciones de montaje NFS separadas por comas.</p> <p>Las opciones de montaje para los volúmenes persistentes de Kubernetes se especifican normalmente en tipos de almacenamiento, pero si no se especifican opciones de montaje en una clase de almacenamiento, Astra Trident se pondrá en contacto con las opciones de montaje especificadas en el archivo de configuración del back-end de almacenamiento.</p> <p>Si no se especifican opciones de montaje en la clase de almacenamiento o el archivo de configuración, Astra Trident no configurará ninguna opción de montaje en un volumen persistente asociado.</p>	""
qtreesPerFlexvol	El número máximo de qtrees por FlexVol debe estar comprendido entre [50, 300]	«200»
smbShare	<p>Puede especificar una de las siguientes opciones: El nombre de un recurso compartido de SMB creado mediante la consola de administración de Microsoft o la interfaz de línea de comandos de ONTAP; un nombre para permitir que Astra Trident cree el recurso compartido de SMB; o bien puede dejar el parámetro en blanco para evitar el acceso de recurso compartido común a los volúmenes.</p> <p>Este parámetro es opcional para ONTAP en las instalaciones.</p> <p>Este parámetro es necesario para los back-ends de Amazon FSx para ONTAP y no puede estar en blanco.</p>	smb-share
useREST	<p>Parámetro booleano para usar las API DE REST de ONTAP. <b>Vista previa técnica</b></p> <p>useREST se proporciona como <b>avance técnico</b> que se recomienda para entornos de prueba y no para cargas de trabajo de producción. Cuando se establece en <code>true</code>, Astra Trident utilizará las API DE REST de ONTAP para comunicarse con el back-end. Esta función requiere ONTAP 9.11.1 o posterior. Además, el rol de inicio de sesión de ONTAP utilizado debe tener acceso a <code>ontap</code> cliente más. Esto está satisfecho por el predefinido <code>vsadmin</code> y <code>cluster-admin</code> funciones.</p> <p>useREST No es compatible con MetroCluster.</p>	falso

## Opciones de configuración de back-end para el aprovisionamiento de volúmenes

Puede controlar el aprovisionamiento predeterminado utilizando estas opciones en la `defaults` sección de la configuración. Para ver un ejemplo, vea los ejemplos de configuración siguientes.

Parámetro	Descripción	Predeterminado
<code>spaceAllocation</code>	Asignación de espacio para las LUN	verdadero
<code>spaceReserve</code>	Modo de reserva de espacio; «ninguno» (fino) o «volumen» (grueso)	ninguno
<code>snapshotPolicy</code>	Política de Snapshot que se debe usar	ninguno
<code>qosPolicy</code>	Grupo de políticas de calidad de servicio que se asignará a los volúmenes creados. Elija uno de <code>qosPolicy</code> o <code>adaptiveQosPolicy</code> por pool/back-end de almacenamiento	""
<code>adaptiveQosPolicy</code>	Grupo de políticas de calidad de servicio adaptativo que permite asignar los volúmenes creados. Elija uno de <code>qosPolicy</code> o <code>adaptiveQosPolicy</code> por pool/back-end de almacenamiento.  no admitido por ontap-nas-Economy.	""
<code>snapshotReserve</code>	Porcentaje de volumen reservado para las Snapshot	«0» si <code>snapshotPolicy</code> no es "ninguno", de lo contrario
<code>splitOnClone</code>	Divida un clon de su elemento principal al crearlo	"falso"
<code>encryption</code>	Habilite el cifrado de volúmenes de NetApp (NVE) en el volumen nuevo; el valor predeterminado es <code>false</code> . Para usar esta opción, debe tener una licencia para NVE y habilitarse en el clúster.  Si NAE está habilitado en el back-end, cualquier volumen aprovisionado en Astra Trident estará habilitado para NAE.  Para obtener más información, consulte: <a href="#">"Cómo funciona Astra Trident con NVE y NAE"</a> .	"falso"
<code>tieringPolicy</code>	Política de organización en niveles para utilizar ninguna	«Solo Snapshot» para la configuración SVM-DR anterior a ONTAP 9,5
<code>unixPermissions</code>	Modo para volúmenes nuevos	«777» para volúmenes NFS; vacío (no aplicable) para volúmenes SMB
<code>snapshotDir</code>	Controla el acceso al <code>.snapshot</code> directorio	"falso"
<code>exportPolicy</code>	Política de exportación que se va a utilizar	"predeterminado"

Parámetro	Descripción	Predeterminado
securityStyle	<p>Estilo de seguridad para nuevos volúmenes.</p> <p>Compatibilidad con NFS <code>mixed</code> y.. <code>unix</code> estilos de seguridad.</p> <p>SMB admite <code>mixed</code> y.. <code>ntfs</code> estilos de seguridad.</p>	<p>El valor predeterminado de NFS es <code>unix</code>.</p> <p>La opción predeterminada de SMB es <code>ntfs</code>.</p>



El uso de grupos de políticas de calidad de servicio con Astra Trident requiere ONTAP 9.8 o posterior. Se recomienda utilizar un grupo de políticas de calidad de servicio no compartido y asegurarse de que el grupo de políticas se aplique a cada componente individualmente. Un grupo de políticas de calidad de servicio compartido hará que se aplique el techo para el rendimiento total de todas las cargas de trabajo.

### Ejemplos de aprovisionamiento de volúmenes

Aquí hay un ejemplo con los valores predeterminados 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` y.. `ontap-nas-flexgroups`, Astra Trident utiliza ahora un nuevo cálculo para garantizar que el tamaño de la FlexVol sea correcto con el porcentaje `snapshotReserve` y la RVP. Cuando el usuario solicita una RVP, Astra Trident crea el FlexVol original con más espacio mediante el nuevo cálculo. Este cálculo garantiza que el usuario recibe el espacio de escritura que solicitó en el PVC y no menos espacio que

el que solicitó. Antes de v21.07, cuando el usuario solicita una RVP (por ejemplo, 5GIB) con el 50 por ciento de snapshotReserve, solo obtiene 2,5 GIB de espacio editable. Esto se debe a que el usuario solicitó es todo el volumen y snapshotReserve es un porcentaje de esta situación. Con Trident 21.07, lo que el usuario solicita es el espacio editable y Astra Trident define el snapshotReserve número como porcentaje del volumen completo. Esto no se aplica a ontap-nas-economy. Vea el siguiente ejemplo para ver cómo funciona:

El cálculo es el siguiente:

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

Para snapshotReserve = 50 % y la solicitud de RVP = 5 GIB, el tamaño total del volumen es  $2/5 = 10$  GIB y el tamaño disponible es de 5 GIB, lo que es lo que solicitó el usuario en la solicitud de RVP. La `volume show` el comando debería mostrar resultados similares a los de este ejemplo:

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.

Los back-ends existentes de instalaciones anteriores aprovisionan volúmenes como se explicó anteriormente al actualizar Astra Trident. En el caso de los volúmenes que creó antes de actualizar, debe cambiar el tamaño de sus volúmenes para que se observe el cambio. Por ejemplo, una RVP de 2 GIB con `snapshotReserve=50` Anteriormente, se produjo un volumen que proporciona 1 GIB de espacio editable. Cambiar el tamaño del volumen a 3 GIB, por ejemplo, proporciona a la aplicación 3 GIB de espacio editable en un volumen de 6 GIB.

### Ejemplos de configuración mínima

Los ejemplos siguientes muestran configuraciones básicas que dejan la mayoría de los parámetros en los valores predeterminados. Esta es la forma más sencilla de definir un back-end.



Si utiliza Amazon FSX en ONTAP de NetApp con Trident, la recomendación es especificar nombres DNS para las LIF en lugar de direcciones IP.

### Ejemplo de economía NAS de ONTAP

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

## Ejemplo de FlexGroup NAS de ONTAP

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

## Ejemplo de MetroCluster

Puede configurar el backend para evitar tener que actualizar manualmente la definición de backend después del switchover y el switchover durante ["Replicación y recuperación de SVM"](#).

Para obtener una conmutación de sitios y una conmutación de estado sin problemas, especifique la SVM con managementLIF y omita la dataLIF y.. svm parámetros. Por ejemplo:

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

## Ejemplo de volúmenes de 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
```

## Ejemplo de autenticación basada en certificados

Este es un ejemplo de configuración de backend mínima. `clientCertificate`, `clientPrivateKey`, y `trustedCACertificate` (Opcional, si se utiliza una CA de confianza) se completan en `backend.json` Y tome los valores codificados base64 del certificado de cliente, la clave privada y el certificado de CA de confianza, 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_
```

## Ejemplo de política de exportación automática

En este ejemplo se muestra cómo puede indicar a Astra Trident que utilice políticas de exportación dinámicas para crear y gestionar automáticamente la directiva de exportación. Esto funciona igual para el `ontap-nas-economy` y `ontap-nas-flexgroup` de `windows`

```
---
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
```



## Ejemplo de direcciones IPv6

Este ejemplo muestra managementLIF Uso de una dirección 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
```

## Ejemplo de Amazon FSx para ONTAP mediante volúmenes de bloque de mensajes del servidor

La smbShare El parámetro es obligatorio para FSx para ONTAP mediante volúmenes de bloque de mensajes del servidor.

```
---
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_
```

## Ejemplos de back-ends con pools virtuales

En los archivos de definición de backend de ejemplo que se muestran a continuación, se establecen valores predeterminados específicos para todos los pools de almacenamiento, como `spaceReserve` en ninguno, `spaceAllocation` en falso, y `encryption` en falso. Los pools virtuales se definen en la sección de almacenamiento.

Astra Trident establece etiquetas de aprovisionamiento en el campo «Comentarios». Los comentarios se establecen en FlexVol para `ontap-nas` O FlexGroup para `ontap-nas-flexgroup`. Astra Trident copia

todas las etiquetas presentes en un pool virtual al volumen de almacenamiento al aprovisionar. Para mayor comodidad, los administradores de almacenamiento pueden definir etiquetas por pool virtual y agrupar volúmenes por etiqueta.

En estos ejemplos, algunos de los pools de almacenamiento establecen sus propios `spaceReserve`, `spaceAllocation`, y `encryption` y algunos pools sustituyen los valores predeterminados.

## Ejemplo de NAS de ONTAP

```
---
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: mysqldb  
  cost: '25'  
  zone: us_east_1d  
  defaults:  
    spaceReserve: volume  
    encryption: 'false'  
    unixPermissions: '0775'
```

## Ejemplo de FlexGroup NAS de 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'
```

## Ejemplo de economía NAS de ONTAP

```
---
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'
```

### Asigne los back-ends a StorageClass

Las siguientes definiciones de StorageClass se refieren a [Ejemplos de back-ends con pools virtuales](#). Con el `parameters.selector` Cada StorageClass llama la atención sobre qué pools virtuales pueden usarse para alojar un volumen. El volumen tendrá los aspectos definidos en el pool virtual elegido.

- La `protection-gold` StorageClass se asignará al primer y segundo pool virtual del `ontap-nas-flexgroup` back-end. Estos son los únicos pools que ofrecen protección de nivel Gold.

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

- La `protection-not-gold` StorageClass se asignará al tercer y cuarto pool virtual del `ontap-nas-flexgroup` back-end. Estos son los únicos pools que ofrecen un nivel de protección distinto al Gold.

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

- La `app-mysqldb` StorageClass se asignará al cuarto pool virtual del `ontap-nas` back-end. Este es el único pool que ofrece configuración de pool de almacenamiento para la aplicación de tipo `mysqldb`.



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

- T. protection-silver-creditpoints-20k StorageClass se asignará al tercer pool virtual del ontap-nas-flexgroup back-end. Este es el único pool que ofrece protección de nivel plata y 20000 puntos 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"
```

- La creditpoints-5k StorageClass se asignará al tercer pool virtual del ontap-nas backend y segundo pool virtual en ontap-nas-economy back-end. Estas son las únicas ofertas de grupo con 5000 puntos de crédito.

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

Astra Trident decidirá qué pool virtual se selecciona y garantizará que se cumplan los requisitos de almacenamiento.

#### **Actualizar dataLIF tras la configuración inicial**

Puede cambiar la LIF de datos tras la configuración inicial ejecutando el siguiente comando para proporcionar el nuevo archivo JSON back-end con LIF de datos actualizadas.

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



Si los RVP están conectados a uno o varios pods, deben recuperar todos los pods correspondientes y, a continuación, traerlos para que surta efecto el nuevo LIF de datos.

## Amazon FSX para ONTAP de NetApp

### Utilice Astra Trident con Amazon FSX para ONTAP de NetApp

"Amazon FSX para ONTAP de NetApp" Es un servicio AWS totalmente gestionado que permite a los clientes iniciar y ejecutar sistemas de archivos con tecnología del sistema operativo de almacenamiento ONTAP de NetApp. FSX para ONTAP le permite aprovechar las funciones, el rendimiento y las funcionalidades administrativas de NetApp con las que ya está familiarizado, a la vez que aprovecha la simplicidad, la agilidad, la seguridad y la escalabilidad de almacenar datos en AWS. FSX para ONTAP es compatible con las funciones del sistema de archivos ONTAP y las API de administración.

#### Descripción general

Un sistema de archivos es el recurso principal de Amazon FSX, similar a un clúster de ONTAP en las instalaciones. En cada SVM, se pueden crear uno o varios volúmenes, que son contenedores de datos que almacenan los archivos y las carpetas en el sistema de archivos. Con Amazon FSX para ONTAP de NetApp, Data ONTAP se proporcionará como un sistema de archivos gestionado en el cloud. El nuevo tipo de sistema de archivos se llama **ONTAP** de NetApp.

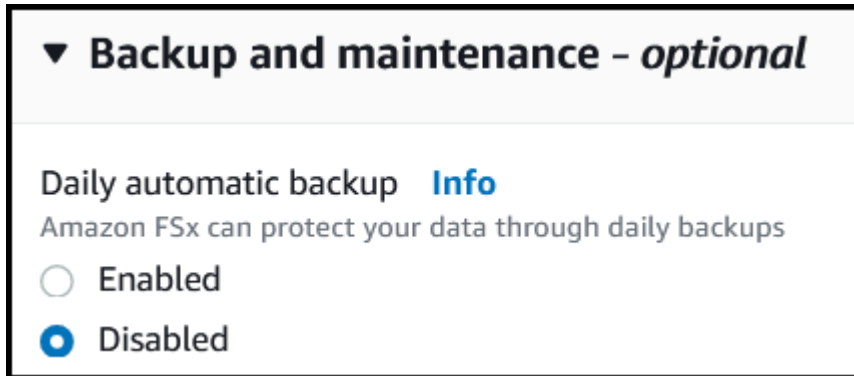
Al utilizar Astra Trident con Amazon FSX para ONTAP de NetApp, puede garantizar que los clústeres de Kubernetes que se ejecutan en Amazon Elastic Kubernetes Service (EKS) pueden aprovisionar volúmenes persistentes de bloques y archivos respaldados por ONTAP.

Usos de Amazon FSX para ONTAP de NetApp "[FabricPool](#)" para gestionar los niveles de almacenamiento. Le permite almacenar datos en un nivel, según la frecuencia de acceso a estos.

#### Consideraciones

- Volúmenes SMB:
  - Se admiten los volúmenes de SMB mediante el `ontap-nas` sólo conductor.
  - Astra Trident admite volúmenes de SMB montados en pods que se ejecutan solo en nodos de Windows.
- Trident no puede eliminar los volúmenes creados en sistemas de archivos Amazon FSX con backups automáticos habilitados. Para eliminar las RVP, es necesario eliminar manualmente el VP y el FSX para el volumen ONTAP. Para evitar este problema:
  - No utilice **creación rápida** para crear el sistema de archivos FSX para ONTAP. El flujo de trabajo de creación rápida permite realizar backups automáticos y no ofrece la opción de anulación de suscripción.
  - Cuando utilice **Standard create**, desactive la copia de seguridad automática. Al deshabilitar los

backups automáticos, Trident puede eliminar correctamente un volumen sin intervención manual adicional.



### FSX para ONTAP detalles del controlador

Puede integrar Astra Trident con Amazon FSX para ONTAP de NetApp mediante los siguientes controladores:

- `ontap-san`: Cada VP aprovisionado es una LUN dentro de su propio Amazon FSX para el volumen ONTAP de NetApp.
- `ontap-san-economy`: Cada VP aprovisionado es un LUN con un número configurable de LUN por Amazon FSX para el volumen ONTAP de NetApp.
- `ontap-nas`: Cada VP aprovisionado es un Amazon FSX completo para el volumen ONTAP de NetApp.
- `ontap-nas-economy`: Cada VP aprovisionado es un qtree, con un número configurable de qtrees por Amazon FSX para el volumen ONTAP de NetApp.
- `ontap-nas-flexgroup`: Cada VP aprovisionado es un Amazon FSX completo para el volumen ONTAP FlexGroup de NetApp.

Para obtener más información sobre el controlador, consulte "[Controladores de NAS](#)" y.. "[Controladores de SAN](#)".

### Autenticación

Astra Trident ofrece dos modos de autenticación.

- Basado en certificados: Astra Trident se comunicará con la SVM en su sistema de archivos FSX mediante un certificado instalado en la SVM.
- Basado en credenciales: Puede utilizar el `fsxadmin` usuario del sistema de archivos o del `vsadmin` Usuario configurado para la SVM.



Astra Trident espera que se ejecute como un `vsadmin` Usuario de SVM o como usuario con un nombre diferente que tenga el mismo rol. Amazon FSX para NetApp ONTAP cuenta con una `fsxadmin` Usuario que es una sustitución limitada de ONTAP `admin` usuario de clúster. Le recomendamos encarecidamente que utilice `vsadmin` Con Astra Trident.

Puede actualizar los back-ends para moverse entre los métodos basados en credenciales y los basados en certificados. Sin embargo, si intenta proporcionar **credenciales y certificados**, la creación de backend fallará. Para cambiar a un método de autenticación diferente, debe eliminar el método existente de la configuración del back-end.

Para obtener más información sobre cómo habilitar la autenticación, consulte la autenticación del tipo de controlador:

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

#### Obtenga más información

- ["Documentación de Amazon FSX para ONTAP de NetApp"](#)
- ["Publicación del blog en Amazon FSX para ONTAP de NetApp"](#)

### Integración de Amazon FSX para ONTAP de NetApp

Puede integrar su sistema de archivos Amazon FSX para ONTAP de NetApp con Astra Trident para garantizar que los clústeres de Kubernetes que se ejecutan en Amazon Elastic Kubernetes Service (EKS) puedan aprovisionar volúmenes persistentes de bloques y archivos respaldados por ONTAP.

#### Requisitos

Además de ["Requisitos de Astra Trident"](#), Para integrar FSX para ONTAP con Astra Trident, necesita:

- Un clúster de Amazon EKS existente o un clúster de Kubernetes autogestionado con `kubectl` instalado.
- Un sistema de archivos Amazon FSx para NetApp ONTAP y una máquina virtual de almacenamiento (SVM) a la que se puede acceder desde los nodos de trabajo del clúster.
- Nodos de trabajo preparados para ["NFS o iSCSI"](#).



Asegúrese de seguir los pasos de preparación de nodos necesarios para Amazon Linux y Ubuntu ["Imágenes de máquina de Amazon"](#) (AMI) en función del tipo de IAM EKS.

- Astra Trident admite volúmenes de SMB montados en pods que se ejecutan solo en nodos de Windows. Consulte [Prepárese para aprovisionar los volúmenes de SMB](#) para obtener más detalles.

### Integración de controladores ONTAP SAN y NAS



Si está configurando para volúmenes SMB, debe leer [Prepárese para aprovisionar los volúmenes de SMB](#) antes de crear el back-end.

#### Pasos

1. Ponga en marcha Astra Trident con una de las ["métodos de implementación"](#).
2. Recoja el nombre de DNS del LIF de gestión de SVM. Por ejemplo, si utiliza la CLI de AWS, busque el `DNSName` entrada en `Endpoints` → `Management` tras ejecutar el siguiente comando:

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

3. Cree e instale certificados para ["Autenticación de back-end NAS"](#) o ["Autenticación de entorno de administración DE SAN"](#).



Puede iniciar sesión en el sistema de archivos (por ejemplo, para instalar certificados) con SSH desde cualquier lugar que pueda llegar al sistema de archivos. Utilice la `fsxadmin` Usuario, la contraseña que configuró al crear el sistema de archivos y el nombre DNS de gestión desde `aws fsx describe-file-systems`.

4. Cree un archivo de entorno de administración mediante sus certificados y el nombre DNS de la LIF de gestión, como se muestra en el ejemplo siguiente:

#### YAML

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

#### JSON

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

Para obtener información sobre la creación de back-ends, consulte estos enlaces:

- ["Configurar un back-end con controladores NAS de ONTAP"](#)
- ["Configuración de un back-end con controladores SAN de ONTAP"](#)

#### Prepárese para aprovisionar los volúmenes de SMB

Puede aprovisionar volúmenes SMB mediante el `ontap-nas` controlador. Antes de completar la tarea [Integración de controladores ONTAP SAN y NAS](#) complete los siguientes pasos.

#### Antes de empezar

Para poder aprovisionar volúmenes de SMB con el `ontap-nas` conductor, debe tener lo siguiente.

- Un clúster de Kubernetes con un nodo de controladora Linux y al menos un nodo de trabajo de Windows que ejecuta Windows Server 2019. Astra Trident admite volúmenes de SMB montados en pods que se ejecutan solo en nodos de Windows.
- Al menos un secreto Astra Trident que contiene sus credenciales de Active Directory. Generar secreto `smbcreds`:

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

- Proxy CSI configurado como servicio de Windows. Para configurar un `csi-proxy`, consulte ["GitHub: Proxy CSI"](#) o ["GitHub: Proxy CSI para Windows"](#) Para nodos Kubernetes que se ejecutan en Windows.

## Pasos

1. Cree recursos compartidos de SMB. Puede crear recursos compartidos de administrador de SMB de una de dos formas mediante el ["Consola de administración de Microsoft"](#) Complemento carpetas compartidas o uso de la CLI de ONTAP. Para crear los recursos compartidos de SMB mediante la CLI de ONTAP:
  - a. Si es necesario, cree la estructura de ruta de acceso de directorio para el recurso compartido.

La `vserver cifs share create` comando comprueba la ruta especificada en la opción `-path` durante la creación del recurso compartido. Si la ruta especificada no existe, el comando falla.

- b. Cree un recurso compartido de SMB asociado con la SVM especificada:

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

- c. Compruebe que se ha creado el recurso compartido:

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



Consulte ["Cree un recurso compartido de SMB"](#) para obtener todos los detalles.

2. Al crear el back-end, debe configurar lo siguiente para especificar volúmenes de SMB. Para obtener información sobre todas las opciones de configuración del entorno de administración de ONTAP, consulte ["Opciones y ejemplos de configuración de FSX para ONTAP"](#).

Parámetro	Descripción	Ejemplo
smbShare	Puede especificar una de las siguientes opciones: El nombre de un recurso compartido de SMB creado con la consola de administración de Microsoft o la interfaz de línea de comandos de ONTAP, o bien un nombre para permitir que Astra Trident cree el recurso compartido de SMB.  Este parámetro es obligatorio para los back-ends de Amazon FSx para ONTAP.	smb-share
nasType	<b>Debe establecer en smb.</b> Si es nulo, el valor predeterminado es nfs.	smb
securityStyle	Estilo de seguridad para nuevos volúmenes.  <b>Debe estar configurado en ntfs o. mixed Para volúmenes SMB.</b>	ntfs o. mixed Para volúmenes de SMB
unixPermissions	Modo para volúmenes nuevos. <b>Se debe dejar vacío para volúmenes SMB.</b>	""

### Opciones y ejemplos de configuración de FSX para ONTAP

Obtenga información acerca de las opciones de configuración de back-end para Amazon FSX para ONTAP. Esta sección proporciona ejemplos de configuración de fondo.

#### Opciones de configuración del back-end

Consulte la siguiente tabla para ver las opciones de configuración del back-end:

Parámetro	Descripción	Ejemplo
version		Siempre 1
storageDriverName	Nombre del controlador de almacenamiento	ontap-nas, ontap-nas-economy, ontap-nas-flexgroup, ontap-san, ontap-san-economy
backendName	Nombre personalizado o el back-end de almacenamiento	Nombre del conductor + “_” + dataLIF

Parámetro	Descripción	Ejemplo
managementLIF	<p>La dirección IP de una LIF de gestión de clústeres o SVM</p> <p>Se puede especificar un nombre de dominio completo (FQDN).</p> <p>Puede configurarse para que utilice direcciones IPv6 si Astra Trident se instaló mediante la marca IPv6. Las direcciones IPv6 deben definirse entre corchetes, como [28e8:d9fb:a825:b7bf:69a8:d02f:9e7b:3555].</p>	"10.0.0.1", "[2001:1234:abcd::fefe]"
dataLIF	<p>Dirección IP de LIF de protocolo.</p> <p><b>Controladores NAS de ONTAP:</b> Recomendamos especificar dataLIF. En caso de no proporcionar esta información, Astra Trident busca las LIF de datos desde la SVM. Puede especificar un nombre de dominio completo (FQDN) para las operaciones de montaje de NFS, lo que permite crear un DNS round-robin para lograr el equilibrio de carga entre varios LIF de datos. Se puede cambiar después del ajuste inicial. Consulte .</p> <p><b>Controladores SAN ONTAP:</b> No se especifica para iSCSI. Astra Trident utiliza la asignación selectiva de LUN de ONTAP para descubrir los LIF iSCSI necesarios para establecer una sesión de varias rutas. Se genera una advertencia si dataLIF se define explícitamente.</p> <p>Puede configurarse para que utilice direcciones IPv6 si Astra Trident se instaló mediante la marca IPv6. Las direcciones IPv6 deben definirse entre corchetes, como [28e8:d9fb:a825:b7bf:69a8:d02f:9e7b:3555].</p>	



Parámetro	Descripción	Ejemplo
autoExportPolicy	Habilite la creación y actualización automática de la política de exportación [Boolean].  Con el autoExportPolicy y.. autoExportCIDRs Astra Trident puede gestionar automáticamente las políticas de exportación.	false
autoExportCIDRs	Lista de CIDRs para filtrar las IP del nodo de Kubernetes contra cuando autoExportPolicy está habilitado.  Con el autoExportPolicy y.. autoExportCIDRs Astra Trident puede gestionar automáticamente las políticas de exportación.	"["0.0.0.0/0", ":/0"]"
labels	Conjunto de etiquetas con formato JSON arbitrario que se aplica en los volúmenes	""
clientCertificate	Valor codificado en base64 del certificado de cliente. Se utiliza para autenticación basada en certificados	""
clientPrivateKey	Valor codificado en base64 de la clave privada de cliente. Se utiliza para autenticación basada en certificados	""
trustedCACertificate	Valor codificado en base64 del certificado de CA de confianza. Opcional. Se utiliza para autenticación basada en certificados.	""
username	El nombre de usuario para conectarse al clúster o SVM. Se utiliza para autenticación basada en credenciales. Por ejemplo, vsadmin.	
password	La contraseña para conectarse al clúster o SVM. Se utiliza para autenticación basada en credenciales.	
svm	Máquina virtual de almacenamiento que usar	Derivado si se especifica una LIF de gestión de SVM.

Parámetro	Descripción	Ejemplo
storagePrefix	<p>El prefijo que se utiliza cuando se aprovisionan volúmenes nuevos en la SVM.</p> <p>No se puede modificar una vez creada. Para actualizar este parámetro, deberá crear un nuevo backend.</p>	trident
limitAggregateUsage	<p><b>No especifique para Amazon FSx para NetApp ONTAP.</b></p> <p>El proporcionado fsxadmin y.. vsadmin No incluya los permisos necesarios para recuperar el uso de agregados y limitarlo mediante Astra Trident.</p>	No utilizar.
limitVolumeSize	<p>Error en el aprovisionamiento si el tamaño del volumen solicitado es superior a este valor.</p> <p>También restringe el tamaño máximo de los volúmenes que gestiona para qtrees y LUN, y la qtreesPerFlexvol Permite personalizar el número máximo de qtrees por FlexVol.</p>	"" (no se aplica de forma predeterminada)
lunsPerFlexvol	<p>El número máximo de LUN por FlexVol debe estar comprendido entre [50 y 200].</p> <p>Solo SAN.</p>	100
debugTraceFlags	<p>Indicadores de depuración que se deben usar para la solución de problemas. Ejemplo, {"api":false, "method":true}</p> <p>No utilizar debugTraceFlags a menos que esté solucionando problemas y necesite un volcado de registro detallado.</p>	nulo

Parámetro	Descripción	Ejemplo
nfsMountOptions	<p>Lista de opciones de montaje NFS separadas por comas.</p> <p>Las opciones de montaje para los volúmenes persistentes de Kubernetes se especifican normalmente en tipos de almacenamiento, pero si no se especifican opciones de montaje en una clase de almacenamiento, Astra Trident se pondrá en contacto con las opciones de montaje especificadas en el archivo de configuración del back-end de almacenamiento.</p> <p>Si no se especifican opciones de montaje en la clase de almacenamiento o el archivo de configuración, Astra Trident no configurará ninguna opción de montaje en un volumen persistente asociado.</p>	""
nasType	<p>Configure la creación de volúmenes NFS o SMB.</p> <p>Las opciones son <code>nfs</code>, <code>smb</code>, o nulo.</p> <p><b>Debe establecer en <code>smb</code> Para volúmenes SMB.</b> el valor predeterminado es <code>null</code> en volúmenes NFS.</p>	<code>nfs</code>
qtreesPerFlexvol	El número máximo de qtrees por FlexVol debe estar comprendido entre [50, 300]	200
smbShare	<p>Puede especificar una de las siguientes opciones: El nombre de un recurso compartido de SMB creado con la consola de administración de Microsoft o la interfaz de línea de comandos de ONTAP, o bien un nombre para permitir que Astra Trident cree el recurso compartido de SMB.</p> <p>Este parámetro es obligatorio para los back-ends de Amazon FSx para ONTAP.</p>	<code>smb-share</code>

Parámetro	Descripción	Ejemplo
useREST	<p>Parámetro booleano para usar las API DE REST de ONTAP. <b>Vista previa técnica</b></p> <p>useREST se proporciona como <b>avance técnico</b> que se recomienda para entornos de prueba y no para cargas de trabajo de producción. Cuando se establece en <code>true</code>, Astra Trident utilizará las API DE REST de ONTAP para comunicarse con el back-end.</p> <p>Esta función requiere ONTAP 9.11.1 o posterior. Además, el rol de inicio de sesión de ONTAP utilizado debe tener acceso a <code>ontap cliente más</code>. Esto está satisfecho por el predeterminado <code>vsadmin</code> y.. <code>cluster-admin</code> funciones.</p>	<code>false</code>

### Actualizar dataLIF tras la configuración inicial

Puede cambiar la LIF de datos tras la configuración inicial ejecutando el siguiente comando para proporcionar el nuevo archivo JSON back-end con LIF de datos actualizadas.

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



Si los RVP están conectados a uno o varios pods, deben recuperar todos los pods correspondientes y, a continuación, traerlos para que surta efecto el nuevo LIF de datos.

### Opciones de configuración de back-end para el aprovisionamiento de volúmenes

Puede controlar el aprovisionamiento predeterminado utilizando estas opciones en la `defaults` sección de la configuración. Para ver un ejemplo, vea los ejemplos de configuración siguientes.

Parámetro	Descripción	Predeterminado
<code>spaceAllocation</code>	Asignación de espacio para las LUN	<code>true</code>
<code>spaceReserve</code>	Modo de reserva de espacio; "none" (thin) o "VOLUME" (grueso)	<code>none</code>
<code>snapshotPolicy</code>	Política de Snapshot que se debe usar	<code>none</code>

Parámetro	Descripción	Predeterminado
qosPolicy	<p>Grupo de políticas de calidad de servicio que se asignará a los volúmenes creados. Elija uno de qosPolicy o adaptiveQosPolicy por pool de almacenamiento o back-end.</p> <p>El uso de grupos de políticas de calidad de servicio con Astra Trident requiere ONTAP 9.8 o posterior.</p> <p>Recomendamos utilizar un grupo de políticas QoS no compartido y garantizar que el grupo de políticas se aplique a cada componente por separado. Un grupo de políticas de calidad de servicio compartido hará que se aplique el techo para el rendimiento total de todas las cargas de trabajo.</p>	""
adaptiveQosPolicy	<p>Grupo de políticas de calidad de servicio adaptativo que permite asignar los volúmenes creados. Elija uno de qosPolicy o adaptiveQosPolicy por pool de almacenamiento o back-end.</p> <p>no admitido por ontap-nas-Economy.</p>	""
snapshotReserve	Porcentaje de volumen reservado para snapshots «0»	Si snapshotPolicy es none, else ""
splitOnClone	Divida un clon de su elemento principal al crearlo	false
encryption	<p>Habilite el cifrado de volúmenes de NetApp (NVE) en el volumen nuevo; el valor predeterminado es false. Para usar esta opción, debe tener una licencia para NVE y habilitarse en el clúster.</p> <p>Si NAE está habilitado en el back-end, cualquier volumen provisionado en Astra Trident estará habilitado para NAE.</p> <p>Para obtener más información, consulte: <a href="#">"Cómo funciona Astra Trident con NVE y NAE"</a>.</p>	false

Parámetro	Descripción	Predeterminado
luksEncryption	Active el cifrado LUKS. Consulte <a href="#">"Usar la configuración de clave unificada de Linux (LUKS)"</a> .  Solo SAN.	""
tieringPolicy	Política de organización en niveles para utilizar none	snapshot-only Para configuraciones anteriores a ONTAP 9,5 SVM-DR
unixPermissions	Modo para volúmenes nuevos.  <b>Dejar vacío para volúmenes SMB.</b>	""
securityStyle	Estilo de seguridad para nuevos volúmenes.  Compatibilidad con NFS mixed y.. unix estilos de seguridad.  SMB admite mixed y.. ntfs estilos de seguridad.	El valor predeterminado de NFS es unix.  La opción predeterminada de SMB es ntfs.

### Ejemplo

Uso `nasType`, `node-stage-secret-name`, y `node-stage-secret-namespace`, Puede especificar un volumen SMB y proporcionar las credenciales necesarias de Active Directory. Se admiten los volúmenes de SMB mediante el `ontap-nas` sólo conductor.

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

### Configura la versión 23,10 del complemento Astra Trident EKS en el clúster de EKS

Astra Trident optimiza la gestión del almacenamiento de Amazon FSx para NetApp ONTAP en Kubernetes para que sus desarrolladores y administradores se centren en la puesta en marcha de aplicaciones. El complemento Astra Trident EKS incluye las últimas revisiones de seguridad, correcciones de errores y AWS lo valida para que funcione con Amazon EKS. El complemento EKS le permite garantizar de forma constante que sus

clústeres de Amazon EKS sean seguros y estables y reducir la cantidad de trabajo que necesita para instalar, configurar y actualizar complementos.

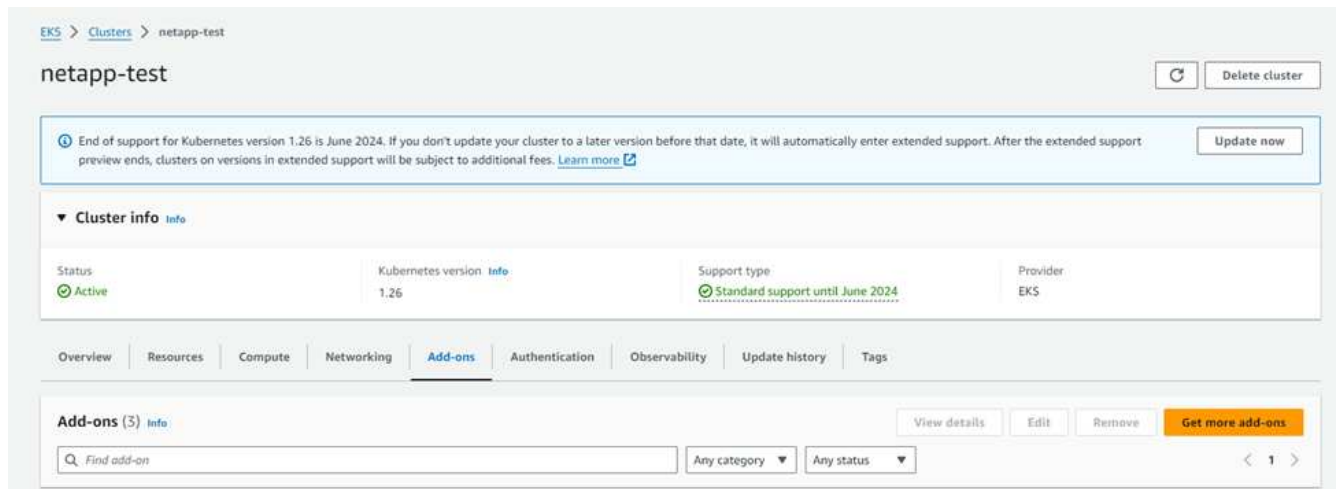
### Requisitos previos

Asegúrate de disponer de lo siguiente antes de configurar el complemento Astra Trident para AWS EKS:

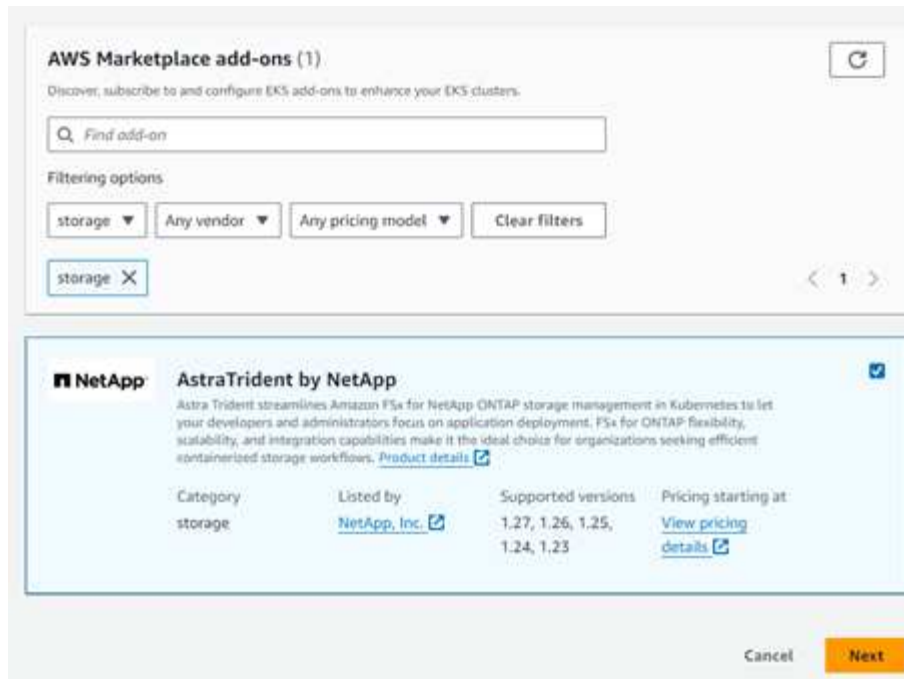
- Una cuenta de clúster de Amazon EKS con suscripción complementaria
- Permisos de AWS para AWS Marketplace:  
"aws-marketplace:ViewSubscriptions",  
"aws-marketplace:Subscribe",  
"aws-marketplace:Unsubscribe"
- Tipo de AMI: Amazon Linux 2 (AL2\_x86\_64) o Amazon Linux 2 Arm (AL2\_ARM\_64)
- Tipo de nodo: AMD o ARM
- Un sistema de archivos Amazon FSx para NetApp ONTAP existente

### Pasos

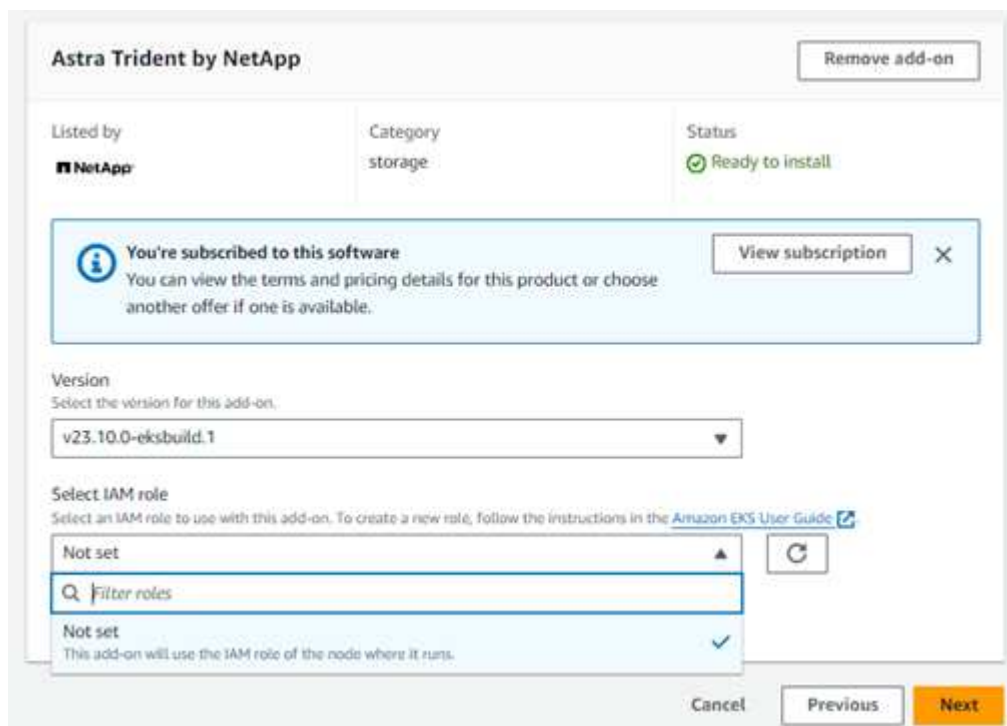
1. En tu clúster de EKS Kubernetes, navega a la pestaña **Add-ons**.



2. Vaya a **AWS Marketplace add-ons** y elija la categoría *storage*.

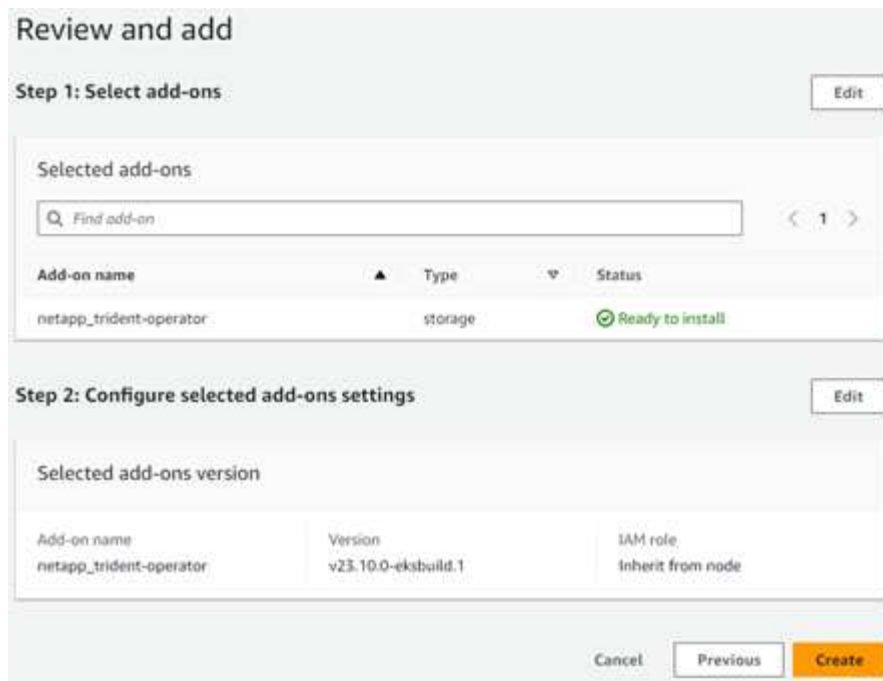


3. Localiza **AstraTrident by NetApp** y selecciona la casilla de verificación para el complemento Astra Trident.
4. Elija la versión deseada del complemento.



5. Seleccione la opción Rol IAM que desea heredar del nodo.
6. Configure cualquier configuración opcional según sea necesario y seleccione **Siguiente**.





7. Seleccione **Crear**.
8. Compruebe que el estado del complemento es *Active*.



### Instalar/desinstalar el complemento Astra Trident EKS mediante la CLI

#### Instale el complemento Astra Trident EKS mediante la CLI:

Los siguientes comandos de ejemplo instalan el complemento Astra Trident EKS:

```
eksctl create addon --cluster K8s-arm --name netapp_trident-operator --version v23.10.0-eksbuild.1
eksctl create addon --cluster K8s-arm --name netapp_trident-operator --version v23.10.0-eksbuild.1 (con una versión dedicada)
```

#### Desinstale el complemento Astra Trident EKS mediante la CLI:

El siguiente comando desinstala el complemento Astra Trident EKS:

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

### Cree back-ends con kubectl

Un back-end define la relación entre Astra Trident y un sistema de almacenamiento. Le indica a Astra Trident cómo se comunica con ese sistema de almacenamiento y cómo debe aprovisionar volúmenes a partir de él. Una vez instalado Astra Trident, el siguiente paso es crear un back-end. La `TridentBackendConfig` Custom Resource Definition (CRD) permite crear y gestionar back-ends de Trident directamente a través de la

interfaz de Kubernetes. Para ello, utilice `kubectl` O la herramienta CLI equivalente para su distribución de Kubernetes.

`TridentBackendConfig`

`TridentBackendConfig` (`tbc`, `tbconfig`, `tbackendconfig`) Es un CRD con nombre y frontend que le permite administrar los back-ends de Astra Trident utilizando `kubectl`. Ahora, los administradores de Kubernetes y almacenamiento pueden crear y gestionar back-ends directamente a través de la CLI de Kubernetes sin necesidad de una utilidad de línea de comandos dedicada (`tridentctl`).

Sobre la creación de un `TridentBackendConfig` objeto, sucede lo siguiente:

- Astra Trident crea automáticamente un back-end en función de la configuración que proporcione. Esto se representa internamente como un `TridentBackend` (`tbe`, `tridentbackend`) CR.
- La `TridentBackendConfig` está vinculado de manera exclusiva a un `TridentBackend` Eso fue creado por Astra Trident.

Cada uno `TridentBackendConfig` mantiene una asignación de uno a uno con un `TridentBackend`. El primero es la interfaz que se ofrece al usuario para diseñar y configurar los back-ends. El segundo es cómo Trident representa el objeto back-end real.



`TridentBackend` Astra Trident crea automáticamente CRS. Usted **no debe** modificarlos. Si desea realizar actualizaciones a los back-ends, modifique el `TridentBackendConfig` objeto.

Consulte el siguiente ejemplo para ver el formato del `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
```

También puede echar un vistazo a los ejemplos de la "[instalador de trident](#)" directorio para configuraciones de ejemplo para la plataforma o servicio de almacenamiento que desee.

La `spec` toma parámetros de configuración específicos del back-end. En este ejemplo, el back-end utiliza el `ontap-san` controlador de almacenamiento y utiliza los parámetros de configuración que se tabulan aquí. Para obtener una lista de las opciones de configuración del controlador de almacenamiento que desee, consulte "[información de configuración del back-end para el controlador de almacenamiento](#)".

La spec la sección también incluye `credentials` y `deletionPolicy` campos, que se introducen recientemente en `TridentBackendConfig` CR:

- `credentials`: Este parámetro es un campo obligatorio y contiene las credenciales utilizadas para autenticarse con el sistema/servicio de almacenamiento. Este juego debe ser un secreto de Kubernetes creado por el usuario. Las credenciales no se pueden pasar en texto sin formato y se producirá un error.
- `deletionPolicy`: Este campo define lo que debe suceder cuando `TridentBackendConfig` se ha eliminado. Puede ser necesario uno de los dos valores posibles:
  - `delete`: Esto resulta en la eliminación de ambos `TridentBackendConfig` CR y el back-end asociado. Este es el valor predeterminado.
  - `retain`: Cuando un `TridentBackendConfig` se elimina la CR, la definición de backend seguirá estando presente y se puede gestionar con `tridentctl`. Establecimiento de la política de eliminación como `retain` permite a los usuarios degradar a una versión anterior (anterior a 21.04) y conservar los back-ends creados. El valor de este campo se puede actualizar después de un `TridentBackendConfig` se ha creado.



El nombre de un back-end se define mediante `spec.backendName`. Si no se especifica, el nombre del backend se establece en el nombre del `TridentBackendConfig` objeto (`metadata.name`). Se recomienda establecer explícitamente nombres de backend mediante `spec.backendName`.



Back-ends creados con `tridentctl` no tienen asociado `TridentBackendConfig` objeto. Se pueden optar por gestionar estos back-ends con `kubectl` mediante la creación de un `TridentBackendConfig` CR. Se debe tener cuidado para especificar parámetros de configuración idénticos (como `spec.backendName`, `spec.storagePrefix`, `spec.storageDriverName`, y así sucesivamente). Astra Trident enlazará automáticamente los recién creados `TridentBackendConfig` con el backend preexistente.

## Descripción general de los pasos

Para crear un nuevo back-end mediante `kubectl`, debe hacer lo siguiente:

1. Cree un "[Secreto Kubernetes](#)". El secreto contiene las credenciales que Astra Trident necesita para comunicarse con el clúster/servicio de almacenamiento.
2. Cree un `TridentBackendConfig` objeto. Este contiene detalles sobre el servicio/clúster de almacenamiento y hace referencia al secreto creado en el paso anterior.

Después de crear un backend, puede observar su estado utilizando `kubectl get tbc <tbc-name> -n <trident-namespace>` y recopile detalles adicionales.

## Paso 1: Cree un secreto de Kubernetes

Cree un secreto que contenga las credenciales de acceso para el back-end. Esto es único para cada servicio/plataforma de almacenamiento. Veamos un ejemplo:

```
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: t@Ax@7q(>
```

Esta tabla resume los campos que deben incluirse en el secreto para cada plataforma de almacenamiento:

Descripción de los campos secretos de la plataforma de almacenamiento	Secreto	Descripción de los campos
Azure NetApp Files	ID del Cliente	El ID de cliente de un registro de aplicación
Cloud Volumes Service para GCP	id_clave_privada	ID de la clave privada. Parte de la clave API de la cuenta de servicio de GCP con el rol de administrador CVS
Cloud Volumes Service para GCP	clave_privada	Clave privada. Parte de la clave API de la cuenta de servicio de GCP con el rol de administrador CVS
Element (HCI/SolidFire de NetApp)	Extremo	MVIP para el clúster de SolidFire con credenciales de inquilino
ONTAP	nombre de usuario	Nombre de usuario para conectarse al clúster/SVM. Se utiliza para autenticación basada en credenciales
ONTAP	contraseña	Contraseña para conectarse al clúster/SVM. Se utiliza para autenticación basada en credenciales
ONTAP	ClientPrivateKey	Valor codificado en base64 de la clave privada de cliente. Se utiliza para autenticación basada en certificados

Descripción de los campos secretos de la plataforma de almacenamiento	Secreto	Descripción de los campos
ONTAP	ChapUsername	Nombre de usuario entrante. Necesario si useCHAP=true. Para ontap-san y.. ontap-san-economy
ONTAP	InitichapatorSecret	Secreto CHAP del iniciador. Necesario si useCHAP=true. Para ontap-san y.. ontap-san-economy
ONTAP	ChapTargetUsername	Nombre de usuario de destino. Necesario si useCHAP=true. Para ontap-san y.. ontap-san-economy
ONTAP	ChapTargetInitiatorSecret	Secreto CHAP del iniciador de destino. Necesario si useCHAP=true. Para ontap-san y.. ontap-san-economy

El secreto creado en este paso será referenciado en el `spec.credentials` del `TridentBackendConfig` objeto creado en el paso siguiente.

## Paso 2: Cree la `TridentBackendConfig` CR

Ya está listo para crear su `TridentBackendConfig` CR. En este ejemplo, un back-end que utiliza ontap-san el controlador se crea mediante `TridentBackendConfig` objeto mostrado a continuación:

```
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

```

### Paso 3: Compruebe el estado del TridentBackendConfig CR

Ahora que creó la TridentBackendConfig CR, puede comprobar el estado. Consulte el siguiente ejemplo:

```

kubect1 -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

```

Se ha creado un backend correctamente y se ha enlazado a TridentBackendConfig CR.

La fase puede tomar uno de los siguientes valores:

- **Bound:** La TridentBackendConfig CR está asociado con un backend, y ese backend contiene configRef establezca en la TridentBackendConfig UID de CR.
- **Unbound:** Representado usando "". La TridentBackendConfig el objeto no está enlazado a un backend. Creadas recientemente TridentBackendConfig CRS se encuentra en esta fase de forma predeterminada. Tras cambiar la fase, no puede volver a «sin límites».
- **Deleting:** La TridentBackendConfig CR deletionPolicy se ha configurado para eliminar. Cuando la TridentBackendConfig La CR se elimina y pasa al estado de supresión.
  - Si no existen reclamaciones de volumen persistente (RVP) en el back-end, eliminando el TridentBackendConfig Como resultado, Astra Trident elimina el back-end, así como el TridentBackendConfig CR.
  - Si uno o más EVs están presentes en el backend, pasa a un estado de supresión. La TridentBackendConfig Posteriormente, CR también entra en fase de eliminación. El back-end y. TridentBackendConfig Se eliminan sólo después de que se hayan eliminado todas las EVs.
- **Lost:** El backend asociado con TridentBackendConfig La CR se eliminó accidental o deliberadamente y la TridentBackendConfig CR todavía tiene una referencia al backend eliminado. La TridentBackendConfig La CR puede ser eliminada independientemente de la deletionPolicy

valor.

- Unknown: Astra Trident no puede determinar el estado o la existencia del backend asociado con TridentBackendConfig CR. Por ejemplo, si el servidor API no responde o si el tridentbackends.trident.netapp.io Falta CRD. Esto puede requerir intervención.

En esta fase, se ha creado un backend. Hay varias operaciones que se pueden realizar además, como ["actualizaciones back-end y eliminaciones backend"](#).

#### (Opcional) Paso 4: Obtener más detalles

Puede ejecutar el siguiente comando para obtener más información acerca de su entorno de administración:

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

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

Además, también puede obtener un volcado YLMA/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` contiene el `backendName` y la `backendUUID` del backend que se creó en respuesta a la `TridentBackendConfig` CR. El campo `lastOperationStatus` representa el estado de la última operación de `TridentBackendConfig` CR, que se puede activar por el usuario (por ejemplo, el usuario ha cambiado algo en `spec`) o activado por Astra Trident (por ejemplo, durante el reinicio de Astra Trident). Puede ser un éxito o un fracaso. `phase` representa el estado de la relación entre el `TridentBackendConfig` CR y el back-end. En el ejemplo anterior, `phase` tiene el valor `enlazado`, lo que significa que `TridentBackendConfig` CR está asociado con el backend.

Puede ejecutar el `kubectl -n trident describe tbc <tbc-cr-name>` comando para obtener detalles de los registros de eventos.



No puede actualizar ni eliminar un backend que contenga un archivo asociado `TridentBackendConfig` objeto con `tridentctl`. Comprender los pasos que implica cambiar entre `tridentctl` y `TridentBackendConfig`, ["ver aquí"](#).



## Gestionar back-ends

### Realice la gestión del entorno de administración con kubectl

Obtenga información sobre cómo realizar operaciones de administración de back-end mediante `kubectl`.

#### Eliminar un back-end

Eliminando una `TridentBackendConfig`, Usted instruye a Astra Trident a que elimine/conserva los back-ends (basados en `deletionPolicy`). Para eliminar un back-end, asegúrese de que `deletionPolicy` está configurado para eliminar. Para eliminar sólo la `TridentBackendConfig`, asegúrese de que `deletionPolicy` se establece en retener. De esta forma se asegurará de que el backend esté todavía presente y se pueda gestionar utilizando `tridentctl`.

Ejecute el siguiente comando:

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

Astra Trident no elimina los secretos de Kubernetes que estaban en uso `TridentBackendConfig`. El usuario de Kubernetes es responsable de limpiar los secretos. Hay que tener cuidado a la hora de eliminar secretos. Solo debe eliminar secretos si no los están utilizando los back-ends.

#### Ver los back-ends existentes

Ejecute el siguiente comando:

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

También puede ejecutar `tridentctl get backend -n trident` o `tridentctl get backend -o yaml -n trident` obtener una lista de todos los back-ends que existen. Esta lista también incluirá los back-ends que se crearon con `tridentctl`.

#### Actualizar un back-end

Puede haber varias razones para actualizar un back-end:

- Las credenciales del sistema de almacenamiento han cambiado. Para actualizar las credenciales, el secreto Kubernetes que se utiliza en la `TridentBackendConfig` el objeto debe actualizarse. Astra Trident actualizará automáticamente el back-end con las últimas credenciales proporcionadas. Ejecute el siguiente comando para actualizar Kubernetes Secret:

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

- Es necesario actualizar los parámetros (como el nombre de la SVM de ONTAP que se está utilizando).
  - Puede actualizar `TridentBackendConfig` Objetos directamente a través de Kubernetes usando el siguiente comando:

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

- Como alternativa, puede realizar cambios en los existentes `TridentBackendConfig` CR con el siguiente comando:

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



- Si falla una actualización de back-end, el back-end continúa en su última configuración conocida. Puede ver los registros para determinar la causa ejecutando `kubectl get tbc <tbc-name> -o yaml -n trident` o `kubectl describe tbc <tbc-name> -n trident`.
- Después de identificar y corregir el problema con el archivo de configuración, puede volver a ejecutar el comando `update`.

## Realizar la administración de back-end con `tridentctl`

Obtenga información sobre cómo realizar operaciones de administración de back-end mediante `tridentctl`.

### Cree un back-end

Después de crear un "[archivo de configuración del back-end](#)", ejecute el siguiente comando:

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

Si se produce un error en la creación del back-end, algo estaba mal con la configuración del back-end. Puede ver los registros para determinar la causa ejecutando el siguiente comando:

```
tridentctl logs -n trident
```

Después de identificar y corregir el problema con el archivo de configuración, simplemente puede ejecutar el `create` comando de nuevo.

### Eliminar un back-end

Para eliminar un back-end de Astra Trident, haga lo siguiente:

1. Recupere el nombre del backend:

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

2. Eliminar el back-end:

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



Si Astra Trident ha provisionado volúmenes y snapshots de este back-end que aún existen, al eliminar el back-end se impiden que el departamento de tecnología provisione nuevos volúmenes. El back-end continuará existiendo en un estado de “eliminación” y Trident seguirá gestionando esos volúmenes y instantáneas hasta que se eliminen.

### Ver los back-ends existentes

Para ver los back-ends que Trident conoce, haga lo siguiente:

- Para obtener un resumen, ejecute el siguiente comando:

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

- Para obtener todos los detalles, ejecute el siguiente comando:

```
tridentctl get backend -o json -n trident
```

### Actualizar un back-end

Después de crear un nuevo archivo de configuración de back-end, ejecute el siguiente comando:

```
tridentctl update backend <backend-name> -f <backend-file> -n trident
```

Si falla la actualización del back-end, algo estaba mal con la configuración del back-end o intentó una actualización no válida. Puede ver los registros para determinar la causa ejecutando el siguiente comando:

```
tridentctl logs -n trident
```

Después de identificar y corregir el problema con el archivo de configuración, simplemente puede ejecutar el update comando de nuevo.

### Identifique las clases de almacenamiento que utilizan un back-end

Este es un ejemplo del tipo de preguntas que puede responder con el JSON que `tridentctl` salidas para objetos backend. Utiliza la `jq` utilidad, que debe instalar.

```
tridentctl get backend -o json | jq '[.items[] | {backend: .name, storageClasses: [.storage[].storageClasses]|unique}]'
```

Esto también se aplica a los back-ends que se crearon con el uso `TridentBackendConfig`.

## Pasar entre las opciones de administración del back-end

Conozca las distintas formas de gestionar los back-ends en Astra Trident.

### Opciones para gestionar back-ends

Con la introducción de `TridentBackendConfig`, los administradores ahora tienen dos formas únicas de administrar los back-ends. Esto plantea las siguientes preguntas:

- Pueden crearse back-ends con `tridentctl` administrarse con `TridentBackendConfig`?
- Pueden crearse back-ends con `TridentBackendConfig` se gestionan mediante `tridentctl`?

### Gestione `tridentctl` con los back-ends `TridentBackendConfig`

En esta sección se describen los pasos necesarios para gestionar los back-ends creados con `tridentctl` directamente mediante la interfaz de Kubernetes creando `TridentBackendConfig` objetos.

Esto se aplica a las siguientes situaciones:

- Back-ends preexistentes, que no tienen un `TridentBackendConfig` porque fueron creados con `tridentctl`.
- Nuevos back-ends que se crearon con `tridentctl`, mientras que otros `TridentBackendConfig` existen objetos.

En ambos escenarios, continuarán presentes los back-ends, con los volúmenes de programación de Astra Trident y el funcionamiento de ellos. A continuación, los administradores tienen una de estas dos opciones:

- Siga utilizando `tridentctl` para gestionar los back-ends que se crearon con él.
- Enlazar los back-ends creados con `tridentctl` a un nuevo `TridentBackendConfig` objeto. Hacerlo significaría que se gestionarán los back-ends `kubectl` y no `tridentctl`.

Para administrar un back-end preexistente mediante `kubectl`, tendrá que crear un `TridentBackendConfig` que enlaza con el backend existente. A continuación se ofrece una descripción general de cómo funciona:

1. Cree un secreto de Kubernetes. El secreto contiene las credenciales que Astra Trident necesita para comunicarse con el clúster/servicio de almacenamiento.
2. Cree un `TridentBackendConfig` objeto. Este contiene detalles sobre el servicio/clúster de almacenamiento y hace referencia al secreto creado en el paso anterior. Se debe tener cuidado para especificar parámetros de configuración idénticos (como `spec.backendName`, `spec.storagePrefix`, `spec.storageDriverName`, y así sucesivamente). `spec.backendName` se debe establecer el nombre del backend existente.

### Paso 0: Identificar el back-end

Para crear un `TridentBackendConfig` que se enlaza a un backend existente, necesitará obtener la configuración de backend. En este ejemplo, supongamos que se ha creado un back-end mediante la siguiente definición 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"
      }
    }
  ]
}

```

```
    }  
  }  
]  
}
```

### Paso 1: Cree un secreto de Kubernetes

Cree un secreto que contenga las credenciales del back-end, como se muestra en este ejemplo:

```
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
```

### Paso 2: Cree un TridentBackendConfig CR

El paso siguiente es crear un `TridentBackendConfig` CR que se enlazará automáticamente a la preexistente `ontap-nas-backend` (como en este ejemplo). Asegurarse de que se cumplen los siguientes requisitos:

- El mismo nombre de fondo se define en `spec.backendName`.
- Los parámetros de configuración son idénticos al backend original.
- Los pools virtuales (si están presentes) deben conservar el mismo orden que en el back-end original.
- Las credenciales se proporcionan a través de un secreto de Kubernetes, pero no en texto sin formato.

En este caso, el `TridentBackendConfig` tendrá este aspecto:

```

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: mysqldb
    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

```

### Paso 3: Compruebe el estado del TridentBackendConfig CR

Después del TridentBackendConfig se ha creado, su fase debe ser Bound. También debería reflejar el mismo nombre de fondo y UUID que el del back-end existente.

```
kubectl get tbc tbc-ontap-nas-backend -n trident
NAME                                BACKEND NAME                BACKEND UUID
PHASE    STATUS
tbc-ontap-nas-backend  ontap-nas-backend          52f2eb10-e4c6-4160-99fc-
96b3be5ab5d7    Bound    Success
```

#confirm that no new backends were created (i.e., TridentBackendConfig did not end up creating a new backend)

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

```
+-----+-----+
+-----+-----+-----+-----+
|          NAME          | STORAGE DRIVER |          UUID          |
| STATE  | VOLUMES |          |
+-----+-----+-----+-----+
| ontap-nas-backend     | ontap-nas      | 52f2eb10-e4c6-4160-99fc-
96b3be5ab5d7 | online |          25 |
+-----+-----+-----+-----+
+-----+-----+-----+-----+
```

El back-end se gestionará completamente mediante el tbc-ontap-nas-backend TridentBackendConfig objeto.

**Gestione TridentBackendConfig con los back-ends tridentctl**

`tridentctl` se puede utilizar para enumerar los back-ends que se crearon con `TridentBackendConfig`. Además, los administradores también pueden optar por gestionar completamente estos back-ends `tridentctl` eliminando `TridentBackendConfig` y eso seguro `spec.deletionPolicy` se establece en `retain`.

### Paso 0: Identificar el back-end

Por ejemplo, supongamos que se ha creado el siguiente back-end mediante TridentBackendConfig:



```
kubectl get tbc backend-tbc-ontap-san -n trident -o wide
NAME                                BACKEND NAME                BACKEND UUID
PHASE  STATUS  STORAGE DRIVER  DELETION POLICY
backend-tbc-ontap-san  ontap-san-backend  81abcb27-ea63-49bb-b606-
0a5315ac5f82  Bound  Success  ontap-san  delete

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 |
+-----+-----+
+-----+-----+-----+-----+
```

Desde la salida, se ve eso TridentBackendConfig Se ha creado correctamente y está enlazado a un backend [observe el UUID del backend].

**Paso 1: Confirmar** deletionPolicy **se establece en** retain

Echemos un vistazo al valor de deletionPolicy. Esto debe definirse como retain. Esto asegurará que cuando un TridentBackendConfig Se elimina la CR, la definición de backend seguirá estando presente y se puede gestionar con tridentctl.

```
kubectl get tbc backend-tbc-ontap-san -n trident -o wide
NAME                                BACKEND NAME                BACKEND UUID
PHASE  STATUS  STORAGE DRIVER  DELETION POLICY
backend-tbc-ontap-san  ontap-san-backend  81abcb27-ea63-49bb-b606-
0a5315ac5f82  Bound  Success  ontap-san  delete

# 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
PHASE  STATUS  STORAGE DRIVER  DELETION POLICY
backend-tbc-ontap-san  ontap-san-backend  81abcb27-ea63-49bb-b606-
0a5315ac5f82  Bound  Success  ontap-san  retain
```



No continúe con el siguiente paso a menos que `deletionPolicy` se establece en `retain`.

## Paso 2: Elimine la `TridentBackendConfig` CR

El paso final es eliminar la `TridentBackendConfig` CR. Tras confirmar la `deletionPolicy` se establece en `retain`, puede utilizar `Adelante` con la eliminación:

```
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 |
+-----+-----+-----+
+-----+-----+-----+
```

Tras la eliminación del `TridentBackendConfig` Astra Trident simplemente la elimina sin eliminar realmente el back-end.

# Crear y gestionar clases de almacenamiento

## Cree una clase de almacenamiento

Configure un objeto `StorageClass` de Kubernetes y cree la clase de almacenamiento para indicar a Astra Trident cómo se aprovisionan los volúmenes.

### Configurar un objeto de Kubernetes `StorageClass`

La "[Objeto de Kubernetes `StorageClass`](#)" Identifica Astra Trident como el proveedor que se usa para esa clase indica a Astra Trident cómo aprovisionar un volumen. Por ejemplo:

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: <Name>
provisioner: csi.trident.netapp.io
mountOptions: <Mount Options>
parameters:
  <Trident Parameters>
allowVolumeExpansion: true
volumeBindingMode: Immediate
```

Consulte "[Objetos de Kubernetes y Trident](#)" si desea obtener información detallada sobre cómo interactúan las clases de almacenamiento con el `PersistentVolumeClaim` Y parámetros para controlar de qué forma Astra Trident aprovisiona volúmenes.

### Cree una clase de almacenamiento

Después de crear el objeto `StorageClass`, puede crear la clase de almacenamiento. [Muestras de clase de almacenamiento](#) proporciona algunas muestras básicas que puede utilizar o modificar.

#### Pasos

1. Este es un objeto de Kubernetes, así que use `kubectl` Para crear en Kubernetes.

```
kubectl create -f sample-input/storage-class-basic-csi.yaml
```

2. Ahora debería ver una clase de almacenamiento `* Basic-csi*` tanto en Kubernetes como en Astra Trident, y Astra Trident debería haber descubierto las piscinas en el back-end.

```

kubect1 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"
        ]
      }
    }
  ]
}

```

### Muestras de clase de almacenamiento

Astra Trident proporciona ["definiciones simples de clase de almacenamiento para back-ends específicos"](#).

Como alternativa, puede editar `sample-input/storage-class-csi.yaml.template` archivo que viene con el instalador y reemplácelo `BACKEND_TYPE` con el nombre del controlador de almacenamiento.

```

./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

```

## Gestione las clases de almacenamiento

Puede ver las clases de almacenamiento existentes, definir una clase de almacenamiento predeterminada, identificar el back-end de la clase de almacenamiento y eliminar clases de almacenamiento.

### Consulte las clases de almacenamiento existentes

- Para ver las clases de almacenamiento Kubernetes existentes, ejecute el siguiente comando:

```
kubectl get storageclass
```

- Para ver la información sobre la clase de almacenamiento Kubernetes, ejecute el siguiente comando:

```
kubectl get storageclass <storage-class> -o json
```

- Para ver las clases de almacenamiento sincronizado de Astra Trident, ejecute el siguiente comando:

```
tridentctl get storageclass
```

- Para ver la información detallada de la clase de almacenamiento sincronizado de Astra Trident, ejecute el siguiente comando:

```
tridentctl get storageclass <storage-class> -o json
```

## Establecer una clase de almacenamiento predeterminada

Kubernetes 1.6 añadió la capacidad de establecer un tipo de almacenamiento predeterminado. Esta es la clase de almacenamiento que se usará para aprovisionar un volumen persistente si un usuario no especifica una en una solicitud de volumen persistente (PVC).

- Defina una clase de almacenamiento predeterminada configurando la anotación `storageclass.kubernetes.io/is-default-class` a `true` en la definición de la clase de almacenamiento. Según la especificación, cualquier otro valor o ausencia de la anotación se interpreta como falso.
- Puede configurar una clase de almacenamiento existente para que sea la clase de almacenamiento predeterminada mediante el siguiente comando:

```
kubectl patch storageclass <storage-class-name> -p '{"metadata": {"annotations":{"storageclass.kubernetes.io/is-default-class":"true"}}}'
```

- De forma similar, puede eliminar la anotación predeterminada de la clase de almacenamiento mediante el siguiente comando:

```
kubectl patch storageclass <storage-class-name> -p '{"metadata": {"annotations":{"storageclass.kubernetes.io/is-default-class":"false"}}}'
```

También hay ejemplos en el paquete del instalador de Trident que incluyen esta anotación.



Solo debe haber una clase de almacenamiento predeterminada en el clúster a la vez. Si no dispone de más de una, técnicamente, Kubernetes no le impide ofrecer más de una, pero funcionará como si no hubiera una clase de almacenamiento predeterminada en absoluto.

## Identifique el back-end para una clase de almacenamiento

Este es un ejemplo del tipo de preguntas que puede responder con el JSON que `tridentctl` Salidas para objetos de backend de Astra Trident. Utiliza la `jq` utilidad, que puede necesitar instalar primero.

```
tridentctl get storageclass -o json | jq '[.items[] | {storageClass: .Config.name, backends: [.storage]|unique}]'
```

## Elimine una clase de almacenamiento

Para eliminar una clase de almacenamiento de Kubernetes, ejecute el siguiente comando:

```
kubectl delete storageclass <storage-class>
```

`<storage-class>` debe sustituirse por su clase de almacenamiento.

Cualquier volumen persistente que se cree a través de esta clase de almacenamiento no cambiará y Astra Trident seguirá gestionarlo.



Astra Trident pone en práctica un espacio en blanco `fsType` para los volúmenes que crea. Para los back-ends de iSCSI, se recomienda aplicar `parameters.fsType` En el tipo de almacenamiento. Debe eliminar las clases de almacenamiento existentes y volver a crearlas con `parameters.fsType` especificado.

## Aprovisione y gestione volúmenes

### Aprovisione un volumen

Cree un volumen persistente (VP) y una reclamación de volumen persistente (RVP) que utilice el tipo de almacenamiento de Kubernetes configurado para solicitar acceso al VP. A continuación, puede montar el VP en un pod.

#### Descripción general

1. "*Volumen persistente*" (PV) es un recurso de almacenamiento físico provisionado por el administrador del clúster en un clúster de Kubernetes. La "*Claim de volumen persistente*" (RVP) es una solicitud para acceder al volumen persistente en el clúster.

La RVP se puede configurar para solicitar almacenamiento de un determinado tamaño o modo de acceso. Mediante el StorageClass asociado, el administrador del clúster puede controlar mucho más que el tamaño de los volúmenes persistentes y el modo de acceso, como el rendimiento o el nivel de servicio.

Después de crear el VP y la RVP, puede montar el volumen en un pod.

#### Manifiestos de muestra

## Manifiesto de muestra de volumen persistente

Este manifiesto de ejemplo muestra un PV básico de 10Gi que está asociado con StorageClass `basic-csi`.

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: pv-storage
  labels:
    type: local
spec:
  storageClassName: basic-csi
  capacity:
    storage: 10Gi
  accessModes:
    - ReadWriteOnce
  hostPath:
    path: "/my/host/path"
```



## Manifiestos de muestra de PersistentVolumeClaim

Estos ejemplos muestran opciones básicas de configuración de PVC.

### PVC con acceso RWO

En este ejemplo se muestra una RVP básica con acceso RWO que está asociada con una clase de almacenamiento denominada `basic-csi`.

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: pvc-storage
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: basic-csi
```

### PVC con NVMe/TCP

En este ejemplo, se muestra una PVC básica para NVMe/TCP con acceso RWO asociado con una clase de almacenamiento denominada `protection-gold`.

```
---
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: pvc-san-nvme
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 300Mi
  storageClassName: protection-gold
```

## Muestras de manifiesto de POD

Estos ejemplos muestran configuraciones básicas para conectar la RVP a un pod.

### Configuración 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
```

## Configuración de NVMe/TCP básica

```
---
apiVersion: v1
kind: Pod
metadata:
  creationTimestamp: null
  labels:
    run: nginx
  name: nginx
spec:
  containers:
  - image: nginx
    name: nginx
    resources: {}
    volumeMounts:
    - mountPath: "/usr/share/nginx/html"
      name: task-pv-storage
  dnsPolicy: ClusterFirst
  restartPolicy: Always
  volumes:
  - name: task-pv-storage
    persistentVolumeClaim:
      claimName: pvc-san-nvme
```

## Cree el VP y la RVP

### Pasos

1. Cree el VP.

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

2. Compruebe el estado de PV.

```
kubectl get pv
NAME          CAPACITY  ACCESS MODES  RECLAIM POLICY  STATUS  CLAIM
STORAGECLASS  REASON    AGE
pv-storage    4Gi       RWO           Retain          Available
7s
```

3. Cree la RVP.

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

#### 4. Verifique el estado de la RVP.

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

#### 5. Monte el volumen en un pod.

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



Puede supervisar el progreso con `kubectl get pod --watch`.

#### 6. Compruebe que el volumen está montado en `/my/mount/path`.

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

#### 7. Ahora puede eliminar el Pod. La aplicación Pod ya no existirá, pero el volumen permanecerá.

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

Consulte "[Objetos de Kubernetes y Trident](#)" si desea obtener información detallada sobre cómo interactúan las clases de almacenamiento con el `PersistentVolumeClaim` Y parámetros para controlar de qué forma Astra Trident aprovisiona volúmenes.

## Expanda los volúmenes

Astra Trident ofrece a los usuarios de Kubernetes la capacidad de ampliar sus volúmenes una vez que se han creado. Encuentre información sobre las configuraciones que se necesitan para ampliar los volúmenes iSCSI y NFS.

### Expanda un volumen iSCSI

Puede expandir un volumen persistente iSCSI (PV) mediante el proveedor CSI.



La ampliación del volumen iSCSI se admite en el `ontap-san`, `ontap-san-economy`, `solidfire-san` Requiere Kubernetes 1.16 o posterior.

#### Paso 1: Configure el tipo de almacenamiento para que admita la ampliación de volumen

Edite la definición de `StorageClass` para establecer el `allowVolumeExpansion` campo a `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
```

En el caso de un tipo de almacenamiento existente, edítelo para incluir el `allowVolumeExpansion` parámetro.

### **Paso 2: Cree una RVP con el tipo de almacenamiento que ha creado**

Edite la definición de PVC y actualice el `spec.resources.requests.storage` para reflejar el nuevo tamaño deseado, que debe ser mayor que el tamaño original.

```
cat pvc-ontapsan.yaml
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: san-pvc
spec:
  accessModes:
  - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: ontap-san
```

Astra Trident crea un volumen persistente (PV) y lo asocia con esta solicitud de volumen persistente (PVC).

```

kubect1 get pvc
NAME          STATUS      VOLUME                                     CAPACITY
ACCESS MODES  STORAGECLASS  AGE
san-pvc      Bound       pvc-8a814d62-bd58-4253-b0d1-82f2885db671  1Gi
RWO          ontap-san    8s

kubect1 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

```

### Paso 3: Defina un pod que fije el PVC

Conecte el VP a un pod para que se cambie su tamaño. Existen dos situaciones a la hora de cambiar el tamaño de un VP iSCSI:

- Si el VP está conectado a un pod, Astra Trident amplía el volumen en el back-end de almacenamiento, vuelve a buscar el dispositivo y cambia el tamaño del sistema de archivos.
- Cuando se intenta cambiar el tamaño de un VP sin conectar, Astra Trident amplía el volumen en el back-end de almacenamiento. Una vez que la RVP está Unido a un pod, Trident vuelve a buscar el dispositivo y cambia el tamaño del sistema de archivos. Kubernetes, después, actualiza el tamaño de RVP después de completar correctamente la operación de ampliación.

En este ejemplo, se crea un pod que utiliza `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
```

#### **Paso 4: Expanda el PV**

Para cambiar el tamaño del VP que se ha creado de 1Gi a 2gi, edite la definición de PVC y actualice el `spec.resources.requests.storage` A 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
  ...
```

### Paso 5: Validar la expansión

Para validar que la ampliación ha funcionado correctamente, compruebe el tamaño del volumen PVC, PV y Astra Trident:



```

kubect1 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
kubect1 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 |
PROTOCOL |          BACKEND UUID          |  STATE  | MANAGED |
+-----+-----+-----+
+-----+-----+-----+-----+
| pvc-8a814d62-bd58-4253-b0d1-82f2885db671 | 2.0 GiB | ontap-san    |
block    | a9b7bfff-0505-4e31-b6c5-59f492e02d33 | online | true    |
+-----+-----+-----+
+-----+-----+-----+-----+

```

## Expanda un volumen NFS

Astra Trident admite la ampliación de volúmenes para los VP de NFS provisionados en `ontap-nas`, `ontap-nas-economy`, `ontap-nas-flexgroup`, `gcp-cvs`, y `azure-netapp-files` back-ends.

### Paso 1: Configure el tipo de almacenamiento para que admita la ampliación de volumen

Para cambiar el tamaño de un VP de NFS, el administrador primero tiene que configurar la clase de almacenamiento para permitir la expansión del volumen estableciendo el `allowVolumeExpansion` campo a `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

```

Si ya ha creado una clase de almacenamiento sin esta opción, puede simplemente editar la clase de almacenamiento existente mediante `kubect1 edit storageclass` para permitir la expansión de volumen.

## Paso 2: Cree una RVP con el tipo de almacenamiento que ha creado

```
cat pvc-ontapnas.yaml
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: ontapnas20mb
spec:
  accessModes:
  - ReadWriteOnce
  resources:
    requests:
      storage: 20Mi
  storageClassName: ontapnas
```

Astra Trident debe crear un PV NFS de 20 MiB para esta RVP:

```
kubectl get pvc
NAME                STATUS    VOLUME
CAPACITY            ACCESS MODES  STORAGECLASS  AGE
ontapnas20mb       Bound     pvc-08f3d561-b199-11e9-8d9f-5254004dfdb7  20Mi
RWO                 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     RWO
Delete             Bound     default/ontapnas20mb  ontapnas
2m42s
```

## Paso 3: Expande el PV

Para cambiar el tamaño del VP de 20 MiB recién creado a 1 GiB, edite el RVP y establezca `spec.resources.requests.storage` a 1GiB:

```
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
  ...
```

#### Paso 4: Validar la expansión

Puede validar que el tamaño de la configuración ha funcionado correctamente comprobando el tamaño del volumen PVC, PV y Astra Trident:

```

kubect1 get pvc ontapnas20mb
NAME                STATUS      VOLUME
CAPACITY    ACCESS MODES   STORAGECLASS   AGE
ontapnas20mb  Bound        pvc-08f3d561-b199-11e9-8d9f-5254004dfdb7  1Gi
RWO                ontapnas                4m44s

kubect1 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      |
+-----+-----+-----+-----+
+-----+-----+-----+-----+

```

## Importar volúmenes

Es posible importar volúmenes de almacenamiento existentes como un VP de Kubernetes mediante `tridentctl import`.

### Descripción general y consideraciones

Es posible importar un volumen en Astra Trident para lo siguiente:

- Agrupe en contenedores una aplicación y vuelva a utilizar su conjunto de datos existente
- Utilice el clon de un conjunto de datos para una aplicación efímera
- Reconstruya un clúster de Kubernetes que haya fallado
- Migración de datos de aplicaciones durante la recuperación ante desastres

### Consideraciones

Antes de importar un volumen, revise las siguientes consideraciones.

- Astra Trident solo puede importar volúmenes de ONTAP de tipo RW (lectura y escritura). Los volúmenes del tipo DP (protección de datos) son volúmenes de destino de SnapMirror. Debe romper la relación de reflejo antes de importar el volumen a Astra Trident.

- Sugerimos importar volúmenes sin conexiones activas. Para importar un volumen que se usa activamente, clone el volumen y, a continuación, realice la importación.



Esto es especialmente importante en el caso de volúmenes de bloque, ya que Kubernetes no sabía que la conexión anterior y podría conectar fácilmente un volumen activo a un pod. Esto puede provocar daños en los datos.

- Sin embargo `StorageClass` Debe especificarse en una RVP, Astra Trident no utiliza este parámetro durante la importación. Durante la creación de volúmenes, se usan las clases de almacenamiento para seleccionar entre los pools disponibles según las características de almacenamiento. Como el volumen ya existe, no se requiere ninguna selección de pool durante la importación. Por lo tanto, la importación no fallará incluso si el volumen existe en un back-end o pool que no coincide con la clase de almacenamiento especificada en la RVP.
- El tamaño del volumen existente se determina y se establece en la RVP. Una vez que el controlador de almacenamiento importa el volumen, se crea el PV con un `ClaimRef` al PVC.
  - La política de reclamaciones se establece inicialmente en `retain` En el PV. Una vez que Kubernetes enlaza correctamente la RVP y el VP, se actualiza la política de reclamaciones para que coincida con la política de reclamaciones de la clase de almacenamiento.
  - Si la política de reclamaciones de la clase de almacenamiento es `delete`, El volumen de almacenamiento se eliminará cuando se elimine el PV.
- De forma predeterminada, Astra Trident gestiona la RVP y cambia el nombre de FlexVol y LUN en el back-end. Puede pasar el `--no-manage` indicador para importar un volumen no gestionado. Si utiliza `--no-manage`, Astra Trident no realiza ninguna operación adicional en el PVC o PV durante el ciclo de vida de los objetos. El volumen de almacenamiento no se elimina cuando se elimina el VP, y también se ignoran otras operaciones como el clon de volumen y el cambio de tamaño de volumen.



Esta opción es útil si desea usar Kubernetes para cargas de trabajo en contenedores, pero de lo contrario desea gestionar el ciclo de vida del volumen de almacenamiento fuera de Kubernetes.

- Se agrega una anotación a la RVP y al VP que tiene el doble propósito de indicar que el volumen se importó y si se administran la PVC y la VP. Esta anotación no debe modificarse ni eliminarse.

## Importe un volumen

Puede utilizar `tridentctl import` para importar un volumen.

### Pasos

1. Cree el archivo de reclamación de volumen persistente (RVP) (por ejemplo, `pvc.yaml`) Que se utilizará para crear la RVP. El archivo PVC debe incluir `name`, `namespace`, `accessModes`, y `storageClassName`. Opcionalmente, se puede especificar `unixPermissions` En su definición de PVC.

A continuación se muestra un ejemplo de una especificación mínima:

```

kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: my_claim
  namespace: my_namespace
spec:
  accessModes:
    - ReadWriteOnce
  storageClassName: my_storage_class

```



No incluya parámetros adicionales, como el nombre del VP o el tamaño del volumen. Esto puede provocar un error en el comando de importación.

- Utilice la `tridentctl import volume` Comando para especificar el nombre del back-end de Astra Trident que contiene el volumen y el nombre que identifica de forma única el volumen en el almacenamiento (por ejemplo, ONTAP FlexVol, Element Volume, ruta Cloud Volumes Service). La `-f` Se necesita un argumento para especificar la ruta al archivo PVC.

```

tridentctl import volume <backendName> <volumeName> -f <path-to-pvc-
file>

```

## Ejemplos

Revise los siguientes ejemplos de importación de volúmenes para los controladores compatibles.

### NAS de ONTAP y NAS FlexGroup de ONTAP

Astra Trident admite la importación de volúmenes mediante el `ontap-nas` y `ontap-nas-flexgroup` de windows



- La `ontap-nas-economy` el controlador no puede importar y gestionar qtrees.
- La `ontap-nas` y `ontap-nas-flexgroup` las controladoras no permiten nombres de volúmenes duplicados.

Cada volumen creado con `ontap-nas` Driver es una FlexVol en el clúster de ONTAP. Importación de FlexVols con `ontap-nas` el controlador funciona igual. Una FlexVol que ya existe en un clúster de ONTAP se puede importar como `ontap-nas` RVP. Del mismo modo, los volúmenes FlexGroup se pueden importar del mismo modo `ontap-nas-flexgroup` EVs.

### Ejemplos de NAS de ONTAP

A continuación, se muestra un ejemplo de un volumen gestionado y una importación de volumen no gestionada.

## Volumen gestionado

En el ejemplo siguiente se importa un volumen llamado `managed_volume` en un backend llamado `ontap_nas`:

```
tridentctl import volume ontap_nas managed_volume -f <path-to-pvc-file>
```

PROTOCOL	NAME	BACKEND UUID	SIZE	STATE	STORAGE CLASS	MANAGED
file	pvc-bf5ad463-afbb-11e9-8d9f-5254004dfdb7	c5a6f6a4-b052-423b-80d4-8fb491a14a22	1.0 GiB	online	standard	true

## Volumen no gestionado

Cuando utilice la `--no-manage` Argumento, Astra Trident no cambia el nombre del volumen.

El siguiente ejemplo importa `unmanaged_volume` en la `ontap_nas` backend:

```
tridentctl import volume nas_blog unmanaged_volume -f <path-to-pvc-file> --no-manage
```

PROTOCOL	NAME	BACKEND UUID	SIZE	STATE	STORAGE CLASS	MANAGED
file	pvc-df07d542-afbc-11e9-8d9f-5254004dfdb7	c5a6f6a4-b052-423b-80d4-8fb491a14a22	1.0 GiB	online	standard	false

## SAN de ONTAP

Astra Trident admite la importación de volúmenes mediante el `ontap-san` controlador. No se admite la importación de volúmenes en el `ontap-san-economy` controlador.

Astra Trident puede importar volúmenes FlexVol de SAN de ONTAP que contengan una única LUN. Esto es consistente con `ontap-san` Controlador, que crea una FlexVol para cada RVP y una LUN dentro del FlexVol. Astra Trident importa el FlexVol y lo asocia con la definición de RVP.

## Ejemplos de SAN de ONTAP

A continuación, se muestra un ejemplo de un volumen gestionado y una importación de volumen no gestionada.

### Volumen gestionado

En el caso de los volúmenes gestionados, Astra Trident cambia el nombre del FlexVol al `pvc-<uuid>` Formatear y la LUN dentro de la FlexVol a `lun0`.

El siguiente ejemplo importa el `ontap-san-managed` FlexVol que está presente en el `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 | MANAGED |
+-----+-----+-----+-----+
+-----+-----+-----+-----+
| pvc-d6ee4f54-4e40-4454-92fd-d00fc228d74a | 20 MiB | basic          |
block    | cd394786-ddd5-4470-adc3-10c5ce4ca757 | online | true        |
+-----+-----+-----+-----+
+-----+-----+-----+-----+
```

### Volumen no gestionado

El siguiente ejemplo importa `unmanaged_example_volume` en la `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 | MANAGED |
+-----+-----+-----+-----+
+-----+-----+-----+-----+
| pvc-1fc999c9-ce8c-459c-82e4-ed4380a4b228 | 1.0 GiB | san-blog      |
block    | e3275890-7d80-4af6-90cc-c7a0759f555a | online | false      |
+-----+-----+-----+-----+
+-----+-----+-----+-----+
```

Si tiene LUN asignadas a iGroups que comparten un IQN con un IQN de nodo de Kubernetes, como se muestra en el ejemplo siguiente, recibirá el error: `LUN already mapped to initiator(s) in this group`. Deberá quitar el iniciador o desasignar la LUN para importar el volumen.



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**

Astra Trident admite el software NetApp Element y la importación de volúmenes de NetApp HCI mediante el `solidfire-san` controlador.



El controlador Element admite los nombres de volúmenes duplicados. Sin embargo, Astra Trident devuelve un error si hay nombres de volúmenes duplicados. Como solución alternativa, clone el volumen, proporcione un nombre de volumen único e importe el volumen clonado.

**Ejemplo de elemento**

El siguiente ejemplo importa un `element-managed` volumen en el back-end `element_default`.

```
tridentctl import volume element_default element-managed -f pvc-basic-import.yaml -n trident -d
```

NAME	SIZE	STORAGE CLASS
pvc-970ce1ca-2096-4ecd-8545-ac7edc24a8fe	10 GiB	basic-element

```
block | d3ba047a-ea0b-43f9-9c42-e38e58301c49 | online | true
```

**Google Cloud Platform**

Astra Trident admite la importación de volúmenes mediante el `gcp-cvs` controlador.



Para importar un volumen respaldado por NetApp Cloud Volumes Service en Google Cloud Platform, identifique el volumen según la ruta de volumen. La ruta del volumen es la parte de la ruta de exportación del volumen después del `:/`. Por ejemplo, si la ruta de exportación es `10.0.0.1:/adroit-jolly-swift`, la ruta de volumen es `adroit-jolly-swift`.

**Ejemplo de Google Cloud Platform**

El siguiente ejemplo importa a. gcp-cvs volumen en el back-end gcpcvs\_YEppr con la ruta del volumen de adroit-jolly-swift.

```
tridentctl import volume gcpcvs_YEppr adroit-jolly-swift -f <path-to-pvc-
file> -n trident

+-----+-----+-----+
+-----+-----+-----+-----+
|          NAME          |  SIZE  | STORAGE CLASS |
PROTOCOL |          BACKEND UUID          |  STATE  | MANAGED |
+-----+-----+-----+
+-----+-----+-----+-----+
| pvc-a46ccab7-44aa-4433-94b1-e47fc8c0fa55 | 93 GiB | gcp-storage   | file
| e1a6e65b-299e-4568-ad05-4f0a105c888f | online | true         |
+-----+-----+-----+
+-----+-----+-----+-----+
```

### Azure NetApp Files

Astra Trident admite la importación de volúmenes mediante el azure-netapp-files controlador.



Para importar un volumen de Azure NetApp Files, identifique el volumen por su ruta de volumen. La ruta del volumen es la parte de la ruta de exportación del volumen después del :/. Por ejemplo, si la ruta de montaje es 10.0.0.2:/importvoll1, la ruta de volumen es importvoll1.

### Ejemplo de Azure NetApp Files

El siguiente ejemplo importa un azure-netapp-files volumen en el back-end azurenetappfiles\_40517 con la ruta del volumen importvoll1.

```
tridentctl import volume azurenetappfiles_40517 importvoll1 -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         |
+-----+-----+-----+
+-----+-----+-----+-----+
```

## Comparta un volumen NFS en espacios de nombres

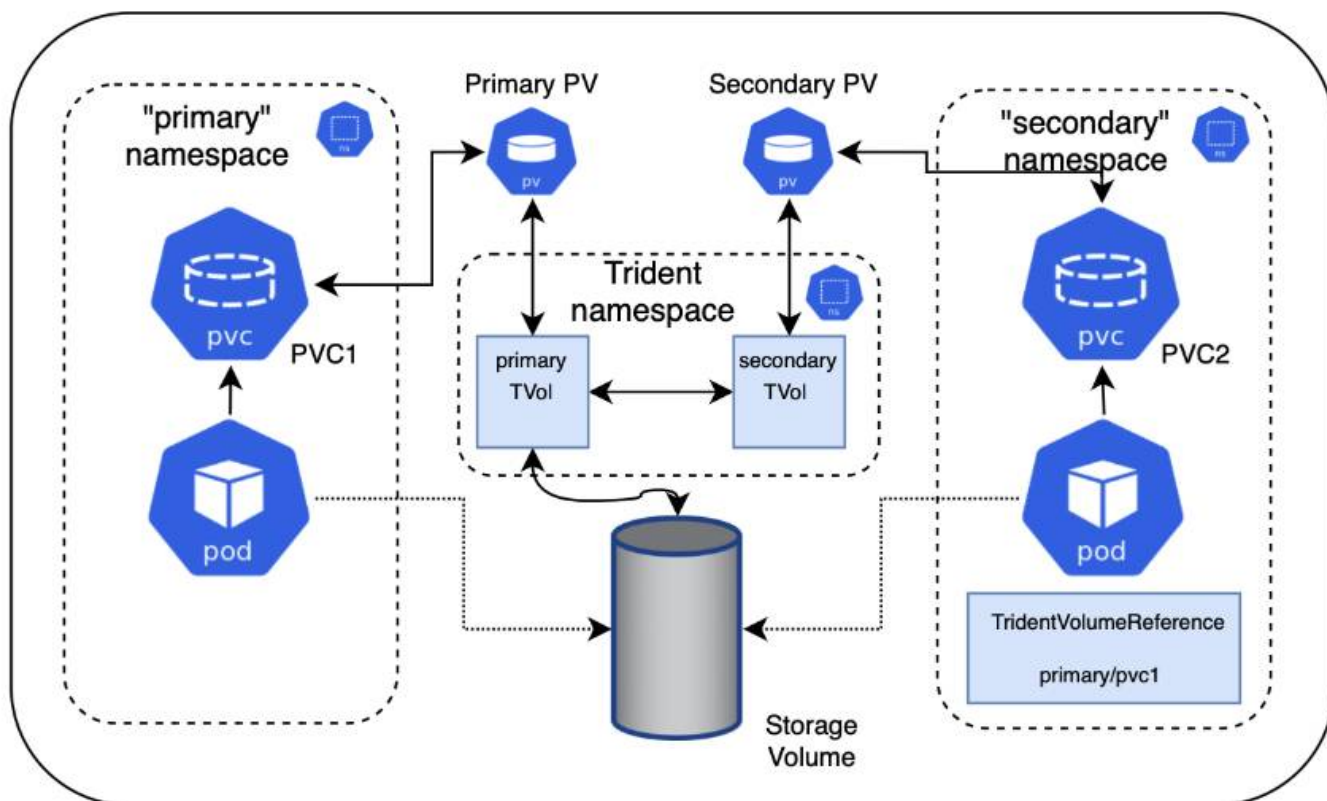
Con Astra Trident, se puede crear un volumen en un espacio de nombres primario y compartirlo en uno o más espacios de nombres secundarios.

### Funciones

La Astra TridentVolumeReference CR le permite compartir de forma segura volúmenes NFS ReadWriteMany (RWX) en uno o más espacios de nombres de Kubernetes. Esta solución nativa de Kubernetes tiene las siguientes ventajas:

- Varios niveles de control de acceso para garantizar la seguridad
- Funciona con todos los controladores de volúmenes NFS de Trident
- No depende de tridentctl ni de ninguna otra función de Kubernetes no nativa

Este diagrama ilustra el uso compartido de volúmenes de NFS en dos espacios de nombres de Kubernetes.



### Inicio rápido

Puede configurar el uso compartido del volumen NFS en unos pocos pasos.

1

#### Configure la RVP de origen para compartir el volumen

El propietario del espacio de nombres de origen concede permiso para acceder a los datos de la RVP de origen.

**2****Conceder permiso para crear una CR en el espacio de nombres de destino**

El administrador del clúster concede permiso al propietario del espacio de nombres de destino para crear el sistema TridentVolumeReference CR.

**3****Cree TridentVolumeReference en el espacio de nombres de destino**

El propietario del espacio de nombres de destino crea el TridentVolumeReference CR para hacer referencia al PVC de origen.

**4****Cree el PVC subordinado en el espacio de nombres de destino**

El propietario del espacio de nombres de destino crea el PVC subordinado para utilizar el origen de datos desde el PVC de origen.

**Configurar los espacios de nombres de origen y destino**

Para garantizar la seguridad, el uso compartido entre espacios de nombres requiere la colaboración y la acción del propietario del espacio de nombres de origen, el administrador de clúster y el propietario del espacio de nombres de destino. La función de usuario se designa en cada paso.

**Pasos**

1. **Propietario del espacio de nombres de origen:** cree el PVC (`pvc1`) en el espacio de nombres de origen que concede permiso para compartir con el espacio de nombres de destino (`namespace2`) utilizando el `shareToNamespace` anotación.

```
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
```

Astra Trident crea el VP y su volumen de almacenamiento NFS back-end.



- Puede compartir el PVC en varios espacios de nombres utilizando una lista delimitada por comas. Por ejemplo: `trident.netapp.io/shareToNamespace: namespace2, namespace3, namespace4`.
- Puede compartir todos los espacios de nombres mediante `*`. Por ejemplo: `trident.netapp.io/shareToNamespace: *`
- Puede actualizar la RVP para incluir el `shareToNamespace` anotación en cualquier momento.

2. **Administrador de clúster:** cree la función personalizada y kubeconfig para conceder permiso al propietario del espacio de nombres de destino para crear el sistema `TridentVolumeReference` CR en el espacio de nombres de destino.
3. **Propietario del espacio de nombres de destino:** cree un sistema `TridentVolumeReference` CR en el espacio de nombres de destino que haga referencia al espacio de nombres de origen `pvc1`.

```
apiVersion: trident.netapp.io/v1
kind: TridentVolumeReference
metadata:
  name: my-first-tvr
  namespace: namespace2
spec:
  pvcName: pvc1
  pvcNamespace: namespace1
```

4. **Propietario del espacio de nombres de destino:** cree un PVC (`pvc2`) en el espacio de nombres de destino (`namespace2`) utilizando el `shareFromPVC` Anotación para designar el PVC de origen.

```
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
```



El tamaño del PVC de destino debe ser menor o igual que el PVC de origen.

## Resultados

Astra Trident lee la `shareFromPVC` Anotación en la RVP de destino y crea el VP de destino como un volumen subordinado sin ningún recurso de almacenamiento propio que apunta al VP de origen y comparte el recurso de almacenamiento VP de origen. La RVP y el VP de destino aparecen vinculados como normales.

## Elimine un volumen compartido

Es posible eliminar un volumen que se comparte en varios espacios de nombres. Astra Trident eliminará el acceso al volumen en el espacio de nombres de origen y mantendrá el acceso a otros espacios de nombres que comparten el volumen. Cuando se eliminan todos los espacios de nombres que hacen referencia al volumen, Astra Trident elimina el volumen.

## Uso `tridentctl get` para consultar volúmenes subordinados

Con el `tridentctl` puede ejecutar la `get` comando para obtener volúmenes subordinados. Para obtener más información, consulte el enlace: [./trident-reference/tridentctl.html](https://trident-reference/tridentctl.html) `tridentctl` comandos y opciones].

```
Usage:
  tridentctl get [option]
```

Indicadores:

- `-h, --help`: Ayuda para volúmenes.
- `--parentOfSubordinate string`: Limite la consulta al volumen de origen subordinado.
- `--subordinateOf string`: Limite la consulta a las subordinadas del volumen.

## Limitaciones

- Astra Trident no puede evitar que los espacios de nombres de destino se escriban en el volumen compartido. Se debe usar el bloqueo de archivos u otros procesos para evitar la sobrescritura de datos de volúmenes compartidos.
- No puede revocar el acceso al PVC de origen quitando el `shareToNamespace` o `shareFromNamespace` anotaciones o eliminar `TridentVolumeReference` CR. Para revocar el acceso, debe eliminar el PVC subordinado.
- Las snapshots, los clones y el mirroring no son posibles en los volúmenes subordinados.

## Si quiere más información

Para obtener más información sobre el acceso de volúmenes entre espacios de nombres:

- Visite ["Uso compartido de volúmenes entre espacios de nombres: Dé la bienvenida al acceso al volumen entre espacios de nombres"](#).
- Vea la demostración en ["NetAppTV"](#).

## Utilice Topología CSI

Astra Trident puede crear y conectar volúmenes a los nodos presentes en un clúster de Kubernetes de forma selectiva mediante el uso de ["Función de topología CSI"](#).

## Descripción general

Con la función de topología CSI, el acceso a los volúmenes puede limitarse a un subconjunto de nodos, en función de regiones y zonas de disponibilidad. En la actualidad, los proveedores de cloud permiten a los administradores de Kubernetes generar nodos basados en zonas. Los nodos se pueden ubicar en diferentes zonas de disponibilidad dentro de una región o en varias regiones. Para facilitar el aprovisionamiento de volúmenes para cargas de trabajo en una arquitectura de varias zonas, Astra Trident utiliza la topología CSI.



Obtenga más información sobre la característica de topología CSI ["aquí"](#).

Kubernetes ofrece dos modos de enlace de volúmenes únicos:

- Con `VolumeBindingMode` establezca en `Immediate`, Astra Trident crea el volumen sin conocimiento de la topología. La vinculación de volúmenes y el aprovisionamiento dinámico se manejan cuando se crea la RVP. Este es el valor predeterminado `VolumeBindingMode` y es adecuado para clústeres que no aplican restricciones de topología. Los volúmenes persistentes se crean sin depender de los requisitos de programación del pod solicitante.
- Con `VolumeBindingMode` establezca en `WaitForFirstConsumer`, La creación y enlace de un volumen persistente para una RVP se retrasa hasta que se programa y crea un pod que usa la RVP. De esta forma, se crean volúmenes con el fin de cumplir las restricciones de programación que se aplican en los requisitos de topología.



La `WaitForFirstConsumer` el modo de encuadernación no requiere etiquetas de topología. Esto se puede utilizar independientemente de la característica de topología CSI.

## Lo que necesitará

Para utilizar la topología CSI, necesita lo siguiente:

- Un clúster de Kubernetes que ejecuta un ["Compatible con la versión de Kubernetes"](#)

```
kubectl version
Client Version: version.Info{Major:"1", Minor:"19",
GitVersion:"v1.19.3",
GitCommit:"1e11e4a2108024935ecfcb2912226cedeadfd99df",
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:"1e11e4a2108024935ecfcb2912226cedeadfd99df",
GitTreeState:"clean", BuildDate:"2020-10-14T12:41:49Z",
GoVersion:"go1.15.2", Compiler:"gc", Platform:"linux/amd64"}
```

- Los nodos del clúster deben tener etiquetas que incluyan el reconocimiento de topología (`topology.kubernetes.io/region` y `topology.kubernetes.io/zone`). Estas etiquetas \* deben estar presentes en los nodos del clúster\* antes de instalar Astra Trident para que Astra Trident tenga en cuenta la topología.

```
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","kuber-
netes.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","kuber-
netes.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","kuber-
netes.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"}]
```

### Paso 1: Cree un backend con detección de topología

Los back-ends de almacenamiento de Astra Trident se pueden diseñar para aprovisionar de forma selectiva volúmenes en función de las zonas de disponibilidad. Cada back-end puede llevar un opcional `supportedTopologies` bloque que representa una lista de zonas y regiones que se deben admitir. En el caso de `StorageClasses` que utilizan dicho back-end, solo se creará un volumen si lo solicita una aplicación programada en una región/zona admitida.

A continuación se muestra un ejemplo de definición 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` se utiliza para proporcionar una lista de regiones y zonas por backend. Estas regiones y zonas representan la lista de valores permitidos que se pueden proporcionar en un `StorageClass`. En el caso de `StorageClasses` que contienen un subconjunto de las regiones y zonas proporcionadas en un back-end, Astra Trident creará un volumen en el back-end.

Puede definir `supportedTopologies` por pool de almacenamiento también. Consulte el siguiente ejemplo:

```

---
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
    region: Iowa-DC
    zone: Iowa-DC-A
    supportedTopologies:
    - topology.kubernetes.io/region: us-centrall
      topology.kubernetes.io/zone: us-centrall-a
- labels:
    workload: dev
    region: Iowa-DC
    zone: Iowa-DC-B
    supportedTopologies:
    - topology.kubernetes.io/region: us-centrall
      topology.kubernetes.io/zone: us-centrall-b

```

En este ejemplo, la `region` y.. `zone` las etiquetas indican la ubicación del pool de almacenamiento. `topology.kubernetes.io/region` y.. `topology.kubernetes.io/zone` dicte desde donde se pueden consumir los pools de almacenamiento.

## Paso 2: Defina las clases de almacenamiento que tienen en cuenta la topología

En función de las etiquetas de topología que se proporcionan a los nodos del clúster, se puede definir `StorageClase` para que contenga información de topología. Esto determinará los pools de almacenamiento que sirven como candidatos para las solicitudes de RVP y el subconjunto de nodos que pueden usar los volúmenes aprovisionados mediante Trident.

Consulte el siguiente ejemplo:

```

apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
name: netapp-san-us-east1
provisioner: csi.trident.netapp.io
volumeBindingMode: WaitForFirstConsumer
allowedTopologies:
- matchLabelExpressions:
- key: topology.kubernetes.io/zone
  values:
- us-east1-a
- us-east1-b
- key: topology.kubernetes.io/region
  values:
- us-east1
parameters:
  fsType: "ext4"

```

En la definición del tipo de almacenamiento que se proporciona anteriormente, `volumeBindingMode` se establece en `WaitForFirstConsumer`. Las RVP solicitadas con este tipo de almacenamiento no se verán en cuestión hasta que se mencionan en un pod. Y, `allowedTopologies` proporciona las zonas y la región que se van a utilizar. La `netapp-san-us-east1 StorageClass` creará EVs en el `san-backend-us-east1 backend` definido anteriormente.

### Paso 3: Cree y utilice un PVC

Con el clase de almacenamiento creado y asignado a un back-end, ahora puede crear RVP.

Vea el ejemplo `spec` a continuación:

```

---
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
name: pvc-san
spec:
accessModes:
- ReadWriteOnce
resources:
  requests:
    storage: 300Mi
storageClassName: netapp-san-us-east1

```

La creación de una RVP con este manifiesto daría como resultado lo siguiente:

```

kubect1 create -f pvc.yaml
persistentvolumeclaim/pvc-san created
kubect1 get pvc
NAME          STATUS      VOLUME   CAPACITY   ACCESS MODES   STORAGECLASS
AGE
pvc-san      Pending
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
  ----      -
  Normal    WaitForFirstConsumer 6s    persistentvolume-controller
waiting
for first consumer to be created before binding

```

Para que Trident cree un volumen y lo enlace a la RVP, use la RVP en un pod. Consulte el siguiente ejemplo:

```

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 indica a Kubernetes que programe el pod de los nodos presentes en el us-east1 region y elija de cualquier nodo que esté presente en el us-east1-a o. us-east1-b zonas.

Consulte la siguiente salida:

```
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
```

### Actualice los back-ends que se incluirán `supportedTopologies`

Se pueden actualizar los back-ends preexistentes para incluir una lista de `supportedTopologies` uso `tridentctl backend update`. Esto no afectará a los volúmenes que ya se han aprovisionado, y sólo se utilizarán en las siguientes CVP.

### Obtenga más información

- ["Gestione recursos para contenedores"](#)
- ["Selector de nodos"](#)
- ["Afinidad y anti-afinidad"](#)
- ["Tolerancias y taints"](#)

## Trabajar con instantáneas

Las snapshots de volúmenes de Kubernetes de Persistent Volumes (VP) permiten copias puntuales de volúmenes. Es posible crear una copia Snapshot de un volumen creado con Astra Trident, importar una copia de Snapshot creada fuera de Astra Trident, crear un volumen nuevo a partir de una copia de Snapshot existente y recuperar datos de volumen de copias Snapshot.

### Descripción general

Admite copias de Snapshot de volumen `ontap-nas`, `ontap-nas-flexgroup`, `ontap-san`, `ontap-san-economy`, `solidfire-san`, `gcp-cvs`, y `azure-netapp-files` de windows

### Antes de empezar

Debe tener un controlador de instantánea externo y definiciones de recursos personalizados (CRD) para trabajar con instantáneas. Esta es la responsabilidad del orquestador de Kubernetes (por ejemplo: Kubeadm, GKE, OpenShift).

Si su distribución de Kubernetes no incluye el controlador de instantáneas ni los CRD, consulte [Implemente una controladora Snapshot de volumen](#).



No cree una controladora Snapshot si crea instantáneas de volumen bajo demanda en un entorno de GKE. GKE utiliza un controlador de instantáneas oculto integrado.

## Cree una copia de Snapshot de volumen

### Pasos

1. Cree un `VolumeSnapshotClass`. Para obtener más información, consulte "[VolumeSnapshotClass](#)".
  - La `driver` Señala el controlador CSI de Astra Trident.
  - `deletionPolicy` puede ser `Delete` o `Retain`. Cuando se establece en `Retain`, la instantánea física subyacente en el clúster de almacenamiento se conserva incluso cuando `VolumeSnapshot` el objeto se ha eliminado.

### Ejemplo

```
cat snap-sc.yaml
apiVersion: snapshot.storage.k8s.io/v1
kind: VolumeSnapshotClass
metadata:
  name: csi-snapclass
driver: csi.trident.netapp.io
deletionPolicy: Delete
```

2. Crear una instantánea de una RVP existente.

### Ejemplos

- En este ejemplo, se crea una copia Snapshot de una RVP existente.

```
cat snap.yaml
apiVersion: snapshot.storage.k8s.io/v1
kind: VolumeSnapshot
metadata:
  name: pvcl-snap
spec:
  volumeSnapshotClassName: csi-snapclass
  source:
    persistentVolumeClaimName: pvcl
```

- En este ejemplo, se crea un objeto Snapshot de volumen para una RVP denominada `pvcl` y el nombre de la copia de snapshot se establece en `pvcl-snap`. Un `VolumeSnapshot` es análogo a un `PVC` y está asociado a un `VolumeSnapshotContent` objeto que representa la instantánea real.

```
kubectl create -f snap.yaml
volumesnapshot.snapshot.storage.k8s.io/pvcl-snap created

kubectl get volumesnapshots
NAME                AGE
pvcl-snap           50s
```

- Puede identificar el `VolumeSnapshotContent` objeto para `pvc1-snap` `VolumeSnapshot`, describiéndolo. La `Snapshot Content Name` Identifica el objeto `VolumeSnapshotContent` que sirve esta snapshot. La `Ready To Use` El parámetro indica que la snapshot se puede usar para crear una nueva RVP.

```
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
.
.
```

## Cree una RVP a partir de una snapshot de volumen

Puede utilizar `dataSource` Para crear una RVP con una Snapshot de volumen llamada `<pvc-name>` como la fuente de los datos. Una vez creada la RVP, se puede conectar a un pod y utilizarla como cualquier otro PVC.



La RVP se creará en el mismo back-end que el volumen de origen. Consulte "[KB: La creación de una RVP a partir de una snapshot de RVP de Trident no se puede crear en un back-end alternativo](#)".

En el siguiente ejemplo se crea la RVP con `pvc1-snap` como origen de datos.



```

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

```

## Importe una copia de Snapshot de volumen

Astra Trident es compatible con el ["Proceso de snapshot aprovisionado previamente de Kubernetes"](#) para habilitar al administrador de clúster para crear un `VolumeSnapshotContent` Objetos e importación de copias de Snapshot creadas fuera de Astra Trident.

### Antes de empezar

Astra Trident debe haber creado o importado el volumen principal del snapshot.

### Pasos

1. **Administrador del clúster:** Crear un `VolumeSnapshotContent` objeto que hace referencia a la instantánea de backend. Esto inicia el flujo de trabajo de las copias Snapshot en Astra Trident.
  - Especifique el nombre de la instantánea de backend en annotations como `trident.netapp.io/internalSnapshotName: <"backend-snapshot-name">`.
  - Especifique `<name-of-parent-volume-in-trident>/<volume-snapshot-content-name>` `pulg snapshotHandle`. Esta es la única información que el snapshot externo proporciona a Astra Trident en la `ListSnapshots` llame.



La `<volumeSnapshotContentName>` No siempre se puede coincidir con el nombre de instantánea de backend debido a restricciones de nomenclatura de CR.

### Ejemplo

En el siguiente ejemplo se crea un `VolumeSnapshotContent` objeto que hace referencia a la instantánea 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>

```

- Administrador del clúster:** Crear el VolumeSnapshot CR que hace referencia al VolumeSnapshotContent objeto. Esto solicita acceso para utilizar el VolumeSnapshot en un espacio de nombres determinado.

### Ejemplo

En el siguiente ejemplo se crea un VolumeSnapshot CR con nombre import-snap que hace referencia a la VolumeSnapshotContent nombre 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

```

- Procesamiento interno (no se requiere ninguna acción):** El Snapshotter externo reconoce el recién creado VolumeSnapshotContent y ejecuta el ListSnapshots llame. Astra Trident crea el TridentSnapshot.
  - El dispositivo de instantáneas externo establece el VolumeSnapshotContent para readyToUse y la VolumeSnapshot para true.
  - Trident vuelve readyToUse=true.
- Cualquier usuario:** Crear a PersistentVolumeClaim para hacer referencia al nuevo VolumeSnapshot, donde spec.dataSource (o spec.dataSourceRef) nombre es el VolumeSnapshot nombre.

### Ejemplo

En el siguiente ejemplo se crea una RVP que hace referencia al VolumeSnapshot nombre 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
```

### Recuperar datos de volumen mediante copias Snapshot

El directorio de snapshots está oculto de forma predeterminada para facilitar la máxima compatibilidad de los volúmenes aprovisionados con el `ontap-nas` y `ontap-nas-economy` de windows Habilite el `.snapshot` directorio para recuperar datos de snapshots directamente.

Use la interfaz de línea de comandos de ONTAP para restaurar un volumen en un estado registrado en una snapshot anterior.

```
cluster1:*> volume snapshot restore -vserver vs0 -volume vol3 -snapshot
vol3_snap_archive
```



Quando se restaura una copia Snapshot, se sobrescribe la configuración de volúmenes existente. Se pierden los cambios que se hagan en los datos del volumen después de crear la copia Snapshot.

### Eliminar un VP con snapshots asociadas

Quando se elimina un volumen persistente con instantáneas asociadas, el volumen Trident correspondiente se actualiza a un “estado de eliminación”. Quite las snapshots de volumen para eliminar el volumen de Astra Trident.

### Implemente una controladora Snapshot de volumen

Si su distribución de Kubernetes no incluye el controlador de snapshots y los CRD, puede implementarlos de la siguiente manera.

#### Pasos

## 1. Crear CRD de snapshot de volumen.

```
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.yam
l
kubectl apply -f https://raw.githubusercontent.com/kubernetes-
csi/external-snapshotter/release-
6.1/client/config/crd/snapshot.storage.k8s.io_volumesnapshots.yaml
```

## 2. Cree la controladora Snapshot.

```
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
```



Si es necesario, abra `deploy/kubernetes/snapshot-controller/rbac-snapshot-controller.yaml` y actualícelo namespace en el espacio de nombres.

### Enlaces relacionados

- ["Copias de Snapshot de volumen"](#)
- ["VolumeSnapshotClass"](#)

## Información de copyright

Copyright © 2024 NetApp, Inc. Todos los derechos reservados. Imprimido en EE. UU. No se puede reproducir este documento protegido por copyright ni parte del mismo de ninguna forma ni por ningún medio (gráfico, electrónico o mecánico, incluidas fotocopias, grabaciones o almacenamiento en un sistema de recuperación electrónico) sin la autorización previa y por escrito del propietario del copyright.

El software derivado del material de NetApp con copyright está sujeto a la siguiente licencia y exención de responsabilidad:

ESTE SOFTWARE LO PROPORCIONA NETAPP «TAL CUAL» Y SIN NINGUNA GARANTÍA EXPRESA O IMPLÍCITA, INCLUYENDO, SIN LIMITAR, LAS GARANTÍAS IMPLÍCITAS DE COMERCIALIZACIÓN O IDONEIDAD PARA UN FIN CONCRETO, CUYA RESPONSABILIDAD QUEDA EXIMIDA POR EL PRESENTE DOCUMENTO. EN NINGÚN CASO NETAPP SERÁ RESPONSABLE DE NINGÚN DAÑO DIRECTO, INDIRECTO, ESPECIAL, EJEMPLAR O RESULTANTE (INCLUYENDO, ENTRE OTROS, LA OBTENCIÓN DE BIENES O SERVICIOS SUSTITUTIVOS, PÉRDIDA DE USO, DE DATOS O DE BENEFICIOS, O INTERRUPTIÓN DE LA ACTIVIDAD EMPRESARIAL) CUALQUIERA SEA EL MODO EN EL QUE SE PRODUJERON Y LA TEORÍA DE RESPONSABILIDAD QUE SE APLIQUE, YA SEA EN CONTRATO, RESPONSABILIDAD OBJETIVA O AGRAVIO (INCLUIDA LA NEGLIGENCIA U OTRO TIPO), QUE SURJAN DE ALGÚN MODO DEL USO DE ESTE SOFTWARE, INCLUSO SI HUBIEREN SIDO ADVERTIDOS DE LA POSIBILIDAD DE TALES DAÑOS.

NetApp se reserva el derecho de modificar cualquiera de los productos aquí descritos en cualquier momento y sin aviso previo. NetApp no asume ningún tipo de responsabilidad que surja del uso de los productos aquí descritos, excepto aquello expresamente acordado por escrito por parte de NetApp. El uso o adquisición de este producto no lleva implícita ninguna licencia con derechos de patente, de marcas comerciales o cualquier otro derecho de propiedad intelectual de NetApp.

Es posible que el producto que se describe en este manual esté protegido por una o más patentes de EE. UU., patentes extranjeras o solicitudes pendientes.

LEYENDA DE DERECHOS LIMITADOS: el uso, la copia o la divulgación por parte del gobierno están sujetos a las restricciones establecidas en el subpárrafo (b)(3) de los derechos de datos técnicos y productos no comerciales de DFARS 252.227-7013 (FEB de 2014) y FAR 52.227-19 (DIC de 2007).

Los datos aquí contenidos pertenecen a un producto comercial o servicio comercial (como se define en FAR 2.101) y son propiedad de NetApp, Inc. Todos los datos técnicos y el software informático de NetApp que se proporcionan en este Acuerdo tienen una naturaleza comercial y se han desarrollado exclusivamente con fondos privados. El Gobierno de EE. UU. tiene una licencia limitada, irrevocable, no exclusiva, no transferible, no sublicenciable y de alcance mundial para utilizar los Datos en relación con el contrato del Gobierno de los Estados Unidos bajo el cual se proporcionaron los Datos. Excepto que aquí se disponga lo contrario, los Datos no se pueden utilizar, desvelar, reproducir, modificar, interpretar o mostrar sin la previa aprobación por escrito de NetApp, Inc. Los derechos de licencia del Gobierno de los Estados Unidos de América y su Departamento de Defensa se limitan a los derechos identificados en la cláusula 252.227-7015(b) de la sección DFARS (FEB de 2014).

## Información de la marca comercial

NETAPP, el logotipo de NETAPP y las marcas que constan en <http://www.netapp.com/TM> son marcas comerciales de NetApp, Inc. El resto de nombres de empresa y de producto pueden ser marcas comerciales de sus respectivos propietarios.