



# **Utilice Astra Trident**

## **Astra Trident**

NetApp  
April 16, 2024

# Tabla de contenidos

- Utilice Astra Trident ..... 1
  - Configurar los back-ends ..... 1
  - Cree back-ends con kubectl ..... 75
  - Realice la gestión del entorno de administración con kubectl ..... 83
  - Realizar la administración de back-end con trimentctl ..... 84
  - Pasar entre las opciones de administración del back-end ..... 86
  - Gestione las clases de almacenamiento ..... 92
  - Realizar operaciones de volumen ..... 94
  - Prepare el nodo de trabajo ..... 119
  - Preparación automática del nodo de trabajo ..... 123
  - Supervisión de Astra Trident ..... 123

# Utilice Astra Trident

## 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 ofrecerá automáticamente pools de almacenamiento de back-ends que cumplan los requisitos definidos por una clase de almacenamiento. Obtenga más información sobre la configuración del back-end en función del tipo de sistema de almacenamiento que tenga.

- ["Configure un back-end de Azure NetApp Files"](#)
- ["Configure un back-end de Cloud Volumes Service para AWS"](#)
- ["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"](#)

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

Descubra cómo configurar Azure NetApp Files (ANF) como back-end para la instalación de Astra Trident con las configuraciones de ejemplo proporcionadas.



El servicio Azure NetApp Files no admite volúmenes de menos de 100 GB. Astra Trident crea automáticamente volúmenes de 100 GB si se solicita un volumen más pequeño.

#### Lo que necesitará

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

- `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 la aplicación debe utilizar la `Owner` o `Contributor` Rol predefinido por Azure.



Para obtener más información acerca de los roles integrados de Azure, consulte ["Documentación de Azure"](#).

- `Azure location` que contiene al menos uno ["subred delegada"](#).
- Si es la primera vez que utiliza Azure NetApp Files o está en una nueva ubicación, se requiere alguna configuración inicial. Consulte ["guía de inicio rápido"](#).

#### Acerca de esta tarea

Según la configuración del back-end (subred, red virtual, nivel de servicio y ubicación), Trident crea volúmenes ANF en los pools de capacidad disponibles en la ubicación solicitada y que coinciden con el nivel de servicio y la subred solicitados.



Astra Trident 21.04.0 y versiones anteriores no admiten pools de capacidad QoS manuales.

## 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	"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	
tenantID	El ID de inquilino de un registro de aplicación	
clientID	El ID de cliente de un registro de aplicación	
clientSecret	El secreto de cliente de un registro de aplicaciones	
serviceLevel	Uno de Standard, Premium, o. Ultra	"" (aleatorio)
location	Nombre de la ubicación de Azure donde se crearán los nuevos volúmenes	"" (aleatorio)
virtualNetwork	Nombre de una red virtual con una subred delegada	"" (aleatorio)
subnet	Nombre de una subred delegada a. Microsoft.Netapp/volumes	"" (aleatorio)
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)
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 registro detallado.	nulo



Modificando el `capacityPools` campo en un back-end existente de tal modo que reduce el número de pools de capacidad utilizados para el aprovisionamiento provoca volúmenes huérfanos, que se aprovisionan en el pool o pools de capacidad que no forman parte del `capacityPools` lista más. Se producirá un error en las operaciones de clonado en estos volúmenes huérfanos.



Si aparece un error que indica que no se han encontrado pools de capacidad al intentar crear un PVC, es probable que el registro de aplicaciones no tenga asociados los permisos y recursos necesarios (subred, red virtual o pool de capacidad). Astra Trident registrará los recursos de Azure que descubrió cuando se crea el back-end al habilitar la depuración. Asegúrese de comprobar si se está utilizando un rol adecuado.



Si desea montar los volúmenes mediante NFS versión 4.1, puede incluir `nfsvers=4` En la lista de opciones de montaje delimitadas por comas para elegir NFS v4.1. Cualquier opción de montaje establecida en una clase de almacenamiento anula las opciones de montaje establecidas en un archivo de configuración de back-end.

Puede controlar de forma predeterminada el modo en que se aprovisiona cada volumen especificando las siguientes opciones en una sección especial del archivo de configuración. Vea los ejemplos de configuración a continuación.

Parámetro	Descripción	Predeterminado
<code>exportRule</code>	Las reglas de exportación de los nuevos volúmenes	"0.0.0.0/0"
<code>size</code>	El tamaño predeterminado de los volúmenes nuevos	"100 G"

La `exportRule` El valor debe ser una lista separada por comas con cualquier combinación de direcciones IPv4 o subredes IPv4 en notación CIDR.



Para todos los volúmenes creados en un back-end de ANF, Astra Trident copia todas las etiquetas presentes en un pool de almacenamiento al volumen de almacenamiento en el momento en el que se aprovisiona. Los administradores de almacenamiento pueden definir etiquetas por pool de almacenamiento y agrupar todos los volúmenes creados en un pool de almacenamiento. Esto proporciona una forma cómoda de diferenciar los volúmenes basándose en un conjunto de etiquetas personalizables que se proporcionan en la configuración del back-end.

### Ejemplo 1: Configuración mínima

Ésta es la configuración mínima absoluta del back-end. Con esta configuración, Astra Trident descubre todas sus cuentas, pools de capacidad y subredes de NetApp delegadas en ANF en cada ubicación del mundo y coloca nuevos volúmenes en una de ellas de forma aleatoria.

Esta configuración es ideal cuando simplemente va a empezar con ANF e intentar cosas, pero en la práctica va a querer proporcionar un ámbito adicional para los volúmenes que debe aprovisionar.

```
{
  "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"
}
```

## Ejemplo 2: Ubicación única y configuración específica de nivel de servicio

Esta configuración de back-end coloca volúmenes en las de Azure eastus ubicación en un Premium pool de capacidad. Astra Trident descubre automáticamente todas las subredes delegadas a ANF en esa ubicación y coloca un nuevo volumen 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": "Premium"
}
```

## Ejemplo 3: 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": "Premium",
  "virtualNetwork": "my-virtual-network",
  "subnet": "my-subnet",
  "nfsMountOptions": "vers=3,proto=tcp,timeo=600",
  "limitVolumeSize": "500Gi",
  "defaults": {
    "exportRule": "10.0.0.0/24,10.0.1.0/24,10.0.2.100",
    "size": "200Gi"
  }
}

```

#### Ejemplo 4: Configuración de pool de almacenamiento 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.

```

{
  "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",
  "nfsMountOptions": "vers=3,proto=tcp,timeo=600",
  "labels": {
    "cloud": "azure"
  },
  "location": "eastus",

  "storage": [
    {
      "labels": {
        "performance": "gold"
      },
      "serviceLevel": "Ultra"
    },
    {
      "labels": {
        "performance": "silver"
      },
      "serviceLevel": "Premium"
    },
    {
      "labels": {
        "performance": "bronze"
      },
      "serviceLevel": "Standard",
    }
  ]
}

```

Lo siguiente `StorageClass` las definiciones hacen referencia a los pools de almacenamiento anteriores. Mediante el uso de `parameters.selector` puede especificar para cada campo `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

```

## 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 CVS para back-end de AWS

Obtenga más información sobre cómo configurar NetApp Cloud Volumes Service (CVS) para AWS como

back-end para su instalación de Astra Trident con las configuraciones de ejemplo proporcionadas.



Cloud Volumes Service para AWS no admite volúmenes de menos de 100 GB. Trident crea automáticamente volúmenes de 100 GB si se solicita un volumen más pequeño.

### Lo que necesitará

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

- Una cuenta de AWS configurada con CVS de NetApp
- Región API, URL y claves para su cuenta CVS

### 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	«aws-cvs»
backendName	Nombre personalizado o el back-end de almacenamiento	Nombre de controlador + "_" + parte de la clave de API
apiRegion	Región de la cuenta CVS. Puede encontrar el valor en el portal web CVS en Configuración de cuentas/acceso API.	
apiURL	URL DE LA API DE LA CUENTA CVS. Puede encontrar el valor en el portal web CVS en Configuración de cuentas/acceso API.	
apiKey	Clave API de cuenta CVS. Puede encontrar el valor en el portal web CVS en Configuración de cuentas/acceso API.	
secretKey	Clave secreta de cuenta CVS. Puede encontrar el valor en el portal web CVS en Configuración de cuentas/acceso API.	
proxyURL	URL de proxy si se requiere servidor proxy para conectarse a la cuenta CVS. El servidor proxy puede ser un proxy HTTP o HTTPS. En el caso de un proxy HTTPS, se omite la validación de certificados para permitir el uso de certificados autofirmados en el servidor proxy. No se admiten los servidores proxy con autenticación habilitada.	

Parámetro	Descripción	Predeterminado
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	El nivel de servicio CVS para nuevos volúmenes. Los valores son "estándar", "premium" y "extremo".	"estándar"
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 registro detallado.	nulo



apiURL es único para cada uno apiRegion. Por ejemplo, el US-West-2 apiRegion tiene la <https://cv.us-west-2.netapp.com:8080/v1/> apiURL. Del mismo modo, el US-East-1 apiRegion tiene la <https://cds-aws-bundles.netapp.com:8080/v1/> apiURL. Asegúrese de comprobar el CVS Dashboard para ver cuál es el correcto apiRegion y.. apiURL parámetros para la configuración de back-end.

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

Puede controlar de forma predeterminada el modo en que se aprovisiona cada volumen especificando las siguientes opciones en una sección especial del archivo de configuración. Vea los ejemplos de configuración a continuación.

Parámetro	Descripción	Predeterminado
exportRule	Las reglas de exportación de los nuevos volúmenes	"0.0.0.0/0"
snapshotDir	Controla la visibilidad de .snapshot directorio	"falso"
snapshotReserve	Porcentaje de volumen reservado para las Snapshot	"" (Aceptar CVS por defecto de 0)
size	El tamaño de los volúmenes nuevos	"100 G"

La exportRule El valor debe ser una lista separada por comas con cualquier combinación de direcciones IPv4 o subredes IPv4 en notación CIDR.



Para todos los volúmenes creados en un back-end de AWS CVS, Astra Trident copia todas las etiquetas presentes en un pool de almacenamiento al volumen de almacenamiento en el momento en que se aprovisiona. Los administradores de almacenamiento pueden definir etiquetas por pool de almacenamiento y agrupar todos los volúmenes creados en un pool de almacenamiento. Esto proporciona una forma cómoda de diferenciar los volúmenes basándose en un conjunto de etiquetas personalizables que se proporcionan en la configuración del back-end.

### Ejemplo 1: Configuración mínima

Ésta es la configuración mínima absoluta del back-end.

Esta configuración es ideal cuando todavía te estás empezando con CVS AWS y quieres probar cosas, pero en la práctica te vas a querer proporcionar un ámbito adicional para los volúmenes que provisionas.

```
{
  "version": 1,
  "storageDriverName": "aws-cvs",
  "apiRegion": "us-east-1",
  "apiURL": "https://cds-aws-bundles.netapp.com:8080/v1",
  "apiKey": "znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGz1zZE",
  "secretKey": "rR0rUmWXfNioN1KhtHisiSAnoTherboGuskey6pU"
}
```

### Ejemplo 2: Configuración de un único nivel de servicio

Este ejemplo muestra un archivo de entorno de administración que aplica los mismos aspectos a todo el almacenamiento creado por Astra Trident en la región US-East-1 de AWS. En este ejemplo también se muestra el uso de `proxyURL` en el archivo de fondo.

```
{
  "version": 1,
  "storageDriverName": "aws-cvs",
  "backendName": "cvs-aws-us-east",
  "apiRegion": "us-east-1",
  "apiURL": "https://cds-aws-bundles.netapp.com:8080/v1",
  "apiKey": "znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGz1zZE",
  "secretKey": "rR0rUmWXfNioN1KhtHisIsAnoTherboGuskey6pU",
  "proxyURL": "http://proxy-server-hostname/",
  "nfsMountOptions": "vers=3,proto=tcp,timeo=600",
  "limitVolumeSize": "50Gi",
  "serviceLevel": "premium",
  "defaults": {
    "snapshotDir": "true",
    "snapshotReserve": "5",
    "exportRule": "10.0.0.0/24,10.0.1.0/24,10.0.2.100",
    "size": "200Gi"
  }
}
```

### Ejemplo 3: Configuración de pool de almacenamiento virtual

En este ejemplo, se muestra el archivo de definición del back-end configurado con pools de almacenamiento virtual junto con StorageClasses que se denominan.

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 `snapshotReserve` con el 5% y la `exportRule` a 0.0.0.0/0. Los pools de almacenamiento virtual se definen en la `storage` sección. En este ejemplo, cada pool de almacenamiento individual establece su propia `serviceLevel`, y algunos pools sobrescriben los valores predeterminados.

```
{
  "version": 1,
  "storageDriverName": "aws-cvs",
  "apiRegion": "us-east-1",
  "apiURL": "https://cds-aws-bundles.netapp.com:8080/v1",
  "apiKey": "EnterYourAPIKeyHere*****",
  "secretKey": "EnterYourSecretKeyHere*****",
  "nfsMountOptions": "vers=3,proto=tcp,timeo=600",

  "defaults": {
    "snapshotReserve": "5",
    "exportRule": "0.0.0.0/0"
  },

  "labels": {
```

```

    "cloud": "aws"
  },
  "region": "us-east-1",

  "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"
}
]
}

```

Las siguientes definiciones de StorageClass se refieren a los pools de almacenamiento anteriores. Mediante el uso de `parameters.selector` Campo, se puede especificar para cada clase de almacenamiento el pool virtual que se usa para alojar un volumen. Los aspectos definidos en el pool elegido serán el volumen.

El primer tipo de almacenamiento (`cvs-extreme-extra-protection`) se asigna al primer grupo de almacenamiento 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 de almacenamiento virtual se selecciona y garantiza que se cumpla el requisito de reserva Snapshot.

```

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

```

```

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

```

## 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 CVS para back-end de GCP

Obtenga más información sobre cómo configurar NetApp Cloud Volumes Service (CVS) para Google Cloud Platform (GCP) como back-end de su instalación de Astra Trident con las configuraciones de muestra proporcionadas.





NetApp Cloud Volumes Service for Google Cloud no admite volúmenes CVS con un tamaño inferior a 100 GiB ni volúmenes CVS con un tamaño inferior a 300 GiB. Astra Trident crea automáticamente volúmenes con el tamaño mínimo si un volumen solicitado es menor que el tamaño mínimo.

### 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 CVS 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 su cuenta de servicio CVS

Astra Trident ahora incluye compatibilidad con volúmenes más pequeños con los valores predeterminados ["Tipo de servicio CVS en GCP"](#). Para los back-ends creados con `storageClass=software`, Los volúmenes tendrán ahora un tamaño de aprovisionamiento mínimo de 300 GiB. CVS actualmente ofrece esta función bajo disponibilidad controlada y no proporciona asistencia técnica. Los usuarios deben registrarse para acceder a volúmenes inferiores a 1 TiB ["aquí"](#). NetApp recomienda a los clientes que consuman volúmenes de menos de 1 TiB para **las cargas de trabajo sin producción**.



Al implementar los back-ends con el tipo de servicio CVS predeterminado (`storageClass=software`), los usuarios deben obtener acceso a la función de volúmenes de sub1 TiB en GCP para los números de proyecto y los ID de proyecto en cuestión. Esto es necesario para que Astra Trident aprovisiona volúmenes inferiores a 1 TiB. Si no es así, se producirá un error al crear el volumen para las RVP que son inferiores a 600 GiB. Obtenga acceso a volúmenes inferiores de 1 TiB mediante el ["este formulario"](#).

Los volúmenes creados por Astra Trident para el nivel de servicio CVS predeterminado se aprovisionan de la siguiente manera:

- Las RVP que son menores de 300 GiB hacen que Astra Trident cree un volumen CVS de 300 GiB.
- Las RVP que están entre 300 GiB y 600 GiB tendrán como resultado que Astra Trident cree un volumen CVS del tamaño solicitado.
- Las RVP que están entre 600 GiB y 1 TiB darán como resultado que Astra Trident crea un volumen CVS de 1 TiB.
- Las RVP que son mayores de 1 TiB darán como resultado que Astra Trident cree un volumen CVS del tamaño solicitado.

### Opciones de configuración del back-end

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

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

Parámetro	Descripción	Predeterminado
storageClass	Tipo de almacenamiento. Elija entre <code>hardware</code> (rendimiento optimizado) o <code>software</code> (Tipo de servicio CVS)	
projectNumber	Número de proyecto de cuenta de Google Cloud. El valor se encuentra en la página de inicio del portal de Google Cloud.	
apiRegion	Región de la cuenta CVS. Es la región en la que el back-end aprovisionará los volúmenes.	
apiKey	Clave de API para la cuenta de servicio de Google Cloud con el <code>netappcloudvolumes.admin</code> función. 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).	
proxyURL	URL de proxy si se requiere servidor proxy para conectarse a la cuenta CVS. El servidor proxy puede ser un proxy HTTP o HTTPS. En el caso de un proxy HTTPS, se omite la validación de certificados para permitir el uso de certificados autofirmados en el servidor proxy. No se admiten los servidores proxy con autenticación habilitada.	
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	El nivel de servicio CVS para nuevos volúmenes. Los valores son "estándar", "premium" y "extremo".	"estándar"
network	GCP se utiliza para volúmenes CVS	"predeterminado"

Parámetro	Descripción	Predeterminado
debugTraceFlags	Indicadores de depuración que se deben usar para la solución de problemas. Ejemplo: <pre>\{"api":false, "method":true\}</pre> . No lo utilice a menos que esté solucionando problemas y necesite un volcado de registro detallado.	nulo

Si se utiliza una red VPC compartida, ambos `projectNumber` y `hostProjectNumber` debe especificarse. En ese caso, `projectNumber` es el proyecto de servicio, y `hostProjectNumber` es el proyecto anfitrión.

La `apiRegion` Representa la región de GCP en la que Astra Trident crea volúmenes CVS. Astra Trident puede montar y conectar volúmenes en nodos Kubernetes que pertenecen a la misma región de GCP. Al crear un back-end, es importante asegurarse `apiRegion` Coincide con la región en la que se implementan los nodos de Kubernetes. Cuando se crean clústeres de Kubernetes en varias regiones, volúmenes CVS que se crean en un determinado `apiRegion` Solo se puede utilizar en cargas de trabajo programadas en nodos que se encuentren en la misma región de GCP.



`storageClass` es un parámetro opcional que puede utilizar para seleccionar el deseado "[Tipo de servicio CVS](#)". Puede elegir entre el tipo de servicio CVS básico (`storageClass=software`) O el tipo de servicio CVS-Performance (`storageClass=hardware`), que Trident utiliza de forma predeterminada. Asegúrese de especificar un `apiRegion` Esto proporciona el CVS correspondiente `storageClass` en su definición de backend.



La integración de Astra Trident con el tipo de servicio CVS básico en Google Cloud es una **funcionalidad beta**, no está pensada para cargas de trabajo de producción. Trident es **totalmente compatible** con el tipo de servicio CVS-Performance y lo usa de forma predeterminada.

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

Puede controlar de forma predeterminada el modo en que se aprovisiona cada volumen especificando las siguientes opciones en una sección especial del archivo de configuración. Vea los ejemplos de configuración a continuación.

Parámetro	Descripción	Predeterminado
exportRule	Las reglas de exportación de los nuevos volúmenes	"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	El tamaño de los volúmenes nuevos	"100Gi"

La `exportRule` El valor debe ser una lista separada por comas con cualquier combinación de direcciones



Para todos los volúmenes creados en un back-end de Google Cloud CVS, Trident copia todas las etiquetas presentes en un pool de almacenamiento en el volumen de almacenamiento en el momento en que se aprovisiona. Los administradores de almacenamiento pueden definir etiquetas por pool de almacenamiento y agrupar todos los volúmenes creados en un pool de almacenamiento. Esto proporciona una forma cómoda de diferenciar los volúmenes basándose en un conjunto de etiquetas personalizables que se proporcionan en la configuración del back-end.

### Ejemplo 1: Configuración mínima

Ésta es la configuración mínima absoluta del back-end.

```
{
  "version": 1,
  "storageDriverName": "gcp-cvs",
  "projectNumber": "012345678901",
  "apiRegion": "us-west2",
  "apiKey": {
    "type": "service_account",
    "project_id": "my-gcp-project",
    "private_key_id": "1234567890123456789012345678901234567890",
    "private_key": "-----BEGIN PRIVATE KEY-----
\nznHczZsrtrHisIsAbOguSaPIKeyAZNchRAGzllZE4jK3bl/qP8B4Kws8zX5ojY9m\nznHczZ
srtrHisIsAbOguSaPIKeyAZNchRAGzllZE4jK3bl/qP8B4Kws8zX5ojY9m\nznHczZsrtrHisI
sAbOguSaPIKeyAZNchRAGzllZE4jK3bl/qP8B4Kws8zX5ojY9m\nznHczZsrtrHisIsAbOguSa
PIKeyAZNchRAGzllZE4jK3bl/qP8B4Kws8zX5ojY9m\nznHczZsrtrHisIsAbOguSaPIKeyAZN
chRAGzllZE4jK3bl/qP8B4Kws8zX5ojY9m\nznHczZsrtrHisIsAbOguSaPIKeyAZNchRAGzll
ZE4jK3bl/qP8B4Kws8zX5ojY9m\nznHczZsrtrHisIsAbOguSaPIKeyAZNchRAGzllZE4jK3bl
/qP8B4Kws8zX5ojY9m\nznHczZsrtrHisIsAbOguSaPIKeyAZNchRAGzllZE4jK3bl/qP8B4K
ws8zX5ojY9m\nznHczZsrtrHisIsAbOguSaPIKeyAZNchRAGzllZE4jK3bl/qP8B4Kws8zX5oj
Y9m\nznHczZsrtrHisIsAbOguSaPIKeyAZNchRAGzllZE4jK3bl/qP8B4Kws8zX5ojY9m\nznHc
zZsrtrHisIsAbOguSaPIKeyAZNchRAGzllZE4jK3bl/qP8B4Kws8zX5ojY9m\nznHczZsrtrHi
sIsAbOguSaPIKeyAZNchRAGzllZE4jK3bl/qP8B4Kws8zX5ojY9m\nznHczZsrtrHisIsAbOgu
SaPIKeyAZNchRAGzllZE4jK3bl/qP8B4Kws8zX5ojY9m\nznHczZsrtrHisIsAbOguSaPIKeyA
ZNchRAGzllZE4jK3bl/qP8B4Kws8zX5ojY9m\nznHczZsrtrHisIsAbOguSaPIKeyAZNchRAGz
llZE4jK3bl/qP8B4Kws8zX5ojY9m\nznHczZsrtrHisIsAbOguSaPIKeyAZNchRAGzllZE4jK3
bl/qP8B4Kws8zX5ojY9m\nznHczZsrtrHisIsAbOguSaPIKeyAZNchRAGzllZE4jK3bl/qP8B4
Kws8zX5ojY9m\nznHczZsrtrHisIsAbOguSaPIKeyAZNchRAGzllZE4jK3bl/qP8B4Kws8zX5o
jY9m\nznHczZsrtrHisIsAbOguSaPIKeyAZNchRAGzllZE4jK3bl/qP8B4Kws8zX5ojY9m\nzn
HczZsrtrHisIsAbOguSaPIKeyAZNchRAGzllZE4jK3bl/qP8B4Kws8zX5ojY9m\nznHczZsrtr
HisIsAbOguSaPIKeyAZNchRAGzllZE4jK3bl/qP8B4Kws8zX5ojY9m\nznHczZsrtrHisIsAbO
guSaPIKeyAZNchRAGzllZE4jK3bl/qP8B4Kws8zX5ojY9m\nznHczZsrtrHisIsAbOguSaPIKe
yAZNchRAGzllZE4jK3bl/qP8B4Kws8zX5ojY9m\nznHczZsrtrHisIsAbOguSaPIKeyAZNchRA
GzllZE4jK3bl/qP8B4Kws8zX5ojY9m\nznHczZsrtrHisIsAbOguSaPIKeyAZNchRAGzllZE4j
K3bl/qP8B4Kws8zX5ojY9m\nXsYg6qyxy4zq70lwWqLwGa==\n-----END PRIVATE
```

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

### Ejemplo 2: Configuración del tipo de servicio CVS base

Este ejemplo muestra una definición de back-end que utiliza el tipo de servicio CVS básico, que está pensada para cargas de trabajo de uso general y ofrece rendimiento ligero/moderado, además de una alta disponibilidad zonal.

```
{
  "version": 1,
  "storageDriverName": "gcp-cvs",
  "projectNumber": "012345678901",
  "storageClass": "software",
  "apiRegion": "us-east4",
  "apiKey": {
    "type": "service_account",
    "project_id": "my-gcp-project",
    "private_key_id": "1234567890123456789012345678901234567890",
    "private_key": "-----BEGIN PRIVATE KEY-----
\nznHczZsrthHisIsAbOguSaPIKeyAZNchRAGzllZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZ
srthHisIsAbOguSaPIKeyAZNchRAGzllZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrthHisI
sAbOguSaPIKeyAZNchRAGzllZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrthHisIsAbOguSa
PIKeyAZNchRAGzllZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrthHisIsAbOguSaPIKeyAZN
chRAGzllZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrthHisIsAbOguSaPIKeyAZNchRAGzll
ZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrthHisIsAbOguSaPIKeyAZNchRAGzllZE4jK3bl
/qp8B4Kws8zX5ojY9m\nznHczZsrthHisIsAbOguSaPIKeyAZNchRAGzllZE4jK3bl/qp8B4K
s8zX5ojY9m\nznHczZsrthHisIsAbOguSaPIKeyAZNchRAGzllZE4jK3bl/qp8B4Kws8zX5ojY
9m\nznHczZsrthHisIsAbOguSaPIKeyAZNchRAGzllZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHc
zZsrthHisIsAbOguSaPIKeyAZNchRAGzllZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrthHi
sIsAbOguSaPIKeyAZNchRAGzllZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrthHisIsAbOgu
SaPIKeyAZNchRAGzllZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrthHisIsAbOguSaPIKeyA
ZNchRAGzllZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrthHisIsAbOguSaPIKeyAZNchRAGz
llZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrthHisIsAbOguSaPIKeyAZNchRAGzllZE4jK3
bl/qp8B4Kws8zX5ojY9m\nznHczZsrthHisIsAbOquSaPIKeyAZNchRAGzllZE4jK3bl/qp8B4
```

```

Kws8zX5ojY9m\nznHczZsrrtHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5o
jY9m\nznHczZsrrtHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m\nzn
HczZsrrtHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrrt
HisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrrtHisIsAbO
guSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrrtHisIsAbOguSaPIKe
yAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrrtHisIsAbOguSaPIKeyAZNchRA
Gz1zZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrrtHisIsAbOguSaPIKeyAZNchRAGz1zZE4j
K3bl/qp8B4Kws8zX5ojY9m\nXsYg6gyxy4zq7OlwWgLwGa==\n-----END PRIVATE
KEY-----\n",
    "client_email": "cloudvolumes-admin-sa@my-gcp-
project.iam.gserviceaccount.com",
    "client_id": "123456789012345678901",
    "auth_uri": "https://accounts.google.com/o/oauth2/auth",
    "token_uri": "https://oauth2.googleapis.com/token",
    "auth_provider_x509_cert_url":
"https://www.googleapis.com/oauth2/v1/certs",
    "client_x509_cert_url":
"https://www.googleapis.com/robot/v1/metadata/x509/cloudvolumes-admin-
sa%40my-gcp-project.iam.gserviceaccount.com"
  }
}

```

### Ejemplo 3: Configuración de un solo nivel de servicio

Este ejemplo muestra un archivo de entorno de administración que aplica los mismos aspectos a todo el almacenamiento creado por Astra Trident en la región Google Cloud US-west2. En este ejemplo también se muestra el uso de proxyURL en el archivo de configuración del back-end.

```

{
  "version": 1,
  "storageDriverName": "gcp-cvs",
  "projectNumber": "012345678901",
  "apiRegion": "us-west2",
  "apiKey": {
    "type": "service_account",
    "project_id": "my-gcp-project",
    "private_key_id": "1234567890123456789012345678901234567890",
    "private_key": "-----BEGIN PRIVATE KEY-----
\nznHczZsrrtHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZ
srrtHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrrtHisI
sAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrrtHisIsAbOguSa
PIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrrtHisIsAbOguSaPIKeyAZN
chRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrrtHisIsAbOguSaPIKeyAZNchRAGz1z
ZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrrtHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl
/qp8B4Kws8zX5ojY9m\nznHczZsrrtHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kw
s8zX5ojY9m\nznHczZsrrtHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY

```



en la `storage` sección. En este ejemplo, cada pool de almacenamiento individual establece su propia `serviceLevel`, y algunos pools sobrescriben los valores predeterminados.

[illegible]



```

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"
    }
  ]
}

```

Las siguientes definiciones de StorageClass se refieren a los pools de almacenamiento anteriores. Mediante el uso de `parameters.selector` Campo, se puede especificar para cada clase de almacenamiento el pool virtual que se usa para alojar un volumen. Los aspectos definidos en el pool elegido serán el volumen.

El primer tipo de almacenamiento (`cvs-extreme-extra-protection`) se asigna al primer grupo de almacenamiento 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 de almacenamiento virtual se selecciona y garantiza que se cumpla el requisito de reserva Snapshot.

```

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

```

```

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

```

## 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

Descubra cómo crear y usar un back-end de Element con su instalación de Astra Trident.

### Lo que necesitará

- 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"](#).

### Lo que usted necesita saber

La `solidfire-san` el controlador de almacenamiento admite ambos modos de volumen: archivo y bloque. 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	No hay sistema de archivos. Dispositivo de bloque RAW.
solidfire-san	ISCSI	Bloque	RWO, ROX, RWX	No hay sistema de archivos. Dispositivo de bloque RAW.
solidfire-san	ISCSI	Sistema de archivos	RWO, ROX	xfs, ext3, ext4
solidfire-san	ISCSI	Sistema de archivos	RWO, ROX	xfs, ext3, ext4



Astra Trident utiliza CHAP cuando funciona como un aprovisionador CSI mejorado. Si está utilizando CHAP (que es el valor predeterminado para CSI), no es necesario realizar ninguna otra preparación. Se recomienda establecer explícitamente el `UseCHAP` Opción para utilizar CHAP con Trident que no sea CSI. De lo contrario, consulte ["aquí"](#).



Los grupos de acceso de volúmenes solo son compatibles con el marco convencional que no es CSI para Astra Trident. Cuando se configura para funcionar en el modo CSI, Astra Trident utiliza CHAP.

Si ninguno `AccessGroups` o `UseCHAP` están definidas, se aplica una de las siguientes reglas:

- Si es el valor predeterminado `trident` se ha detectado el grupo de acceso; se utilizan los grupos de acceso.
- Si no se detecta ningún grupo de acceso y la versión de Kubernetes es 1.7 o posterior, se utiliza CHAP.

## Opciones de configuración del back-end

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

Parámetro	Descripción	Predeterminado
<code>version</code>		Siempre 1
<code>storageDriverName</code>	Nombre del controlador de almacenamiento	Siempre "solidfire-san"
<code>backendName</code>	Nombre personalizado o el back-end de almacenamiento	Dirección IP "SolidFire_" + almacenamiento (iSCSI)
<code>Endpoint</code>	MVIP para el clúster de SolidFire con credenciales de inquilino	
<code>SVIP</code>	La dirección IP y el puerto de almacenamiento (iSCSI)	
<code>labels</code>	Conjunto de etiquetas con formato JSON arbitrario que se aplica en los volúmenes.	""
<code>TenantName</code>	Nombre de inquilino que se va a usar (creado si no se encuentra)	
<code>InitiatorIFace</code>	Restringir el tráfico de iSCSI a una interfaz de host específica	"predeterminado"
<code>UseCHAP</code>	Utilice CHAP para la autenticación de iSCSI	verdadero
<code>AccessGroups</code>	Lista de ID de grupos de acceso que se van a usar	Busca el código de un grupo de acceso denominado "trident".
<code>Types</code>	Especificaciones de calidad de servicio	
<code>limitVolumeSize</code>	Error en el aprovisionamiento si el tamaño del volumen solicitado es superior a este valor	"" (no se aplica de forma predeterminada)
<code>debugTraceFlags</code>	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.



Para todos los volúmenes creados, Astra Trident copiará todas las etiquetas presentes en un pool de almacenamiento a la LUN de almacenamiento de respaldo en el momento en el que se aprovisiona. Los administradores de almacenamiento pueden definir etiquetas por pool de almacenamiento y agrupar todos los volúmenes creados en un pool de almacenamiento. Esto proporciona una forma cómoda de diferenciar los volúmenes basándose en un conjunto de etiquetas personalizables que se proporcionan en la configuración del back-end.

### 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 de almacenamiento virtual

En este ejemplo, se muestra el archivo de definición del back-end configurado con pools de almacenamiento virtual junto con `StorageClasses` que se denominan.

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 de almacenamiento virtual se definen en la `storage` sección. En este ejemplo, algunos del pool de almacenamiento establecen su propio tipo, y algunos pools sobrescriben los valores predeterminados establecidos 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 de almacenamiento virtual 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 de almacenamiento

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 de almacenamiento 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"](#)

## Configuración de un back-end con controladores SAN de ONTAP

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

- ["Preparación"](#)
- ["Configuración y ejemplos"](#)

### 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.

### Preparación

Descubra cómo preparar un back-end de ONTAP con controladores DE SAN de ONTAP. 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 ["aquí"](#) para obtener más detalles.

### Autenticación

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).

Los usuarios también pueden optar por actualizar los back-ends existentes, optar por pasar de basado en credenciales a basado en certificados y viceversa. Si **se proporcionan tanto las credenciales como los certificados**, Astra Trident utilizará por defecto los certificados mientras emite una advertencia para eliminar las credenciales de la definición de backend.

## 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:

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

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=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",
  "dataLIF": "1.2.3.8",
  "svm": "vserver_test",
  "clientCertificate": "Faaaakkkkeeee...Vaaalllluuuuueeee",
  "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, utilice una actualización `backend.json` archivo 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",
  "dataLIF": "1.2.3.8",
  "svm": "vserver_test",
  "username": "vsadmin",
  "password": "secret",
  "storagePrefix": "myPrefix_"
}

#Update backend with tridentctl
$ tridentctl update backend SanBackend -f cert-backend-updated.json -n
trident

+-----+-----+-----+-----+
+-----+-----+
|      NAME      | STORAGE DRIVER |          UUID          |
STATE | VOLUMES |
+-----+-----+-----+-----+
+-----+-----+
| SanBackend | ontap-san      | 586b1cd5-8cf8-428d-a76c-2872713612c1 |
online |          9 |
+-----+-----+-----+-----+
+-----+-----+
```



Cuando 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.

### Especifique iGroups

Astra Trident utiliza iGroups para controlar el acceso a los volúmenes (LUN) que aprovisiona. Los administradores tienen dos opciones cuando se trata de especificar iGroups para los back-ends:

- Astra Trident puede crear y gestionar automáticamente un igroup por back-end. Si `igroupName` No se incluye en la definición de back-end, Astra Trident crea un igroup llamado `trident-<backend-UUID>` En la SVM. De este modo, cada back-end cuenta con un igroup dedicado y manejar la adición/eliminación automatizada de IQN de nodos de Kubernetes.
- De forma alternativa, los iGroups creados previamente también se pueden proporcionar en una definición

de back-end. Esto se puede hacer usando `igroupName` parámetro config. Astra Trident añadirá/eliminará IQN de nodos de Kubernetes al `igroup` preexistente.

Para los back-ends que tengan `igroupName` definida, el `igroupName` se puede eliminar con un `tridentctl backend update`. Para tener `iGroups` de gestión automática Astra Trident. Esto no interrumpirá el acceso a volúmenes que ya están conectados a las cargas de trabajo. Futuras conexiones se gestionarán con el `igroup` Astra Trident creado.



Dedicar un `igroup` para cada instancia única de Astra Trident es una práctica recomendada que beneficia al administrador de Kubernetes y al administrador de almacenamiento. CSI Trident automatiza la adición y la eliminación de IQN de nodos de clúster al `igroup`, por lo que simplifica en gran medida su gestión. Cuando se utiliza la misma SVM en entornos de Kubernetes (y instalaciones de Astra Trident), el uso de un `igroup` dedicado garantiza que los cambios realizados en un clúster de Kubernetes no afecten a los `iGroups` asociados a otro. Además, también es importante garantizar que cada nodo del clúster de Kubernetes tenga un IQN único. Como se ha mencionado anteriormente, Astra Trident se encarga automáticamente de la adición y eliminación de IQN. La reutilización de IQN entre hosts puede provocar situaciones no deseadas en las que los hosts se confunden entre sí y se deniega el acceso a las LUN.

Si Astra Trident está configurada para que funcione como un aprovisionador de nodos CSI, los IQN de nodos de Kubernetes se añaden o eliminan automáticamente del `igroup`. Cuando se añaden nodos a un clúster de Kubernetes, `trident-csi` DemonSet despliega un pod (`trident-csi-xxxxx`) en los nodos recién añadidos y registra los nuevos nodos a los que puede asociar volúmenes. Los IQN de nodos también se agregan al `igroup` del backend. Un conjunto de pasos similares tratan de la eliminación de IQN cuando se acortan, drenan y se eliminan nodos de Kubernetes.

Si Astra Trident no se ejecuta como un aprovisionador CSI, el `igroup` se debe actualizar manualmente para contener los IQN iSCSI de cada nodo de trabajo del clúster de Kubernetes. Se deberán añadir al `igroup` varios IQN de nodos que se unen al clúster de Kubernetes. De igual manera, los IQN de nodos que se quitan del clúster de Kubernetes se deben quitar del `igroup`.

#### **Auténtica 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 predeterminada de la SVM en CHAP bidireccional y establece el nombre de usuario y los secretos del archivo de entorno de administración. 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": "FaKePaSsWoRd",
  "igroupName": "trident",
  "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 del iniciador predeterminado de la SVM es `none` (establecido de forma predeterminada) y no hay LUN preexistentes 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 esta forma se garantiza que el acceso a las LUN que ya están presentes en la SVM no esté restringido.
- 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).
- Gestionar la adición de iniciadores a la `igroupName` dado en el backend. Si no se especifica, el valor predeterminado es `trident`.

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": "FaKePaSsWoRd",
  "igroupName": "trident",
  "chapInitiatorSecret": "cl9qxUpDaTeD",
  "chapTargetInitiatorSecret": "rqxigXgkeUpDaTeD",
  "chapTargetUsername": "iJF4heBRT0TCwxyz",
  "chapUsername": "uh2aNCLSD6cNwxyz",
}
```

```
$ ./tridentctl update backend ontap_san_chap -f backend-san.json -n
trident
```

NAME	STORAGE DRIVER	UUID
ontap_san_chap	ontap-san	aa458f3b-ad2d-4378-8a33-1a472ffbeb5c

```
tridentctl update backend ontap_san_chap -f backend-san.json -n
trident
```

NAME	STORAGE DRIVER	UUID
ontap_san_chap	ontap-san	aa458f3b-ad2d-4378-8a33-1a472ffbeb5c

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 de configuración y ejemplos

Descubra cómo crear y usar controladores SAN de ONTAP con su instalación de Astra Trident. En esta sección, se ofrecen ejemplos de configuración del back-end y detalles sobre cómo asignar 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
storageDriverName	Nombre del controlador de almacenamiento	"ontap-nas", "ontap-nas-economy", "ontap-nas-flexgroup", "ontap-san" y "ontap-san-economy"
backendName	Nombre personalizado o el back-end de almacenamiento	Nombre del conductor + "_" + dataLIF
managementLIF	La dirección IP de una LIF de gestión de clústeres o SVM	"10.0.0.1", "[2001:1234:abcd::fefe]"
dataLIF	Dirección IP de LIF de protocolo. Use corchetes para IPv6. No se puede actualizar después de configurarlo	Derivado de la SVM a menos que se especifique
useCHAP	Usar CHAP para la autenticación de iSCSI para los controladores SAN de ONTAP [booleano]	falso
chapInitiatorSecret	Secreto CHAP del iniciador. Obligatorio si useCHAP=true	""
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	""

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	""
svm	Máquina virtual de almacenamiento que usar	Derivado si una SVM managementLIF está especificado
igroupName	Nombre del igroup para volúmenes DE SAN que usar	"Trident-<backend-UUID>"
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.	"" (no se aplica de forma predeterminada)
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}	nulo
useREST	Parámetro booleano para usar las API DE REST de ONTAP. <b>Vista previa técnica</b>	falso



useREST se proporciona como **avance técnico** que se recomienda para entornos de prueba y no para cargas de trabajo de producción. Cuando se establece en `true`, Astra Trident utilizará las API DE REST de ONTAP para comunicarse con el back-end. Esta función requiere ONTAP 9.9 o posterior. Además, el rol de inicio de sesión de ONTAP utilizado debe tener acceso a `ontap` cliente más. Esto está satisfecho por el predefinido `vsadmin` y `cluster-admin` funciones.

Para comunicarse con el clúster ONTAP, debe proporcionar los parámetros de autenticación. Puede ser el nombre de usuario o la contraseña de un inicio de sesión de seguridad o un certificado instalado.



Si utiliza un entorno de administración de Amazon FSX para ONTAP de NetApp, no especifique el `limitAggregateUsage` parámetro. La `fsxadmin` y `vsadmin` Las funciones que ofrece Amazon FSX para ONTAP de NetApp no incluyen los permisos de acceso necesarios para recuperar el uso de agregados y limitarla a través de Astra Trident.



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

Para la `ontap-san` Controladores, el valor predeterminado es utilizar todas las IP de LIF de datos de la SVM y para utilizar la multivía iSCSI. Especificar una dirección IP para la LIF de datos del `ontap-san` los controladores les obligan a deshabilitar la multivía y a usar solo la dirección especificada.



Al crear un back-end, recuerde eso `dataLIF` y `storagePrefix` no se puede modificar una vez creada. Para actualizar estos parámetros, deberá crear un nuevo backend.

`igroupName` Puede establecerse en un `igroup` que ya se creó en el clúster de ONTAP. Si no se especifica, Astra Trident crea automáticamente un `igroup` llamado `Trident-<backend-UUID>`. Si proporciona un nombre de canal medio predefinido, NetApp recomienda usar un `igroup` por clúster de Kubernetes, si la SVM se va a compartir entre entornos. Esto es necesario para que Astra Trident mantenga automáticamente las adiciones y eliminaciones por IQN.

Los back-ends también pueden tener `iGroups` actualizados después de la creación:

- Se puede actualizar el nombre de `lfe` para que apunte a un nuevo `igroup` que se crea y gestiona en la SVM fuera de Astra Trident.
- Se puede omitir el nombre de la pila. En este caso, Astra Trident creará y gestionará automáticamente un `igroup` `trident-<backend-UUID>`.

En ambos casos, los archivos adjuntos de volumen seguirán siendo accesibles. Los futuros archivos adjuntos de volumen utilizarán el `igroup` actualizado. Esta actualización no interrumpe el acceso a los volúmenes presentes en el back-end.

Se puede especificar un nombre de dominio completo (FQDN) para el `managementLIF` opción.

``managementLIF`` Para todos los controladores ONTAP también se puede establecer en direcciones IPv6. Asegúrese de instalar Trident con el ``--use-ipv6`` bandera. Hay que tener cuidado para definir ``managementLIF`` La dirección IPv6 entre corchetes.



Cuando se usen direcciones IPv6, asegúrese de `managementLIF` y `dataLIF` (si se incluye en su definición de backend) se definen entre corchetes, como `[28e8:d9fb:a825:b7bf:69a8:d02f:9e7b:3555]`. Si `dataLIF` No se proporciona; Astra Trident recuperará las LIF de datos IPv6 desde la SVM.

Para habilitar los controladores `ontap-san` para que usen CHAP, configure el `useCHAP` parámetro a `true` en su definición de backend. A continuación, Astra Trident configurará y utilizará CHAP bidireccional como la

autenticación predeterminada para la SVM proporcionada en el back-end. Consulte ["aquí"](#) para descubrir cómo funciona.

Para la `ontap-san-economy` controlador, el `limitVolumeSize` Opción también restringirá el tamaño máximo de los volúmenes que gestiona para `qtrees` y `LUN`.



Astra Trident establece etiquetas de aprovisionamiento en el campo "Comentarios" de todos los volúmenes creados mediante `ontap-san` controlador. Para cada volumen creado, el campo "Comentarios" del FlexVol se rellenará con todas las etiquetas presentes en el pool de almacenamiento en el que se haya colocado. Los administradores de almacenamiento pueden definir etiquetas por pool de almacenamiento y agrupar todos los volúmenes creados en un pool de almacenamiento. Esto proporciona una forma cómoda de diferenciar los volúmenes basándose en un conjunto de etiquetas personalizables que se proporcionan en la configuración del back-end.

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

Puede controlar cómo se aprovisiona cada volumen de forma predeterminada mediante estas opciones de una sección especial 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; "none" (thin) o "VOLUME" (grueso)	"ninguna"
<code>snapshotPolicy</code>	Política de Snapshot que se debe usar	"ninguna"
<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	""
<code>snapshotReserve</code>	Porcentaje del volumen reservado para instantáneas "0"	Si <code>snapshotPolicy</code> no es "ninguno", sino ""
<code>splitOnClone</code>	Divida un clon de su elemento principal al crearlo	"falso"
<code>splitOnClone</code>	Divida un clon de su elemento principal al crearlo	"falso"
<code>encryption</code>	Habilite el cifrado de volúmenes de NetApp	"falso"

Parámetro	Descripción	Predeterminado
securityStyle	Estilo de seguridad para nuevos volúmenes	"unix"
tieringPolicy	Política de organización en niveles para usar "ninguno"	"Solo Snapshot" para configuración previa a ONTAP 9.5 SVM-DR



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.

A continuación se muestra un ejemplo con valores predeterminados definidos:

```
{
  "version": 1,
  "storageDriverName": "ontap-san",
  "managementLIF": "10.0.0.1",
  "dataLIF": "10.0.0.2",
  "svm": "trident_svm",
  "username": "admin",
  "password": "password",
  "labels": {"k8scluster": "dev2", "backend": "dev2-sanbackend"},
  "storagePrefix": "alternate-trident",
  "igroupName": "custom",
  "debugTraceFlags": {"api":false, "method":true},
  "defaults": {
    "spaceReserve": "volume",
    "qosPolicy": "standard",
    "spaceAllocation": "false",
    "snapshotPolicy": "default",
    "snapshotReserve": "10"
  }
}
```



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:

```
Total volume size = [(PVC requested size) / (1 - (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 se utiliza Amazon FSX en ONTAP de NetApp con Astra Trident, se recomienda especificar los nombres DNS para las LIF en lugar de las direcciones IP.

#### ontap-san **controlador con 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,
  "storageDriverName": "ontap-san",
  "backendName": "DefaultSANBackend",
  "managementLIF": "10.0.0.1",
  "dataLIF": "10.0.0.3",
  "svm": "svm_iscsi",
  "useCHAP": true,
  "chapInitiatorSecret": "cl9qxIm36DKyawxy",
  "chapTargetInitiatorSecret": "rqxigXgkesIpwxyz",
  "chapTargetUsername": "iJF4heBRT0TCwxyz",
  "chapUsername": "uh2aNCLSD6cNwxyz",
  "igroupName": "trident",
  "clientCertificate": "ZXR0ZXJwYXB...ICMgJ3BhcGVyc2",
  "clientPrivateKey": "vciwKIyAgZG...0cnksIGRlc2NyaX",
  "trustedCACertificate": "zcyBbaG...b3Igb3duIGNsYXNz"
}
```

#### ontap-san **Controlador con CHAP bidireccional**

Este es un ejemplo de configuración de backend mínima. Esta configuración básica crea un ontap-san backend con useCHAP establezca en true.

```
{
  "version": 1,
  "storageDriverName": "ontap-san",
  "managementLIF": "10.0.0.1",
  "dataLIF": "10.0.0.3",
  "svm": "svm_iscsi",
  "labels": {"k8scluster": "test-cluster-1", "backend": "testcluster1-sanbackend"},
  "useCHAP": true,
  "chapInitiatorSecret": "cl9qxIm36DKyawxy",
  "chapTargetInitiatorSecret": "rqxigXgkesIpwxyz",
  "chapTargetUsername": "iJF4heBRT0TCwxyz",
  "chapUsername": "uh2aNCLSD6cNwxyz",
  "igroupName": "trident",
  "username": "vsadmin",
  "password": "secret"
}
```

#### ontap-san-economy **controlador**



```
{
  "version": 1,
  "storageDriverName": "ontap-san-economy",
  "managementLIF": "10.0.0.1",
  "svm": "svm_iscsi_eco",
  "useCHAP": true,
  "chapInitiatorSecret": "cl9qxIm36DKyawxy",
  "chapTargetInitiatorSecret": "rqxigXgkesIpwxyz",
  "chapTargetUsername": "iJF4heBRT0TCwxyz",
  "chapUsername": "uh2aNCLSD6cNwxyz",
  "igroupName": "trident",
  "username": "vsadmin",
  "password": "secret"
}
```

### Ejemplos de back-ends con pools de almacenamiento virtuales

En el archivo de definición del back-end de ejemplo que se muestra a continuación, se establecen valores predeterminados específicos para todos los grupos de almacenamiento, como `spaceReserve` en ninguno, `spaceAllocation` en falso, y `encryption` en falso. Los pools de almacenamiento virtual se definen en la sección de almacenamiento.

En este ejemplo, algunos de los recursos compartidos de almacenamiento son los suyos propios `spaceReserve`, `spaceAllocation`, y `encryption` los valores y algunos pools sobrescriben los valores predeterminados establecidos anteriormente.

```
{
  "version": 1,
  "storageDriverName": "ontap-san",
  "managementLIF": "10.0.0.1",
  "dataLIF": "10.0.0.3",
  "svm": "svm_iscsi",
  "useCHAP": true,
  "chapInitiatorSecret": "cl9qxIm36DKyawxy",
  "chapTargetInitiatorSecret": "rqxigXgkesIpwxyz",
  "chapTargetUsername": "iJF4heBRT0TCwxyz",
  "chapUsername": "uh2aNCLSD6cNwxyz",
  "igroupName": "trident",
  "username": "vsadmin",
  "password": "secret",

  "defaults": {
    "spaceAllocation": "false",
    "encryption": "false",
    "qosPolicy": "standard"
  },
}
```

```

    "labels":{"store": "san_store", "kubernetes-cluster": "prod-cluster-1"},
    "region": "us_east_1",
    "storage": [
      {
        "labels":{"protection":"gold", "creditpoints":"40000"},
        "zone":"us_east_1a",
        "defaults": {
          "spaceAllocation": "true",
          "encryption": "true",
          "adaptiveQosPolicy": "adaptive-extreme"
        }
      },
      {
        "labels":{"protection":"silver", "creditpoints":"20000"},
        "zone":"us_east_1b",
        "defaults": {
          "spaceAllocation": "false",
          "encryption": "true",
          "qosPolicy": "premium"
        }
      },
      {
        "labels":{"protection":"bronze", "creditpoints":"5000"},
        "zone":"us_east_1c",
        "defaults": {
          "spaceAllocation": "true",
          "encryption": "false"
        }
      }
    ]
  }
}

```

A continuación, se muestra un ejemplo de iSCSI para el `ontap-san-economy` controlador:

```

{
  "version": 1,
  "storageDriverName": "ontap-san-economy",
  "managementLIF": "10.0.0.1",
  "svm": "svm_iscsi_eco",
  "useCHAP": true,
  "chapInitiatorSecret": "cl9qxIm36DKyawxy",
  "chapTargetInitiatorSecret": "rqxigXgkesIpwxyz",
  "chapTargetUsername": "iJF4heBRT0TCwxyz",
  "chapUsername": "uh2aNCLSD6cNwxyz",

```

```

"igroupName": "trident",
"username": "vsadmin",
"password": "secret",

"defaults": {
    "spaceAllocation": "false",
    "encryption": "false"
},
"labels":{"store":"san_economy_store"},
"region": "us_east_1",
"storage": [
    {
        "labels":{"app":"oracledb", "cost":"30"},
        "zone":"us_east_1a",
        "defaults": {
            "spaceAllocation": "true",
            "encryption": "true"
        }
    },
    {
        "labels":{"app":"postgresdb", "cost":"20"},
        "zone":"us_east_1b",
        "defaults": {
            "spaceAllocation": "false",
            "encryption": "true"
        }
    },
    {
        "labels":{"app":"mysqldb", "cost":"10"},
        "zone":"us_east_1c",
        "defaults": {
            "spaceAllocation": "true",
            "encryption": "false"
        }
    }
]
}

```

### Asigne los back-ends a StorageClass

Las siguientes definiciones de StorageClass se refieren a los pools de almacenamiento virtual 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 (`protection-gold`) se asignará al primer, segundo grupo de almacenamiento virtual del `ontap-nas-flexgroup` back-end y el primer pool de almacenamiento virtual del `ontap-san` back-end. Se trata de la única piscina que ofrece protección de nivel Gold.

- El segundo tipo de almacenamiento (`protection-not-gold`) se asignará al tercer y cuarto bloque de almacenamiento virtual en `ontap-nas-flexgroup` back-end y el segundo, tercer pool de almacenamiento virtual del `ontap-san` back-end. Estos son los únicos pools que ofrecen un nivel de protección distinto al Gold.
- El tercer tipo de almacenamiento (`app-mysqldb`) se asignará al cuarto bloque de almacenamiento virtual en `ontap-nas` back-end y el tercer pool de almacenamiento virtual de `ontap-san-economy` back-end. Estos son los únicos grupos que ofrecen la configuración del pool de almacenamiento para la aplicación de tipo `mysqldb`.
- El cuarto tipo de almacenamiento (`protection-silver-creditpoints-20k`) se asignará al tercer grupo de almacenamiento virtual en `ontap-nas-flexgroup` back-end y el segundo pool de almacenamiento virtual de `ontap-san` back-end. Estas son las únicas piscinas que ofrecen protección de nivel Gold con 20000 puntos de crédito.
- El quinto tipo de almacenamiento (`creditpoints-5k`) se asignará al segundo grupo de almacenamiento virtual en `ontap-nas-economy` back-end y el tercer pool de almacenamiento virtual de `ontap-san` back-end. Se trata de la única oferta de pool en 5000 puntos de crédito.

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

```

apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: protection-gold
provisioner: netapp.io/trident
parameters:
  selector: "protection=gold"
  fsType: "ext4"
---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: protection-not-gold
provisioner: netapp.io/trident
parameters:
  selector: "protection!=gold"
  fsType: "ext4"
---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: app-mysqldb
provisioner: netapp.io/trident
parameters:
  selector: "app=mysqldb"
  fsType: "ext4"
---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: protection-silver-creditpoints-20k
provisioner: netapp.io/trident
parameters:
  selector: "protection=silver; creditpoints=20000"
  fsType: "ext4"
---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: creditpoints-5k
provisioner: netapp.io/trident
parameters:
  selector: "creditpoints=5000"
  fsType: "ext4"

```

## Configurar un back-end con controladores NAS de ONTAP

Obtenga información acerca de la configuración de un back-end de ONTAP con controladores NAS de ONTAP o Cloud Volumes ONTAP.

- ["Preparación"](#)
- ["Configuración y ejemplos"](#)

### 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.

### Preparación

Descubra cómo preparar un back-end de ONTAP con controladores NAS de ONTAP. Para todos los back-ends de ONTAP, Astra Trident requiere al menos un agregado asignado a la SVM.

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 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.

### Autenticación

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).

Los usuarios también pueden optar por actualizar los back-ends existentes, optar por pasar de basado en credenciales a basado en certificados y viceversa. Si **se proporcionan tanto las credenciales como los certificados**, Astra Trident utilizará por defecto los certificados mientras emite una advertencia para eliminar las credenciales de la definición de backend.

## 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:

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

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

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

+-----+-----+-----+
+-----+-----+
|      NAME      | STORAGE DRIVER |                      UUID                      |
STATE | VOLUMES |
+-----+-----+-----+
+-----+-----+
| 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, utilice una actualización `backend.json` archivo que contiene los parámetros necesarios para ejecutarse `tridentctl backend update`.

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

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

+-----+-----+-----+-----+
+-----+-----+
|      NAME      | STORAGE DRIVER |                      UUID                      |
STATE | VOLUMES |
+-----+-----+-----+-----+
+-----+-----+
| NasBackend | ontap-nas      | 98e19b74-aec7-4a3d-8dcf-128e5033b214 |
online |          9 |
+-----+-----+-----+-----+
+-----+-----+
```



Cuando 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

La versión 20.04 de CSI Trident ofrece la capacidad de gestionar dinámicamente 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 que tienen IP en el rango especificado, por lo que admite una gestión automatizada y finegrada.



La gestión dinámica de las políticas de exportación sólo está disponible para CSI Trident. Es importante asegurarse de que los nodos de trabajo no estén siendo atados.

## Ejemplo

Hay dos opciones de configuración que deben utilizarse. A continuación se muestra 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": "FaKePaSsWoRd",
  "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 predefinida con una regla de exportación que permite al bloque CIDR de nodo (como la política de exportación predeterminada). Siga siempre la mejor práctica recomendada por 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 denominada después de que el UUID del back-end y los volúmenes presentes en el back-end utilicen 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.

## Opciones de configuración y ejemplos

Obtenga más información sobre cómo crear y utilizar controladores NAS de ONTAP con su instalación de Astra Trident. En esta sección, se ofrecen ejemplos de configuración del back-end y detalles sobre cómo asignar 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
<code>version</code>		Siempre 1
<code>storageDriverName</code>	Nombre del controlador de almacenamiento	"ontap-nas", "ontap-nas-economy", "ontap-nas-flexgroup", "ontap-san" y "ontap-san-economy"
<code>backendName</code>	Nombre personalizado o el back-end de almacenamiento	Nombre del conductor + " _ " + <code>dataLIF</code>
<code>managementLIF</code>	La dirección IP de una LIF de gestión de clústeres o SVM	"10.0.0.1", "[2001:1234:abcd::fefe]"
<code>dataLIF</code>	Dirección IP de LIF de protocolo. Use corchetes para IPv6. No se puede actualizar después de configurarlo	Derivado de la SVM a menos que se especifique
<code>autoExportPolicy</code>	Habilitar la creación y actualización automática de la política de exportación [booleano]	falso
<code>autoExportCIDRs</code>	Lista de CIDR para filtrar las IP de nodo de Kubernetes contra cuándo <code>autoExportPolicy</code> está habilitado	["0.0.0.0/0", ":::/0"]
<code>labels</code>	Conjunto de etiquetas con formato JSON arbitrario que se aplica en los volúmenes	""

Parámetro	Descripción	Predeterminado
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	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	
svm	Máquina virtual de almacenamiento que usar	Derivado si una SVM managementLIF está especificado
igroupName	Nombre del igroup para volúmenes DE SAN que usar	"Trident-<backend-UUID>"
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.	"" (no se aplica de forma predeterminada)
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}	nulo
nfsMountOptions	Lista de opciones de montaje NFS separadas por comas	""

Parámetro	Descripción	Predeterminado
qtreesPerFlexvol	El número máximo de qtrees por FlexVol debe estar comprendido entre [50, 300]	"200"
useREST	Parámetro booleano para usar las API DE REST de ONTAP. <b>Vista previa técnica</b>	falso



useREST se proporciona como **avance técnico** que se recomienda para entornos de prueba y no para cargas de trabajo de producción. Cuando se establece en `true`, Astra Trident utilizará las API DE REST de ONTAP para comunicarse con el back-end. Esta función requiere ONTAP 9.9 o posterior. Además, el rol de inicio de sesión de ONTAP utilizado debe tener acceso a `ontap` cliente más. Esto está satisfecho por el predefinido `vsadmin` y `cluster-admin` funciones.

Para comunicarse con el clúster ONTAP, debe proporcionar los parámetros de autenticación. Puede ser el nombre de usuario o la contraseña de un inicio de sesión de seguridad o un certificado instalado.



Si utiliza un entorno de administración de Amazon FSX para ONTAP de NetApp, no especifique el `limitAggregateUsage` parámetro. La `fsxadmin` y `vsadmin` Las funciones que ofrece Amazon FSX para ONTAP de NetApp no incluyen los permisos de acceso necesarios para recuperar el uso de agregados y limitarla a través de Astra Trident.



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



Al crear un back-end, recuerde que `dataLIF` y `storagePrefix` no se puede modificar una vez creada. Para actualizar estos parámetros, deberá crear un nuevo backend.

Se puede especificar un nombre de dominio completo (FQDN) para el `managementLIF` opción. También se puede especificar un FQDN para el `dataLIF` Opción, en cuyo caso, se utilizará el FQDN para las operaciones de montaje de NFS. De esta forma puede crear un DNS round-robin para lograr un equilibrio entre la carga en múltiples LIF de datos.

``managementLIF`` Para todos los controladores ONTAP también se puede establecer en direcciones IPv6. Asegúrese de instalar Astra Trident con el ``--use-ipv6`` bandera. Hay que tener cuidado para definir el ``managementLIF`` La dirección IPv6 entre corchetes.



Cuando se usen direcciones IPv6, asegúrese de `managementLIF` y `dataLIF` (si se incluye en su definición de backend) se definen entre corchetes, como `[28e8:d9fb:a825:b7bf:69a8:d02f:9e7b:3555]`. Si `dataLIF` No se proporciona; Astra Trident recuperará las LIF de datos IPv6 desde la SVM.

Con el `autoExportPolicy` y `autoExportCIDRs` Opciones, CSI Trident puede gestionar automáticamente las políticas de exportación. Esto es compatible con todos los controladores `ontap-nas-*`.

Para la `ontap-nas-economy` controlador, el `limitVolumeSize` Esta opción también restringirá el tamaño máximo de los volúmenes que gestiona para `qtrees` y `LUN`, y el `qtreesPerFlexvol` Permite personalizar el número máximo de `qtrees` por `FlexVol`.

La `nfsMountOptions` el parámetro puede utilizarse para especificar opciones de montaje. Normalmente, las opciones de montaje para los volúmenes persistentes de Kubernetes se especifican 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. Si no se especifican opciones de montaje en la clase de almacenamiento o el archivo de configuración, Astra Trident no establecerá ninguna opción de montaje en un volumen persistente asociado.



Astra Trident establece etiquetas de aprovisionamiento en el campo "Comentarios" de todos los volúmenes creados mediante `ontap-nas` y `ontap-nas-flexgroup`. Según el controlador utilizado, los comentarios se establecen en `FlexVol` (`ontap-nas`) O `FlexGroup` (`ontap-nas-flexgroup`). Astra Trident copiará todas las etiquetas presentes en un pool de almacenamiento al volumen de almacenamiento en el momento en que se aprovisiona. Los administradores de almacenamiento pueden definir etiquetas por pool de almacenamiento y agrupar todos los volúmenes creados en un pool de almacenamiento. Esto proporciona una forma cómoda de diferenciar los volúmenes basándose en un conjunto de etiquetas personalizables que se proporcionan en la configuración del back-end.

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

Puede controlar cómo se aprovisiona cada volumen de forma predeterminada mediante estas opciones de una sección especial 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; "none" (thin) o "VOLUME" (grueso)	"ninguna"
<code>snapshotPolicy</code>	Política de Snapshot que se debe usar	"ninguna"
<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 <code>ontap-nas-Economy</code> .	""
<code>snapshotReserve</code>	Porcentaje del volumen reservado para instantáneas "0"	Si <code>snapshotPolicy</code> no es "ninguno", sino ""



Parámetro	Descripción	Predeterminado
splitOnClone	Divida un clon de su elemento principal al crearlo	"falso"
encryption	Habilite el cifrado de volúmenes de NetApp	"falso"
securityStyle	Estilo de seguridad para nuevos volúmenes	"unix"
tieringPolicy	Política de organización en niveles para usar "ninguno"	"Solo Snapshot" para configuración previa a ONTAP 9.5 SVM-DR
Permisos univalados	Modo para volúmenes nuevos	"777"
Copias Snapshot Dir	Controla la visibilidad de .snapshot directorio	"falso"
Política de exportoPolicy	Política de exportación que se va a utilizar	"predeterminado"
SecurityStyle	Estilo de seguridad para nuevos volúmenes	"unix"



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.

A continuación se muestra un ejemplo con 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:

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

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.

#### ontap-nas **controlador con 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_"
}
```

#### ontap-nas **controlador con 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": "secret",
  "nfsMountOptions": "nfsvers=4",
}
```

ontap-nas-flexgroup **controlador**

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

ontap-nas **Controlador con 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": "netapp123"
}
```

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

### Ejemplos de back-ends con pools de almacenamiento virtuales

En el archivo de definición del back-end de ejemplo que se muestra a continuación, se establecen valores predeterminados específicos para todos los grupos de almacenamiento, como `spaceReserve` en ninguno, `spaceAllocation` en falso, y `encryption` en falso. Los pools de almacenamiento virtual se definen en la sección de almacenamiento.

En este ejemplo, algunos de los recursos compartidos de almacenamiento son los suyos propios `spaceReserve`, `spaceAllocation`, y `encryption` los valores y algunos pools sobrescriben los valores predeterminados establecidos anteriormente.

ontap-nas **controlador**

```
{
  {
    "version": 1,
    "storageDriverName": "ontap-nas",
    "managementLIF": "10.0.0.1",
    "dataLIF": "10.0.0.2",
    "svm": "svm_nfs",
    "username": "admin",
    "password": "secret",
    "nfsMountOptions": "nfsvers=4",

    "defaults": {
      "spaceReserve": "none",
      "encryption": "false",
      "qosPolicy": "standard"
    },
    "labels": {"store": "nas_store", "k8scluster": "prod-cluster-1"},
    "region": "us_east_1",
    "storage": [
      {
        "labels": {"app": "msoffice", "cost": "100"},

```

```

    "zone": "us_east_1a",
    "defaults": {
      "spaceReserve": "volume",
      "encryption": "true",
      "unixPermissions": "0755",
      "adaptiveQosPolicy": "adaptive-premium"
    }
  },
  {
    "labels": {"app": "slack", "cost": "75"},
    "zone": "us_east_1b",
    "defaults": {
      "spaceReserve": "none",
      "encryption": "true",
      "unixPermissions": "0755"
    }
  },
  {
    "labels": {"app": "wordpress", "cost": "50"},
    "zone": "us_east_1c",
    "defaults": {
      "spaceReserve": "none",
      "encryption": "true",
      "unixPermissions": "0775"
    }
  },
  {
    "labels": {"app": "mysqldb", "cost": "25"},
    "zone": "us_east_1d",
    "defaults": {
      "spaceReserve": "volume",
      "encryption": "false",
      "unixPermissions": "0775"
    }
  }
]
}

```

ontap-nas-flexgroup **controlador**

```

{
  "version": 1,
  "storageDriverName": "ontap-nas-flexgroup",
  "managementLIF": "10.0.0.1",
  "dataLIF": "10.0.0.2",

```

```

"svm": "svm_nfs",
"username": "vsadmin",
"password": "secret",

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

```

```

    }
  ]
}

```

ontap-nas-economy **controlador**

```

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

  "defaults": {
    "spaceReserve": "none",
    "encryption": "false"
  },
  "labels": {"store": "nas_economy_store"},
  "region": "us_east_1",
  "storage": [
    {
      "labels": {"department": "finance", "creditpoints": "6000"},
      "zone": "us_east_1a",
      "defaults": {
        "spaceReserve": "volume",
        "encryption": "true",
        "unixPermissions": "0755"
      }
    },
    {
      "labels": {"department": "legal", "creditpoints": "5000"},
      "zone": "us_east_1b",
      "defaults": {
        "spaceReserve": "none",
        "encryption": "true",
        "unixPermissions": "0755"
      }
    },
    {
      "labels": {"department": "engineering", "creditpoints": "3000"},
      "zone": "us_east_1c",
      "defaults": {
        "spaceReserve": "none",

```



```

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

```

### Asigne los back-ends a StorageClass

Las siguientes definiciones de StorageClass se refieren a los pools de almacenamiento virtual 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 (`protection-gold`) se asignará al primer, segundo grupo de almacenamiento virtual del `ontap-nas-flexgroup` back-end y el primer pool de almacenamiento virtual del `ontap-san` back-end. Se trata de la única piscina que ofrece protección de nivel Gold.
- El segundo tipo de almacenamiento (`protection-not-gold`) se asignará al tercer y cuarto bloque de almacenamiento virtual en `ontap-nas-flexgroup` back-end y el segundo, tercer pool de almacenamiento virtual del `ontap-san` back-end. Estos son los únicos pools que ofrecen un nivel de protección distinto al Gold.
- El tercer tipo de almacenamiento (`app-mysqldb`) se asignará al cuarto bloque de almacenamiento virtual en `ontap-nas` back-end y el tercer pool de almacenamiento virtual de `ontap-san-economy` back-end. Estos son los únicos grupos que ofrecen la configuración del pool de almacenamiento para la aplicación de tipo `mysqldb`.
- El cuarto tipo de almacenamiento (`protection-silver-creditpoints-20k`) se asignará al tercer grupo de almacenamiento virtual en `ontap-nas-flexgroup` back-end y el segundo pool de almacenamiento virtual de `ontap-san` back-end. Estas son las únicas piscinas que ofrecen protección de nivel Gold con 20000 puntos de crédito.
- El quinto tipo de almacenamiento (`creditpoints-5k`) se asignará al segundo grupo de almacenamiento virtual en `ontap-nas-economy` back-end y el tercer pool de almacenamiento virtual de `ontap-san` back-end. Se trata de la única oferta de pool en 5000 puntos de crédito.

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

```

apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: protection-gold
provisioner: netapp.io/trident
parameters:
  selector: "protection=gold"
  fsType: "ext4"
---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: protection-not-gold
provisioner: netapp.io/trident
parameters:
  selector: "protection!=gold"
  fsType: "ext4"
---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: app-mysqldb
provisioner: netapp.io/trident
parameters:
  selector: "app=mysqldb"
  fsType: "ext4"
---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: protection-silver-creditpoints-20k
provisioner: netapp.io/trident
parameters:
  selector: "protection=silver; creditpoints=20000"
  fsType: "ext4"
---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: creditpoints-5k
provisioner: netapp.io/trident
parameters:
  selector: "creditpoints=5000"
  fsType: "ext4"

```

## 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 lanzar y ejecutar sistemas de archivos con el sistema operativo de almacenamiento ONTAP de NetApp. Amazon FSX para ONTAP de NetApp le permite aprovechar las funciones, el rendimiento y las funcionalidades administrativas de NetApp con las que está familiarizado, a la vez que aprovecha la simplicidad, la agilidad, la seguridad y la escalabilidad de almacenar datos en AWS. FSX es compatible con muchas de las funciones del sistema de archivos y API de administración de ONTAP.

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 sus clústeres de Kubernetes que se ejecutan en Amazon Elastic Kubernetes Service (EKS) pueden aprovisionar volúmenes persistentes de bloques y archivos respaldados por ONTAP.

### Obtenga más información sobre Astra Trident

Si es nuevo en Astra Trident, familiarícese con los siguientes enlaces:

- ["Preguntas frecuentes"](#)
- ["Requisitos para usar Astra Trident"](#)
- ["Ponga en marcha Astra Trident"](#)
- ["Prácticas recomendadas para configurar ONTAP, Cloud Volumes ONTAP y Amazon FSX para ONTAP de NetApp"](#)
- ["Integre Astra Trident"](#)
- ["Configuración de entorno de administración DE SAN ONTAP"](#)
- ["Configuración de back-end NAS de ONTAP"](#)

Obtenga más información sobre las capacidades del controlador ["aquí"](#).

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.

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. No se recomienda utilizar el `fsxadmin` Usuario, con Trident, como `vsadmin` El usuario de SVM tiene acceso a más funcionalidades de Astra Trident.

### De Windows

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.

## 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.



Le recomendamos encarecidamente que utilice `vsadmin` usuario en lugar de `fsxadmin` para configurar el back-end. Astra Trident se comunicará con el sistema de archivos FSX mediante este nombre de usuario y contraseña.

Para obtener más información acerca de la autenticación, consulte estos enlaces:

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

## Ponga en marcha y configure Astra Trident en EKS con Amazon FSX para ONTAP de NetApp

### Lo que necesitará

- Un clúster de Amazon EKS existente o un clúster de Kubernetes autogestionado con `kubectl` instalado.
- Un Amazon FSX existente para el sistema de archivos ONTAP de NetApp y una máquina virtual de almacenamiento (SVM) accesible desde los nodos de trabajo del clúster.
- Nodos de trabajo preparados para ["NFS y/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.

Para ver otros requisitos de Astra Trident, consulte ["aquí"](#).

### Pasos

1. Implemente Astra Trident utilizando uno de los métodos de implementación `../trident-get-starting/kubernetes-deployment.html`.
2. Configure Astra Trident de la siguiente manera:
  - a. Recopile el nombre DNS de LIF de gestión de la SVM. Por ejemplo, utilice 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 la autenticación. Si está utilizando un `ontap-san` back-end, consulte ["aquí"](#). Si está utilizando un `ontap-nas` back-end, consulte ["aquí"](#).



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:

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

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"](#)



No especifique `dataLIF` para la `ontap-san` y.. `ontap-san-economy` Controladores para permitir que Astra Trident utilice multivía.



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.

Después de la implementación, lleve a cabo los pasos para crear un ["clase de almacenamiento, aprovisiona un volumen y monte el volumen en un pod"](#).

### Obtenga más información

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

## 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. A continuación, se muestra 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 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 AWS	AciKey	Clave API de cuenta CVS
Cloud Volumes Service para AWS	Clave secreta	Clave secreta de cuenta CVS
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



Descripción de campos secretos de la plataforma de almacenamiento	Secreto	Descripción de los campos
ONTAP	ClientPrivateKey	Valor codificado en base64 de la clave privada de cliente. Se utiliza para autenticación basada en certificados
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:

```

$ kubectl -n trident get tbc backend-tbc-ontap-san

```

NAME	BACKEND NAME	BACKEND UUID
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 El uid de la 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 podría requerir la intervención del usuario.

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

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. La lastOperationStatus el campo 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"](#).

# 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). En este caso, `TridentBackendConfig` Los objetos se pueden actualizar directamente mediante Kubernetes.

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

Como alternativa, realice cambios en el existente `TridentBackendConfig` CR ejecutando 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. 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:

- Los back-ends preexistentes, que no tienen `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 back-end existente, necesitará obtener la configuración del back-end. 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 de almacenamiento virtual (si existen) 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 -n trident 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 creó 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.

# Gestione las clases de almacenamiento

Busque información sobre cómo crear una clase de almacenamiento, eliminar una clase de almacenamiento y ver las clases de almacenamiento existentes.

## Diseñe una clase de almacenamiento

Consulte ["aquí"](#) para obtener más información acerca de las clases de almacenamiento y cómo las configura.

## Cree una clase de almacenamiento

Después de tener un archivo de clase de almacenamiento, ejecute el siguiente comando:

```
kubectl create -f <storage-class-file>
```

`<storage-class-file>` debe sustituirse por el nombre de archivo de clase de almacenamiento.

## 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 esixting StorageClasses y volver a crearlos con `parameters.fsType` especificado.

## 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 tener una clase de almacenamiento predeterminada en el clúster en un momento dado. 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}] '
```

## Realizar operaciones de volumen

Más información sobre las funciones que ofrece Astra Trident para la gestión de volúmenes.

- ["Utilice Topología CSI"](#)
- ["Trabajar con instantáneas"](#)
- ["Expandir los volúmenes"](#)
- ["Importar volúmenes"](#)

### 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"](#). 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 pueden estar ubicados en distintas regiones de una zona de disponibilidad o en varias zonas de disponibilidad. 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 dependencia alguna 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 funciona con la versión 1.17 o posterior.

```
$ kubectl version
Client Version: version.Info{Major:"1", Minor:"19",
GitVersion:"v1.19.3",
GitCommit:"1e11e4a2108024935ecfcb2912226cedeafd99df",
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:"1e11e4a2108024935ecfcb2912226cedeafd99df",
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","kubernetes.io/arch":"amd64","kubernetes.io/hostname":"node1","kubernetes.io/os":"linux","node-role.kubernetes.io/master":"","topology.kubernetes.io/region":"us-east1","topology.kubernetes.io/zone":"us-east1-a"}]
[node2,
{"beta.kubernetes.io/arch":"amd64","beta.kubernetes.io/os":"linux","kubernetes.io/arch":"amd64","kubernetes.io/hostname":"node2","kubernetes.io/os":"linux","node-role.kubernetes.io/worker":"","topology.kubernetes.io/region":"us-east1","topology.kubernetes.io/zone":"us-east1-b"}]
[node3,
{"beta.kubernetes.io/arch":"amd64","beta.kubernetes.io/os":"linux","kubernetes.io/arch":"amd64","kubernetes.io/hostname":"node3","kubernetes.io/os":"linux","node-role.kubernetes.io/worker":"","topology.kubernetes.io/region":"us-east1","topology.kubernetes.io/zone":"us-east1-c"}]
```

### 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.

Este es el aspecto de una definición de backend de ejemplo:

```
{
  "version": 1,
  "storageDriverName": "ontap-san",
  "backendName": "san-backend-us-east1",
  "managementLIF": "192.168.27.5",
  "svm": "iscsi_svm",
  "username": "admin",
  "password": "xxxxxxxxxxxx",
  "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-central1",
"managementLIF": "172.16.238.5",
"svm": "nfs_svm",
"username": "admin",
"password": "Netapp123",
"supportedTopologies": [
  {"topology.kubernetes.io/region": "us-central1",
"topology.kubernetes.io/zone": "us-central1-a"},
  {"topology.kubernetes.io/region": "us-central1",
"topology.kubernetes.io/zone": "us-central1-b"}
]
"storage": [
  {
    "labels": {"workload":"production"},
    "region": "Iowa-DC",
    "zone": "Iowa-DC-A",
    "supportedTopologies": [
      {"topology.kubernetes.io/region": "us-central1",
"topology.kubernetes.io/zone": "us-central1-a"}
    ]
  },
  {
    "labels": {"workload":"dev"},
    "region": "Iowa-DC",
    "zone": "Iowa-DC-B",
    "supportedTopologies": [
      {"topology.kubernetes.io/region": "us-central1",
"topology.kubernetes.io/zone": "us-central1-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:

```

$ kubectl create -f pvc.yaml
persistentvolumeclaim/pvc-san created
$ kubectl get pvc
NAME          STATUS    VOLUME   CAPACITY   ACCESS MODES   STORAGECLASS
AGE
pvc-san      Pending                netapp-san-us-east1
2s
$ kubectl 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: vol1
      persistentVolumeClaim:
        claimName: pvc-san
  containers:
    - name: sec-ctx-demo
      image: busybox
      command: [ "sh", "-c", "sleep 1h" ]
      volumeMounts:
        - name: vol1
          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

A partir del lanzamiento de Astra Trident 20.01, se pueden crear snapshots de VP en la capa de Kubernetes. Puede utilizar estas snapshots para mantener copias de un momento específico de los volúmenes que haya creado Astra Trident y programar la creación de volúmenes adicionales (clones). La copia de Snapshot de volumen es compatible con `ontap-nas`, `ontap-nas-flexgroup`, `ontap-san`, `ontap-san-economy`, `solidfire-san`, `aws-cvs`, `gcp-cvs`, y `azure-netapp-files` de windows



Esta función está disponible en Kubernetes 1.17 (beta) y es GA a partir de 1.20. Para comprender los cambios relacionados con el paso de beta a GA, consulte ["el blog del lanzamiento"](#). Con la graduación a GA, el `v1` Se introduce la versión API y es compatible con versiones anteriores `v1beta1` snapshot.

### Lo que necesitará

- Para crear snapshots de volumen, es necesario crear una controladora de snapshot externa, así como algunas definiciones de recursos personalizados (CRD). Esta es la responsabilidad del orquestador de Kubernetes que se está utilizando (por ejemplo: Kubeadm, GKE, OpenShift).

Es posible crear una controladora Snapshot externa y CRD Snapshot de la siguiente manera:

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-3.0/client/config/crd/snapshot.storage.k8s.io_volumesnapshotclasses.yaml
kubectl apply -f https://raw.githubusercontent.com/kubernetes-csi/external-snapshotter/release-3.0/client/config/crd/snapshot.storage.k8s.io_volumesnapshotcontents.yaml
kubectl apply -f https://raw.githubusercontent.com/kubernetes-csi/external-snapshotter/release-3.0/client/config/crd/snapshot.storage.k8s.io_volumesnapshots.yaml
```

2. Crear la controladora Snapshot en el espacio de nombres que desee. Edite los manifiestos YAML a continuación para modificar el espacio de nombres.



No cree un controlador de instantáneas si configura instantáneas de volumen bajo demanda en un entorno GKE. GKE utiliza un controlador de instantáneas oculto integrado.

```
kubectl apply -f https://raw.githubusercontent.com/kubernetes-csi/external-snapshotter/release-3.0/deploy/kubernetes/snapshot-controller/rbac-snapshot-controller.yaml
kubectl apply -f https://raw.githubusercontent.com/kubernetes-csi/external-snapshotter/release-3.0/deploy/kubernetes/snapshot-controller/setup-snapshot-controller.yaml
```



CSI Snapshotter proporciona un ["validando webhook"](#) ayudar a los usuarios a validar las instantáneas v1beta1 existentes y confirmar que son objetos de recursos válidos. El gancho web validador etiqueta automáticamente objetos de instantánea no válidos e impide la creación de futuros objetos no válidos. Kubernetes orchestrator pone en marcha el enlace validando webhook. Consulte las instrucciones para implementar manualmente el enlace web validador ["aquí"](#). Busque ejemplos de manifiestos de instantáneas no válidos ["aquí"](#).

El ejemplo detallado a continuación explica las construcciones necesarias para trabajar con instantáneas y muestra cómo se pueden crear y usar las instantáneas.

### Paso 1: Configurar un VolumeSnapshotClass

Antes de crear una snapshot de volumen, establezca un enlace: `./trident-reference/objects.html[VolumeSnapshotClass#]`.

```
$ cat snap-sc.yaml
#Use apiVersion v1 for Kubernetes 1.20 and above. For Kubernetes 1.17 -
1.19, use apiVersion v1beta1.
apiVersion: snapshot.storage.k8s.io/v1
kind: VolumeSnapshotClass
metadata:
  name: csi-snapclass
driver: csi.trident.netapp.io
deletionPolicy: Delete
```

La driver Apunta al driver 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.

## Paso 2: Crear una instantánea de una RVP existente

```
$ cat snap.yaml
#Use apiVersion v1 for Kubernetes 1.20 and above. For Kubernetes 1.17 -
1.19, use apiVersion v1beta1.
apiVersion: snapshot.storage.k8s.io/v1
kind: VolumeSnapshot
metadata:
  name: pvcl-snap
spec:
  volumeSnapshotClassName: csi-snapclass
  source:
    persistentVolumeClaimName: pvcl
```

La copia de Snapshot se está creando para una RVP llamada `pvcl`, y el nombre de la instantánea se establece en `pvcl-snap`.

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

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

Esto creó un `VolumeSnapshot` objeto. Un `VolumeSnapshot` es análogo a un `PVC` y está asociado a un `VolumeSnapshotContent` objeto que representa la instantánea real.

Es posible identificar la `VolumeSnapshotContent` objeto para `pvcl-snap VolumeSnapshot`, describiéndolo.

```

$ kubectl describe volumesnapshots pvcl-snap
Name:          pvcl-snap
Namespace:     default
.
.
.
Spec:
  Snapshot Class Name:  pvcl-snap
  Snapshot Content Name: snapcontent-e8d8a0ca-9826-11e9-9807-525400f3f660
  Source:
    API Group:
    Kind:      PersistentVolumeClaim
    Name:      pvcl
Status:
  Creation Time:  2019-06-26T15:27:29Z
  Ready To Use:   true
  Restore Size:   3Gi
.
.

```

La Snapshot Content Name Identifica el objeto VolumeSnapshotContent que sirve esta snapshot. La Ready To Use Parámetro indica que la Snapshot se puede usar para crear una RVP nueva.

### Paso 3: Creación de EVs a partir de VolumeSnapshots

Consulte el siguiente ejemplo para crear una RVP con una Snapshot:

```

$ 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

```

dataSource Muestra que la RVP debe crearse con un VolumeSnapshot llamado pvcl-snap como la fuente

de los datos. Esto le indica a Astra Trident que cree una RVP a partir de la snapshot. Una vez creada la RVP, se puede conectar a un pod y utilizarla como cualquier otro PVC.



Cuando se elimina un volumen persistente con instantáneas asociadas, el volumen Trident correspondiente se actualiza a un “estado de eliminación”. Para eliminar el volumen Astra Trident, deben eliminarse las snapshots del volumen.

## Obtenga más información

- ["Copias de Snapshot de volumen"](#)
- [enlace:../trident-reference/objects.html\[VolumeSnapshotClass#\]](#)

## 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.

### Descripción general

Para expandir un VP iSCSI, se deben realizar los siguientes pasos:

- Editar la definición de StorageClass para establecer el `allowVolumeExpansion` campo a `true`.
- Edición de la definición de PVC y actualización de `spec.resources.requests.storage` para reflejar el nuevo tamaño deseado, que debe ser mayor que el tamaño original.
- Para que se pueda cambiar el tamaño, se debe conectar el PV a un pod. 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.

El ejemplo siguiente muestra cómo funcionan las VP iSCSI.

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

```
$ 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

```
$ 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).

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

$ kubectl get pv
NAME          CAPACITY  ACCESS MODES
RECLAIM POLICY  STATUS    CLAIM                                STORAGECLASS  REASON  AGE
pvc-8a814d62-bd58-4253-b0d1-82f2885db671  1Gi      RWO
Delete         Bound     default/san-pvc                     ontap-san     10s
```

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

En este ejemplo, se crea un pod que utiliza san-pvc.

```
$ kubectl get pod
NAME          READY   STATUS    RESTARTS   AGE
centos-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:    centos-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:

```
$ kubectl get pvc san-pvc
NAME          STATUS    VOLUME                                     CAPACITY
ACCESS MODES  STORAGECLASS  AGE
san-pvc      Bound       pvc-8a814d62-bd58-4253-b0d1-82f2885db671  2Gi
RWO           ontap-san    11m

$ kubectl get pv
NAME          CAPACITY  ACCESS MODES
RECLAIM POLICY STATUS    CLAIM          STORAGECLASS  REASON    AGE
pvc-8a814d62-bd58-4253-b0d1-82f2885db671  2Gi        RWO
Delete              Bound       default/san-pvc  ontap-san    12m

$ tridentctl get volumes -n trident
+-----+-----+-----+
+-----+-----+-----+-----+
|          NAME          | SIZE  | STORAGE CLASS |
+-----+-----+-----+-----+
|          BACKEND UUID  | STATE | MANAGED  |
+-----+-----+-----+-----+
| pvc-8a814d62-bd58-4253-b0d1-82f2885db671 | 2.0 GiB | ontap-san |
| block      | a9b7bfff-0505-4e31-b6c5-59f492e02d33 | online | true  |
+-----+-----+-----+-----+
+-----+-----+-----+-----+
```

## Expanda un volumen NFS

Astra Trident admite la ampliación de volúmenes para los VP de NFS aprovisionados en `ontap-nas`, `ontap-nas-economy`, `ontap-nas-flexgroup`, `aws-cvs`, `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 `kubectl 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: Expanda 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 1 GB:

```

$ 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:

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

$ kubectl get pv pvc-08f3d561-b199-11e9-8d9f-5254004dfdb7
NAME          CAPACITY  ACCESS MODES
RECLAIM POLICY STATUS    CLAIM          STORAGECLASS  REASON
AGE
pvc-08f3d561-b199-11e9-8d9f-5254004dfdb7  1Gi      RWO
Delete          Bound      default/ontapnas20mb  ontapnas
5m35s

$ tridentctl get volume pvc-08f3d561-b199-11e9-8d9f-5254004dfdb7 -n
trident
+-----+-----+-----+
+-----+-----+-----+-----+
|          NAME          | SIZE | STORAGE CLASS |
PROTOCOL |          BACKEND UUID          | STATE | MANAGED |
+-----+-----+-----+-----+
+-----+-----+-----+-----+
| pvc-08f3d561-b199-11e9-8d9f-5254004dfdb7 | 1.0 GiB | ontapnas      |
file      | c5a6f6a4-b052-423b-80d4-8fb491a14a22 | online | true      |
+-----+-----+-----+-----+
+-----+-----+-----+-----+
```

## Importar volúmenes

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

### Controladores que admiten la importación de volúmenes

En esta tabla se muestran los controladores que admiten la importación de volúmenes y la versión en la que se introdujeron.

Controlador	Liberar
ontap-nas	19.04
ontap-nas-flexgroup	19.04
solidfire-san	19.04
aws-cvs	19.04

Controlador	Liberar
azure-netapp-files	19.04
gcp-cvs	19.04
ontap-san	19.04

## ¿Por qué debo importar volúmenes?

Existen varios casos de uso para importar un volumen en Trident:

- Contenerización de una aplicación y reutilización del conjunto de datos existente
- Usar un clon de un conjunto de datos para una aplicación efímera
- Reconstruir un clúster de Kubernetes con fallos
- Migración de datos de aplicaciones durante la recuperación tras siniestros

## ¿Cómo funciona la importación?

El proceso de importación de volúmenes utiliza el archivo de solicitud de volumen persistente (PVC) para crear la RVP. Como mínimo, el archivo PVC debe incluir los campos name, Namespace, accessModes y storageClassName como se muestra en el ejemplo siguiente.

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

La `tridentctl` el cliente se utiliza para importar un volumen de almacenamiento existente. Trident importa el volumen persiste en los metadatos del volumen y crea la RVP y el VP.

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

Para importar un volumen de almacenamiento, especifique 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, CVS Volume path). El volumen de almacenamiento debe permitir el acceso de lectura/escritura y debe ser accesible desde el back-end de Astra Trident especificado. La `-f` El argumento String es necesario y especifica la ruta al archivo YLMA o PVC JSON.

Cuando Astra Trident recibe la solicitud de importación de volumen, se determina el tamaño de volumen

existente 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.

Cuando se importa un volumen con `--no-manage` Argumento, Trident no realiza ninguna operación adicional en la RVP o el VP durante el ciclo de vida de los objetos. Dado que Trident ignora los eventos VP y RVP para `--no-manage` Los objetos, el volumen de almacenamiento no se elimina cuando se elimina el VP. 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.

Trident 19.07 y versiones posteriores gestionan el adjunto de los VP y monta el volumen como parte de la importación. Para las importaciones con versiones anteriores de Astra Trident, no habrá ninguna operación en la ruta de datos y la importación de volúmenes no verificará si es posible montar el volumen. Si se produce un error con la importación de volumen (por ejemplo, StorageClass es incorrecto), puede recuperar cambiando la política de reclamación en el VP a `retain`, Eliminando el PVC y el VP y volviendo a intentar el comando de importación de volumen.

#### ontap-nas y.. ontap-nas-flexgroup importaciones

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.



Un volumen de ONTAP debe ser del tipo `rw` que haya que importar Trident. Si un volumen es del tipo `dp`, es un volumen de destino de SnapMirror, se debe interrumpir la relación de mirroring antes de importar el volumen a Trident.



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

Por ejemplo, para importar un volumen llamado `managed_volume` en un backend llamado `ontap_nas`, utilice el siguiente comando:

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

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

Para importar un volumen denominado `unmanaged_volume` (en la `ontap_nas` backend), que Trident no administrará, utilice el siguiente comando:

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

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

Cuando utilice la `--no-manage` Argumento, Trident no cambia el nombre del volumen ni se valida si se montó el volumen. Se produce un error en la operación de importación de volumen si el volumen no se montó manualmente.



Se ha solucionado un error existente con la importación de volúmenes con `UnixPermissions` personalizado. Puede especificar `unixPermissions` en la definición de PVC o en la configuración de back-end, e indicar a Astra Trident que importe el volumen según corresponda.

## ontap-san importar

Astra Trident también puede importar SAN FlexVols de ONTAP que contienen una única LUN. Esto es consistente con `ontap-san` Controlador, que crea una FlexVol para cada RVP y una LUN dentro del FlexVol. Puede utilizar el `tridentctl import` comando de la misma forma que en otros casos:

- Incluya el nombre del `ontap-san` back-end.

- Escriba el nombre de la FlexVol que se debe importar. Recuerde, esta FlexVol solo contiene una LUN que es necesario importar.
- Proporcione la ruta de la definición de PVC que debe utilizarse con el `-f` bandera.
- Elija entre administrar o no administrar el PVC. De forma predeterminada, Trident gestionará la RVP y cambiará el nombre de los FlexVol y LUN en el back-end. Para importar como volumen no administrado, pase el `--no-manage` bandera.



Al importar un no administrado `ontap-san` Volumen, debe asegurarse de que el nombre de la LUN de la FlexVol sea `lun0` y se asigna a un `igroup` con los iniciadores deseados. Astra Trident se encarga automáticamente de esto en una importación gestionada.

A continuación, Astra Trident importará el FlexVol y lo asociará con la definición de PVC. Astra Trident también cambia el nombre de FlexVol al `pvc-<uuid>` Formatear y la LUN dentro de la FlexVol a `lun0`.



Se recomienda importar volúmenes que no tengan conexiones activas existentes. Si desea importar un volumen que está utilizado activamente, Clone el volumen primero y, a continuación, realice la importación.

### Ejemplo

Para importar la `ontap-san-managed` FlexVol que está presente en el `ontap_san_default` back-end, ejecute el `tridentctl import` comando como:

```
$ tridentctl import volume ontapsan_san_default ontap-san-managed -f pvc-
basic-import.yaml -n trident -d
```

PROTOCOL	NAME	SIZE	STORAGE CLASS
block	pvc-d6ee4f54-4e40-4454-92fd-d00fc228d74a	20 MiB	basic
	cd394786-ddd5-4470-adc3-10c5ce4ca757	online	true



Un volumen ONTAP debe ser del tipo `rw` que importe Astra Trident. Si un volumen es del tipo `dp`, es un volumen de destino de SnapMirror, se debe interrumpir la relación de mirroring antes de importar el volumen a Astra Trident.

### element **importar**

Es posible importar el software NetApp Element/volúmenes de HCI de NetApp en el clúster de Kubernetes con Trident. Necesita el nombre de su entorno de administración Astra Trident, y el nombre único del volumen y el archivo PVC como argumentos para `tridentctl import` comando.

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

```
+-----+-----+-----+-----+
+-----+-----+-----+-----+
|          NAME          | SIZE | STORAGE CLASS |
PROTOCOL |          BACKEND UUID          | STATE | MANAGED |
+-----+-----+-----+-----+
+-----+-----+-----+-----+
| pvc-970ce1ca-2096-4ecd-8545-ac7edc24a8fe | 10 GiB | basic-element |
block    | d3ba047a-ea0b-43f9-9c42-e38e58301c49 | online | true      |
+-----+-----+-----+-----+
+-----+-----+-----+-----+
```



El controlador Element admite los nombres de volúmenes duplicados. Si hay nombres de volúmenes duplicados, el proceso de importación de volúmenes de Trident devuelve un error. Como solución alternativa, Clone el volumen y proporcione un nombre de volumen único. A continuación, importe el volumen clonado.

#### aws-cvs importar



Para importar un volumen respaldado por Cloud Volumes Service de NetApp en AWS, identifique el volumen por su ruta de volumen en lugar de su nombre.

Para importar una `aws-cvs` volumen en el backend llamado `awscvs_YEppr` con la ruta del volumen de `adroit-jolly-swift`, utilice el siguiente comando:

```
$ tridentctl import volume awscvs_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 | aws-storage   | file
| e1a6e65b-299e-4568-ad05-4f0a105c888f | online | true          |
+-----+-----+-----+-----+
+-----+-----+-----+-----+
```



La ruta del volumen es la parte de la ruta de exportación del volumen después de `:/`. Por ejemplo, si la ruta de exportación es `10.0.0.1:/adroit-jolly-swift`, la ruta de volumen es `adroit-jolly-swift`.



## gcp-cvs **importar**

Importar un gcp-cvs el volumen funciona igual que la importación de un aws-cvs volumen.

## azure-netapp-files **importar**

Para importar una azure-netapp-files volumen en el backend llamado azurenetappfiles\_40517 con la ruta del volumen importvol1, ejecute el siguiente comando:

```
$ tridentctl import volume azurenetappfiles_40517 importvol1 -f <path-to-pvc-file> -n trident
```

```
+-----+-----+-----+-----+
+-----+-----+-----+-----+
|          NAME          |  SIZE  | STORAGE CLASS |
+-----+-----+-----+-----+
|          BACKEND UUID  |        | STATE         |
+-----+-----+-----+-----+
|          NAME          |  SIZE  | STORAGE CLASS |
+-----+-----+-----+-----+
| pvc-0ee95d60-fd5c-448d-b505-b72901b3a4ab | 100 GiB | anf-storage   |
+-----+-----+-----+-----+
| file          | 1c01274f-d94b-44a3-98a3-04c953c9a51e | online | true         |
+-----+-----+-----+-----+
+-----+-----+-----+-----+
```



La ruta de volumen para el volumen ANF está presente en la ruta de montaje después de :/. Por ejemplo, si la ruta de montaje es 10.0.0.2:/importvol1, la ruta de volumen es importvol1.

## Prepare el nodo de trabajo

Todos los nodos de trabajo del clúster de Kubernetes deben poder montar los volúmenes que haya aprovisionado para los pods. Si está utilizando la ontap-nas, ontap-nas-economy, o. ontap-nas-flexgroup Controlador para uno de los back-ends, los nodos de trabajador necesitan las herramientas NFS. De lo contrario, se necesitan las herramientas iSCSI.

Las versiones recientes de RedHat CoreOS tienen instaladas de forma predeterminada NFS e iSCSI.



Siempre debe reiniciar los nodos de trabajo después de instalar las herramientas NFS o iSCSI, o bien es posible que se produzca un error en la asociación de volúmenes a contenedores.

## Volúmenes de NFS

Protocolo	De NetApp	Comandos
NFS	RHEL/CentOS	<pre>sudo yum install -y nfs-utils</pre>

Protocolo	De NetApp	Comandos
NFS	Ubuntu/Debian	<code>sudo apt-get install -y nfs-common</code>



Debe asegurarse de que el servicio NFS se haya iniciado durante el arranque.

## Volúmenes iSCSI

Tenga en cuenta lo siguiente al usar volúmenes iSCSI:

- 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, o RHEL o CentOS versión 8.2 o posterior con el `solidfire-san` Controlador, asegúrese de que el algoritmo de autenticación CHAP esté establecido en MD5 in `/etc/iscsi/iscsid.conf`.

```
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, asegúrese de especificar el `discard MountOption` en `StorageClass` para realizar un reclamación de espacio en línea. Consulte "[La documentación de redhat](#)".

Protocolo	De NetApp	Comandos
ISCSI	RHEL/CentOS	<ol style="list-style-type: none"> <li>1. Instale los siguientes paquetes del sistema: <pre>sudo yum install -y lsscsi iscsi-initiator- utils sg3_utils device- mapper-multipath</pre> </li> <li>2. Compruebe que la versión de iscsi-initiator-utils sea 6.2.0.874-2.el7 o posterior: <pre>rpm -q iscsi-initiator- utils</pre> </li> <li>3. Configure el escaneo en manual: <pre>sudo sed -i 's/^\(node.session.scan \).*\/1 = manual/' /etc/iscsi/iscsid.conf</pre> </li> <li>4. Activar accesos múltiples: <pre>sudo mpathconf --enable --with_multipathd y --find_multipaths n</pre> <div>  <p>Asegúrese etc/multipat h.conf contiene find_multipa ths no inferior defaults.</p> </div> </li> <li>5. Asegúrese de que así sea iscsid y.. multipathd están en ejecución: <pre>sudo systemctl enable --now iscsid multipathd</pre> </li> <li>6. Activar e iniciar iscsi: <pre>sudo systemctl enable --now iscsi</pre> </li> </ol>

Protocolo	De NetApp	Comandos
ISCSI	Ubuntu/Debian	<p>1. Instale los siguientes paquetes del sistema:</p> <pre>sudo apt-get install -y open-iscsi lsscsi sg3- utils multipath-tools scsitools</pre> <p>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):</p> <pre>dpkg -l open-iscsi</pre> <p>3. Configure el escaneo en manual:</p> <pre>sudo sed -i 's/^\(node.session.scan \).*\/\1 = manual/' /etc/iscsi/iscsid.conf</pre> <p>4. Activar accesos múltiples:</p> <pre>sudo tee /etc/multipath.conf &lt; ←'EOF' defaults { user_friendly_names yes find_multipaths no } EOF sudo systemctl enable --now multipath- tools.service sudo service multipath- tools restart</pre> <div>  <p>Asegúrese etc/multipat h.conf contiene find_multipa ths no inferior defaults.</p> </div> <p>5. Asegúrese de que así sea open-iscsi y.. multipath-tools están habilitadas y en ejecución:</p> <pre>sudo systemctl status multipath-tools</pre>



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.



Si desea obtener más información acerca de la preparación automática del nodo de trabajo, que es una función beta, consulte ["aquí"](#).

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

## Preparación automática del nodo de trabajo

Astra Trident puede instalar automáticamente el necesario NFS y.. iSCSI Herramientas en los nodos presentes en el clúster de Kubernetes. Se trata de una **función beta** y **no está pensada para** clusters de producción. Hoy en día, la función está disponible para nodos que ejecutan **CentOS, RHEL y Ubuntu**.

Para esta función, Astra Trident incluye una nueva Marca de instalación: `--enable-node-prep` para las instalaciones implementadas con `tridentctl`. Para implementaciones con el operador Trident, use la opción Boolean `enableNodePrep`.



La `--enable-node-prep` La opción de instalación le indica a Astra Trident que instale y asegúrese de que los paquetes y/o servicios NFS e iSCSI se estén ejecutando cuando un volumen está montado en un nodo de trabajo. Se trata de una **característica beta** pensada para ser utilizada en entornos de prueba/desarrollo que **no está calificada** para su uso en producción.

Cuando la `--enable-node-prep` Flag se incluye en las instalaciones de Astra Trident implementadas con `tridentctl`, esto es lo que sucede:

1. Como parte de la instalación, Astra Trident registra los nodos en los que se ejecuta.
2. Cuando se realiza una solicitud de reclamación de volumen persistente (PVC), Astra Trident crea un VP a partir de uno de los back-ends que gestiona.
3. Para usar la RVP en un pod, es necesario que Astra Trident monte el volumen en el nodo en el que se ejecuta el pod. Astra Trident intenta instalar las utilidades de cliente NFS/iSCSI necesarias y garantizar que los servicios necesarios están activos. Esto se realiza antes de montar el volumen.

La preparación de un nodo de trabajo sólo se realiza una vez como parte del primer intento de montaje de un volumen. Todos los montajes de volumen subsiguientes deben tener éxito siempre que no se produzcan cambios fuera de Astra Trident touch el NFS y.. iSCSI servicios públicos.

De este modo, Astra Trident puede garantizar que todos los nodos de un clúster de Kubernetes tengan las utilidades necesarias para montar y conectar volúmenes. En el caso de los volúmenes NFS, la política de exportación también debe permitir el montaje del volumen. Trident puede gestionar automáticamente políticas de exportación por back-end; como alternativa, los usuarios pueden gestionar políticas de exportación fuera de banda.

## Supervisión de Astra Trident

Astra Trident proporciona un conjunto de extremos de métricas Prometheus que puede utilizar para supervisar el rendimiento de Astra Trident.

Las métricas proporcionadas por Astra Trident le permiten hacer lo siguiente:

- Mantenga pestañas sobre el estado y la configuración de Astra Trident. Puede examinar la eficacia de las operaciones y si puede comunicarse con los back-ends como se esperaba.
- Examine la información de uso del back-end, y comprenda cuántos volúmenes se aprovisionan en un entorno de administración y la cantidad de espacio consumido, etc.
- Mantenga una asignación de la cantidad de volúmenes aprovisionados en los back-ends disponibles.
- Seguimiento del rendimiento. Podrá observar el tiempo que tarda Astra Trident en comunicarse con los back-ends y realizar operaciones.



De forma predeterminada, las métricas de Trident se exponen en el puerto de destino 8001 en la `/metrics` extremo. Estas métricas están **activadas de forma predeterminada** cuando se instala Trident.

### Lo que necesitará

- Un clúster de Kubernetes con Astra Trident instalado.
- Una instancia Prometheus. Esto puede ser un ["Puesta en marcha de Prometeo en contenedores"](#) También puede optar por ejecutar Prometheus como a. ["aplicación nativa"](#).

## Paso 1: Definir un objetivo Prometheus

Debe definir un destino Prometheus para recopilar las métricas y obtener información sobre los back-ends que administra Astra Trident, los volúmenes que crea, etc. Este ["blog"](#) Explica cómo puede usar Prometheus y Grafana con Astra Trident para recuperar métricas. En el blog se explica cómo puede ejecutar Prometheus como operador de su clúster de Kubernetes y la creación de un ServiceMonitor para obtener las métricas de Astra Trident.

## Paso 2: Cree un Prometheus ServiceMonitor

Para usar las métricas de Trident, debe crear un Prometheus ServiceMonitor que vaya a ver el `trident-csi` el servicio y escucha el `metrics` puerto. Un ejemplo de ServiceMonitor tiene este aspecto:

```

apiVersion: monitoring.coreos.com/v1
kind: ServiceMonitor
metadata:
  name: trident-sm
  namespace: monitoring
  labels:
    release: prom-operator
spec:
  jobLabel: trident
  selector:
    matchLabels:
      app: controller.csi.trident.netapp.io
  namespaceSelector:
    matchNames:
      - trident
  endpoints:
    - port: metrics
      interval: 15s

```

Esta definición de ServiceMonitor recupera las métricas devueltas por `trident-csi` servicio y busca específicamente la `metrics` extremo del servicio. Como resultado, Prometheus ahora está configurado para comprender las métricas de Astra Trident.

Además de las métricas disponibles directamente de Astra Trident, kubelet expone muchas `kubelet_volume_*` métricas a través de su propio extremo de métricas. Kubelet puede proporcionar información sobre los volúmenes adjuntos y los pods y otras operaciones internas que realiza. Consulte ["aquí"](#).

### Paso 3: Consulte las métricas de Trident con PromQL

PromQL es adecuado para crear expresiones que devuelvan datos tabulares o de series temporales.

A continuación se muestran algunas consultas PromQL que se pueden utilizar:

#### Obtenga información de estado de Trident

- **Porcentaje de respuestas HTTP 2XX de Astra Trident**

```

(sum (trident_rest_ops_seconds_total_count{status_code=~"2.."} OR on()
vector(0)) / sum (trident_rest_ops_seconds_total_count)) * 100

```

- **Porcentaje de respuestas DE DESCANSO de Astra Trident a través del código de estado**

```

(sum (trident_rest_ops_seconds_total_count) by (status_code) / scalar
(sum (trident_rest_ops_seconds_total_count))) * 100

```

- **Duración media en ms de operaciones realizadas por Astra Trident**

```
sum by (operation)
(trident_operation_duration_milliseconds_sum{success="true"}) / sum by
(operation)
(trident_operation_duration_milliseconds_count{success="true"})
```

## Obtenga la información de uso de Astra Trident

- **Tamaño medio del volumen**

```
trident_volume_allocated_bytes/trident_volume_count
```

- **Espacio total por volumen aprovisionado por cada backend**

```
sum (trident_volume_allocated_bytes) by (backend_uuid)
```

## Obtenga el uso de cada volumen



Esto solo se habilita si también se recopilan las métricas Kubelet.

- **Porcentaje de espacio usado para cada volumen**

```
kubelet_volume_stats_used_bytes / kubelet_volume_stats_capacity_bytes *
100
```

## Obtenga más información sobre la telemetría Astra Trident AutoSupport

De forma predeterminada, Astra Trident envía a NetApp métricas y información básica sobre los back-end a través de una cadencia diaria.

- Para que Astra Trident deje de enviar métricas Prometheus e información básica del back-end a NetApp, pase el `--silence-autosupport` Durante la instalación de Astra Trident.
- Astra Trident también puede enviar registros de contenedores al soporte de NetApp bajo demanda a través `tridentctl send autosupport`. Deberá activar Astra Trident para cargar los registros. Antes de enviar los registros, debe aceptar las <https://www.netapp.com/company/legal/privacy-policy/> ["política de privacidad"^].
- A menos que se especifique lo contrario, Astra Trident recupera los registros de las últimas 24 horas.
- Se puede especificar el plazo de retención del registro con el `--since` bandera. Por ejemplo: `tridentctl send autosupport --since=1h`. Esta información se recopila y se envía a través de un `trident-autosupport` Contenedor instalado junto a Astra Trident. Puede obtener la imagen del contenedor en ["AutoSupport de Trident"](#).
- Trident AutoSupport no recopila ni transmite información personal identificable (PII) ni Información personal. Viene con un ["CLUF"](#) Esto no se aplica a la imagen del contenedor Trident. Puede obtener más



información sobre el compromiso de NetApp con la seguridad y la confianza de los datos ["aquí"](#).

Una carga útil de ejemplo enviada por Astra Trident tiene el siguiente aspecto:

```
{
  "items": [
    {
      "backendUUID": "ff3852e1-18a5-4df4-b2d3-f59f829627ed",
      "protocol": "file",
      "config": {
        "version": 1,
        "storageDriverName": "ontap-nas",
        "debug": false,
        "debugTraceFlags": null,
        "disableDelete": false,
        "serialNumbers": [
          "nwkvzfanek_SN"
        ],
        "limitVolumeSize": ""
      },
      "state": "online",
      "online": true
    }
  ]
}
```

- Los mensajes de AutoSupport se envían al extremo AutoSupport de NetApp. Si está utilizando un Registro privado para almacenar imágenes contenedoras, puede utilizar `--image-registry` bandera.
- También puede configurar direcciones URL proxy generando los archivos YLMA de instalación. Esto se puede hacer usando `tridentctl install --generate-custom-yaml` Para crear los archivos YAML y agregar `--proxy-url` argumento para `trident-autosupport` contenedor en `trident-deployment.yaml`.

## Deshabilite las métricas de Astra Trident

Para **desactivar las métricas** de ser reportadas, debe generar YAMLs personalizados (utilizando la `--generate-custom-yaml` y editarlas para eliminar `--metrics` no se invoca el indicador para el ``trident-main`` contenedor.

## 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 INTERRUPCIÓN 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.