



Gestione y supervise Astra Trident

Astra Trident

NetApp

January 14, 2026

This PDF was generated from <https://docs.netapp.com/es-es/trident-2406/trident-managing-k8s/upgrade-trident.html> on January 14, 2026. Always check docs.netapp.com for the latest.

Tabla de contenidos

- Gestione y supervise Astra Trident 1
 - Actualice Astra Trident 1
 - Actualice Astra Trident 1
 - Actualizar con el operador 2
 - Actualice con tridentctl 7
 - Gestión de Astra Trident con tridentctl 7
 - Comandos e indicadores globales 7
 - Opciones de comando y indicadores 9
 - Supervisión de Astra Trident 15
 - Descripción general 15
 - Paso 1: Definir un objetivo Prometheus 15
 - Paso 2: Cree un Prometheus ServiceMonitor 15
 - Paso 3: Consulte las métricas de Trident con PromQL 16
 - Obtenga más información sobre la telemetría Astra Trident AutoSupport 17
 - Deshabilite las métricas de Astra Trident 18
 - Desinstale Astra Trident 18
 - Determine el método de instalación original 19
 - Desinstale una instalación del operador Trident 19
 - Desinstale una tridentctl instalación 20

Gestione y supervise Astra Trident

Actualice Astra Trident

Actualice Astra Trident

A partir del lanzamiento de la versión 24,02, Astra Trident sigue una cadencia de lanzamiento de cuatro meses y entrega tres versiones principales cada año. Cada nueva versión se basa en versiones anteriores y proporciona nuevas funciones, mejoras de rendimiento, correcciones de errores y mejoras. Le animamos a que realice una actualización al menos una vez al año para aprovechar las nuevas funciones de Astra Trident.

Consideraciones antes de la actualización

Cuando actualice a la versión más reciente de Astra Trident, tenga en cuenta lo siguiente:

- Solo debe haber una instancia de Astra Trident instalada en todos los espacios de nombres en un clúster de Kubernetes determinado.
- Astra Trident 23,07 y versiones posteriores requieren v1 copias Snapshot de volumen y ya no admite copias Snapshot alfa o beta.
- Si has creado Cloud Volumes Service para Google Cloud en ["Tipo de servicio CVS"](#), debes actualizar la configuración de back-end para utilizar `standardsw` el nivel de servicio o `zoneredundantstandardsw` al actualizar desde Astra Trident 23,01. Si no se actualiza el `serviceLevel` en el backend, se podrían producir errores en los volúmenes. Consulte ["Muestras de tipo de servicio CVS"](#) para obtener más información.
- Cuando se realiza la actualización, es importante que `parameter.fsType StorageClasses` utilices Astra Trident. Puede eliminar y volver a crear `StorageClasses` sin interrumpir los volúmenes existentes anteriores.
 - Este es un **requisito** de la aplicación de ["contextos de seguridad"](#) volúmenes SAN.
 - El directorio [sample input](#) contiene ejemplos, como `storage-class-basic.yaml.template` y `link:https://github.com/NetApp/Trident/blob/master/Trident-installer/sample-input/storage-class-bronze-class[storage-class-bronze-default.yaml ^`.
 - Para obtener más información, consulte ["Problemas conocidos"](#).

Paso 1: Seleccione una versión

Las versiones de Astra Trident siguen una convención de nomenclatura basada en fechas `YY.MM`, donde «YY» son los dos últimos dígitos del año y «MM» es el mes. Las versiones de DOT siguen `YY.MM.X` una convención, donde «X» es el nivel de parche. Deberá seleccionar la versión a la que se actualizará en función de la versión desde la que se actualice.

- Puede realizar una actualización directa a cualquier versión de destino que esté dentro de una ventana de cuatro versiones de la versión instalada. Por ejemplo, puede actualizar directamente de 23,04 (o cualquier versión de 23,04 puntos) a 24,06.
- Si va a actualizar desde una versión fuera de la ventana de cuatro versiones, realice una actualización de varios pasos. Utilice las instrucciones de actualización de la ["versión anterior"](#) que va a actualizar para actualizar a la versión más reciente que se ajuste a la ventana de cuatro versiones. Por ejemplo, si utiliza

22,01 y desea actualizar a la versión 24,06:

- a. Primera actualización de 22,07 a 23,04.
- b. A continuación, actualice de 23,04 a 24,06.



Cuando se actualice con el operador Trident en OpenShift Container Platform, debe actualizar a Trident 21.01.1 o una versión posterior. El operador Trident publicado con 21.01.0 contiene un problema conocido que se ha solucionado en 21.01.1. Para obtener más información, consulte la ["Detalles del problema en GitHub"](#).

Paso 2: Determine el método de instalación original

Para determinar qué versión solías instalar originalmente Astra Trident:

1. Se utiliza `kubectl get pods -n trident` para examinar los pods.
 - Si no hay ningún pod de operador, se instaló Astra Trident mediante `tridentctl`.
 - Si hay un pod de operador, se instaló Astra Trident mediante el operador Trident manualmente o mediante Helm.
2. Si hay un pod de operador, utilícelo `kubectl describe torc` para determinar si Astra Trident se instaló mediante Helm.
 - Si hay una etiqueta Helm, Astra Trident se instaló usando Helm.
 - Si no hay ninguna etiqueta Helm, Astra Trident se instaló manualmente mediante el operador Trident.

Paso 3: Seleccione un método de actualización

Por lo general, debe actualizar utilizando el mismo método que utilizó para la instalación inicial, sin embargo, puede ["desplazarse entre los métodos de instalación"](#). Hay dos opciones para actualizar Astra Trident.

- ["Actualice con el operador Trident"](#)



Le sugerimos que lo revise ["Comprender el flujo de trabajo de actualización del operador"](#) antes de actualizar con el operador.

*

Actualizar con el operador

Comprender el flujo de trabajo de actualización del operador

Antes de usar el operador Trident para actualizar Astra Trident, debe comprender los procesos en segundo plano que se producen durante la actualización. Esto incluye cambios en la controladora Trident, en el pod de controladora y en los pods de nodos, así como en el DaemonSet de nodos que permiten actualizaciones graduales.

Manejo de actualizaciones del operador Trident

Uno de los muchos ["Ventajas del uso del operador Trident"](#) que instalan y actualizan Astra Trident es la gestión automática de los objetos de Astra Trident y Kubernetes sin interrumpir los volúmenes montados existentes. De esta forma, Astra Trident puede admitir renovaciones sin tiempos de inactividad o

"[actualizaciones sucesivas](#)". En concreto, el operador Trident se comunica con el clúster de Kubernetes para:

- Elimine y vuelva a crear la implementación de Trident Controller y DaemonSet de nodos.
- Sustituya el pod de la controladora de Trident y los pods de nodos de Trident por nuevas versiones.
 - Si no se actualiza un nodo, no impide que se actualicen los nodos restantes.
 - Solo los nodos con un nodo de Trident en ejecución pueden montar volúmenes.



Para obtener más información sobre la arquitectura de Astra Trident en el clúster de Kubernetes, consulte "[Arquitectura de Astra Trident](#)".

Flujo de trabajo de actualización del operador

Cuando inicie una actualización con el operador Trident:

1. El operador **Trident**:
 - a. Detecta la versión instalada actualmente de Astra Trident (versión n).
 - b. Actualiza todos los objetos de Kubernetes, incluidos CRD, RBAC y Trident SVC.
 - c. Elimina la implementación de Trident Controller para la versión n .
 - d. Crea la implementación de Trident Controller para la versión $n+1$.
2. **Kubernetes** crea Trident Controller Pod para $n+1$.
3. El operador **Trident**:
 - a. Elimina el conjunto de cambios de nodo Trident para n . El operador no espera la terminación del Node Pod.
 - b. Crea el inicio del demonio del nodo Trident para $n+1$.
4. **Kubernetes** crea pods de nodos Trident en nodos que no ejecutan Trident Node Pod n . De este modo se garantiza que nunca haya más de un pod de nodo de Trident, de ninguna versión, en un nodo.

Actualiza una instalación de Astra Trident mediante el operador Trident o Helm

Puede actualizar Astra Trident de forma manual o mediante Helm mediante el operador Trident. Puede actualizar de una instalación del operador de Trident a otra instalación del operador de Trident o actualizar de una `tridentctl` instalación a una versión del operador de Trident. Revise "[Seleccione un método de actualización](#)" antes de actualizar una instalación del operador Trident.

Actualizar una instalación manual

Puede actualizar desde una instalación de operadores Trident en el ámbito del clúster a otra instalación del operador Trident en el ámbito del clúster. Todas las versiones 21.01 y posteriores de Astra Trident utilizan un operador con ámbito de clúster.



Para actualizar desde Astra Trident que se ha instalado con el operador de espacio de nombres (de la 20,07 a la 20,10), utiliza las instrucciones de actualización de "[la versión instalada](#)" Astra Trident.

Acerca de esta tarea

Trident proporciona un archivo de paquete que se puede utilizar para instalar el operador y crear objetos

asociados para la versión de Kubernetes.

- Para los clústeres que ejecutan Kubernetes 1,24, utilice ["bundle_pre_1_25.yaml"](#).
- Para los clústeres que ejecutan Kubernetes 1,25 o posterior, utilice ["bundle_post_1_25.yaml"](#).

Antes de empezar

Asegúrese de que está utilizando un clúster de Kubernetes en ejecución ["Una versión de Kubernetes compatible"](#).

Pasos

1. Compruebe su versión de Astra Trident:

```
./tridentctl -n trident version
```

2. Elimine el operador Trident que se ha utilizado para instalar la instancia actual de Astra Trident. Por ejemplo, si va a actualizar desde 23,07, ejecute el siguiente comando:

```
kubectl delete -f 23.07.0/trident-installer/deploy/<bundle.yaml> -n trident
```

3. Si personalizó la instalación inicial mediante `TridentOrchestrator` atributos, puede editar el `TridentOrchestrator` objeto para modificar los parámetros de instalación. Esto podría incluir cambios realizados para especificar registros de imágenes de Trident y CSI reflejados para el modo sin conexión, habilitar registros de depuración o especificar secretos de extracción de imágenes.
4. Instale Astra Trident con el archivo YAML del paquete correcto para su entorno, donde *<bundle.yaml>* está `bundle_pre_1_25.yaml` o `bundle_post_1_25.yaml` se basa en su versión de Kubernetes. Por ejemplo, si está instalando Astra Trident 24,06, ejecute el siguiente comando:

```
kubectl create -f 24.06.0/trident-installer/deploy/<bundle.yaml> -n trident
```

Actualizar una instalación Helm

Puede actualizar una instalación de Astra Trident Helm.



Cuando actualice un clúster de Kubernetes de 1,24 a la versión 1,25 o una versión posterior que tiene Astra Trident instalado, debe actualizar los valores.yaml para establecer `excludePodSecurityPolicy true` o añadir `--set excludePodSecurityPolicy=true` al `helm upgrade` comando antes de actualizar el clúster.

Pasos

1. Si ["Instalar Astra Trident mediante Helm"](#) utiliza `helm upgrade trident netapp-trident/trident-operator --version 100.2406.0` para actualizar en un solo paso. Si no ha añadido el repositorio Helm o no puede utilizarlo para actualizar:
 - a. Descargue la última versión de Astra Trident de ["La sección Assets de GitHub"](#).

- b. Utilice `helm upgrade` el comando donde `trident-operator-24.06.0.tgz` refleja la versión a la que desea actualizar.

```
helm upgrade <name> trident-operator-24.06.0.tgz
```



Si establece opciones personalizadas durante la instalación inicial (como la especificación de registros privados y reflejados para imágenes Trident y CSI), agregue el `helm upgrade` comando Using `--set` para asegurarse de que esas opciones se incluyen en el comando UPGRADE; de lo contrario, los valores se restablecerán a los valores predeterminados.

2. Ejecute `helm list` para verificar que la tabla y la versión de la aplicación se han actualizado. Ejecute `tridentctl logs` para revisar los mensajes de depuración.

Actualización de una `tridentctl` instalación al operador Trident

Puede actualizar a la última versión del operador Trident desde una `tridentctl` instalación. Los back-ends y EVs existentes estarán disponibles automáticamente.



Antes de cambiar entre los métodos de instalación, revise ["Moverse entre los métodos de instalación"](#).

Pasos

1. Descargue la última versión de Astra Trident.

```
# Download the release required [24.060.0]
mkdir 24.06.0
cd 24.06.0
wget
https://github.com/NetApp/trident/releases/download/v24.06.0/trident-
installer-24.06.0.tar.gz
tar -xf trident-installer-24.06.0.tar.gz
cd trident-installer
```

2. Cree el `tridentorchestrator` CRD a partir del manifiesto.

```
kubectl create -f
deploy/crds/trident.netapp.io_tridentorchestrators_crd_post1.16.yaml
```

3. Despliegue el operador de ámbito de cluster en el mismo espacio de nombres.

```
kubectl create -f deploy/<bundle-name.yaml>
```

```
serviceaccount/trident-operator created
clusterrole.rbac.authorization.k8s.io/trident-operator created
clusterrolebinding.rbac.authorization.k8s.io/trident-operator created
deployment.apps/trident-operator created
podsecuritypolicy.policy/tridentoperatorpods created
```

```
#Examine the pods in the Trident namespace
```

NAME	READY	STATUS	RESTARTS	AGE
trident-controller-79df798bdc-m79dc	6/6	Running	0	150d
trident-node-linux-xrst8	2/2	Running	0	150d
trident-operator-5574dbbc68-nthjv	1/1	Running	0	1m30s

4. Crea un TridentOrchestrator CR para instalar Astra Trident.

```
cat deploy/crds/tridentorchestrator_cr.yaml
```

```
apiVersion: trident.netapp.io/v1
```

```
kind: TridentOrchestrator
```

```
metadata:
```

```
  name: trident
```

```
spec:
```

```
  debug: true
```

```
  namespace: trident
```

```
kubectl create -f deploy/crds/tridentorchestrator_cr.yaml
```

```
#Examine the pods in the Trident namespace
```

NAME	READY	STATUS	RESTARTS	AGE
trident-csi-79df798bdc-m79dc	6/6	Running	0	1m
trident-csi-xrst8	2/2	Running	0	1m
trident-operator-5574dbbc68-nthjv	1/1	Running	0	5m41s

5. Confirmar que Trident se ha actualizado a la versión prevista.

```
kubectl describe torc trident | grep Message -A 3
```

```
Message:          Trident installed
Namespace:        trident
Status:           Installed
Version:          v24.06.0
```

Actualice con tridentctl

Puede actualizar fácilmente una instalación existente de Astra Trident con `tridentctl`.

Acerca de esta tarea

La desinstalación y reinstalación de Astra Trident actúa como una actualización. Cuando desinstala Trident, la reclamación de volumen persistente (PVC) y el volumen persistente (PV) que utiliza la puesta en marcha de Astra Trident no se eliminan. Las RVP que ya se han aprovisionado seguirán disponibles mientras Astra Trident está offline y Astra Trident aprovisiona volúmenes para cualquier RVP que se crean interanualmente una vez que vuelve a estar online.

Antes de empezar

Revise ["Seleccione un método de actualización"](#) antes de actualizar con `tridentctl`.

Pasos

1. Ejecute el comando `uninstall` en `tridentctl` para quitar todos los recursos asociados con Astra Trident, excepto los CRD y los objetos relacionados.

```
./tridentctl uninstall -n <namespace>
```

2. Vuelva a instalar Astra Trident. Consulte ["Instalar Astra Trident mediante tridentctl"](#).



No interrumpa el proceso de actualización. Asegúrese de que el instalador se ejecuta hasta su finalización.

Gestión de Astra Trident con tridentctl

```
https://github.com/NetApp/trident/releases["Paquete de instalación de Trident"] Incluye la `tridentctl` utilidad de línea de comandos para proporcionar un acceso simple a Astra Trident. Los usuarios de Kubernetes que cuentan con suficientes privilegios pueden utilizarlo para instalar Astra Trident o gestionar el espacio de nombres que contiene el pod de Astra Trident.
```

Comandos e indicadores globales

Puede ejecutar `tridentctl help` para obtener una lista de comandos disponibles `tridentctl` o agregar el `--help` indicador a cualquier comando para obtener una lista de opciones y indicadores para ese comando específico.

```
tridentctl [command] [--optional-flag]
```

La utilidad Astra Trident `tridentctl` admite los siguientes comandos y avisos globales.

Comandos

create

Añade un recurso a Astra Trident.

delete

Quita uno o varios recursos de Astra Trident.

get

Obtén uno o más recursos de Astra Trident.

help

Ayuda sobre cualquier comando.

images

Imprime una tabla de las imágenes de contenedores que Astra Trident necesita.

import

Importar un recurso existente a Astra Trident.

install

Instala Astra Trident.

logs

Imprime los registros desde Astra Trident.

send

Enviar un recurso desde Astra Trident.

uninstall

Desinstale Astra Trident.

update

Modificar un recurso en Astra Trident.

update backend state

Suspender temporalmente las operaciones de backend.

upgrade

Actualiza un recurso en Astra Trident.

version

Imprime la versión de Astra Trident.

Indicadores globales

-d, --debug

Salida de depuración.

-h, --help

Ayuda para `tridentctl`.

-k, --kubeconfig string

Especifique `KUBECONFIG` la ruta para ejecutar comandos localmente o desde un clúster de Kubernetes a otro.



También puede exportar la `KUBECONFIG` variable para que apunte a un clúster de Kubernetes específico y emitir `tridentctl` comandos a ese clúster.

-n, --namespace string

Espacio de nombres de puesta en marcha de Astra Trident.

-o, --output string

Formato de salida. Uno de `json|yaml|name|Wide|ps` (predeterminado).

-s, --server string

Dirección/puerto de la interfaz REST DE Astra Trident.



La interfaz DE REST de Trident se puede configurar para escuchar y servir únicamente en `127.0.0.1` (para IPv4) o `:::1` (para IPv6).

Opciones de comando y indicadores

cree

Utilice `create` el comando para añadir un recurso a Astra Trident.

```
tridentctl create [option]
```

Opciones

`backend`: Añadir un backend a Astra Trident.

eliminar

Usa `delete` el comando para quitar uno o varios recursos de Astra Trident.

```
tridentctl delete [option]
```

Opciones

`backend`: Eliminar uno o varios back-ends de almacenamiento de Astra Trident.

`snapshot`: Elimina una o varias instantáneas de volumen de Astra Trident.

`storageclass`: Elimina una o varias clases de almacenamiento de Astra Trident.

volume: Elimina uno o varios volúmenes de almacenamiento de Astra Trident.

obtenga

Usa `get` el comando para obtener uno o varios recursos de Astra Trident.

```
tridentctl get [option]
```

Opciones

backend: Obtenga uno o varios back-ends de almacenamiento de Astra Trident.

snapshot: Obtenga una o más instantáneas de Astra Trident.

storageclass: Obtenga una o varias clases de almacenamiento de Astra Trident.

volume: Obtenga uno o varios volúmenes de Astra Trident.

Indicadores

-h, --help: Ayuda para volúmenes.

--parentOfSubordinate string: Limite la consulta al volumen fuente subordinado.

--subordinateOf string: Limite la consulta a los subordinados de volumen.

imágenes

Utiliza `images` banderas para imprimir una tabla de las imágenes de contenedores que Astra Trident necesita.

```
tridentctl images [flags]
```

Indicadores

-h, --help: Ayuda para imágenes.

-v, --k8s-version string: Versión semántica del clúster de Kubernetes.

importe volumen

Use `import volume` el comando para importar un volumen existente a Astra Trident.

```
tridentctl import volume <backendName> <volumeName> [flags]
```

Alias

volume, v

Indicadores

-f, --filename string: Ruta a YAML o archivo JSON PVC.

-h, --help: Ayuda para el volumen.

--no-manage: Crear PV/PVC solamente. No asuma que se gestiona el ciclo de vida de los volúmenes.

instale

Utiliza `install` los indicadores para instalar Astra Trident.

```
tridentctl install [flags]
```

Indicadores

`--autosupport-image string`: La imagen del contenedor para la telemetría AutoSupport (por defecto "NetApp/Trident AutoSupport:<current-version>").

`--autosupport-proxy string`: La dirección/puerto de un proxy para enviar telemetría AutoSupport.

`--enable-node-prep`: Intenta instalar los paquetes necesarios en los nodos.

`--generate-custom-yaml`: Generar archivos YAML sin instalar nada.

`-h, --help`: Ayuda para la instalación.

`--http-request-timeout`: Sustituya el tiempo de espera de la solicitud HTTP para la API REST del controlador Trident (por defecto 1m30s).

`--image-registry string`: La dirección/puerto de un registro interno de imágenes.

`--k8s-timeout duration`: El tiempo de espera para todas las operaciones de Kubernetes (por defecto 3m0s).

`--kubelet-dir string`: La ubicación del anfitrión del estado interno de kubelet (por defecto "/var/lib/kubelet").

`--log-format string`: El formato de registro de Astra Trident (texto, json) (texto predeterminado).

`--pv string`: El nombre del PV heredado utilizado por Astra Trident, se asegura de que no exista (por defecto, «Trident»).

`--pvc string`: El nombre del PVC heredado utilizado por Astra Trident, se asegura de que esto no exista (por defecto "Trident").

`--silence-autosupport`: No enviar paquetes AutoSupport a NetApp automáticamente (predeterminado true).

`--silent`: Deshabilitar la mayoría de la salida durante la instalación.

`--trident-image string`: La imagen de Astra Trident para instalar.

`--use-custom-yaml`: Utilice cualquier archivo YAML existente que exista en el directorio de configuración.

`--use-ipv6`: Utilice IPv6 para la comunicación de Astra Trident.

registros

Utiliza `logs` marcas para imprimir los registros de Astra Trident.

```
tridentctl logs [flags]
```

Indicadores

`-a, --archive`: Crear un archivo de soporte con todos los registros a menos que se especifique lo contrario.

`-h, --help`: Ayuda para registros.

`-l, --log string`: Registro de Astra Trident para mostrar. Uno de Trident|auto|Trident-operator|all (automático por defecto).

`--node string`: El nombre del nodo de Kubernetes desde el que se recopilan los registros de pod de nodo.

`-p, --previous`: Obtenga los logs de la instancia de contenedor anterior si existe.

`--sidecars`: Obtenga los registros para los contenedores sidecar.

enviar

Utilice `send` el comando para enviar un recurso desde Astra Trident.

```
tridentctl send [option]
```

Opciones

`autosupport`: Enviar un archivo AutoSupport a NetApp.

desinstalar

Utiliza `uninstall` flags para desinstalar Astra Trident.

```
tridentctl uninstall [flags]
```

Indicadores

- `-h, --help`: Ayuda para desinstalar.
- `--silent`: Deshabilitar la mayoría de la salida durante la desinstalación.

actualizar

Utilice `update` el comando para modificar un recurso en Astra Trident.

```
tridentctl update [option]
```

Opciones

`backend`: Actualizar un backend en Astra Trident.

actualizar estado de backend

Utilice `update backend state` el comando para suspender o reanudar operaciones de back-end.

```
tridentctl update backend state <backend-name> [flag]
```

Puntos que considerar

- Si se crea un backend con un `TridentBackendConfig` (tbc), el backend no se puede actualizar con un `backend.json` archivo.
- Si el `userState` se ha establecido en una tbc, no se puede modificar mediante el `tridentctl update backend state <backend-name> --user-state suspended/normal` comando.
- Para recuperar la capacidad de establecer el `userState` `tridentctl` vía una vez que se ha establecido a través de tbc, el `userState` campo debe eliminarse del tbc. Esto se puede hacer usando `kubectyl edit tbc` el comando. Una vez eliminado el `userState` campo, puede utilizar `tridentctl update backend state` el comando para cambiar el `userState` de un backend.
- Utilice el `tridentctl update backend state` para cambiar la `userState`. También puede actualizar el `userState` archivo Using `TridentBackendConfig` OR `backend.json`; esto desencadena una reinicialización completa del backend y puede llevar mucho tiempo.

Indicadores

- `-h, --help`: Ayuda para el estado de backend.
- `--user-state`: Establecer `suspended` para pausar las operaciones de backend. Establezca esta opción `normal` para reanudar las operaciones de backend. Cuando se establece en `suspended`:

- `AddVolume Import Volume` y se ponen en pausa.
- `CloneVolume, , , ResizeVolume PublishVolume UnPublishVolume, , CreateSnapshot, GetSnapshot RestoreSnapshot, , , DeleteSnapshot RemoveVolume, , GetVolumeExternal,`

`ReconcileNodeAccess` seguir estando disponible.

También puede actualizar el estado del backend utilizando `userState` el campo en el archivo de configuración de backend `TridentBackendConfig` o `backend.json`. Para obtener más información, consulte ["Opciones para gestionar back-ends"](#) y ["Realice la gestión del entorno de administración con kubectl"](#)

Ejemplo:

JSON

Siga estos pasos para actualizar el `userState` utilizando el `backend.json` archivo:

1. Edite el `backend.json` archivo para incluir el `userState` campo con su valor establecido en 'SUSPENDED'.
2. Actualice el backend con el `tridentctl backend update` comando y la ruta de acceso al archivo actualizado `backend.json`.

Ejemplo: `tridentctl backend update -f /<path to backend JSON file>/backend.json`

```
{
  "version": 1,
  "storageDriverName": "ontap-nas",
  "managementLIF": "<redacted>",
  "svm": "nas-svm",
  "backendName": "customBackend",
  "username": "<redacted>",
  "password": "<redacted>",
  "userState": "suspended",
}
```

YAML

Puede editar el tbc después de que se haya aplicado con el `kubectl edit <tbc-name> -n <namespace>` comando. En el ejemplo siguiente se actualiza el estado del back-end para suspender con la `userState: suspended` opción:

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-ontap-nas
spec:
  version: 1
  backendName: customBackend
  storageDriverName: ontap-nas
  managementLIF: <redacted>
  svm: nas-svm
  userState: suspended
  credentials:
    name: backend-tbc-ontap-nas-secret
```

versión

Utilice `version` indicadores para imprimir la versión de `tridentctl` y el servicio Trident en ejecución.

```
tridentctl version [flags]
```

Indicadores

- `--client`: Solo versión de cliente (no se requiere servidor).
- `-h`, `--help`: Ayuda para la versión.

Supervisión de Astra Trident

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

Descripción general

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 muestran en el puerto de destino 8001 en el `/metrics` punto final. 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"](#) o usted puede elegir ejecutar Prometeo como un ["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. Esto ["blog"](#) explica cómo puedes usar Prometheus y Grafana con Astra Trident para recuperar métricas. El blog explica cómo puede ejecutar Prometheus como operador en su clúster de Kubernetes y la creación de un ServiceMonitor para obtener métricas de Astra Trident.

Paso 2: Cree un Prometheus ServiceMonitor

Para consumir las métricas de Trident, debe crear un ServiceMonitor de Prometheus que vigile `trident-csi` el servicio y escuche en 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 el `trident-csi` servicio y busca específicamente el `metrics` punto final del servicio. Como resultado, Prometheus ahora está configurado para comprender las métricas de Astra Trident.

Además de las métricas disponibles directamente desde Astra Trident, kubelet expone muchas `kubelet_volume_*` métricas a través de su propio punto final 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 evitar que Astra Trident envíe métricas de Prometheus e información de back-end básica a NetApp, pase el `--silence-autosupport` indicador durante la instalación de Astra Trident.
- Astra Trident también puede enviar registros de contenedor a Soporte de NetApp bajo demanda a través de `tridentctl send autosupport`de``. Deberá activar Astra Trident para cargar los registros. Antes de enviar registros, debe aceptar NetApp <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.
- Puede especificar el plazo de retención del registro con `--since` el indicador. Por ejemplo `tridentctl send autosupport --since=1h:`. Esta información se recopila y se envía a través de un `trident-autosupport` contenedor instalado junto con 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. Incluye una ["CLUF"](#) que no es aplicable a la propia imagen del contenedor de 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:
    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 de contenedor, puede utilizar el `--image-registry` indicador.
- También puede configurar direcciones URL proxy generando los archivos YLMA de instalación. Esto se puede hacer `tridentctl install --generate-custom-yaml` usando para crear los archivos YAML y agregando el `--proxy-url` argumento para el `trident-autosupport` contenedor en `trident-deployment.yaml`.

Deshabilite las métricas de Astra Trident

Para **desactivar las métricas** de ser reportadas, debe generar YAML personalizados (usando el `--generate-custom-yaml` indicador) y editarlos para eliminar el `--metrics` indicador de ser invocado para el `trident-main` contenedor.

Desinstale Astra Trident

Debe utilizar el mismo método para desinstalar Astra Trident que utilizó para instalar Astra Trident.

Acerca de esta tarea

- Si necesita una corrección de los errores observados después de una actualización, problemas de dependencia o una actualización fallida o incompleta, debe desinstalar Astra Trident y volver a instalar la versión anterior siguiendo las instrucciones específicas para ese ["versión"](#). Esta es la única forma recomendada de *downgrade* a una versión anterior.
- Para facilitar la actualización y la reinstalación, la desinstalación de Astra Trident no quita los CRD ni los objetos relacionados que ha creado Astra Trident. Si necesita eliminar por completo Astra Trident y todos sus datos, consulte ["Elimina por completo Astra Trident y CRD"](#)el .

Antes de empezar

Si va a decomisionar clústeres de Kubernetes, debe eliminar todas las aplicaciones que usan volúmenes creados por Astra Trident antes de desinstalar. De este modo se garantiza la eliminación de las RVP en los nodos de Kubernetes antes de que se eliminen.

Determine el método de instalación original

Debe usar el mismo método para desinstalar Astra Trident que utilizó para instalarlo. Antes de la desinstalación, compruebe la versión que usaba para instalar Astra Trident originalmente.

1. Se utiliza `kubectl get pods -n trident` para examinar los pods.
 - Si no hay ningún pod de operador, se instaló Astra Trident mediante `tridentctl`.
 - Si hay un pod de operador, se instaló Astra Trident mediante el operador Trident manualmente o mediante Helm.
2. Si hay un pod de operador, utilícelo `kubectl describe tproc trident` para determinar si Astra Trident se instaló mediante Helm.
 - Si hay una etiqueta Helm, Astra Trident se instaló usando Helm.
 - Si no hay ninguna etiqueta Helm, Astra Trident se instaló manualmente mediante el operador Trident.

Desinstale una instalación del operador Trident

Puede desinstalar una instalación de operador trident manualmente o usando Helm.

Desinstale la instalación manual

Si instaló Astra Trident mediante el operador, puede desinstalarlo siguiendo una de las siguientes acciones:

1. **Editar `TridentOrchestrator` CR y establecer el indicador de desinstalación:**

```
kubectl patch torc <trident-orchestrator-name> --type=merge -p
'{"spec":{"uninstall":true}}'
```

Cuando el `uninstall` indicador está definido en `true`, el operador Trident desinstala Trident, pero no elimina el propio `TridentOrchestrator`. Debe limpiar el `TridentOrchestrator` y crear uno nuevo si desea volver a instalar Trident.

2. **Eliminar `TridentOrchestrator`:** Al eliminar el `TridentOrchestrator` CR que se utilizó para implementar Astra Trident, le indicas al operador que desinstale Trident. El operador procesa la eliminación `TridentOrchestrator` y procede a eliminar la implementación de Astra Trident y el inicio de datos, eliminando los pods de Trident que había creado como parte de la instalación.

```
kubectl delete -f deploy/<bundle.yaml> -n <namespace>
```

Desinstale la instalación de Helm

Si ha instalado Astra Trident mediante Helm, puede desinstalarlo utilizando `helm uninstall`.

```
#List the Helm release corresponding to the Astra Trident install.
helm ls -n trident
NAME                NAMESPACE      REVISION      UPDATED
STATUS              CHART           APP VERSION
trident             trident         1             2021-04-20
00:26:42.417764794 +0000 UTC deployed  trident-operator-21.07.1
21.07.1

#Uninstall Helm release to remove Trident
helm uninstall trident -n trident
release "trident" uninstalled
```

Desinstale una tridentctl instalación

Use `uninstall` el comando en `tridentctl` para quitar todos los recursos asociados con Astra Trident, excepto los CRD y los objetos relacionados:

```
./tridentctl uninstall -n <namespace>
```

Información de copyright

Copyright © 2026 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.