



Actualice Astra Trident

Astra Trident

NetApp

November 14, 2025

Tabla de contenidos

Actualice Astra Trident	1
Actualice Astra Trident	1
Seleccione una versión	1
Seleccione una opción de actualización	1
Cambios en el operador	2
Actualizar con el operador	2
Actualice una instalación de operador de Trident de ámbito de clúster	3
Actualice la instalación de un operador de ámbito de espacio de nombres	4
Actualice la instalación de un operador basado en Helm	8
Actualizar desde una instalación que no sea del operador	8
Actualice con trimentctl	10
Consideraciones antes de la actualización	10
Siguientes pasos después de la actualización	11

Actualice Astra Trident

Actualice Astra Trident

Astra Trident sigue una cadencia de lanzamientos trimestrales y ofrece cuatro lanzamientos principales cada año. Cada versión nueva se basa en las versiones anteriores, y ofrece nuevas funciones y mejoras de rendimiento, así como 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.

Seleccione una versión

Las versiones de Astra Trident siguen una fecha basada YY.MM Convención de nomenclatura, donde "YY" es los dos últimos dígitos del año y "MM" es el mes. Las versiones de puntos siguen a YY.MM.X 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 a 23.01 desde 22.01 (incluidas las versiones de puntos, como 22.01.1).
 - Si dispone de una versión anterior, debe realizar una actualización en varios pasos con la documentación de la versión correspondiente para obtener instrucciones específicas. Esto requiere que primero actualice a la versión más reciente que se ajuste a su ventana de cuatro lanzamientos. Por ejemplo, si ejecuta 18.07 y desea actualizar a la versión 20.07, siga el proceso de actualización de varios pasos como se indica a continuación:
 - a. Primera actualización de 18.07 a 19.07.
 - b. A continuación, actualice de 19.07 a 20.07.
-  • Todas las actualizaciones de las versiones 19.04 y anteriores requieren la migración de los metadatos de Astra Trident `etc` Para crear objetos. Asegúrese de consultar la documentación de la versión para saber cómo funciona la actualización.
- Al actualizar, es importante que proporcione `parameter.fsType` `pulg StorageClasses` Utilizado por Astra Trident. Puede eliminar y volver a crear `StorageClasses` sin interrumpir los volúmenes preexistentes. Este es un **requisito** para la aplicación de contextos `security` para volúmenes SAN. El directorio `sample input` contiene ejemplos, como `storage-class-basic.yaml` y [LINK:https://github.com/NetApp/trident/blob/master/trident-installer/sample-input/storage-class-samples/storage-class-bronze-default.yaml](https://github.com/NetApp/trident/blob/master/trident-installer/sample-input/storage-class-samples/storage-class-bronze-default.yaml)[`storage-class-bronze-default.yaml`]. Para obtener más información, consulte "["Problemas conocidos"](#)".

Seleccione una opción de actualización

Hay dos opciones para actualizar Astra Trident. Por lo general, utilizará la misma opción que utilizó para la instalación inicial, sin embargo, puede hacerlo "["desplazarse entre los métodos de instalación"](#)".

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

*



Las instantáneas de volumen CSI son ahora una función de GA que comienza con Kubernetes 1.20. Al actualizar Astra Trident, deben eliminarse todas las CR y CRD de snapshot alfa anteriores (clases de snapshot para volumen, instantáneas de volumen y contenido de snapshots para volumen) antes de realizar la actualización. Consulte "[este blog](#)" Comprender los pasos que implica la migración de instantáneas alfa a la especificación beta/GA.

Cambios en el operador

La versión 21.01 de Astra Trident presenta algunos cambios clave en la arquitectura al operador, es decir, los siguientes:

- El operador ahora está **ámbito de clúster**. Las instancias anteriores del operador Trident (versiones 20.04 a 20.10) eran **espacio de nombres**. Un operador con ámbito de clúster puede ser ventajoso por los siguientes motivos:
 - Responsabilidad de recursos: Ahora el operador gestiona los recursos asociados con una instalación de Astra Trident a nivel de clúster. Como parte de la instalación de Astra Trident, el operador crea y mantiene varios recursos mediante el uso `ownerReferences`. Mantenimiento `ownerReferences` En los recursos de ámbito de clúster pueden generar errores en determinados distribuidores de Kubernetes como OpenShift. Esto se mitiga con un operador con ámbito de clúster. Para la reparación automática y parches de recursos de Trident, este es un requisito esencial.
 - Limpieza durante la desinstalación: Una eliminación completa de Astra Trident requeriría que se eliminen todos los recursos asociados. Un operador con espacio de nombres puede experimentar problemas con la eliminación de recursos con ámbito de clúster (como `clusterRole`, `ClusterRoleBinding` y `PodSecurityPolicy`) y dar lugar a una limpieza incompleta. Un operador con ámbito de clúster elimina este problema. Los usuarios pueden desinstalar por completo Astra Trident e instalar de nuevo si es necesario.
- `TridentProvisioner` se sustituye ahora por `TridentOrchestrator` Como recurso personalizado utilizado para instalar y gestionar Astra Trident. Además, se introduce un nuevo campo en el `TridentOrchestrator` espec. Los usuarios pueden especificar que el espacio de nombres Trident debe instalarse o actualizarse desde mediante el `spec.namespace` campo. Puede echar un vistazo a un ejemplo "[aqui](#)".

Actualizar con el operador

Puede actualizar fácilmente una instalación existente de Astra Trident con el operador.

Antes de empezar

Para realizar la actualización con el operador, se deben cumplir las siguientes condiciones:

- Debe disponer de una instalación de Astra Trident basada en CSI. Todas las versiones de 19.07 en adelante se basan en CSI. Es posible examinar los pods en el espacio de nombres de Trident para verificarlos.
 - La asignación de nombres de POD en versiones anteriores a la 23.01 sigue a `trident-csi-*` convención.
 - La nomenclatura de POD en 23.01 y versiones posteriores utiliza: `trident-controller-<generated id>` para el pod de la controladora; `trident-node-<operating system>-<generated id>` para los pods de nodo; `trident-operator-<generated id>` para el podómetro del operador.
- Si ha desinstalado CSI Trident y los metadatos de la instalación persisten, puede realizar la actualización

mediante el operador.

- Solo debe haber una instalación de Astra Trident en todos los espacios de nombres en un clúster de Kubernetes determinado.
- Debe usar un clúster de Kubernetes en ejecución "[Una versión de Kubernetes compatible](#)".
- Si hay CRD de instantánea alfa, debe eliminarlos con `tridentctl oblivate alpha-snapshot-crd`. De esta forma se eliminan los CRD para la especificación de instantánea alfa. Para obtener información sobre las snapshots existentes que se deben eliminar/migrar, consulte ["este blog"](#).

-  • Al actualizar Trident con el operador 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. Si quiere más detalles, consulte "["Detalles del problema en GitHub"](#)".
- Si utiliza una, no utilice el operador para actualizar Trident `etcd` Versión de Trident basada en (19.04 o anterior).

Actualice una instalación de operador de Trident de ámbito de clúster

Siga estos pasos para actualizar una instalación de operador de Trident de ámbito del clúster. Todas las versiones 21.01 y posteriores de Astra Trident utilizan un operador con ámbito de clúster.

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 22.01, ejecute el siguiente comando:

```
kubectl delete -f 22.01/trident-installer/deploy/bundle.yaml -n trident
```

3. Si ha personalizado la instalación inicial mediante `TridentOrchestrator` atributos, puede editar `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 de paquete correcto para su entorno y la versión Astra Trident. Por ejemplo, si está instalando Astra Trident 23.01 para Kubernetes 1.26, ejecute el siguiente comando:

```
kubectl create -f 23.01.1/trident-installer/deploy/bundle_post_1_25.yaml -n trident
```

Trident proporciona un archivo de paquete que se puede usar para instalar el operador y crear objetos asociados para la versión de Kubernetes.



- Para los clústeres que ejecutan Kubernetes 1.24 o inferior, utilice "["bundle_pre_1_25.yaml"](#)".
- Utilice el para clústeres que ejecuten Kubernetes 1.25 o superior "["bundle_post_1_25.yaml"](#)".

Resultados

El operador de Trident identifica una instalación existente de Astra Trident y la actualiza a la misma versión que el operador.

Actualice la instalación de un operador de ámbito de espacio de nombres

Siga estos pasos para actualizar desde una instancia de Astra Trident instalada mediante el operador Namespace (versiones 20.07 a 20.10).

Pasos

- Compruebe el estado de la instalación existente de Trident. Para ello, compruebe el estado * de TridentProvisioner. El estado debe ser Installed.

```
kubectl describe tprov trident -n trident | grep Message: -A 3
Message:  Trident installed
Status:   Installed
Version:  v20.10.1
```



Si el estado muestra Updating, asegúrese de resolverlo antes de continuar. Para obtener una lista de los posibles valores de estado, consulte ["aqui"](#).

- Cree el TridentOrchestrator CRD mediante el manifiesto proporcionado con el instalador de Trident.

```
# Download the release required [23.01.1]
mkdir 23.01.1
cd 23.01.1
wget
https://github.com/NetApp/trident/releases/download/v23.01.1/trident-
installer-23.01.1.tar.gz
tar -xf trident-installer-23.01.1.tar.gz
cd trident-installer
kubectl create -f
deploy/crds/trident.netapp.io_tridentorchestrators_crd_post1.16.yaml
```

- Elimine el operador de ámbito del espacio de nombres mediante su manifiesto. Para completar este paso, necesita el archivo YLMA de paquete utilizado para desplegar el operador de ámbito de espacio de nombres de <https://github.com/NetApp/trident/tree/stable/vXX.XX> /deploy/BUNDLE.YAML donde vXX.XX es el número de versión y. BUNDLE.YAML Es el nombre del

archivo YAML del grupo.



Debe realizar los cambios necesarios en los parámetros de instalación de Trident (por ejemplo, cambiar los valores para `tridentImage`, `autosupportImage`, repositorio privado de imágenes y provisión `imagePullSecrets`) después de eliminar el operador de ámbito de espacio de nombres y antes de instalar el operador de ámbito de clúster. Para obtener una lista completa de los parámetros que se pueden actualizar, consulte la "opciones de configuración".

```
#Ensure you are in the right directory
pwd
/root/20.10.1/trident-installer

#Delete the namespace-scoped operator
kubectl delete -f deploy/<BUNDLE.YAML> -n trident
serviceaccount "trident-operator" deleted
clusterrole.rbac.authorization.k8s.io "trident-operator" deleted
clusterrolebinding.rbac.authorization.k8s.io "trident-operator" deleted
deployment.apps "trident-operator" deleted
podsecuritypolicy.policy "tridentoperatorpods" deleted

#Confirm the Trident operator was removed
kubectl get all -n trident
NAME                                     READY   STATUS    RESTARTS   AGE
pod/trident-csi-68d979fb85-dsrmn      6/6     Running   12          99d
pod/trident-csi-8jfhf                  2/2     Running   6           105d
pod/trident-csi-jtnjz                 2/2     Running   6           105d
pod/trident-csi-lcxvh                 2/2     Running   8           105d

NAME              TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)
AGE
service/trident-csi   ClusterIP   10.108.174.125 <none>
34571/TCP, 9220/TCP   105d

NAME          DESIRED  CURRENT  READY  UP-TO-DATE   AGE
AVAILABLE  NODE SELECTOR
daemonset.apps/trident-csi   3         3         3       3          3          3
kubernetes.io/arch=amd64, kubernetes.io/os=linux   105d

NAME          READY  UP-TO-DATE  AVAILABLE  AGE
deployment.apps/trident-csi   1/1     1          1          105d

NAME          DESIRED  CURRENT  READY  AGE
replicaset.apps/trident-csi-68d979fb85   1         1         1          105d
```

En esta etapa, la `trident-operator-xxxxxxxxxx-xxxx` pod se ha eliminado.

4. (Opcional) Si es necesario modificar los parámetros de instalación, actualice `TridentProvisioner` espec. Estos pueden ser cambios como modificar el registro de imágenes privadas para extraer imágenes de contenedores, habilitar registros de depuración o especificar secretos de extracción de imágenes.

```
kubectl patch tprov <trident-provisioner-name> -n <trident-namespace>  
--type=merge -p '{"spec":{"debug":true}}'
```

5. Instale el operador Trident.



La instalación del operador de ámbito del clúster inicia la migración de `TridentProvisioner` objetos a `TridentOrchestrator` objetos, eliminaciones `TridentProvisioner` los objetos y la `tridentprovisioner` CRD y actualiza Astra Trident a la versión del operador de ámbito del clúster que se utiliza. En el ejemplo siguiente, Trident se actualiza a 23.01.1.



La actualización de Astra Trident con el operador Trident provoca la migración de `tridentProvisioner` a `tridentOrchestrator` objeto con el mismo nombre. El operador lo maneja automáticamente. La actualización también tendrá instalada Astra Trident en el mismo espacio de nombres que antes.

```

#Ensure you are in the correct directory
pwd
/root/23.01.1/trident-installer

#Install the cluster-scoped operator in the **same namespace**
kubectl create -f deploy/<BUNDLE.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

#All tridentProvisioners will be removed, including the CRD itself
kubectl get tprov -n trident
Error from server (NotFound): Unable to list "trident.netapp.io/v1,
Resource=tridentprovisioners": the server could not find the requested
resource (get tridentprovisioners.trident.netapp.io)

#tridentProvisioners are replaced by tridentOrchestrator
kubectl get torc
NAME      AGE
trident   13s

#Examine Trident pods in the namespace
kubectl get pods -n trident
NAME                           READY   STATUS    RESTARTS
AGE
trident-controller-79df798bdc-m79dc   6/6     Running   0
1m41s
trident-node-linux-xrst8           2/2     Running   0
1m41s
trident-operator-5574dbbc68-nthjv   1/1     Running   0
1m52s

#Confirm Trident has been updated to the desired version
kubectl describe torc trident | grep Message -A 3
Message:          Trident installed
Namespace:        trident
Status:           Installed
Version:          v23.01.1

```



La trident-controller y los nombres de pod reflejan la convención de nomenclatura introducida en 23.01.

Actualice la instalación de un operador basado en Helm

Realice los pasos siguientes para actualizar la instalación de un operador basado en Helm.



Cuando actualice un clúster de Kubernetes de 1.24 a 1.25 o posterior que tenga instalado Astra Trident, debe actualizar `Values.yaml` para establecer `excludePodSecurityPolicy` para `true` o agregar `--set excludePodSecurityPolicy=true` para la `helm upgrade` comando antes de poder actualizar el clúster.

Pasos

1. Descargue la última versión de Astra Trident.
2. Utilice la `helm upgrade` comando donde `trident-operator-23.01.1.tgz` refleja la versión a la que desea actualizar.

```
helm upgrade <name> trident-operator-23.01.1.tgz
```

Si establece cualquier opción no predeterminada durante la instalación inicial (como especificar registros privados reflejados para imágenes Trident y CSI), utilice `--set` para asegurarse de que estas opciones están incluidas en el comando `upgrade`, de lo contrario, los valores se restablecerán a los valores predeterminados.



Por ejemplo, para cambiar el valor predeterminado de `tridentDebug`, ejecute el siguiente comando:

```
helm upgrade <name> trident-operator-23.01.1-custom.tgz --set  
tridentDebug=true
```

3. Ejecución `helm list` para comprobar que la versión de la gráfica y de la aplicación se han actualizado. Ejecución `tridentctl logs` para revisar cualquier mensaje de depuración.

Resultados

El operador de Trident identifica una instalación existente de Astra Trident y la actualiza a la misma versión que el operador.

Actualizar desde una instalación que no sea del operador

Puede actualizarlo a la versión más reciente del operador de Trident desde un `tridentctl` instalación.

Pasos

1. Descargue la última versión de Astra Trident.

```
# Download the release required [23.01.1]
mkdir 23.01.1
cd 23.01.1
wget
https://github.com/NetApp/trident/releases/download/v22.01.1/trident-
installer-23.01.1.tar.gz
tar -xf trident-installer-23.01.1.tar.gz
cd trident-installer
```

2. Cree el `tridentorchestrator` CRD del manifiesto.

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

3. Despliegue el operador.

```
#Install the cluster-scoped operator in the **same namespace**
kubectl create -f deploy/<BUNDLE.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. Cree un `TridentOrchestrator` CR para instalar Astra Trident.

```

#Create a tridentOrchestrator to initiate a Trident install
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

#Confirm Trident was upgraded to the desired version
kubectl describe torc trident | grep Message -A 3
Message:                  Trident installed
Namespace:                trident
Status:                   Installed
Version:                 v23.01.1

```

Resultados

Los back-ends y las CVP existentes están disponibles automáticamente.

Actualice con tridentctl

Puede actualizar fácilmente una instalación de Astra Trident existente mediante tridentctl.

Consideraciones antes de la actualización

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

- A partir de Trident 20.01, solo la versión en beta de "[copias de snapshot de volumen](#)" compatible. Los administradores de Kubernetes deben tener cuidado de realizar un backup o convertir de forma segura los objetos Snapshot alfa en beta para conservar las snapshots alfa heredadas.
- La versión beta de instantáneas de volumen introduce un conjunto modificado de CRD y una controladora Snapshot, que deben configurarse antes de instalar Astra Trident. "[Este blog](#)" explica los pasos que se deben seguir para migrar instantáneas de volúmenes alfa al formato beta.
- 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.



Cuando actualice Astra Trident, no interrumpa el proceso de actualización. Asegúrese de que el instalador se ejecuta hasta la finalización.

Siguientes pasos después de la actualización

Para usar el conjunto enriquecido de funciones que están disponibles en las versiones Trident más recientes (como Snapshots de volumen bajo demanda), se pueden actualizar los volúmenes mediante la `tridentctl upgrade` comando.

Si hay volúmenes heredados, debe actualizarlos de un tipo NFS/iSCSI al tipo CSI para poder utilizar el conjunto completo de nuevas funciones de Astra Trident. Un VP heredado que ha sido aprovisionado por Trident admite el conjunto tradicional de funciones.

Tenga en cuenta lo siguiente al decidir actualizar volúmenes al tipo CSI:

- Es posible que no sea necesario actualizar todos los volúmenes. Los volúmenes creados previamente seguirán siendo accesibles y funcionarán normalmente.
- Un PV se puede montar como parte de un despliegue/Statilusionados al actualizar. No es necesario que los ilusionados traigan el despliegue/StatSet.
- **No puede** conectar un PV a un pod independiente al realizar la actualización. Debe apagar el pod antes de actualizar el volumen.
- Solo puede actualizar un volumen vinculado a una RVP. Los volúmenes que no están enlazados a PVC deben eliminarse e importarse antes de actualizar.

Ejemplo de actualización de volumen

A continuación se muestra un ejemplo que muestra cómo se lleva a cabo una actualización de volumen.

1. Ejecución `kubectl get pv` Para enumerar los VP.

kubectl get pv					
NAME	STATUS	CLAIM	CAPACITY	ACCESS MODES	RECLAIM POLICY
					STORAGECLASS REASON AGE
default-pvc-1-a8475	Bound	default/pvc-1	1073741824	RWO	standard Delete 19h
default-pvc-2-a8486	Bound	default/pvc-2	1073741824	RWO	standard Delete 19h
default-pvc-3-a849e	Bound	default/pvc-3	1073741824	RWO	standard Delete 19h
default-pvc-4-a84de	Bound	default/pvc-4	1073741824	RWO	standard Delete 19h
trident	Bound	trident/trident	2Gi	RWO	standard Retain 19h

Actualmente hay cuatro VP creados por Trident 20.07 con la netapp.io/trident aprovisionador.

2. Ejecución `kubectl describe pv` Para obtener los detalles del PV.

```
kubectl describe pv default-pvc-2-a8486

Name:           default-pvc-2-a8486
Labels:         <none>
Annotations:   pv.kubernetes.io/provisioned-by: netapp.io/trident
               volume.beta.kubernetes.io/storage-class: standard
Finalizers:    [kubernetes.io/pv-protection]
StorageClass:  standard
Status:        Bound
Claim:         default/pvc-2
Reclaim Policy: Delete
Access Modes:  RWO
VolumeMode:   Filesystem
Capacity:     1073741824
Node Affinity: <none>
Message:
Source:
  Type:      NFS (an NFS mount that lasts the lifetime of a pod)
  Server:    10.xx.xx.xx
  Path:      /trid_1907_alpha_default_pvc_2_a8486
  ReadOnly:  false
```

El VP se creó mediante la netapp.io/trident provisioner y es del tipo NFS. Para admitir todas las nuevas funciones proporcionadas por Astra Trident, este PV debe actualizarse al tipo CSI.

3. Ejecute el `tridentctl upgrade volume <name-of-trident-volume>` Comando para actualizar un volumen heredado de Astra Trident a la especificación CSI.

```

./tridentctl get volumes -n trident
+-----+-----+-----+
+-----+-----+-----+
|           NAME      |  SIZE   | STORAGE CLASS | PROTOCOL |
BACKEND UUID          | STATE   | MANAGED      |
+-----+-----+-----+
+-----+-----+-----+
| default-pvc-2-a8486 | 1.0 GiB | standard     | file    | c5a6f6a4-
b052-423b-80d4-8fb491a14a22 | online | true        |
| default-pvc-3-a849e | 1.0 GiB | standard     | file    | c5a6f6a4-
b052-423b-80d4-8fb491a14a22 | online | true        |
| default-pvc-1-a8475 | 1.0 GiB | standard     | file    | c5a6f6a4-
b052-423b-80d4-8fb491a14a22 | online | true        |
| default-pvc-4-a84de | 1.0 GiB | standard     | file    | c5a6f6a4-
b052-423b-80d4-8fb491a14a22 | online | true        |
+-----+-----+-----+
+-----+-----+-----+
./tridentctl upgrade volume default-pvc-2-a8486 -n trident
+-----+-----+-----+
+-----+-----+-----+
|           NAME      |  SIZE   | STORAGE CLASS | PROTOCOL |
BACKEND UUID          | STATE   | MANAGED      |
+-----+-----+-----+
+-----+-----+-----+
| default-pvc-2-a8486 | 1.0 GiB | standard     | file    | c5a6f6a4-
b052-423b-80d4-8fb491a14a22 | online | true        |
+-----+-----+-----+
+-----+-----+-----+

```

- Ejecute un `kubectl describe pv` Para verificar que el volumen es un volumen CSI.

```

kubectl describe pv default-pvc-2-a8486
Name:           default-pvc-2-a8486
Labels:         <none>
Annotations:   pv.kubernetes.io/provisioned-by: csi.trident.netapp.io
               volume.beta.kubernetes.io/storage-class: standard
Finalizers:    [kubernetes.io/pv-protection]
StorageClass: standard
Status:        Bound
Claim:         default/pvc-2
Reclaim Policy: Delete
Access Modes:  RWO
VolumeMode:   Filesystem
Capacity:     1073741824
Node Affinity: <none>
Message:
Source:
  Type:          CSI (a Container Storage Interface (CSI) volume
source)
    Driver:       csi.trident.netapp.io
    VolumeHandle: default-pvc-2-a8486
    ReadOnly:     false
    VolumeAttributes: backendUUID=c5a6f6a4-b052-423b-80d4-
                    8fb491a14a22

internalName=trid_1907_alpha_default_pvc_2_a8486
          name=default-pvc-2-a8486
          protocol=file
Events:      <none>

```

De esta forma, puede actualizar volúmenes del tipo NFS/iSCSI que Astra Trident creó al tipo CSI por volumen.

Información de copyright

Copyright © 2025 NetApp, Inc. Todos los derechos reservados. Impreso 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.