



Gestione Astra Trident

Astra Trident

NetApp
April 16, 2024

Tabla de contenidos

- Gestione Astra Trident 1
 - Actualice Astra Trident 1
 - Actualizar con el operador 3
 - Actualice con trimentctl 11
 - Desinstale Astra Trident 14
 - Degradar Astra Trident 16

Gestione 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 recomendamos que actualice al menos una vez al año para aprovechar las nuevas funciones de Astra Trident.



La actualización a una versión que tenga cinco versiones por delante requerirá que realice una actualización en varios pasos.

Determine la versión a la que se va a actualizar

- Puede actualizar a la YY.MM suelte de la YY-1.MM release y cualquier versión intermedio. Por ejemplo, es posible realizar una actualización directa a 20.07 desde 19.07 y versiones posteriores (incluidas versiones de puntos como 19.07.1).
- Si tiene una versión anterior, debe realizar una actualización de varios pasos. Esto requiere que primero actualice a la versión más reciente que se ajuste a su ventana de cuatro lanzamientos. Por ejemplo, si está ejecutando 18.07 y desea actualizar a la versión 20.07, siga el proceso de actualización de varios pasos como se indica a continuación:
 - Primera actualización de 18.07 a 19.07. Consulte la documentación de la versión correspondiente para obtener instrucciones específicas para la actualización.
 - 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 `etcd` Para crear objetos. Asegúrese de comprobar la documentación de la versión para comprender 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 hacer cumplir "contextos de seguridad" Para volúmenes SAN. La "entrada de muestra" el directorio contiene ejemplos como `[storage-class-basic.yaml.template]` y `[storage-class-bronze-default.yaml]`. Para obtener más información, consulte "Problemas conocidos".

¿Qué ruta de actualización debo elegir?

La actualización se puede realizar mediante una de las siguientes rutas:

- Utilice el operador Trident.
- Uso `tridentctl`.



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.

Si se cumplen las siguientes condiciones, puede utilizar el operador de Trident para actualizar:

- Está ejecutando CSI Trident (19.07 y posterior).
- Tiene una versión de Trident basada en CRD (19.07 y posterior).
- Usted está **no** realizando una instalación personalizada (usando YAMLs personalizados).



Si utiliza una, no utilice el operador para actualizar Trident `etcd` Versión de Trident basada en (19.04 o anterior).

Si no desea utilizar el operador o tiene una instalación personalizada que no puede ser soportada por el operador, puede realizar la actualización mediante `tridentctl`. Este es el método preferido para actualizaciones de las versiones 19.04 y anteriores de Trident.

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 ámbito de 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` `spec`. 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 ["aquí"](#).

Obtenga más información

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

*

Actualizar con el operador

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

Lo que necesitará

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

- Debe tener una instalación de Astra Trident basada en CSI. Para comprobar si está ejecutando CSI Trident, examine los pods del espacio de nombres Trident. Si siguen la `trident-csi-*` Patrón de nombres, está ejecutando CSI Trident.
- Debe disponer de una instalación de Trident basada en CRD. Esto representa todas las versiones de la versión 19.07 y posteriores. Si tiene una instalación basada en CSI, lo más probable es que tenga una instalación basada en CRD.
- 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 que ejecute ["Una versión de Kubernetes compatible"](#).
- Si hay CRD de instantánea alfa, debe eliminarlos con `tridentctl obliviate 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 de 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"](#).

Actualice una instalación de operador de ámbito del clúster

Siga estos pasos para actualizar desde **Trident 21.01 y posterior**.

Pasos

1. Elimine el operador Trident que se ha utilizado para instalar la instancia actual de Astra Trident. Por ejemplo, si va a actualizar desde 21.01, ejecute el siguiente comando:

```
kubectrl delete -f 22.01/trident-installer/deploy/BUNDLE.YAML -n trident
```

2. Opcionalmente, puede editar la `TridentOrchestrator` Objeto creado al instalar Trident para modificar los parámetros de instalación. Esto incluye cambios como la modificación de la imagen personalizada de Trident, el registro de imágenes privadas para extraer imágenes de contenedores, la habilitación de los registros de depuración o la especificación de secretos de extracción de imágenes.
3. Instale Astra Trident con el archivo YAML de paquete correcto para su entorno y la versión Astra Trident de <https://github.com/NetApp/trident/tree/stable/vXX.XX/deploy/BUNDLE.YAML> donde `vXX.XX` es el número de versión (por ejemplo `v22.10`) y `BUNDLE.YAML` Es el nombre del archivo YAML del grupo.



- 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"`.

Por ejemplo, si está instalando Astra Trident 22.10 para Kubernetes 1.25, ejecute el siguiente comando:

```
kubectl create -f 22.10.0/trident-installer/deploy/bundle_post_1_25.yaml
-n trident
```

Como parte de este paso, el operador de Trident identificará una instalación existente de Astra Trident y la actualizará con la misma versión que el operador.

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

Para actualizar desde una instancia de Astra Trident instalada mediante el operador Namespace-scoped (versiones 20.07 a 20.10), a continuación encontrará el conjunto de pasos que hay que seguir:

Pasos

1. 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 ["aquí"](#).

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

```
# Download the release required [22.10.0]
mkdir 22.10.0
cd 22.10.0
wget
https://github.com/NetApp/trident/releases/download/v22.10.0/trident-
installer-22.10.0.tar.gz
tar -xf trident-installer-22.10.0.tar.gz
cd trident-installer
kubectl create -f
deploy/crds/trident.netapp.io_tridentorchestrators_crd_post1.16.yaml
```

3. 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 (por ejemplo `v22.10`) 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>
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
daemonset.apps/trident-csi	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-xxxxx` pod se ha eliminado.

- (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 de ámbito del clúster.



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 22.10.0.



La actualización de Astra Trident mediante el uso del operador con ámbito del clúster da como resultado 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/22.10.0/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-csi-79df798bdc-m79dc                          6/6     Running   0           1m41s
trident-csi-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:         v22.10.0

```

Actualice la instalación de un operador basado en Helm

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

Pasos

1. Descargue la última versión de Astra Trident.
2. Utilice la `helm upgrade` comando. Consulte el siguiente ejemplo:

```
helm upgrade <name> trident-operator-22.10.0.tgz
```

donde `trident-operator-22.10.0.tgz` refleja la versión a la que desea actualizar.

3. Ejecución `helm list` para comprobar que la versión de la gráfica y de la aplicación se han actualizado.



Para pasar los datos de configuración durante la actualización, utilice `--set`.

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

```
helm upgrade <name> trident-operator-22.10.0-custom.tgz --set  
tridentDebug=true
```

Si usted corre `tridentctl logs`, puede ver los mensajes de depuración.



Si establece cualquier opción no predeterminada durante la instalación inicial, asegúrese de que las opciones se incluyen en el comando `upgrade` o, de lo contrario, los valores se restablecerán a sus valores predeterminados.

Actualizar desde una instalación que no sea del operador

Si dispone de una instancia de CSI Trident que cumpla con los requisitos previos indicados anteriormente, puede actualizar a la versión más reciente del operador de Trident.

Pasos

1. Descargue la última versión de Astra Trident.

```
# Download the release required [22.10.0]  
mkdir 22.10.0  
cd 22.10.0  
wget  
https://github.com/NetApp/trident/releases/download/v22.10.0/trident-  
installer-22.10.0.tar.gz  
tar -xf trident-installer-22.10.0.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-csi-79df798bdc-m79dc        6/6     Running   0           150d
trident-csi-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:          v22.10.0
```

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 `kubect1 get pv` Para enumerar los VP.

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

```

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

```

kubect1 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.

Desinstale Astra Trident

En función de la instalación de Astra Trident, hay varias opciones para desinstalarla.

Desinstalar utilizando Helm

Si ha instalado Astra Trident mediante Helm, puede desinstalarlo mediante `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 mediante el operador Trident

Si ha instalado Astra Trident mediante el operador, puede desinstalarlo realizando una de las siguientes acciones:

- **Edición `TridentOrchestrator` Para establecer el indicador de desinstalación:** puede editar `TridentOrchestrator` y ajustar `spec.uninstall=true`. Edite el `TridentOrchestrator` CR y ajuste la `uninstall` marcar como se muestra a continuación:

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

Cuando la `uninstall` el indicador se establece en `true`, El operador Trident desinstala Trident, pero no quita el propio `TridentOrchestrator`. Debe limpiar el `TridentOrchestrator` y crear uno nuevo si desea volver a instalar Trident.

- **Eliminar `TridentOrchestrator`:** extrayendo el `TridentOrchestrator` CR utilizado para implementar Astra Trident, indica al operador que desinstale Trident. El operador procesa la eliminación de `TridentOrchestrator` Y procede a eliminar la implementación y el demonset de Astra Trident, con la eliminación de los pods de Trident que ha creado como parte de la instalación. Para eliminar completamente Astra Trident (incluidos los CRD que crea) y borrar la pizarra de forma efectiva, puede editar `TridentOrchestrator` para pasar la `wipeout` opción. Consulte el siguiente ejemplo:

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

Esto desinstala Astra Trident por completo y borra todos los metadatos relacionados con los back-ends y los volúmenes que gestiona. Las instalaciones posteriores se tratan como instalaciones frescas.



Sólo debe considerar borrar los CRD al realizar una desinstalación completa. Esta acción no se puede deshacer. **No limpie los CRD a menos que esté buscando empezar y crear una nueva instalación de Astra Trident.**

Desinstale mediante `tridentctl`

Ejecute el `uninstall` comando en `tridentctl`. A continuación, se eliminan todos los recursos asociados con Astra Trident, excepto los CRD y los objetos relacionados, lo que facilita la ejecución del instalador de nuevo para actualizar a una versión más reciente.

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

Para realizar una eliminación completa de Astra Trident, debe eliminar los finalizadores de los CRD creados por Astra Trident y eliminar los CRD.

Degradar Astra Trident

Conozca los pasos que se deben seguir para cambiar a una versión anterior de Astra Trident.

Cuándo degradar

Es posible que tenga en cuenta la degradación por varios motivos, como los siguientes:

- Planificación de contingencia
- Solución inmediata de errores observados como resultado de una actualización
- Problemas de dependencia, actualizaciones incorrectas e incompletas

Debe considerar una degradación al cambiar a una versión de Astra Trident que utiliza CRD. Como Astra Trident utiliza CRD para el mantenimiento del estado, todas las entidades de almacenamiento creadas (back-ends, clases de almacenamiento, VP y snapshots de volúmenes) tienen objetos CRD asociados en lugar de los datos escritos en la `trident PV` (utilizado por la versión anterior de Astra Trident). Las clases de almacenamiento, los back-ends y los VP recién creados se mantienen como objetos CRD.

Solo intente realizar una degradación a una versión de Astra Trident que se ejecute con CRD (19.07 y posterior). Esto garantiza que las operaciones realizadas en la versión actual de Astra Trident sean visibles una vez que se produce la degradación.

Cuando no se debe degradar

No debe degradar a una versión de Trident que utilice `etcd` mantener el estado (19.04 y anteriores). Todas las operaciones realizadas con la versión actual de Astra Trident no se reflejan después de la degradación. Los VP recién creados no se pueden utilizar al volver a una versión anterior. Los cambios que se realizan en objetos como los back-ends, VP, las clases de almacenamiento y las snapshots de volúmenes (creadas, actualizadas o eliminadas) no son visibles para Astra Trident al volver a una versión anterior. Volver a una versión anterior no interrumpe el acceso a los VP que ya se habían creado con la versión anterior, a menos que se hayan actualizado.

Proceso de degradación cuando se instala Astra Trident mediante el operador

En el caso de las instalaciones realizadas mediante el operador Trident, el proceso de degradación es diferente y no requiere el uso de `tridentctl`.

En el caso de las instalaciones realizadas mediante el operador Trident, es posible reclasificar Astra Trident a

uno de los siguientes:

- Versión que se instala mediante el operador namespace-scoped (20.07 - 20.10).
- Versión que se instala mediante el operador de ámbito del clúster (21.01 y posteriores).

Degradar al operador de ámbito del clúster

Para degradar Astra Trident a una versión que utilice el operador de ámbito del clúster, siga los pasos que se mencionan a continuación.

Pasos

1. **"Desinstale Astra Trident". No elimine los CRD a menos que desee eliminar completamente una instalación existente.**
2. El operador de Trident puede eliminarse mediante el manifiesto de operador asociado con su versión de Trident. Por ejemplo: `https://github.com/NetApp/trident/tree/stable/vXX.XX/deploy/bundle.yaml` donde `vXX.XX` es el número de versión (por ejemplo `v22.10`) y `bundle.yaml` Es el nombre del archivo YAML del grupo.
3. Continúe con la degradación instalando la versión deseada de Astra Trident. Siga la documentación para la versión deseada.

Degradar al operador de ámbito de espacio de nombres

En esta sección se resumen los pasos necesarios para la degradación a una versión de Astra Trident que está dentro del intervalo comprendido entre el 20.07 y el 20.10, que se instalará utilizando el operador de ámbito del espacio de nombres.

Pasos

1. **"Desinstale Astra Trident". No extraiga los CRD a menos que desee eliminar completamente una instalación existente.** Asegúrese de que la `tridentorchestrator` se ha eliminado.

```
#Check to see if there are any tridentorchestrators present
kubectl get torc
NAME          AGE
trident      20h

#Looks like there is a tridentorchestrator that needs deleting
kubectl delete torc trident
tridentorchestrator.trident.netapp.io "trident" deleted
```

2. El operador de Trident puede eliminarse mediante el manifiesto de operador asociado con su versión de Trident. Por ejemplo: `https://github.com/NetApp/trident/tree/stable/vXX.XX/deploy/bundle.yaml` donde `vXX.XX` es el número de versión (por ejemplo `v22.10`) y `bundle.yaml` Es el nombre del archivo YAML del grupo.
3. Elimine el `tridentorchestrator` CRD.

```
#Check to see if ``tridentorchestrators.trident.netapp.io`` CRD is present and delete it.
```

```
kubectl get crd tridentorchestrators.trident.netapp.io
```

```
NAME                                CREATED AT
tridentorchestrators.trident.netapp.io 2021-01-21T21:11:37Z
```

```
kubectl delete crd tridentorchestrators.trident.netapp.io
```

```
customresourcedefinition.apiextensions.k8s.io
"tridentorchestrators.trident.netapp.io" deleted
```

Astra Trident se ha desinstalado.

4. Continúe con la degradación instalando la versión deseada. Siga la documentación para la versión deseada.

Bajar utilizando Helm

Para degradar, utilice `helm rollback` comando. Consulte el siguiente ejemplo:

```
helm rollback trident [revision #]
```

Proceso de degradación cuando Astra Trident se instala mediante `tridentctl`

Si instaló Astra Trident mediante `tridentctl`, el proceso de degradación implica los siguientes pasos. Esta secuencia le guiará por el proceso de degradación para pasar de Astra Trident 21.07 a 20.07.



Antes de iniciar la degradación, debe tomar una instantánea del clúster de Kubernetes `etcd`. Esto le permite realizar una copia de seguridad del estado actual de los CRD de Astra Trident.

Pasos

1. Compruebe que Trident se instala mediante `tridentctl`. Si no está seguro de cómo está instalado Astra Trident, ejecute esta sencilla prueba:
 - a. Enumere los pods presentes en el espacio de nombres de Trident.
 - b. Identifique la versión de Astra Trident que se ejecuta en su clúster. Puede utilizar cualquiera de estos dos usos `tridentctl`. También puede ver la imagen utilizada en los pods de Trident.
 - c. Si **no ve** a `tridentOrchestrator`, (o) un `tridentprovisioner`, (o) un pod llamado `trident-operator-xxxxxxxx-xxxxx`, Astra Trident **está instalado** con `tridentctl`.
2. Desinstale Astra Trident con el existente `tridentctl` binario. En este caso, se desinstalará con el binario 21.07.

```

tridentctl version -n trident
+-----+-----+
| SERVER VERSION | CLIENT VERSION |
+-----+-----+
| 21.07.0       | 21.07.0       |
+-----+-----+

tridentctl uninstall -n trident
INFO Deleted Trident deployment.
INFO Deleted Trident daemonset.
INFO Deleted Trident service.
INFO Deleted Trident secret.
INFO Deleted Trident cluster role binding.
INFO Deleted Trident cluster role.
INFO Deleted Trident service account.
INFO Deleted Trident pod security policy.
podSecurityPolicy=tridentpods
INFO The uninstaller did not delete Trident's namespace in case it is
going to be reused.
INFO Trident uninstallation succeeded.

```

3. Una vez finalizado este proceso, obtenga el binario de Trident correspondiente a la versión deseada (en este ejemplo, 20.07) y utilícelo para instalar Astra Trident. Puede generar YAML personalizados para un ["instalación personalizada"](#) si es necesario.

```

cd 20.07/trident-installer/
./tridentctl install -n trident-ns
INFO Created installer service account.
serviceaccount=trident-installer
INFO Created installer cluster role.                clusterrole=trident-
installer
INFO Created installer cluster role binding.
clusterrolebinding=trident-installer
INFO Created installer configmap.                  configmap=trident-
installer
...
...
INFO Deleted installer cluster role binding.
INFO Deleted installer cluster role.
INFO Deleted installer service account.

```

Se completó el proceso de degradación.

Información de copyright

Copyright © 2024 NetApp, Inc. Todos los derechos reservados. Imprimido en EE. UU. No se puede reproducir este documento protegido por copyright ni parte del mismo de ninguna forma ni por ningún medio (gráfico, electrónico o mecánico, incluidas fotocopias, grabaciones o almacenamiento en un sistema de recuperación electrónico) sin la autorización previa y por escrito del propietario del copyright.

El software derivado del material de NetApp con copyright está sujeto a la siguiente licencia y exención de responsabilidad:

ESTE SOFTWARE LO PROPORCIONA NETAPP «TAL CUAL» Y SIN NINGUNA GARANTÍA EXPRESA O IMPLÍCITA, INCLUYENDO, SIN LIMITAR, LAS GARANTÍAS IMPLÍCITAS DE COMERCIALIZACIÓN O IDONEIDAD PARA UN FIN CONCRETO, CUYA RESPONSABILIDAD QUEDA EXIMIDA POR EL PRESENTE DOCUMENTO. EN NINGÚN CASO NETAPP SERÁ RESPONSABLE DE NINGÚN DAÑO DIRECTO, INDIRECTO, ESPECIAL, EJEMPLAR O RESULTANTE (INCLUYENDO, ENTRE OTROS, LA OBTENCIÓN DE BIENES O SERVICIOS SUSTITUTIVOS, PÉRDIDA DE USO, DE DATOS O DE BENEFICIOS, O INTERRUPTIÓN DE LA ACTIVIDAD EMPRESARIAL) CUALQUIERA SEA EL MODO EN EL QUE SE PRODUJERON Y LA TEORÍA DE RESPONSABILIDAD QUE SE APLIQUE, YA SEA EN CONTRATO, RESPONSABILIDAD OBJETIVA O AGRAVIO (INCLUIDA LA NEGLIGENCIA U OTRO TIPO), QUE SURJAN DE ALGÚN MODO DEL USO DE ESTE SOFTWARE, INCLUSO SI HUBIEREN SIDO ADVERTIDOS DE LA POSIBILIDAD DE TALES DAÑOS.

NetApp se reserva el derecho de modificar cualquiera de los productos aquí descritos en cualquier momento y sin aviso previo. NetApp no asume ningún tipo de responsabilidad que surja del uso de los productos aquí descritos, excepto aquello expresamente acordado por escrito por parte de NetApp. El uso o adquisición de este producto no lleva implícita ninguna licencia con derechos de patente, de marcas comerciales o cualquier otro derecho de propiedad intelectual de NetApp.

Es posible que el producto que se describe en este manual esté protegido por una o más patentes de EE. UU., patentes extranjeras o solicitudes pendientes.

LEYENDA DE DERECHOS LIMITADOS: el uso, la copia o la divulgación por parte del gobierno están sujetos a las restricciones establecidas en el subpárrafo (b)(3) de los derechos de datos técnicos y productos no comerciales de DFARS 252.227-7013 (FEB de 2014) y FAR 52.227-19 (DIC de 2007).

Los datos aquí contenidos pertenecen a un producto comercial o servicio comercial (como se define en FAR 2.101) y son propiedad de NetApp, Inc. Todos los datos técnicos y el software informático de NetApp que se proporcionan en este Acuerdo tienen una naturaleza comercial y se han desarrollado exclusivamente con fondos privados. El Gobierno de EE. UU. tiene una licencia limitada, irrevocable, no exclusiva, no transferible, no sublicenciable y de alcance mundial para utilizar los Datos en relación con el contrato del Gobierno de los Estados Unidos bajo el cual se proporcionaron los Datos. Excepto que aquí se disponga lo contrario, los Datos no se pueden utilizar, desvelar, reproducir, modificar, interpretar o mostrar sin la previa aprobación por escrito de NetApp, Inc. Los derechos de licencia del Gobierno de los Estados Unidos de América y su Departamento de Defensa se limitan a los derechos identificados en la cláusula 252.227-7015(b) de la sección DFARS (FEB de 2014).

Información de la marca comercial

NETAPP, el logotipo de NETAPP y las marcas que constan en <http://www.netapp.com/TM> son marcas comerciales de NetApp, Inc. El resto de nombres de empresa y de producto pueden ser marcas comerciales de sus respectivos propietarios.