



# **Replicar volúmenes mediante SnapMirror**

Astra Trident

NetApp

January 14, 2026

# Tabla de contenidos

- Replicar volúmenes mediante SnapMirror . . . . . 1
  - Requisitos previos de replicación . . . . . 1
  - Cree una RVP reflejada . . . . . 1
  - Estados de replicación de volúmenes . . . . . 4
  - Promocione la RVP secundaria durante una conmutación al respaldo no planificada . . . . . 5
  - Promocione la RVP secundaria durante una conmutación al respaldo planificada . . . . . 5
  - Restaurar una relación de mirroring después de una conmutación al nodo de respaldo . . . . . 5
  - Operaciones adicionales . . . . . 6
    - Replica la PVC primaria a una nueva PVC secundaria . . . . . 6
    - Cambie el tamaño de una RVP reflejada, primaria o secundaria . . . . . 6
    - Elimine la replicación de una RVP . . . . . 6
    - Eliminar una RVP (que se había duplicado previamente) . . . . . 6
    - Eliminar un TMR . . . . . 6
  - Actualice las relaciones de reflejo cuando el ONTAP esté en línea . . . . . 7
  - Actualice las relaciones de reflejo cuando la ONTAP esté sin conexión . . . . . 7
  - Habilita el proveedor de Astra Control . . . . . 7

# Replicar volúmenes mediante SnapMirror

Con Astra Control Provisioning, puede crear relaciones de mirroring entre un volumen de origen en un clúster y el volumen de destino en el clúster con relación de paridad para replicar datos para la recuperación de desastres. Puede utilizar una definición de recursos personalizados (CRD) con nombre para realizar las siguientes operaciones:

- Crear relaciones de mirroring entre volúmenes (RVP)
- Elimine las relaciones de reflejo entre volúmenes
- Rompa las relaciones de reflejo
- Promocionar el volumen secundario durante condiciones de desastre (conmutaciones al respaldo).
- Realice una transición de las aplicaciones sin pérdidas de un clúster a otro (durante las migraciones y las conmutaciones al respaldo planificadas).

## Requisitos previos de replicación

Asegúrese de que se cumplen los siguientes requisitos previos antes de comenzar:

### Clústeres ONTAP

- **Astra Control Provisionador:** Astra Control Provisionador versión 23,10 o posterior debe existir en los clústeres de Kubernetes de origen y destino que utilizan ONTAP como backend.
- **Licencias:** Las licencias asíncronas de SnapMirror de ONTAP que utilizan el paquete de protección de datos deben estar habilitadas en los clústeres de ONTAP de origen y de destino. Consulte ["Información general sobre las licencias de SnapMirror en ONTAP"](#) si desea obtener más información.

### Interconexión

- **Cluster y SVM:** Los back-ends de almacenamiento ONTAP deben ser peered. Consulte ["Información general sobre relaciones entre iguales de clústeres y SVM"](#) si desea obtener más información.



Compruebe que los nombres de las SVM utilizados en la relación de replicación entre dos clústeres de ONTAP sean únicos.

- **Astra Control Provisionador y SVM:** Las SVM remotas entre iguales deben estar disponibles para Astra Control Provisionador en el clúster de destino.

### Controladores compatibles

- La replicación de volúmenes es compatible con los controladores ontap-nas y ontap-san.

## Cree una RVP reflejada

Siga estos pasos y utilice los ejemplos de CRD para crear una relación de reflejo entre los volúmenes primario y secundario.

### Pasos

1. Realice los siguientes pasos en el clúster de Kubernetes principal:
  - a. Cree un objeto StorageClass con el `trident.netapp.io/replication: true` parámetro.

### Ejemplo

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: csi-nas
provisioner: csi.trident.netapp.io
parameters:
  backendType: "ontap-nas"
  fsType: "nfs"
  trident.netapp.io/replication: "true"
```

- b. Cree una RVP con el tipo de almacenamiento creado anteriormente.

### Ejemplo

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: csi-nas
spec:
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 1Gi
  storageClassName: csi-nas
```

- c. Cree un CR de MirrorRelationship con información local.

### Ejemplo

```
kind: TridentMirrorRelationship
apiVersion: trident.netapp.io/v1
metadata:
  name: csi-nas
spec:
  state: promoted
  volumeMappings:
    - localPVCName: csi-nas
```

Astra Control Provisioner obtiene la información interna del volumen y el estado actual de protección de datos (DP) del volumen y, a continuación, rellena el campo de estado del MirrorRelationship.

- d. Obtenga el TridentMirrorRelationship CR para obtener el nombre interno y SVM de la PVC.

```
kubectl get tmr csi-nas
```

```
kind: TridentMirrorRelationship
apiVersion: trident.netapp.io/v1
metadata:
  name: csi-nas
  generation: 1
spec:
  state: promoted
  volumeMappings:
  - localPVCName: csi-nas
status:
  conditions:
  - state: promoted
    localVolumeHandle:
"datavserver:trident_pvc_3bedd23c_46a8_4384_b12b_3c38b313c1e1"
    localPVCName: csi-nas
    observedGeneration: 1
```

2. Realice los siguientes pasos en el clúster de Kubernetes secundario:

- a. Cree una StorageClass con el parámetro trident.netapp.io/replication: true.

#### Ejemplo

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: csi-nas
provisioner: csi.trident.netapp.io
parameters:
  trident.netapp.io/replication: true
```

- b. Cree un CR de MirrorRelationship con información de destino y origen.

### Ejemplo

```
kind: TridentMirrorRelationship
apiVersion: trident.netapp.io/v1
metadata:
  name: csi-nas
spec:
  state: established
  volumeMappings:
    - localPVCName: csi-nas
      remoteVolumeHandle:
        "datavserver:trident_pvc_3bedd23c_46a8_4384_b12b_3c38b313c1e1"
```

El aprovisionador de control de Astra creará una relación de SnapMirror con el nombre de la política de relaciones configurada (o predeterminado para ONTAP) e inicializarla.

- c. Crear una RVP con StorageClass creado anteriormente para que actúe como secundario (destino de SnapMirror).

### Ejemplo

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: csi-nas
  annotations:
    trident.netapp.io/mirrorRelationship: csi-nas
spec:
  accessModes:
    - ReadWriteMany
resources:
  requests:
    storage: 1Gi
  storageClassName: csi-nas
```

El aprovisionador de control de Astra comprobará el CRD de TridentMirrorRelationship y no podrá crear el volumen si la relación no existe. Si existe la relación, el aprovisionador de Astra Control se asegurará de que el nuevo volumen de FlexVol se coloque en una SVM vinculada con la SVM remota definida en MirrorRelationship.

## Estados de replicación de volúmenes

Una relación de mirroring de Trident (TMR) es un CRD que representa un extremo de una relación de replicación entre RVP. El TMR de destino tiene un estado, que le dice a Astra Control Provisioner cuál es el estado deseado. El TMR de destino tiene los siguientes estados:

- **Establecido:** El PVC local es el volumen de destino de una relación de espejo, y esta es una nueva relación.

- **Promocionado:** El PVC local es ReadWrite y montable, sin relación de espejo actualmente en vigor.
- **Reestablecido:** El PVC local es el volumen de destino de una relación de espejo y también estaba anteriormente en esa relación de espejo.
  - El estado reestablecido se debe usar si el volumen de destino alguna vez mantuvo una relación con el volumen de origen debido a que sobrescribe el contenido del volumen de destino.
  - El estado reestablecido generará un error si el volumen no mantuvo una relación anteriormente con el origen.

## Promocione la RVP secundaria durante una conmutación al respaldo no planificada

Realice el siguiente paso en el clúster de Kubernetes secundario:

- Actualice el campo `spec.state` de `TridentMirrorRelationship` a `promoted`.

## Promocione la RVP secundaria durante una conmutación al respaldo planificada

Durante una conmutación al respaldo planificada (migración), realice los siguientes pasos para promocionar la RVP secundaria:

### Pasos

1. En el clúster de Kubernetes principal, cree una snapshot de la RVP y espere hasta que se cree la snapshot.
2. En el clúster de Kubernetes principal, cree `SnapshotInfo` CR para obtener información interna.

### Ejemplo

```
kind: SnapshotInfo
apiVersion: trident.netapp.io/v1
metadata:
  name: csi-nas
spec:
  snapshot-name: csi-nas-snapshot
```

3. En el clúster de Kubernetes secundario, actualice el campo `spec.state` de `TridentMirrorRelationship` CR a `promoted` y `spec.promotedSnapshotHandle` para que sea `InternalName` de la snapshot.
4. En un clúster de Kubernetes secundario, confirme el estado (campo `status.state`) de `TridentMirrorRelationship` a `Promoted`.

## Restaurar una relación de mirroring después de una conmutación al nodo de respaldo

Antes de restaurar una relación de reflejo, elija el lado que desea realizar como el nuevo primario.

### Pasos

1. En el clúster de Kubernetes secundario, compruebe que se actualicen los valores del campo *spec.remoteVolumeHandle* del *TridentMirrorRelationship*.
2. En el clúster de Kubernetes secundario, actualice el campo *spec.mirror* de *TridentMirrorRelationship* a *reestablished*.

## Operaciones adicionales

Astra Control Provisioning admite las siguientes operaciones en los volúmenes primarios y secundarios:

### Replica la PVC primaria a una nueva PVC secundaria

Asegúrese de que ya tiene un PVC primario y un PVC secundario.

#### Pasos

1. Elimine los CRD de *PersistentVolumeClaim* y *TridentMirrorRelationship* del clúster secundario (destino) establecido.
2. Elimine el CRD de *TridentMirrorRelationship* del clúster primario (origen).
3. Cree un nuevo CRD de *TridentMirrorRelationship* en el clúster primario (de origen) para la nueva PVC secundaria (de destino) que desea establecer.

### Cambie el tamaño de una RVP reflejada, primaria o secundaria

El PVC se puede cambiar de tamaño como normal, ONTAP expandirá automáticamente cualquier flexvols de destino si la cantidad de datos excede el tamaño actual.

### Elimine la replicación de una RVP

Para eliminar la replicación, realice una de las siguientes operaciones en el volumen secundario actual:

- Elimine el *MirrorRelationship* en la RVP secundaria. Esto interrumpe la relación de replicación.
- O bien, actualice el campo *spec.state* a *Promoted*.

### Eliminar una RVP (que se había duplicado previamente)

Astra Control Provisioning comprueba si existen las RVP replicadas y libera la relación de replicación antes de intentar eliminar el volumen.

### Eliminar un TMR

Al eliminar un TMR en un lado de una relación reflejada, el TMR restante pasará al estado *Promoted* antes de que Astra Control Provisioner complete la eliminación. Si el TMR seleccionado para eliminación ya se encuentra en el estado *Promoted*, no existe ninguna relación de reflejo y el TMR se eliminará y el aprovisionador de Astra Control promoverá la RVP local a *ReadWrite*. Esta eliminación libera los metadatos de *SnapMirror* del volumen local en ONTAP. Si este volumen se utiliza en una relación de reflejo en el futuro, debe utilizar un nuevo TMR con un estado de replicación de volumen *established* al crear la nueva relación de reflejo.



# Actualice las relaciones de reflejo cuando el ONTAP esté en línea

Las relaciones de reflejos se pueden actualizar en cualquier momento una vez establecidas. Puede utilizar los `state: promoted` campos o `state: reestablished` para actualizar las relaciones. Al promocionar un volumen de destino a un volumen de ReadWrite normal, se puede usar *promotedSnapshotHandle* para especificar una snapshot específica a la que restaurar el volumen actual.

## Actualice las relaciones de reflejo cuando la ONTAP esté sin conexión

Puede utilizar un CRD para realizar una actualización de SnapMirror sin Astra Control para tener conectividad directa con el clúster de ONTAP. Consulte el siguiente formato de ejemplo de `TridentActionMirrorUpdate`:

### Ejemplo

```
apiVersion: trident.netapp.io/v1
kind: TridentActionMirrorUpdate
metadata:
  name: update-mirror-b
spec:
  snapshotHandle: "pvc-1234/snapshot-1234"
  tridentMirrorRelationshipName: mirror-b
```

`status.state` Refleja el estado del CRD `TridentActionMirrorUpdate`. Puede tomar un valor de *succeeded*, *in progress* o *failed*.

## Habilita el proveedor de Astra Control

Las versiones 23,10 y posteriores de Trident incluyen la opción de usar Astra Control Provisioning, que permite a los usuarios de Astra Control con licencia acceder a funcionalidades avanzadas de aprovisionamiento del almacenamiento. El proveedor Astra Control ofrece esta funcionalidad ampliada, además de la funcionalidad estándar basada en CSI de Astra Trident. Puedes usar este procedimiento para habilitar e instalar el proveedor de Astra Control.

Tu suscripción al servicio de Astra Control incluye automáticamente la licencia para el uso del proveedor de Astra Control.

En las próximas actualizaciones de Astra Control, el proveedor de Astra Control reemplazará a Astra Trident como proveedor de almacenamiento y orquestador y será obligatorio para su uso en Astra Control. Por este motivo, se recomienda encarecidamente que los usuarios de Astra Control habiliten el proveedor de Astra Control. Astra Trident seguirá siendo de código abierto y se seguirá lanzando, manteniendo, admitiendo y actualizando con las nuevas funciones CSI y otras de NetApp.

### ¿Cómo puedo saber si debo habilitar Astra Control Provisioner?

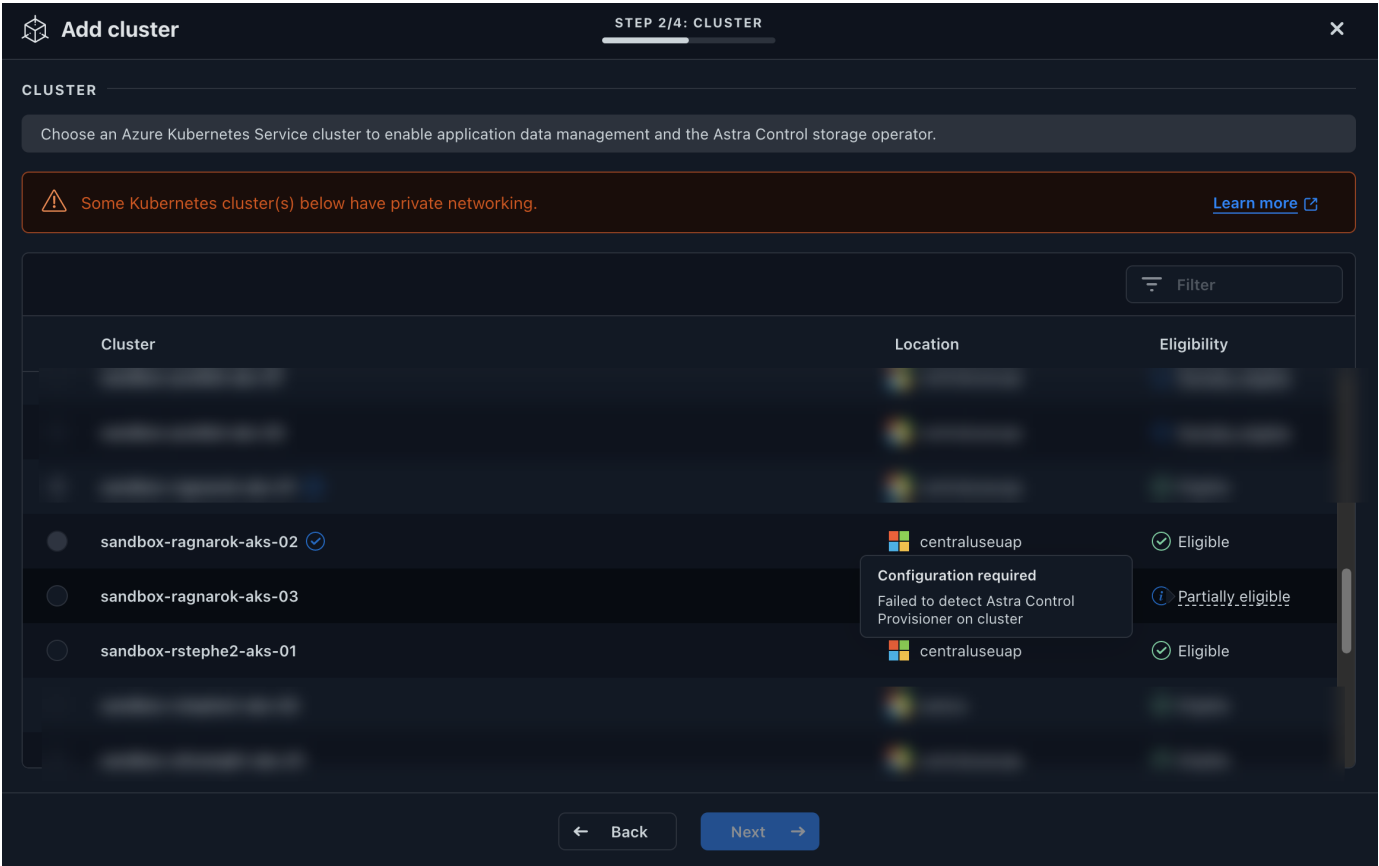
Si agrega un clúster a Astra Control Service que no tiene Astra Trident instalado previamente, el clúster se marcará como `Eligible`. Una vez que usted ["Añada el clúster a Astra Control"](#), Astra Control Provisioner se

habilitará automáticamente.

Si su clúster no está marcado `Eligible`, se marcará `Partially eligible` debido a una de las siguientes acciones:

- Está usando una versión anterior de Astra Trident
- Se utiliza un Astra Trident 23,10 que aún no tiene habilitada la opción de aprovisionador
- Se trata de un tipo de clúster que no permite la habilitación automática

En `Partially eligible` casos, sigue estas instrucciones para habilitar manualmente el aprovisionador de control Astra para tu clúster.



**Antes de habilitar Astra Control Provisioner**

Si ya tienes un Astra Trident sin el aprovisionador de Astra Control y quieres habilitar el aprovisionador de Astra Control, haz lo siguiente primero:

- **Si tienes Astra Trident instalado, confirma que su versión está dentro de una ventana de cuatro versiones:** Puedes realizar una actualización directa a Astra Trident 24,02 con el aprovisionador de control de Astra si tu Astra Trident está dentro de una ventana de cuatro versiones de la versión 24,02. Por ejemplo, puedes actualizar directamente de Astra Trident 23,04 a 24,02.
- **Confirme que su clúster tiene una arquitectura de sistema AMD64:** La imagen del aprovisionador de Astra Control se proporciona en las arquitecturas de CPU AMD64 y ARM64, pero solo AMD64 es compatible con Astra Control.

**Pasos**

1. Acceda al registro de imágenes de Astra Control de NetApp:

- a. Inicia sesión en la interfaz de usuario de Astra Control Service y registra tu ID de cuenta de Astra Control.
  - i. Seleccione el icono de figura en la parte superior derecha de la página.
  - ii. Seleccione **acceso API**.
  - iii. Escriba su ID de cuenta.
- b. En la misma página, selecciona **Generar token de API** y copia la cadena de token de API en el portapapeles y guárdalo en tu editor.
- c. Inicia sesión en el registro de Astra Control usando el método que prefieras:

```
docker login cr.astra.netapp.io -u <account-id> -p <api-token>
```

```
crane auth login cr.astra.netapp.io -u <account-id> -p <api-token>
```

2. (Solo registros personalizados) Siga estos pasos para mover la imagen a su registro personalizado. Si no está utilizando un registro, siga los pasos del operador Trident en el [siguiente sección](#).



Puede usar Podman en lugar de Docker para los siguientes comandos. Si se utiliza un entorno de Windows, se recomienda PowerShell.

## Docker

- a. Extrae la imagen del aprovisionador de Astra Control del registro:



La imagen extraída no soportará múltiples plataformas y solo soportará la misma plataforma que el host que sacó la imagen, como Linux AMD64.

```
docker pull cr.astra.netapp.io/astra/trident-acp:24.02.0
--platform <cluster platform>
```

### Ejemplo:

```
docker pull cr.astra.netapp.io/astra/trident-acp:24.02.0
--platform linux/amd64
```

- b. Etiqueta la imagen:

```
docker tag cr.astra.netapp.io/astra/trident-acp:24.02.0
<my_custom_registry>/trident-acp:24.02.0
```

- c. Introduzca la imagen en el registro personalizado:

```
docker push <my_custom_registry>/trident-acp:24.02.0
```

## Grúa

- a. Copie el manifiesto de Astra Control Provisioner en su registro personalizado:

```
crane copy cr.astra.netapp.io/astra/trident-acp:24.02.0
<my_custom_registry>/trident-acp:24.02.0
```

3. Determine si el método de instalación original de Astra Trident utilizó un.
4. Habilita el aprovisionamiento de Astra Control en Astra Trident con el método de instalación que solías originalmente:

## Operador Astra Trident

- a. ["Descarga el instalador de Astra Trident y extraígalo"](#).
- b. Complete estos pasos si todavía no ha instalado Astra Trident o si ha quitado el operador de la implementación original de Astra Trident:
  - i. Cree el CRD:

```
kubectl create -f
deploy/crds/trident.netapp.io_tridentorchestrators_crd_post1.1
6.yaml
```

- ii. Cree el espacio de nombres trident (`kubectl create namespace trident`) o confirme que el espacio de nombres trident aún existe (`kubectl get all -n trident`). Si el espacio de nombres se ha eliminado, vuelva a crearlo.

- c. Actualice Astra Trident a 24.02.0:



Para los clústeres que ejecutan Kubernetes 1,24 o una versión anterior, `bundle_pre_1_25.yaml` utilice . Para los clústeres que ejecutan Kubernetes 1,25 o posterior, utilice `bundle_post_1_25.yaml`.

```
kubectl -n trident apply -f trident-installer/deploy/<bundle-
name.yaml>
```

- d. Compruebe que Astra Trident está ejecutando:

```
kubectl get torc -n trident
```

Respuesta:

NAME	AGE
trident	21m

- e. Si tienes un registro que usa secretos, crea un secreto para extraer la imagen del proveedor de Astra Control:

```
kubectl create secret docker-registry <secret_name> -n trident
--docker-server=<my_custom_registry> --docker-username=<username>
--docker-password=<token>
```

- f. Edite el CR de TridentOrchestrator y realice las siguientes modificaciones:

```
kubectl edit torc trident -n trident
```

- i. Establezca una ubicación de registro personalizada para la imagen de Astra Trident o extráigala del registro de Astra Control (tridentImage: <my\_custom\_registry>/trident:24.02.0 o tridentImage: netapp/trident:24.02.0).
- ii. Habilite Astra Control Provisioner (enableACP: true).
- iii. Establezca la ubicación de registro personalizada para la imagen del aprovisionador de Astra Control o extráigala del registro de Astra Control (acpImage: <my\_custom\_registry>/trident-acp:24.02.0 o acpImage: cr.astra.netapp.io/astra/trident-acp:24.02.0).
- iv. Si estableció [la imagen descubre los secretos](#) anteriormente en este procedimiento, puede establecerlos aquí (imagePullSecrets: - <secret\_name>). Utilice el mismo nombre secreto que estableció en los pasos anteriores.

```
apiVersion: trident.netapp.io/v1
kind: TridentOrchestrator
metadata:
  name: trident
spec:
  debug: true
  namespace: trident
  tridentImage: <registry>/trident:24.02.0
  enableACP: true
  acpImage: <registry>/trident-acp:24.02.0
  imagePullSecrets:
    - <secret_name>
```

- g. Guarde y salga del archivo. El proceso de despliegue comenzará automáticamente.
- h. Compruebe que se han creado el operador, el despliegue y los replicaset.

```
kubectl get all -n trident
```



Solo debe haber **una instancia** del operador en un clúster de Kubernetes. No cree varias implementaciones del operador Trident de Astra.

- i. Compruebe que trident-acp el contenedor se está ejecutando y que acpVersion tiene 24.02.0 un estado de Installed:

```
kubectl get torc -o yaml
```

Respuesta:

```
status:
  acpVersion: 24.02.0
  currentInstallationParams:
    ...
    acpImage: <registry>/trident-acp:24.02.0
    enableACP: "true"
    ...
  ...
status: Installed
```

### tridentctl

- "Descarga el instalador de Astra Trident y extraígalo".
- "Si ya tiene un Astra Trident existente, desinstálelo del clúster que lo aloja".
- Instalar Astra Trident con el aprovisionador de control de Astra habilitado (`--enable-acp=true`):

```
./tridentctl -n trident install --enable-acp=true --acp
-image=mycustomregistry/trident-acp:24.02
```

- Confirme que se ha habilitado el aprovisionador de Astra Control:

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

Respuesta:

```
+-----+-----+-----+ | SERVER
VERSION | CLIENT VERSION | ACP VERSION | +-----+
+-----+-----+-----+ | 24.02.0 | 24.02.0 | 24.02.0. |
+-----+-----+-----+
```

### Timón

- Si tienes Astra Trident 23.07.1 o anterior instalado, "[desinstalar](#)" el operador y otros componentes.
- Si tu clúster de Kubernetes ejecuta la versión 1,24 o anterior, elimina psp:

```
kubectl delete psp tridentoperatorpod
```

- Añada el repositorio de Astra Trident Helm:

```
helm repo add netapp-trident https://netapp.github.io/trident-helm-chart
```

d. Actualice el gráfico Helm:

```
helm repo update netapp-trident
```

Respuesta:

```
Hang tight while we grab the latest from your chart
repositories...
...Successfully got an update from the "netapp-trident" chart
repository
Update Complete. ☐Happy Helming!☐
```

e. Enumere las imágenes:

```
./tridentctl images -n trident
```

Respuesta:

```
| v1.28.0          | netapp/trident:24.02.0|
|                  | docker.io/netapp/trident-
autosupport:24.02|
|                  | registry.k8s.io/sig-storage/csi-
provisioner:v4.0.0|
|                  | registry.k8s.io/sig-storage/csi-
attacher:v4.5.0|
|                  | registry.k8s.io/sig-storage/csi-
resizer:v1.9.3|
|                  | registry.k8s.io/sig-storage/csi-
snapshotter:v6.3.3|
|                  | registry.k8s.io/sig-storage/csi-node-
driver-registrar:v2.10.0 |
|                  | netapp/trident-operator:24.02.0 (optional)
```

f. Asegúrese de que el trident-operator 24.02.0 esté disponible:

```
helm search repo netapp-trident/trident-operator --versions
```

Respuesta:



NAME	CHART VERSION	APP VERSION	
DESCRIPTION			
netapp-trident/trident-operator	100.2402.0	24.02.0	A

g. Utilice `helm install` y ejecute una de las siguientes opciones que incluyen estos valores:

- Un nombre para la ubicación de despliegue
- La versión de Trident de Astra
- El nombre de la imagen del aprovisionador de Astra Control
- La marca para habilitar el aprovisionador
- (Opcional) Una ruta de registro local. Si está utilizando un registro local, "[Imágenes de Trident](#)" puede estar ubicado en un registro o registros diferentes, pero todas las imágenes CSI deben estar ubicadas en el mismo registro.
- El espacio de nombres de Trident

### Opciones

- Imágenes sin registro

```
helm install trident netapp-trident/trident-operator --version
100.2402.0 --set acpImage=cr.astra.netapp.io/astra/trident-
acp:24.02.0 --set enableACP=true --set operatorImage=netapp/trident-
operator:24.02.0 --set
tridentAutosupportImage=docker.io/netapp/trident-autosupport:24.02
--set tridentImage=netapp/trident:24.02.0 --namespace trident
```

- Imágenes en uno o más registros

```
helm install trident netapp-trident/trident-operator --version
100.2402.0 --set acpImage=<your-registry>:<acp image> --set
enableACP=true --set imageRegistry=<your-registry>/sig-storage --set
operatorImage=netapp/trident-operator:24.02.0 --set
tridentAutosupportImage=docker.io/netapp/trident-autosupport:24.02
--set tridentImage=netapp/trident:24.02.0 --namespace trident
```

Puede utilizar `helm list` para revisar detalles de la instalación como nombre, espacio de nombres, gráfico, estado, versión de la aplicación, y el número de revisión.

Si tiene problemas para poner en marcha Trident mediante Helm, ejecute este comando para desinstalar completamente Astra Trident:

```
./tridentctl uninstall -n trident
```

No ["Elimina por completo los CRD de Astra Trident"](#) como parte de tu desinstalación antes de intentar habilitar Astra Control Provisionador de nuevo.

Resultado

Está habilitada la funcionalidad de aprovisionamiento de Astra Control y es posible usar cualquier función disponible para la versión que esté ejecutando.

Después de instalar el aprovisionador de Astra Control, el clúster que aloja el aprovisionador en la interfaz de usuario de Astra Control mostrará un `ACP version` número de versión instalada actual y un campo en lugar de `Trident version`.

⚡ CLUSTER STATUS

✔ Available

Version  
v1.24.9+rke2r2

Managed  
2024/03/15 17:32 UTC

Kube-system namespace UID

ACP Version

Private route identifier  

...

Cloud instance  
private

Default bucket  
astra-bucket1 (inherited)

Overview

Namespaces

Storage

Activity

Si quiere más información

- ["Documentación sobre actualizaciones de Astra Trident"](#)

## 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.