



Actualiza con el operador

Trident

NetApp
October 29, 2026

Tabla de contenidos

- Actualiza con el operador 1
 - Comprende el flujo de trabajo de actualización del operador Trident 1
 - Gestión de la actualización del operador Trident 1
 - Flujo de trabajo de actualización de Operator 1
- Actualiza una instalación de Trident usando el operador Trident o Helm 2
 - Actualizar una instalación manual 2
 - Actualizar una instalación de Helm 3
 - Actualiza desde una `tridentctl` instalación a Trident operator 4

Actualiza con el operador

Comprende el flujo de trabajo de actualización del operador Trident

Antes de usar el operador Trident para actualizar Trident, debes entender los procesos en segundo plano que ocurren durante la actualización. Esto incluye cambios en el controlador Trident, el Pod de controlador y los Pods de nodo, y el nodo DaemonSet que permiten actualizaciones continuas.

Gestión de la actualización del operador Trident

Uno de los muchos "[ventajas de usar el operador Trident](#)" para instalar y actualizar Trident es el manejo automático de objetos de Trident y Kubernetes sin interrumpir los volúmenes montados existentes. De esta manera, Trident puede soportar actualizaciones con cero tiempo de inactividad o "[actualizaciones continuas](#)". En concreto, el operador de Trident se comunica con el clúster de Kubernetes para:

- Elimina y vuelve a crear el deployment del controlador Trident y el DaemonSet de nodo.
- Reemplaza el Trident Controller Pod y los Trident Node Pods por nuevas versiones.
 - Si un nodo no se actualiza, no impide que los nodos restantes se actualicen.
 - Solo los nodos con un Trident Node Pod en ejecución pueden montar volúmenes.



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

Flujo de trabajo de actualización de Operator

Cuando inicias una actualización usando el operador Trident:

1. **El Trident operator:**
 - a. Detecta la versión de Trident que tienes instalada actualmente (versión n).
 - b. Actualiza todos los objetos de Kubernetes, incluidos CRDs, RBAC y Trident SVC.
 - c. Elimina la implementación del controlador Trident para la versión n .
 - d. Crea el despliegue del controlador Trident para la versión $n+1$.
2. **Kubernetes** crea Trident Controller Pod para $n+1$.
3. **El Trident operator:**
 - a. Elimina el DaemonSet de nodo de Trident para n . El operador no espera a que termine el Pod de nodo.
 - b. Crea el Daemonset del nodo Trident para $n+1$.
4. **Kubernetes** crea Trident Node Pods en nodos que no están ejecutando Trident Node Pod n . Esto garantiza que nunca haya más de un Trident Node Pod, de cualquier versión, en un nodo.

Actualiza una instalación de Trident usando el operador Trident o Helm

Puedes actualizar Trident usando el operador Trident, ya sea manualmente o usando Helm. Puedes actualizar de una instalación del operador Trident a otra instalación del operador Trident o actualizar de una instalación de `tridentctl` a una versión del operador Trident. Revisa ["Selecciona un método de actualización"](#) antes de actualizar una instalación del operador Trident.

Actualizar una instalación manual

Puedes actualizar desde una instalación de operador Trident con alcance de clúster a otra instalación de operador Trident con alcance de clúster. Todas las versiones de Trident usan un operador con alcance de clúster.



Para actualizar desde Trident que se instaló utilizando el operador de espacio de nombres (versiones 20.07 a 20.10), usa las instrucciones de actualización para ["tu versión instalada"](#) de Trident.

Acerca de esta tarea

Trident proporciona un archivo de paquete que puedes usar para instalar el operador y crear objetos asociados para tu versión de Kubernetes.

- Para clústeres que ejecutan Kubernetes 1.25 o posterior, usa ["bundle_post_1_25.yaml"](#).

Antes de empezar

Asegúrate de estar usando un clúster de Kubernetes que esté ejecutando ["una versión compatible de Kubernetes"](#).

Pasos

1. Verifica tu versión de Trident:

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

2. Actualiza el `operator.yaml`, `tridentorchestrator_cr.yaml` y `post_1_25_bundle.yaml` con el registro y las rutas de imagen para la versión a la que estás actualizando (por ejemplo, 25.06), y el secreto correcto.
3. Elimina el operador Trident que se usó para instalar la instancia actual de Trident. Por ejemplo, si estás actualizando desde 25.02, ejecuta el siguiente comando:

```
kubectl delete -f 25.02.0/trident-installer/deploy/<bundle.yaml> -n trident
```

4. Si personalizaste tu instalación inicial usando `TridentOrchestrator` atributos, puedes editar el objeto `TridentOrchestrator` para modificar los parámetros de instalación. Esto puede incluir cambios 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.

5. Instala Trident usando el archivo YAML del paquete correcto para tu entorno, donde `<bundle.yaml>` es `bundle_pre_1_25.yaml` o `bundle_post_1_25.yaml` según tu versión de Kubernetes. Por ejemplo, si estás instalando Trident 25.06.0, ejecuta el siguiente comando:

```
kubectl create -f 25.06.0/trident-installer/deploy/<bundle.yaml> -n trident
```

6. Edita el trident torc para incluir la imagen 25.06.0.

Actualizar una instalación de Helm

Puedes actualizar una instalación de Trident Helm.



Al actualizar un clúster de Kubernetes de 1.24 a 1.25 o posterior que tenga Trident instalado, debes actualizar `values.yaml` para establecer `excludePodSecurityPolicy` en `true` o agregar `--set excludePodSecurityPolicy=true` al comando `helm upgrade` antes de poder actualizar el clúster.

Si ya actualizaste tu clúster de Kubernetes de la versión 1.24 a la 1.25 sin actualizar el helm de Trident, la actualización de helm fallará. Para que la actualización de helm funcione, realiza estos pasos como requisitos previos:

1. Instala el complemento `helm-mapkubeapis` desde <https://github.com/helm/helm-mapkubeapis>.
2. Haz una prueba en seco de la versión de Trident en el espacio de nombres donde está instalado. Esto muestra los recursos que se limpiarán.

```
helm mapkubeapis --dry-run trident --namespace trident
```

3. Realiza una ejecución completa con helm para hacer la limpieza.

```
helm mapkubeapis trident --namespace trident
```

Pasos

1. Si "[instalé Trident usando Helm](#)", puedes usar `helm upgrade trident netapp-trident/trident-operator --version 100.2606.0` para actualizar en un solo paso. Si no agregaste el repositorio de Helm o no puedes usarlo para actualizar:
 - a. Descarga la última versión de Trident desde "[la sección Assets en GitHub](#)".
 - b. Usa el `helm upgrade` comando donde `trident-operator-26.06.0.tgz` refleja la versión a la que quieres actualizar.

```
helm upgrade <name> trident-operator-26.06.0.tgz
```



Si configuras opciones personalizadas durante la instalación inicial (como especificar registros privados o reflejados para las imágenes de Trident y CSI), agrega el `helm`

upgrade comando usando `--set` para asegurarte de que esas opciones estén incluidas en el comando de actualización, de lo contrario los valores se restablecerán a los predeterminados.

2. Ejecuta `helm list` para verificar que tanto el chart como la versión de la app se hayan actualizado. Ejecuta `tridentctl logs` para revisar cualquier mensaje de depuración.

Actualiza desde una `tridentctl` instalación a Trident operator

Puedes actualizar a la última versión del operador Trident desde una `tridentctl` instalación. Los backends y PVC existentes estarán disponibles automáticamente.



Antes de cambiar entre métodos de instalación, revisa ["Cambiar entre métodos de instalación"](#).

Pasos

1. Descarga la última versión de Trident.

```
# Download the release required [26.06.0]
mkdir 26.06.0
cd 26.06.0
wget
https://github.com/NetApp/trident/releases/download/v26.06.0/trident-
installer-26.06.0.tar.gz
tar -xf trident-installer-26.06.0.tar.gz
cd trident-installer
```

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

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

3. Implementa el operador con ámbito de clúster en el mismo espacio de nombres.

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

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

#Examine the pods in the Trident namespace
NAME                                READY   STATUS    RESTARTS   AGE
trident-controller-79df798bdc-m79dc 6/6     Running   0           150d
trident-node-linux-xrst8             2/2     Running   0           150d
```

```
trident-operator-5574dbbc68-nthjv    1/1    Running    0    1m30s
```

4. Crea un `TridentOrchestrator` CR para instalar Trident.

```
cat deploy/crds/tridentorchestrator_cr.yaml
apiVersion: trident.netapp.io/v1
kind: TridentOrchestrator
metadata:
  name: trident
spec:
  debug: true
  namespace: trident

kubectl create -f deploy/crds/tridentorchestrator_cr.yaml

#Examine the pods in the Trident namespace
NAME                                READY   STATUS    RESTARTS   AGE
trident-csi-79df798bdc-m79dc        6/6     Running   0           1m
trident-csi-xrst8                    2/2     Running   0           1m
trident-operator-5574dbbc68-nthjv    1/1     Running   0          5m41s
```

5. Confirma que Trident se actualizó a la versión prevista.

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

Message:          Trident installed
Namespace:        trident
Status:           Installed
Version:          v26.06.0
```

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 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.