Instala Trident Protect

Trident

NetApp
July 01, 2026

Tabla de contenidos

Instala Trident Protect

Requisitos de Trident Protect

Empieza verificando la preparación de tu entorno operativo, clústeres de aplicaciones, aplicaciones y licencias. Asegúrate de que tu entorno cumple estos requisitos para desplegar y operar Trident Protect.

Compatibilidad de Trident Protect Kubernetes cluster

Trident Protect es compatible con una amplia gama de ofertas de Kubernetes totalmente gestionadas y autogestionadas, incluyendo:

- Amazon Elastic Kubernetes Service (EKS)
- Google Kubernetes Engine (GKE)
- Microsoft Azure Kubernetes Service (AKS)
- Red Hat OpenShift
- SUSE Harvester 1.7.0 (ONTAP iSCSI)
- SUSE Rancher
- VMware Tanzu Portfolio
- Kubernetes upstream

- Las copias de seguridad de Trident Protect solo son compatibles con los nodos de cómputo Linux. Los nodos de cómputo Windows no son compatibles para las operaciones de backup.
- Asegúrate de que el clúster en el que instales Trident Protect esté configurado con un controlador de instantáneas en funcionamiento y los CRD relacionados. Para instalar un controlador de instantáneas, consulta "estas instrucciones".
- Asegúrate de que existe al menos un VolumeSnapshotClass. Para obtener más información, consulta "VolumeSnapshotClass".
- Se requiere Helm 4.x o posterior para instalar Trident Protect.

Compatibilidad del backend de almacenamiento Trident Protect

Trident Protect es compatible con los siguientes storage backends:

- Amazon FSx for NetApp ONTAP
- Cloud Volumes ONTAP
- Matrices de almacenamiento ONTAP
- Google Cloud NetApp Volumes
- Azure NetApp Files

Asegúrate de que tu backend de almacenamiento cumple con los siguientes requisitos:

- Asegúrate de que el almacenamiento NetApp conectado al clúster está usando Trident 24.02 o una

versión más nueva (se recomienda Trident 24.10).

- Asegúrate de que tienes un backend de almacenamiento NetApp ONTAP.
- Asegúrate de que has configurado un bucket de almacenamiento de objetos para guardar los backups.
- Crea cualquier espacio de nombres de aplicaciones que planees usar para aplicaciones o para operaciones de gestión de datos de aplicaciones. Trident Protect no crea estos espacios de nombres por ti; si especificas un espacio de nombres inexistente en un recurso personalizado, la operación fallará.

Requisitos para los volúmenes de nas-economy

Trident Protect admite operaciones de backup y restauración en volúmenes nas-economy. Las snapshots, los clones y la replicación de SnapMirror en volúmenes nas-economy no son compatibles actualmente. Necesitas habilitar un directorio de snapshots para cada volumen nas-economy que planees usar con Trident Protect.

Algunas aplicaciones no son compatibles con volúmenes que utilizan un directorio de instantáneas. Para estas aplicaciones, necesitas ocultar el directorio de instantáneas ejecutando el siguiente comando en el sistema de almacenamiento ONTAP:

```
nfs modify -vserver <svm> -v3-hide-snapshot enabled
```

Puedes activar el directorio de instantáneas ejecutando el siguiente comando para cada volumen nas-economy, reemplazando `<volume-UUID>` por el UUID del volumen que quieres cambiar:

```
tridentctl update volume <volume-UUID> --snapshot-dir=true --pool-level
=true -n trident
```

Puedes habilitar los directorios de instantáneas por defecto para los nuevos volúmenes configurando la opción del backend de Trident `snapshotDir` a `true`. Los volúmenes existentes no se ven afectados.

Protegiendo datos con KubeVirt VMs

Trident Protect proporciona capacidades de congelación y descongelación del sistema de archivos para las máquinas virtuales KubeVirt durante las operaciones de protección de datos para garantizar la consistencia de los datos. El método de configuración y el comportamiento predeterminado para las operaciones de congelación de máquinas virtuales varía entre las versiones de Trident Protect, y las versiones más recientes ofrecen una configuración simplificada mediante parámetros del Helm chart.

Durante las operaciones de restauración, cualquier `VirtualMachineSnapshots` creado para una máquina virtual (VM) no se restaura.

Trident Protect 25.10 y versiones más recientes

Trident Protect congela y descongela automáticamente los sistemas de archivos KubeVirt durante las operaciones de protección de datos para garantizar la coherencia. A partir de Trident Protect 25.10, puedes desactivar este comportamiento usando el parámetro `vm.freeze` durante la instalación del chart de Helm. El parámetro está activado por defecto.

```
helm install ... --set vm.freeze=false ...
```

Trident Protect 24.10.1 a 25.06

A partir de Trident Protect 24.10.1, Trident Protect congela y descongela automáticamente los sistemas de archivos KubeVirt durante las operaciones de protección de datos. Opcionalmente, puedes desactivar este comportamiento automático usando el siguiente comando:

```
kubectl set env deployment/trident-protect-controller-manager
NEPTUNE_VM_FREEZE=false -n trident-protect
```

Trident Protect 24.10

Trident Protect 24.10 no garantiza automáticamente un estado consistente para los sistemas de archivos de KubeVirt VM durante las operaciones de protección de datos. Si quieres proteger tus datos de KubeVirt VM usando Trident Protect 24.10, necesitas habilitar manualmente la funcionalidad de congelar/descongelar para los sistemas de archivos antes de la operación de protección de datos. Esto asegura que los sistemas de archivos estén en un estado consistente.

Puedes configurar Trident Protect 24.10 para gestionar la congelación y descongelación del sistema de archivos de la máquina virtual durante las operaciones de protección de datos mediante "configurando la virtualización" y luego usar el siguiente comando:

```
kubectl set env deployment/trident-protect-controller-manager
NEPTUNE_VM_FREEZE=true -n trident-protect
```

Requisitos para SnapMirror replication

NetApp SnapMirror replication está disponible para su uso con Trident Protect para las siguientes soluciones ONTAP:

- Sistemas NetApp FAS, AFF y ASA en las instalaciones. La replicación SnapMirror con Trident protect no es compatible actualmente para los sistemas ASA r2.
- NetApp ONTAP Select
- NetApp Cloud Volumes ONTAP
- Amazon FSx for NetApp ONTAP

Requisitos del clúster ONTAP para la replicación de SnapMirror

Asegúrate de que tu clúster ONTAP cumple los siguientes requisitos si planeas usar la replicación SnapMirror:

- **NetApp Trident**: NetApp Trident debe existir en ambos clústeres Kubernetes de origen y destino que utilizan ONTAP como backend. Trident Protect admite la replicación con la tecnología NetApp SnapMirror usando clases de almacenamiento respaldadas por los siguientes controladores:
  - `ontap-nas`: NFS
  - `ontap-san`: iSCSI
  - `ontap-san`: FC
  - `ontap-san`: NVMe/TCP (requiere ONTAP versión mínima 9.15.1)
- **Licencias**: Las licencias asíncronas de ONTAP SnapMirror usando el paquete Data Protection deben estar habilitadas tanto en el clúster ONTAP de origen como en el clúster de destino. Consulta "Resumen de licencias de SnapMirror en ONTAP" para más información.

  A partir de ONTAP 9.10.1, todas las licencias se entregan como un archivo de licencia NetApp (NLF), que es un único archivo que habilita varias funciones. Consulta "Licencias incluidas con ONTAP One" para más información.

  > Solo se admite la protección asíncrona de SnapMirror.

Consideraciones de peering para la replicación SnapMirror

Asegúrate de que tu entorno cumple los siguientes requisitos si planeas usar el peering de backend de almacenamiento:

- **Clúster y SVM**: Los backends de almacenamiento de ONTAP deben estar interconectados. Consulta "Descripción general de clúster y SVM peering" para más información.

  > Asegúrate de que los nombres de SVM usados en la relación de replicación entre dos clústeres ONTAP sean únicos.

- **NetApp Trident y SVM**: Las SVM remotas peered deben estar disponibles para NetApp Trident en el clúster de destino.
- **Backends gestionados**: necesitas agregar y gestionar ONTAP storage backends en Trident Protect para crear una relación de replicación.

Configuración de Trident / ONTAP para SnapMirror replication

Trident Protect requiere que configures al menos un backend de almacenamiento que admita la replicación tanto para el clúster de origen como para el clúster de destino. Si el clúster de origen y el clúster de destino son el mismo, la aplicación de destino debería usar un backend de almacenamiento diferente al de la aplicación de origen para lograr la mejor resiliencia.

Requisitos del clúster de Kubernetes para la replicación de SnapMirror

Asegúrate de que tus clústeres Kubernetes cumplen los siguientes requisitos:

- **Accesibilidad de AppVault**: tanto el clúster de origen como el de destino deben tener acceso a la red para leer y escribir en el AppVault para la replicación de objetos de aplicación.

- **Conectividad de red**: configura las reglas del cortafuegos, los permisos de los buckets y las listas de IP permitidas para permitir la comunicación entre ambos clústeres y el AppVault a través de WANs.

> Muchos entornos empresariales aplican estrictas políticas de cortafuegos en las conexiones WAN. Verifica estos requisitos de red con tu equipo de infraestructura antes de configurar la replicación.

Instala y configura Trident Protect

Si tu entorno cumple los requisitos para Trident Protect, puedes seguir estos pasos para instalar Trident Protect en tu clúster. Puedes obtener Trident Protect de NetApp o instalarlo desde tu propio registro privado. Instalarlo desde un registro privado es útil si tu clúster no puede acceder a Internet.

Instala Trident Protect

Instala Trident Protect de NetApp

Pasos

1. Agrega el repositorio Trident Helm:

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

2. Usa Helm para instalar Trident Protect. Reemplaza `<name-of-cluster>` por un nombre de clúster, que se asignará al clúster y se usará para identificar los backups y las snapshots del clúster:

```
helm install trident-protect netapp-trident-protect/trident-protect
--set clusterName=<name-of-cluster> --version 100.2602.0 --create
-namespace --namespace trident-protect
```

3. Opcionalmente, para activar el registro de depuración (recomendado para la solución de problemas), usa:

```
helm install trident-protect netapp-trident-protect/trident-protect
--set clusterName=<name-of-cluster> --set logLevel=debug --version
100.2602.0 --create-namespace --namespace trident-protect
```

   El registro de depuración ayuda a soporte de NetApp a solucionar problemas sin requerir cambios en el nivel de registro ni la reproducción del problema.

Instala Trident Protect desde un registro privado

Puedes instalar Trident Protect desde un registro de imágenes privado si tu clúster de Kubernetes no puede acceder a Internet. En estos ejemplos, reemplaza los valores entre corchetes por información de tu entorno:

Pasos

1. Descarga las siguientes imágenes en tu máquina local, actualiza las etiquetas y luego súbelas a tu registro privado:

```
docker.io/netapp/controller:26.02.0
docker.io/netapp/restic:26.02.0
docker.io/netapp/kopia:26.02.0
docker.io/netapp/kopiablockrestore:26.02.0
docker.io/netapp/trident-autosupport:26.02.0
docker.io/netapp/exechook:26.02.0
docker.io/netapp/resourcebackup:26.02.0
docker.io/netapp/resourcerestore:26.02.0
docker.io/netapp/resourcedelete:26.02.0
docker.io/netapp/trident-protect-utils:v1.0.0
```

Por ejemplo:

```
docker pull docker.io/netapp/controller:26.02.0
```

```
docker tag docker.io/netapp/controller:26.02.0 <private-registry-url>/controller:26.02.0
```

```
docker push <private-registry-url>/controller:26.02.0
```

> Para obtener el gráfico Helm, primero descarga el gráfico Helm en una máquina con acceso a internet usando `helm pull trident-protect --version 100.2602.0 --repo https://netapp.github.io/trident-protect-helm-chart`, luego copia el archivo `trident-protect-100.2602.0.tgz` resultante a tu entorno sin conexión e instala usando `helm install trident-protect ./trident-protect-100.2602.0.tgz` en vez de la referencia del repositorio en el paso final.

2. Crea el espacio de nombres del sistema Trident Protect:

```
kubectl create ns trident-protect
```

3. Inicia sesión en el registro:

```
helm registry login <private-registry-url> -u <account-id> -p <api-token>
```

4. Crea un pull secret para usarlo en la autenticación del registro privado:

```
kubectl create secret docker-registry regcred --docker
-username=<registry-username> --docker-password=<api-token> -n
trident-protect --docker-server=<private-registry-url>
```

5. Agrega el repositorio Trident Helm:

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

6. Crea un archivo llamado `protectValues.yaml`. Asegúrate de que contiene la siguiente configuración de Trident Protect:

```
---
imageRegistry: <private-registry-url>
imagePullSecrets:
  - name: regcred
```

> Los valores `imageRegistry` y `imagePullSecrets` se aplican a todas las imágenes de componentes, incluyendo `resourcebackup` y `resourcerestore`. Si envías imágenes a una ruta de repositorio específica dentro de tu registro (por ejemplo, `example.com:443/my-repo`), incluye la ruta completa en el campo de registro. Esto asegurará que todas las imágenes se extraigan de `<private-registry-url>/<image-name>:<tag>`.

7. Usa Helm para instalar Trident Protect. Reemplaza `<name_of_cluster>` por un nombre de clúster, que se asignará al clúster y se usará para identificar los backups y las snapshots del clúster:

```
helm install trident-protect netapp-trident-protect/trident-protect
--set clusterName=<name_of_cluster> --version 100.2602.0 --create
-namespace --namespace trident-protect -f protectValues.yaml
```

8. Opcionalmente, para activar el registro de depuración (recomendado para la solución de problemas), usa:

```
helm install trident-protect netapp-trident-protect/trident-protect
--set clusterName=<name-of-cluster> --set logLevel=debug --version
100.2602.0 --create-namespace --namespace trident-protect -f
protectValues.yaml
```

El registro de depuración ayuda a soporte de NetApp a solucionar problemas sin requerir cambios en el nivel de registro ni la reproducción del problema.

Para más opciones de configuración del gráfico de Helm, incluidos los ajustes de AutoSupport y el filtrado de espacios de nombres, consulta "Personaliza la instalación de Trident Protect".

Instala el plugin CLI de Trident Protect

Puedes usar el complemento de línea de comandos de Trident Protect, que es una extensión de la utilidad Trident `tridentctl` para crear e interactuar con los recursos personalizados (CRs) de Trident Protect.

Instala el plugin CLI de Trident Protect

Antes de usar la utilidad de línea de comandos, necesitas instalarla en la máquina que usas para acceder a tu clúster. Sigue estos pasos, dependiendo de si tu máquina usa una CPU x64 o ARM.

Descarga el plugin para CPUs Linux AMD64

Pasos

1. Descarga el plugin CLI de Trident Protect:

```
curl -L -o tridentctl-protect https://github.com/NetApp/tridentctl-protect/releases/download/26.02.0/tridentctl-protect-linux-amd64
```

Descarga el plugin para CPUs Linux ARM64

Pasos

1. Descarga el plugin CLI de Trident Protect:

```
curl -L -o tridentctl-protect https://github.com/NetApp/tridentctl-protect/releases/download/26.02.0/tridentctl-protect-linux-arm64
```

Descarga el plugin para Mac AMD64 CPUs

Pasos

1. Descarga el plugin CLI de Trident Protect:

```
curl -L -o tridentctl-protect https://github.com/NetApp/tridentctl-protect/releases/download/26.02.0/tridentctl-protect-macos-amd64
```

Descarga el plugin para Mac ARM64 CPUs

Pasos

1. Descarga el plugin CLI de Trident Protect:

```
curl -L -o tridentctl-protect https://github.com/NetApp/tridentctl-protect/releases/download/26.02.0/tridentctl-protect-macos-arm64
```

1. Habilita los permisos de ejecución para el binario del plugin:

```
chmod +x tridentctl-protect
```

2. Copia el archivo binario del complemento en una ubicación que esté definida en tu variable PATH. Por ejemplo, `/usr/bin` o `/usr/local/bin` (puede que necesites privilegios elevados):

```
cp ./tridentctl-protect /usr/local/bin/
```

3. Opcionalmente, puedes copiar el binario del plugin en una ubicación de tu directorio inicial. En este caso, se recomienda asegurarte de que la ubicación sea parte de tu variable PATH:

```
cp ./tridentctl-protect ~/bin/
```

Copiar el complemento en una ubicación en tu variable PATH te permite usar el complemento escribiendo `tridentctl-protect` o `tridentctl protect` desde cualquier ubicación.

Ver la ayuda del plugin CLI de Trident

Puedes usar las funciones de ayuda integradas en el plugin para obtener ayuda detallada sobre las capacidades del plugin:

Pasos

1. Utiliza la función de ayuda para ver la guía de uso:

```
tridentctl-protect help
```

Habilitar el autocompletado de comandos

Después de que hayas instalado el complemento CLI de Trident Protect, puedes habilitar el autocompletado para ciertos comandos.

Habilita el autocompletado para la shell Bash

Pasos

1. Crea el script de finalización:

```
tridentctl-protect completion bash > tridentctl-completion.bash
```

2. Crea un nuevo directorio en tu directorio inicial para contener el script:

```
mkdir -p ~/.bash/completions
```

3. Mueve el script descargado al directorio `~/.bash/completions`:

```
mv tridentctl-completion.bash ~/.bash/completions/
```

4. Agrega la siguiente línea al archivo `~/.bashrc` en tu directorio inicial:

```
source ~/.bash/completions/tridentctl-completion.bash
```

Habilita el autocompletado para la Z shell

Pasos

1. Crea el script de finalización:

```
tridentctl-protect completion zsh > tridentctl-completion.zsh
```

2. Crea un nuevo directorio en tu directorio inicial para contener el script:

```
mkdir -p ~/.zsh/completions
```

3. Mueve el script descargado al directorio `~/.zsh/completions`:

```
mv tridentctl-completion.zsh ~/.zsh/completions/
```

4. Agrega la siguiente línea al archivo `~/.zprofile` en tu directorio inicial:

```
source ~/.zsh/completions/tridentctl-completion.zsh
```

Resultado

En tu próximo inicio de sesión en el shell, puedes usar el autocompletado de comandos con el plugin tridentctl-protect.

Personaliza la instalación de Trident Protect

Puedes personalizar la configuración predeterminada de Trident Protect para satisfacer los requisitos específicos de tu entorno.

Especifica los límites de recursos del contenedor Trident Protect

Puedes usar un archivo de configuración para especificar los límites de recursos para los contenedores de Trident Protect después de instalar Trident Protect. Configurar los límites de recursos te permite controlar cuántos recursos del clúster consumen las operaciones de Trident Protect.

Pasos

1. Crea un archivo llamado `resourceLimits.yaml`.
2. Rellena el archivo con las opciones de límite de recursos para los contenedores Trident Protect según las necesidades de tu entorno.

   El siguiente archivo de configuración de ejemplo muestra los ajustes disponibles y contiene los valores predeterminados para cada límite de recursos:

```
---
jobResources:
  defaults:
    limits:
      cpu: 8000m
      memory: 10000Mi
      ephemeralStorage: ""
    requests:
      cpu: 100m
      memory: 100Mi
      ephemeralStorage: ""
  resticVolumeBackup:
    limits:
      cpu: ""
      memory: ""
      ephemeralStorage: ""
    requests:
      cpu: ""
      memory: ""
      ephemeralStorage: ""
  resticVolumeRestore:
    limits:
      cpu: ""
      memory: ""
```

```
      ephemeralStorage: ""
    requests:
      cpu: ""
      memory: ""
      ephemeralStorage: ""
  kopiaVolumeBackup:
    limits:
      cpu: ""
      memory: ""
      ephemeralStorage: ""
    requests:
      cpu: ""
      memory: ""
      ephemeralStorage: ""
  kopiaVolumeRestore:
    limits:
      cpu: ""
      memory: ""
      ephemeralStorage: ""
    requests:
      cpu: ""
      memory: ""
      ephemeralStorage: ""
```

3. Aplica los valores del archivo `resourceLimits.yaml`:

```
helm upgrade trident-protect -n trident-protect netapp-trident-
protect/trident-protect -f resourceLimits.yaml --reuse-values
```

Personaliza las restricciones del contexto de seguridad

Puedes usar un archivo de configuración para modificar las security context constraint (SCCs) de OpenShift para los contenedores de Trident Protect después de instalar Trident Protect. Estas constraints definen restricciones de seguridad para los pods en un clúster de Red Hat OpenShift.

Pasos

1. Crea un archivo llamado `sccconfig.yaml`.
2. Agrega la opción SCC al archivo y modifica los parámetros según las necesidades de tu entorno.

   El siguiente ejemplo muestra los valores predeterminados de los parámetros para la opción SCC:

```
scc:
  create: true
  name: trident-protect-job
  priority: 1
```

Esta tabla describe los parámetros para la opción SCC:

Parámetro	Descripción	Predeterminado
crear	Determina si se puede crear un recurso SCC. Se creará un recurso SCC solo si `scc.create` está configurado en `true` y el proceso de instalación de Helm identifica un entorno OpenShift. Si no se está operando en OpenShift o si `scc.create` está configurado en `false`, no se creará ningún recurso SCC.	verdadero
nombre	Especifica el nombre del SCC.	trident-protect-job
prioridad	Define la prioridad del SCC. Los SCC con valores de prioridad más altos se evalúan antes que los que tienen valores más bajos.	1

3. Aplica los valores del archivo `sccconfig.yaml`:

```
helm upgrade trident-protect -n trident-protect netapp-trident-
protect/trident-protect -f sccconfig.yaml --reuse-values
```

Esto sustituirá los valores por defecto por los especificados en el `sccconfig.yaml` archivo.

Configura los ajustes adicionales del helm chart de Trident Protect

Puedes personalizar la configuración de AutoSupport y el filtrado de espacios de nombres para satisfacer tus requisitos específicos. La siguiente tabla describe los parámetros de configuración disponibles:

Parámetro	Tipo	Descripción
autoSupport.proxy	cadena	Configura una URL de proxy para las conexiones de NetApp AutoSupport. Usa esto para enrutar las cargas de paquetes de soporte a través de un servidor proxy. Ejemplo: `http://my.proxy.url`.

Parámetro	Tipo	Descripción
autoSupport.insecure	booleano	Omite la verificación TLS para las conexiones proxy de AutoSupport cuando se establece en `true`. Úsalo solo para conexiones proxy no seguras. (predeterminado: `false`)
autoSupport.enabled	booleano	Habilita o deshabilita las cargas diarias de paquetes de Trident Protect AutoSupport. Cuando se establece en `false`, las cargas diarias programadas se deshabilitan, pero aún puedes generar manualmente paquetes de soporte. (predeterminado: `true`)
restoreSkipNamespaceAnnotations	cadena	Lista separada por comas de anotaciones de espacios de nombres que se excluirán de las operaciones de copia de seguridad y restauración. Te permite filtrar los espacios de nombres según las anotaciones.
restoreSkipNamespaceLabels	cadena	Lista separada por comas de etiquetas de espacios de nombres que se excluirán de las operaciones de copia de seguridad y restauración. Te permite filtrar los espacios de nombres según las etiquetas.

Puedes configurar estas opciones usando un archivo de configuración YAML o indicadores de línea de comandos:

Usa el archivo YAML

Pasos

1. Crea un archivo de configuración y ponle el nombre `values.yaml`.
2. En el archivo que creaste, agrega las opciones de configuración que quieras personalizar.

```
autoSupport:
  enabled: false
  proxy: http://my.proxy.url
  insecure: true
restoreSkipNamespaceAnnotations: "annotation1,annotation2"
restoreSkipNamespaceLabels: "label1,label2"
```

3. Después de rellenar el archivo `values.yaml` con los valores correctos, aplica el archivo de configuración:

```
helm upgrade trident-protect -n trident-protect netapp-trident-
protect/trident-protect -f values.yaml --reuse-values
```

Usa el indicador CLI

Pasos

1. Usa el siguiente comando con la bandera `--set` para especificar parámetros individuales:

```
helm upgrade trident-protect -n trident-protect netapp-trident-
protect/trident-protect \
  --set autoSupport.enabled=false \
  --set autoSupport.proxy=http://my.proxy.url \
  --set-string
restoreSkipNamespaceAnnotations="{annotation1,annotation2}" \
  --set-string restoreSkipNamespaceLabels="{label1,label2}" \
  --reuse-values
```

Restringe los pods de Trident Protect a nodos específicos

Puedes usar la restricción de selección de nodos de Kubernetes nodeSelector para controlar cuáles de tus nodos pueden ejecutar pods de Trident Protect, según las etiquetas de los nodos. Por defecto, Trident Protect está restringido a los nodos que ejecutan Linux. Puedes personalizar aún más estas restricciones según lo que necesites.

Pasos

1. Crea un archivo llamado `nodeSelectorConfig.yaml`.
2. Agrega la opción nodeSelector al archivo y modifica el archivo para añadir o cambiar las etiquetas de nodo

y restringir según las necesidades de tu entorno. Por ejemplo, el siguiente archivo contiene la restricción de SO por defecto, pero también apunta a una región y un nombre de app específicos:

```
nodeSelector:
  kubernetes.io/os: linux
  region: us-west
  app.kubernetes.io/name: mysql
```

3. Aplica los valores del archivo `nodeSelectorConfig.yaml`:

```
helm upgrade trident-protect -n trident-protect netapp-trident-
protect/trident-protect -f nodeSelectorConfig.yaml --reuse-values
```

Esto reemplaza las restricciones predeterminadas por las que especificaste en el archivo `nodeSelectorConfig.yaml`.