Gestionar y supervisar Trident

Trident

NetApp
January 15, 2026

Tabla de contenidos

Gestionar y supervisar Trident

Mejora el Trident

Mejora el Trident

A partir de la versión 24.02, Trident sigue una cadencia de lanzamientos de cuatro meses, ofreciendo tres versiones principales cada año calendario. Cada nueva versión se basa en las versiones anteriores y proporciona nuevas funciones, mejoras de rendimiento, correcciones de errores y mejoras generales. Le recomendamos que actualice al menos una vez al año para aprovechar las nuevas funciones de Trident.

Consideraciones previas a la actualización

Al actualizar a la última versión de Trident, tenga en cuenta lo siguiente:

- Solo debe haber una instancia de Trident instalada en todos los espacios de nombres de un clúster de Kubernetes determinado.
- Trident 23.07 y versiones posteriores requieren instantáneas de volumen v1 y ya no admiten instantáneas alfa o beta.
- Si creó un Cloud Volumes Service para Google Cloud en el"Tipo de servicio CVS" , debes actualizar la configuración del backend para usar el `standardsw` o `zoneredundantstandardsw` Nivel de servicio al actualizar desde Trident 23.01. No actualizar el `serviceLevel` en el backend podría provocar fallos en los volúmenes. Referirse a "Ejemplos de tipos de servicio de CVS" Para más detalles.
- Al actualizar, es importante que proporcione `parameter.fsType` en `StorageClasses` Utilizado por Trident. Puedes eliminar y volver a crear `StorageClasses` sin alterar los volúmenes preexistentes.
  - Este es un **requisito** para hacer cumplir la ley. "contextos de seguridad" para volúmenes SAN.
  - El directorio https://github.com/NetApp/trident/tree/master/trident-installer/sample-input [ejemplo de entrada^] contiene ejemplos, como `storage-class-basic.yaml.templ` y enlace: `storage-class-bronze-default.yaml` .
  - Para obtener más información, consulte"Problemas conocidos" .

Paso 1: Seleccione una versión

Las versiones de Trident siguen una fecha. `YY.MM` Convención de nomenclatura, donde "YY" son los dos últimos dígitos del año y "MM" es el mes. Los lanzamientos de Dot siguen a `YY.MM.X` convención, donde "X" es el nivel de parche. Seleccionarás la versión a la que deseas actualizar en función de la versión desde la que estás actualizando.

- Puede realizar una actualización directa a cualquier versión objetivo que se encuentre dentro de un rango de cuatro versiones con respecto a su versión instalada. Por ejemplo, puede actualizar directamente desde la versión 24.06 (o cualquier versión derivada de la 24.06) a la versión 25.06.
- Si está actualizando desde una versión que no se encuentra dentro del período de cuatro versiones, realice una actualización en varios pasos. Utilice las instrucciones de actualización para"versión anterior" Estás actualizando a la versión más reciente que se ajusta al período de cuatro versiones. Por ejemplo, si está utilizando la versión 23.07 y desea actualizar a la versión 25.06:
  a. Primera actualización de la versión 23.07 a la 24.06.

b. Luego, actualice de la versión 24.06 a la 25.06.

> Al actualizar utilizando el operador Trident en OpenShift Container Platform, debe actualizar a Trident 21.01.1 o posterior. El operador Trident lanzado con la versión 21.01.0 contiene un problema conocido que se ha corregido en la versión 21.01.1. Para obtener más detalles, consulte el "Detalles del problema en GitHub" .

Paso 2: Determinar el método de instalación original

Para determinar qué versión utilizó para instalar Trident originalmente:

1. Usar `kubectl get pods -n trident` para examinar las vainas.
   - Si no hay ningún pod de operador, Trident se instaló mediante `tridentctl` .
   - Si existe un pod de operador, Trident se instaló utilizando el operador Trident , ya sea manualmente o utilizando Helm.
2. Si hay un pod de operador, utilice `kubectl describe torc` para determinar si Trident se instaló utilizando Helm.
   - Si aparece una etiqueta Helm, Trident se instaló utilizando Helm.
   - Si no hay ninguna etiqueta Helm, Trident se instaló manualmente utilizando el operador Trident .

Paso 3: Seleccione un método de actualización

Generalmente, debes actualizar utilizando el mismo método que usaste para la instalación inicial; sin embargo, puedes"cambiar entre métodos de instalación" . Existen dos opciones para actualizar Trident.

- "Actualizar usando el operador Trident"

> Le sugerimos que revise"Comprender el flujo de trabajo de actualización del operador" antes de actualizar con el operador.

*

Actualiza con el operador

Comprender el flujo de trabajo de actualización del operador

Antes de utilizar el operador Trident para actualizar Trident, debe comprender los procesos en segundo plano que ocurren durante la actualización. Esto incluye cambios en el controlador Trident , el Pod del controlador y los Pods de nodo, y el DaemonSet del nodo que habilitan las actualizaciones continuas.

Manejo de la actualización del operador Trident

Una de las muchas"Beneficios de usar el operador Trident" La instalación y actualización de Trident consiste en el manejo automático de objetos de Trident y Kubernetes sin interrumpir los volúmenes montados existentes. De esta forma, Trident puede admitir actualizaciones sin tiempo de inactividad, o"*actualizaciones continuas*" . En particular, el operador Trident se comunica con el clúster de Kubernetes para:

- Elimine y vuelva a crear el despliegue del controlador Trident y el DaemonSet del nodo.

- Reemplace los módulos Trident Controller Pod y Trident Node Pod con las nuevas versiones.
  - Si un nodo no se actualiza, esto no impide que se actualicen los nodos restantes.
  - Solo los nodos con un Trident Node Pod en ejecución pueden montar volúmenes.

Para obtener más información sobre la arquitectura Trident en el clúster de Kubernetes, consulte"Arquitectura Trident" .

Flujo de trabajo de actualización del operador

Cuando se inicia una actualización mediante el operador Trident :

1. El operador * Trident *:
   a. Detecta la versión actualmente instalada de Trident (versión *n*).
   b. Actualiza todos los objetos de Kubernetes, incluidos CRD, RBAC y Trident SVC.
   c. Elimina la implementación del controlador Trident para la versión *n*.
   d. Crea la implementación del controlador Trident para la versión *n+1*.
2. **Kubernetes** crea un Pod de Controlador Trident para *n+1*.
3. El operador * Trident *:
   a. Elimina el DaemonSet de nodo Trident para *n*. El operador no espera a que finalice el Node Pod.
   b. Crea el conjunto de demonios de nodo Trident para *n+1*.
4. **Kubernetes** crea pods de nodo Trident en nodos que no ejecutan el pod de nodo Trident *n*. Esto garantiza que nunca haya más de un Trident Node Pod, de ninguna versión, en un nodo.

Actualizar una instalación de Trident usando el operador de Trident o Helm.

Puedes actualizar Trident usando el operador Trident, ya sea manualmente o usando Helm. Puede actualizar de una instalación de operador Trident a otra instalación de operador Trident o actualizar desde una `tridentctl` Instalación en una versión de operador Trident . Revisar"Seleccione un método de actualización" antes de actualizar una instalación de operador Trident .

Actualizar una instalación manual

Puede actualizar desde una instalación de operador Trident con ámbito de clúster a otra instalación de operador Trident con ámbito de clúster. Todas las versiones de Trident utilizan un operador con ámbito de clúster.

Para actualizar desde Trident instalado mediante el operador con ámbito de espacio de nombres (versiones 20.07 a 20.10), siga las instrucciones de actualización para"tu versión instalada" del Trident.

Acerca de esta tarea

Trident proporciona un archivo de paquete que puede utilizar para instalar el operador y crear objetos asociados para su versión de Kubernetes.

- Para clústeres que ejecutan Kubernetes 1.24, utilice"bundle_pre_1_25.yaml" .

- Para clústeres que ejecutan Kubernetes 1.25 o posterior, utilice"bundle_post_1_25.yaml" .

Antes de empezar

Asegúrese de estar utilizando un clúster de Kubernetes en ejecución."una versión compatible de Kubernetes" .

Pasos

1. Verifique su versión de Trident :

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

2. Actualizar 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 está actualizando (ej. 25.06) y el secreto correcto.
3. Elimine el operador Trident que se utilizó para instalar la instancia actual de Trident . Por ejemplo, si está actualizando desde la versión 25.02, ejecute el siguiente comando:

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

4. Si personalizó su instalación inicial utilizando `TridentOrchestrator` atributos, puedes editar los `TridentOrchestrator` objeto para modificar los parámetros de instalación. Esto podría incluir cambios realizados para especificar registros de imágenes Trident y CSI duplicados para el modo sin conexión, habilitar registros de depuración o especificar secretos de extracción de imágenes.
5. Instala Trident utilizando el archivo YAML de 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 está instalando Trident 25.06.0, ejecute el siguiente comando:

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

6. Edite el torque tridente 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 la versión 1.24 a la 1.25 o posterior que tenga Trident instalado, debe actualizar el archivo values.yaml para configurar `excludePodSecurityPolicy` a `true` o añadir `--set excludePodSecurityPolicy=true` hacia `helm upgrade` comando antes de poder actualizar el clúster.

Si ya has actualizado tu clúster de Kubernetes de la versión 1.24 a la 1.25 sin actualizar el Helm de Trident , la actualización del Helm fallará. Para que la actualización de Helm se realice correctamente, siga estos pasos como requisitos previos:

1. Instala el plugin helm-mapkubeapis desde https://github.com/helm/helm-mapkubeapis .

2. Realice una prueba en seco de la versión de Trident en el espacio de nombres donde está instalado Trident . Esta lista incluye los recursos que serán depurados.

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

3. Ejecuta una limpieza completa con Helm.

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

Pasos

1. Si usted"Instalé Trident usando Helm." , puedes usar `helm upgrade trident netapp-trident/trident-operator --version 100.2506.0` Actualizar en un solo paso. Si no has añadido el repositorio de Helm o no puedes usarlo para actualizar:
   a. Descarga la última versión de Trident desde"la sección *Assets* en GitHub" .
   b. Utilice el `helm upgrade` comando donde `trident-operator-25.06.0.tgz` refleja la versión a la que desea actualizar.

```
helm upgrade <name> trident-operator-25.06.0.tgz
```

> Si configura opciones personalizadas durante la instalación inicial (como especificar registros privados y duplicados para imágenes Trident y CSI), añada lo siguiente: `helm upgrade` comando usando `--set` para garantizar que esas opciones se incluyan en el comando de actualización; de lo contrario, los valores se restablecerán a los valores predeterminados.

2. Correr `helm list` para verificar que tanto el gráfico como la versión de la aplicación se hayan actualizado. Correr `tridentctl logs` para revisar cualquier mensaje de depuración.

Actualización desde un `tridentctl` Instalación en el operador Trident

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

> Antes de cambiar entre métodos de instalación, revise"Cambio entre métodos de instalación" .

Pasos

1. Descarga la última versión de Trident .

```
# Download the release required [25.06.0]
mkdir 25.06.0
cd 25.06.0
wget
https://github.com/NetApp/trident/releases/download/v25.06.0/trident-
installer-25.06.0.tar.gz
tar -xf trident-installer-25.06.0.tar.gz
cd trident-installer
```

2. Crea el `tridentorchestrator` CRD del manifiesto.

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

3. Implemente 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. Crear una `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. Confirmo que Trident se actualizó a la versión prevista.

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

Message:                Trident installed
Namespace:              trident
Status:                 Installed
Version:                v25.06.0
```

Actualizar con tridentctl

Puede actualizar fácilmente una instalación de Trident existente usando `tridentctl` .

Acerca de esta tarea

Desinstalar y reinstalar Trident funciona como una actualización. Al desinstalar Trident, la reclamación de volumen persistente (PVC) y el volumen persistente (PV) utilizados por la implementación de Trident no se eliminan. Los PV que ya se hayan aprovisionado seguirán estando disponibles mientras Trident esté fuera de línea, y Trident aprovisionará volúmenes para cualquier PVC que se cree durante ese período una vez que vuelva a estar en línea.

Antes de empezar

Revisar"Seleccione un método de actualización" antes de actualizar usando `tridentctl` .

Pasos

1. Ejecuta el comando de desinstalación en `tridentctl` Eliminar todos los recursos asociados con Trident, excepto los CRD y los objetos relacionados.

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

2. Reinstale Trident. Referirse a"Instala Trident usando tridentctl" .

> No interrumpa el proceso de actualización. Asegúrese de que el instalador se ejecute hasta completarse.

Gestiona Trident usando tridentctl

El "Paquete de instalación de Trident" incluye el `tridentctl` Utilidad de línea de comandos para proporcionar un acceso sencillo a Trident. Los usuarios de Kubernetes con privilegios suficientes pueden usarlo para instalar Trident o administrar el espacio de nombres que contiene el pod de Trident .

Comandos y banderas globales

Puedes correr `tridentctl help` para obtener una lista de los comandos disponibles para `tridentctl` o añadir el `--help` Añada la bandera `--flag` a cualquier comando para obtener una lista de opciones y banderas para ese comando específico.

```
tridentctl [command] [--optional-flag]
```

El Trident `tridentctl` Esta utilidad admite los siguientes comandos y opciones globales.

Comandos

`create`
Agregar un recurso a Trident.

`delete`
Retira uno o más recursos de Trident.

`get`
Obtén uno o más recursos de Trident.

`help`
Ayuda sobre cualquier comando.

`images`
Imprime una tabla con las imágenes de contenedor que necesita Trident .

`import`
Importe un recurso existente a Trident.

`install`
Instala Trident.

`logs`
Imprime los registros de Trident.

`send`
Enviar un recurso desde Trident.

`uninstall`
Desinstala Trident.

`update`
Modificar un recurso en Trident.

`update backend state`
Suspender temporalmente las operaciones del servidor.

`upgrade`
Mejora un recurso en Trident.

`version`
Imprime la versión de Trident.

banderas mundiales

`-d, --debug`
Salida de depuración.

`-h, --help`
Ayuda para `tridentctl` .

`-k, --kubeconfig string`
Especifique el `KUBECONFIG` Ruta para ejecutar comandos localmente o desde un clúster de Kubernetes a otro.

> Alternativamente, puede exportar el `KUBECONFIG` variable para apuntar a un clúster de Kubernetes específico y un problema `tridentctl` comandos para ese clúster.

`-n, --namespace string`
Espacio de nombres de la implementación de Trident .

`-o, --output string`
Formato de salida. Uno de los siguientes: json|yaml|name|wide|ps (predeterminado).

`-s, --server string`
Dirección/puerto de la interfaz REST de Trident .

> La interfaz REST de Trident se puede configurar para escuchar y servir solo en 127.0.0.1 (para IPv4) o [::1] (para IPv6).

Opciones de comando y banderas

crear

Utilice el `create` comando para agregar un recurso a Trident.

```
tridentctl create [option]
```

Opciones

`backend`Agregar un backend a Trident.

borrar

Utilice el `delete` comando para eliminar uno o más recursos de Trident.

```
tridentctl delete [option]
```

Opciones

`backend`Eliminar uno o más backends de almacenamiento de Trident.
`snapshot` Eliminar una o más instantáneas de volumen de Trident.
`storageclass` Eliminar una o más clases de almacenamiento de Trident.

`volume` Eliminar uno o más volúmenes de almacenamiento de Trident.

conseguir

Utilice el `get` comando para obtener uno o más recursos de Trident.

```
tridentctl get [option]
```

Opciones

`backend`Obtenga uno o más backends de almacenamiento de Trident.
`snapshot` Obtenga una o más instantáneas de Trident.
`storageclass` Obtén una o más clases de almacenamiento de Trident.
`volume` Obtén uno o más volúmenes de Trident.

Banderas

`-h`, `--help` Ayuda para volúmenes.
`--parentOfSubordinate string` Limitar la consulta al volumen de origen subordinado.
`--subordinateOf string` Limitar la consulta a los subordinados del volumen.

imágenes

Usar `images` flags para imprimir una tabla con las imágenes de contenedor que necesita Trident .

```
tridentctl images [flags]
```

Banderas

`-h`, `--help` Ayuda para imágenes.
`-v` , `--k8s-version string` Versión semántica de clúster de Kubernetes.

volumen de importación

Utilice el `import volume` comando para importar un volumen existente a Trident.

```
tridentctl import volume <backendName> <volumeName> [flags]
```

Alias

`volume`, `v`

Banderas

`-f`, `--filename string` Ruta al archivo PVC YAML o JSON.
`-h` , `--help` Ayuda para el volumen.
`--no-manage` Crear solo PV/PVC. No dé por sentada la gestión del ciclo de vida del volumen.

instalar

Utilice el `install` flags para instalar Trident.

```
tridentctl install [flags]
```

Banderas

`--autosupport-image string`: La imagen del contenedor para la telemetría de soporte automático (por defecto "netapp/trident autosupport:<versión-actual>").
`--autosupport-proxy string` : La dirección/puerto de un proxy para enviar telemetría de Autosupport.
`--enable-node-prep` Intento de instalar los paquetes necesarios en los nodos.
`--generate-custom-yaml` Genera archivos YAML sin instalar nada.
`-h` , `--help` Ayuda para la instalación.
`--http-request-timeout` : Anular el tiempo de espera de la solicitud HTTP para la API REST del controlador Trident (por defecto 1m30s).
`--image-registry string` : La dirección/puerto de un registro de imágenes interno.
`--k8s-timeout duration` : El tiempo de espera para todas las operaciones de Kubernetes (por defecto 3m0s).
`--kubelet-dir string` : La ubicación del host del estado interno de kubelet (por defecto "/var/lib/kubelet").
`--log-format string` Formato de registro de Trident (texto, json) (por defecto "texto").
`--node-prep` Permite a Trident preparar los nodos del clúster de Kubernetes para administrar volúmenes utilizando el protocolo de almacenamiento de datos especificado. **Actualmente, `iscsi` es el único valor admitido. A partir de OpenShift 4.19, la versión mínima de Trident compatible con esta función es la 25.06.1.**
`--pv string` : El nombre del PV heredado utilizado por Trident, asegura que esto no exista (por defecto "trident").
`--pvc string` : El nombre del PVC heredado utilizado por Trident, asegura que esto no exista (por defecto "trident").
`--silence-autosupport` No enviar automáticamente paquetes de soporte automático a NetApp (valor predeterminado: verdadero).
`--silent` Deshabilitar la mayor parte de la salida durante la instalación.
`--trident-image string` La imagen de Trident que se va a instalar.
`--k8s-api-qps` : El límite de consultas por segundo (QPS) para las solicitudes de la API de Kubernetes (por defecto 100; opcional).
`--use-custom-yaml` Utilice cualquier archivo YAML existente que se encuentre en el directorio de configuración.
`--use-ipv6` Utilice IPv6 para la comunicación de Trident.

registros

Usar `logs` flags para imprimir los registros de Trident.

```
tridentctl logs [flags]
```

Banderas

`-a`, `--archive` Crear un archivo de soporte con todos los registros a menos que se especifique lo contrario.
`-h` , `--help` Ayuda para los registros.
`-l` , `--log string` Registro de Trident para mostrar. Una de las siguientes opciones: trident|auto|trident-operator|all (por defecto "auto").
`--node string` : El nombre del nodo de Kubernetes desde el cual se recopilarán los registros del pod del nodo.
`-p` , `--previous` Obtener los registros de la instancia de contenedor anterior, si existe.
`--sidecars` Obtén los registros de los contenedores sidecar.

enviar

Utilice el `send` comando para enviar un recurso desde Trident.

```
tridentctl send [option]
```

Opciones

`autosupport`Enviar un archivo de Autosupport a NetApp.

desinstalar

Usar `uninstall` flags para desinstalar Trident.

```
tridentctl uninstall [flags]
```

Banderas

`-h, --help`Ayuda para la desinstalación.
`--silent` Deshabilitar la mayor parte de la salida durante la desinstalación.

actualizar

Utilice el `update` comando para modificar un recurso en Trident.

```
tridentctl update [option]
```

Opciones

`backend`Actualizar un backend en Trident.

actualizar el estado del backend

Utilice el `update backend state` comando para suspender o reanudar las operaciones del servidor.

```
tridentctl update backend state <backend-name> [flag]
```

Puntos a considerar

- Si se crea un backend utilizando un TridentBackendConfig (por confirmar), el backend no se puede actualizar utilizando un `backend.json` archivo.
- Si el `userState` Se ha configurado en un archivo TBC, por lo que no se puede modificar mediante el mismo. `tridentctl update backend state <backend-name> --user-state suspended/normal` dominio.
- Para recuperar la capacidad de establecer el `userState` a través de tridentctl después de haber sido configurado a través de tbc, el `userState` El campo debe eliminarse del tbc. Esto se puede hacer utilizando el `kubectl edit tbc` dominio. Después de `userState` Se ha eliminado el campo, puede usar el `tridentctl update backend state` comando para cambiar el `userState` de un backend.
- Utilice el `tridentctl update backend state` para cambiar el `userState` . También puedes actualizar el `userState` usando `TridentBackendConfig` o `backend.json` archivo; esto desencadena una reinicialización completa del backend y puede llevar mucho tiempo.

Banderas

`-h, --help` Ayuda para el estado del backend.

`--user-state` : Configurado a `suspended` pausar las operaciones del servidor. Empezar a `normal` para reanudar las operaciones de backend. Cuando se configura para `suspended` :

- `AddVolume`y `Import Volume` están en pausa.
- `CloneVolume`, `ResizeVolume` , `PublishVolume` , `UnPublishVolume` , `CreateSnapshot` , `GetSnapshot` , `RestoreSnapshot` , `DeleteSnapshot` , `RemoveVolume` , `GetVolumeExternal` , `ReconcileNodeAccess` permanecen disponibles.

También puedes actualizar el estado del backend usando `userState` campo en el archivo de configuración del backend `TridentBackendConfig` o `backend.json` . Para obtener más información, consulte"Opciones para la gestión de backends" y"Realizar la gestión del backend con kubectl" .

Ejemplo:

JSON

Siga estos pasos para actualizar el `userState` utilizando el `backend.json` archivo:

1. Editar el `backend.json` archivo para incluir el `userState` campo con su valor establecido en 'suspendido'.
2. Actualiza el backend usando el `tridentctl update backend` comando y la ruta a la versión actualizada `backend.json` archivo.

   Ejemplo: `tridentctl update backend -f /<path to backend JSON file>/backend.json -n trident`

```
{
  "version": 1,
  "storageDriverName": "ontap-nas",
  "managementLIF": "<redacted>",
  "svm": "nas-svm",
  "backendName": "customBackend",
  "username": "<redacted>",
  "password": "<redacted>",
  "userState": "suspended"
}
```

YAML

Puedes editar el tbc después de que se haya aplicado usando el `kubectl edit <tbc-name> -n <namespace>` dominio. El siguiente ejemplo actualiza el estado del backend a suspendido utilizando el `userState: suspended` opción:

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-ontap-nas
spec:
  version: 1
  backendName: customBackend
  storageDriverName: ontap-nas
  managementLIF: <redacted>
  svm: nas-svm
  userState: suspended
  credentials:
    name: backend-tbc-ontap-nas-secret
```

versión

Usar `version` banderas para imprimir la versión de `tridentctl` y el servicio Trident en funcionamiento.

```
tridentctl version [flags]
```

Banderas

`--client`: Solo versión cliente (no requiere servidor).
`-h, --help` Ayuda para la versión.

Compatibilidad con complementos

Tridentctl admite complementos similares a los de kubectl. Tridentctl detecta un complemento si el nombre del archivo binario del complemento sigue el esquema "tridentctl-<plugin>" y el binario se encuentra en una carpeta que figura en la variable de entorno PATH. Todos los plugins detectados se enumeran en la sección de plugins de la ayuda de tridentctl. Opcionalmente, también puede limitar la búsqueda especificando una carpeta de complementos en la variable de entorno TRIDENTCTL_PLUGIN_PATH (Ejemplo: `TRIDENTCTL_PLUGIN_PATH=~/tridentctl-plugins/`). Si se utiliza la variable, tridenctl buscará solo en la carpeta especificada.

Monitor Trident

Trident proporciona un conjunto de endpoints de métricas de Prometheus que puede utilizar para supervisar el rendimiento de Trident .

Descripción general

Las métricas proporcionadas por Trident le permiten hacer lo siguiente:

- Controla el estado y la configuración de Trident. Puedes examinar el éxito de las operaciones y si se comunican con los sistemas backend según lo previsto.
- Examine la información de uso del backend y comprenda cuántos volúmenes se aprovisionan en un backend y la cantidad de espacio consumido, etc.
- Mantenga un registro de la cantidad de volúmenes aprovisionados en los backends disponibles.
- Rendimiento de la pista. Puedes comprobar cuánto tiempo tarda Trident en comunicarse con los backends y realizar operaciones.

Por defecto, las métricas de Trident se exponen en el puerto de destino. `8001` al `/metrics` punto final. Estas métricas están **habilitadas por defecto** cuando se instala Trident .

Lo que necesitarás

- Un clúster de Kubernetes con Trident instalado.
- Una instancia de Prometheus. Esto puede ser un "Despliegue de Prometheus en contenedores" o puedes optar por ejecutar Prometheus como un "aplicación nativa" .

Paso 1: Definir un objetivo de Prometheus

Debes definir un objetivo de Prometheus para recopilar las métricas y obtener información sobre los backends que gestiona Trident , los volúmenes que crea, etc. Este "blog" Explica cómo puedes usar Prometheus y Grafana con Trident para recuperar métricas. El blog explica cómo puedes ejecutar Prometheus como

operador en tu clúster de Kubernetes y la creación de un ServiceMonitor para obtener métricas de Trident .

Paso 2: Crear un ServiceMonitor de Prometheus

Para consumir las métricas de Trident , debes crear un ServiceMonitor de Prometheus que supervise el `trident-csi` servicio y escucha en el `metrics` puerto. Un ejemplo de ServiceMonitor tiene el siguiente aspecto:

```
apiVersion: monitoring.coreos.com/v1
kind: ServiceMonitor
metadata:
  name: trident-sm
  namespace: monitoring
  labels:
    release: prom-operator
spec:
  jobLabel: trident
  selector:
    matchLabels:
      app: controller.csi.trident.netapp.io
  namespaceSelector:
    matchNames:
      - trident
  endpoints:
    - port: metrics
      interval: 15s
```

Esta definición de ServiceMonitor recupera las métricas devueltas por el `trident-csi` servicio y específicamente busca el `metrics` punto final del servicio. Como resultado, Prometheus ahora está configurado para comprender las métricas de Trident.

Además de las métricas disponibles directamente desde Trident, kubelet expone muchas otras. `kubelet_volume_*` métricas a través de su propio punto final de métricas. Kubelet puede proporcionar información sobre los volúmenes conectados, los pods y otras operaciones internas que gestiona. Referirse a "aquí" .

Paso 3: Consultar las métricas de Trident con PromQL

PromQL es útil para crear expresiones que devuelven datos de series temporales o tabulares.

Aquí tienes algunas consultas PromQL que puedes usar:

Obtén información de salud de Trident

- **Porcentaje de respuestas HTTP 2XX de Trident**

```
(sum (trident_rest_ops_seconds_total_count{status_code=~"2.."} OR on()
vector(0)) / sum (trident_rest_ops_seconds_total_count)) * 100
```

- **Porcentaje de respuestas REST de Trident mediante código de estado**

```
(sum (trident_rest_ops_seconds_total_count) by (status_code)  / scalar
(sum (trident_rest_ops_seconds_total_count))) * 100
```

- **Duración media en ms de las operaciones realizadas por Trident**

```
sum by (operation)
(trident_operation_duration_milliseconds_sum{success="true"}) / sum by
(operation)
(trident_operation_duration_milliseconds_count{success="true"})
```

Obtén información sobre el uso de Trident

- **Tamaño de volumen promedio**

```
trident_volume_allocated_bytes/trident_volume_count
```

- **Espacio de volumen total aprovisionado por cada backend**

```
sum (trident_volume_allocated_bytes) by (backend_uuid)
```

Obtenga el uso de volumen individual

> Esto solo se habilita si también se recopilan las métricas de kubelet.

- **Porcentaje de espacio utilizado para cada volumen**

```
kubelet_volume_stats_used_bytes / kubelet_volume_stats_capacity_bytes *
100
```

Obtenga información sobre la telemetría de Trident AutoSupport .

Por defecto, Trident envía métricas de Prometheus e información básica del backend a NetApp diariamente.

- Para evitar que Trident envíe métricas de Prometheus e información básica del backend a NetApp, pase el `--silence-autosupport` bandera durante la instalación del Trident .
- Trident también puede enviar registros de contenedores al soporte de NetApp bajo demanda a través de

`tridentctl send autosupport` . Deberás configurar Trident para que suba sus registros. Antes de enviar los registros, debe aceptar los términos de NetApp.https://www.netapp.com/company/legal/privacy-policy/["política de privacidad"^] .

- A menos que se especifique lo contrario, Trident recupera los registros de las últimas 24 horas.
- Puede especificar el período de retención de registros con el `--since` bandera. Por ejemplo: `tridentctl send autosupport --since=1h` . Esta información se recopila y se envía a través de un `trident-autosupport` contenedor que se instala junto con Trident. Puedes obtener la imagen del contenedor en "AutoSupport Trident" .
- Trident AutoSupport no recopila ni transmite información de identificación personal (PII) ni información personal. Viene con un "CLUF" Eso no es aplicable a la propia imagen del contenedor Trident . Puedes obtener más información sobre el compromiso de NetApp con la seguridad y la confianza en los datos. "aquí" .

Un ejemplo de carga útil enviada por Trident tiene el siguiente aspecto:

```
---
items:
  - backendUUID: ff3852e1-18a5-4df4-b2d3-f59f829627ed
    protocol: file
    config:
      version: 1
      storageDriverName: ontap-nas
      debug: false
      debugTraceFlags: null
      disableDelete: false
      serialNumbers:
        - nwkvzfanek_SN
      limitVolumeSize: ""
    state: online
    online: true
```

- Los mensajes de AutoSupport se envían al punto final de AutoSupport de NetApp. Si utiliza un registro privado para almacenar imágenes de contenedores, puede usar el `--image-registry` bandera.
- También puedes configurar las URL del proxy generando los archivos YAML de instalación. Esto se puede hacer utilizando `tridentctl install --generate-custom-yaml` para crear los archivos YAML y agregar el `--proxy-url` argumento a favor del `trident-autosupport` contenedor en `trident-deployment.yaml` .

Deshabilitar las métricas de Trident

Para **deshabilitar** la generación de informes de métricas, debe generar archivos YAML personalizados (utilizando el `--generate-custom-yaml` bandera) y edítelas para eliminar la `--metrics` bandera para que no se invoque para el `trident-main` recipiente.

Desinstalar Trident

Debes utilizar el mismo método para desinstalar Trident que el que utilizaste para Trident.

Acerca de esta tarea

- Si necesita solucionar errores detectados tras una actualización, problemas de dependencias o una actualización fallida o incompleta, debe desinstalar Trident y reinstalar la versión anterior siguiendo las instrucciones específicas para ello."versión" . Esta es la única forma recomendada de *revertir* a una versión anterior.
- Para facilitar la actualización y reinstalación, la desinstalación de Trident no elimina los CRD ni los objetos relacionados creados por Trident. Si necesita eliminar completamente Trident y todos sus datos, consulte"Eliminar completamente Trident y CRD" .

Antes de empezar

Si va a desmantelar clústeres de Kubernetes, debe eliminar todas las aplicaciones que utilizan volúmenes creados por Trident antes de la desinstalación. Esto garantiza que los PVC se despubliquen en los nodos de Kubernetes antes de ser eliminados.

Determinar el método de instalación original

Debes utilizar el mismo método para desinstalar Trident que el que usaste para instalarlo. Antes de desinstalar, verifique qué versión utilizó para instalar Trident originalmente.

1. Usar `kubectl get pods -n trident` para examinar las vainas.
   - Si no hay ningún pod de operador, Trident se instaló mediante `tridentctl` .
   - Si existe un pod de operador, Trident se instaló utilizando el operador Trident , ya sea manualmente o utilizando Helm.
2. Si hay un pod de operador, utilice `kubectl describe tproc trident` para determinar si Trident se instaló utilizando Helm.
   - Si aparece una etiqueta Helm, Trident se instaló utilizando Helm.
   - Si no hay ninguna etiqueta Helm, Trident se instaló manualmente utilizando el operador Trident .

Desinstalar una instalación de operador de Trident

Puedes desinstalar una instalación de operador Trident manualmente o usando Helm.

Desinstalación de la instalación manual

Si instalaste Trident usando el operador, puedes desinstalarlo realizando una de las siguientes acciones:

1. **Editar `TridentOrchestrator` CR y establecer la bandera de desinstalación**:

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

   Cuando el `uninstall` La bandera está configurada para `true` El operador de Trident desinstala Trident,

pero no elimina el propio TridentOrchestrator. Deberás eliminar el archivo TridentOrchestrator y crear uno nuevo si deseas reinstalar Trident .

2. **Borrar `TridentOrchestrator`** : Al eliminar el `TridentOrchestrator` Si el CR se utilizó para implementar Trident, debe indicar al operador que desinstale Trident. El operador procesa la eliminación de `TridentOrchestrator` y procede a eliminar la implementación de Trident y el daemonset, borrando los pods de Trident que había creado como parte de la instalación.

```
kubectl delete -f deploy/<bundle.yaml> -n <namespace>
```

Desinstalar la instalación de Helm

Si instalaste Trident usando Helm, puedes desinstalarlo usando `helm uninstall` .

```
#List the Helm release corresponding to the 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
```

Desinstalar un `tridentctl` instalación

Utilice el `uninstall` comando en `tridentctl` Eliminar todos los recursos asociados a Trident, excepto los CRD y los objetos relacionados:

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