Documentación de Trident 25.06

Trident

NetApp
January 15, 2026

This PDF was generated from https://docs.netapp.com/es-es/trident-2506/index.html on January 15, 2026. Always check docs.netapp.com for the latest.

Tabla de contenidos

Documentación de Trident 25.06 . . . 1
Notas de la versión . . . 2
Qué hay de nuevo . . . 2
Novedades de la versión 25.06.2 . . . 2
Cambios en 25.06.1 . . . 2
Cambios en 25.06 . . . 2
Cambios en 25.02.1 . . . 5
Cambios en 25.02 . . . 5
Cambios en 24.10.1 . . . 7
Cambios en 24.10 . . . 7
Cambios en 24.06 . . . 9
Cambios en 24.02 . . . 10
Cambios en 23.10 . . . 10
Cambios en 23.07.1 . . . 11
Cambios en 23.07 . . . 11
Cambios en 23.04 . . . 12
Cambios en 23.01.1 . . . 13
Cambios en 23.01 . . . 14
Cambios en 22.10 . . . 14
Cambios en 22.07 . . . 16
Cambios en 22.04 . . . 17
Cambios en 22.01.1 . . . 17
Cambios en 22.01.0 . . . 18
Cambios en 21.10.1 . . . 18
Cambios en 21.10.0 . . . 19
Problemas conocidos . . . 20
Encuentra más información . . . 21
Versiones anteriores de la documentación . . . 21
Problemas conocidos . . . 21
La restauración de copias de seguridad de archivos grandes de Restic puede fallar. . . . 21
Empezar . . . 22
Aprende sobre Trident . . . 22
Aprende sobre Trident . . . 22
Arquitectura Trident . . . 23
Conceptos . . . 26
Guía de inicio rápido para Trident . . . 30
¿Que sigue? . . . 31
Requisitos . . . 32
Información crítica sobre Trident . . . 32
Front-ends compatibles (orquestadores) . . . 32
Sistemas de almacenamiento compatibles . . . 33
Compatibilidad de Trident con la virtualización de KubeVirt y OpenShift . . . 33
Requisitos de las funciones . . . 34

Sistemas operativos host probados . . . 34
Configuración del host . . . 34
Configuración del sistema de almacenamiento . . . 35
Puertos Trident . . . 35
Imágenes de contenedores y versiones de Kubernetes correspondientes . . . 35
Instalar Trident . . . 37
Instalar usando el operador Trident . . . 37
Instalar usando tridentctl . . . 37
Instalar utilizando un operador certificado de OpenShift . . . 37
Usar Trident . . . 38
Preparar el nodo de trabajo . . . 38
Seleccionar las herramientas adecuadas . . . 38
Descubrimiento de servicios de nodo . . . 38
Volúmenes de NFS . . . 39
volúmenes iSCSI . . . 39
volúmenes NVMe/TCP . . . 43
SCSI sobre volúmenes FC . . . 44
Configurar y administrar backends . . . 47
Configurar backends . . . 47
Azure NetApp Files . . . 47
Google Cloud NetApp Volumes . . . 67
Configurar un Cloud Volumes Service para el backend de Google Cloud . . . 84
Configure un backend de NetApp HCI o SolidFire . . . 96
Controladores SAN de ONTAP . . . 101
Controladores NAS ONTAP . . . 131
Amazon FSx for NetApp ONTAP . . . 167
Crea backends con kubectl . . . 202
Gestionar backends . . . 210
Crear y gestionar clases de almacenamiento . . . 220
Crear una clase de almacenamiento . . . 220
Gestionar clases de almacenamiento . . . 223
Provisión y gestión de volúmenes . . . 225
Provisión de un volumen . . . 225
Ampliar volúmenes . . . 229
volúmenes de importación . . . 240
Personaliza los nombres y etiquetas de los volúmenes. . . . 248
Compartir un volumen NFS entre espacios de nombres . . . 251
Clonar volúmenes entre espacios de nombres . . . 255
Replicar volúmenes usando SnapMirror . . . 258
Utilizar la topología CSI . . . 265
Trabajar con instantáneas . . . 272
Trabajar con instantáneas de grupos de volúmenes . . . 280
Gestionar y supervisar Trident . . . 285
Mejora el Trident . . . 285
Mejora el Trident . . . 285

Actualiza con el operador . . . 286
Actualizar con tridentctl . . . 291
Gestiona Trident usando tridentctl . . . 292
Comandos y banderas globales . . . 292
Opciones de comando y banderas . . . 294
Compatibilidad con complementos . . . 300
Monitor Trident . . . 300
Descripción general . . . 300
Paso 1: Definir un objetivo de Prometheus . . . 300
Paso 2: Crear un ServiceMonitor de Prometheus . . . 301
Paso 3: Consultar las métricas de Trident con PromQL . . . 301
Obtenga información sobre la telemetría de Trident AutoSupport . . . 302
Deshabilitar las métricas de Trident . . . 303
Desinstalar Trident . . . 304
Determinar el método de instalación original . . . 304
Desinstalar una instalación de operador de Trident . . . 304
Desinstalar un `tridentctl` instalación . . . 305
Trident para Docker . . . 306
Requisitos previos para la implementación . . . 306
Verificar los requisitos . . . 306
Herramientas NVMe . . . 308
Herramientas FC . . . 309
Despliegue Trident . . . 311
Método de complemento gestionado por Docker (versión 1.13/17.03 y posteriores) . . . 311
Método tradicional (versión 1.12 o anterior) . . . 313
Inicie Trident al arrancar el sistema. . . . 315
Actualizar o desinstalar Trident . . . 315
Mejora . . . 316
Desinstalar . . . 317
Trabajar con volúmenes . . . 317
Crear un volumen . . . 317
Eliminar un volumen . . . 318
Clonar un volumen . . . 318
Acceso a volúmenes creados externamente . . . 320
Opciones de volumen específicas del conductor . . . 320
Recopilar registros . . . 326
Recopilar registros para la resolución de problemas . . . 326
Consejos generales para la solución de problemas . . . 327
Gestionar múltiples instancias de Trident . . . 327
Pasos para el plugin gestionado por Docker (versión 1.13/17.03 o posterior) . . . 327
Pasos para la versión tradicional (versión 1.12 o anterior) . . . 328
Opciones de configuración de almacenamiento . . . 328
Opciones de configuración global . . . 328
Configuración de ONTAP . . . 329
Configuración del software Element . . . 337

Problemas y limitaciones conocidos . . . 339
La actualización del complemento Trident Docker Volume a la versión 20.10 o posterior desde versiones anteriores produce un fallo en la actualización con el error "no existe el archivo o directorio". . . . 339
Los nombres de los volúmenes deben tener una longitud mínima de 2 caracteres. . . . 340
Docker Swarm presenta ciertos comportamientos que impiden que Trident lo admita con todas las combinaciones de almacenamiento y controladores. . . . 340
Si se está aprovisionando un FlexGroup , ONTAP no aprovisiona un segundo FlexGroup si este FlexGroup tiene uno o más agregados en común con el FlexGroup que se está aprovisionando. . . . 340
Mejores prácticas y recomendaciones . . . 341
Despliegue . . . 341
Implementar en un espacio de nombres dedicado . . . 341
Utilice cuotas y límites de rango para controlar el consumo de almacenamiento. . . . 341
Configuración de almacenamiento . . . 341
Descripción general de la plataforma . . . 341
Mejores prácticas de ONTAP y Cloud Volumes ONTAP . . . 341
Mejores prácticas de SolidFire . . . 346
¿Dónde puedo encontrar más información? . . . 348
Integrar Trident . . . 348
Selección y despliegue de controladores . . . 349
Diseño de clase de almacenamiento . . . 352
Diseño de piscina virtual . . . 353
Operaciones de volumen . . . 354
Servicio de métricas . . . 358
Protección de datos y recuperación ante desastres . . . 359
replicación y recuperación de Trident . . . 359
replicación y recuperación de SVM . . . 360
replicación y recuperación de volumen . . . 361
Protección de datos de instantáneas . . . 361
Seguridad . . . 361
Seguridad . . . 361
Configuración de clave unificada de Linux (LUKS) . . . 363
Cifrado en vuelo Kerberos . . . 369
Proteja las aplicaciones con Trident Protect . . . 377
Obtenga más información sobre Trident Protect . . . 377
¿Que sigue? . . . 377
Instalar Trident Protect . . . 377
Requisitos de Trident Protect . . . 377
Instalar y configurar Trident Protect . . . 381
Instalar el complemento CLI de Trident Protect . . . 384
Personalizar la instalación de Trident Protect . . . 388
Administrar Trident Protect . . . 393
Administrar la autorización y el control de acceso de Trident Protect . . . 393
Supervisar los recursos de Trident Protect . . . 400
Generar un paquete de soporte de Trident Protect . . . 405

Actualizar Trident Protect 407
Gestionar y proteger las aplicaciones 408
Utilice objetos Trident Protect AppVault para administrar depósitos 408
Definir una aplicación para la gestión con Trident Protect 422
Proteja las aplicaciones con Trident Protect 426
Restaurar aplicaciones 437
Replicar aplicaciones utilizando NetApp SnapMirror y Trident Protect 455
Migrar aplicaciones usando Trident Protect 472
Administrar los ganchos de ejecución de Trident Protect 476
Desinstalar Trident Protect 488
Blogs de Trident y Trident Protect 489
Blogs de Trident 489
Blogs de Trident Protect 489
Conocimiento y apoyo 491
Preguntas frecuentes 491
Preguntas generales 491
Instala y usa Trident en un clúster de Kubernetes. 491
Solución de problemas y soporte 492
Mejora el Trident 494
Gestionar backends y volúmenes 494
Solución de problemas 498
Solución de problemas generales 498
Despliegue fallido de Trident utilizando el operador 500
Despliegue fallido de Trident utilizando `tridentctl` 502
Eliminar completamente Trident y CRD 502
Fallo en la desorganización del nodo NVMe con espacios de nombres de bloques sin formato RWX en Kubernetes 1.26 503
Los clientes NFSv4.2 informan "argumento no válido" después de actualizar ONTAP cuando esperaban que "v4.2-xattrs" estuviera habilitado 504
Soporte 504
Soporte Trident 504
Autosuficiencia 505
Apoyo comunitario 505
Soporte técnico de NetApp 505
Para más información 505
Referencia 506
Puertos Trident 506
Puertos Trident 506
API REST de Trident 506
Cuándo usar la API REST 506
Utilizando la API REST 506
Opciones de línea de comandos 507
Explotación florestal 507
Kubernetes 507
Docker 508

DESCANSAR 508
Objetos de Kubernetes y Trident 508
¿Cómo interactúan los objetos entre sí? 508
Kubernetes `PersistentVolumeClaim` objetos 509
Kubernetes `PersistentVolume` objetos 511
Kubernetes `StorageClass` objetos 511
Kubernetes `VolumeSnapshotClass` objetos 515
Kubernetes `VolumeSnapshot` objetos 515
Kubernetes `VolumeSnapshotContent` objetos 516
Kubernetes `VolumeGroupSnapshotClass` objetos 516
Kubernetes `VolumeGroupSnapshot` objetos 517
Kubernetes `VolumeGroupSnapshotContent` objetos 517
Kubernetes `CustomResourceDefinition` objetos 518
Trident `StorageClass` objetos 518
objetos de backend de Trident 518
Trident `StoragePool` objetos 519
Trident `Volume` objetos 519
Trident `Snapshot` objetos 520
Trident `ResourceQuota` objeto 521
Estándares de seguridad de pods (PSS) y restricciones de contexto de seguridad (SCC) 522
Contexto de seguridad de Kubernetes requerido y campos relacionados 523
Estándares de seguridad de pods (PSS) 523
Políticas de seguridad de pods (PSP) 524
Restricciones del contexto de seguridad (SCC) 525
Avisos legales 528
Copyright 528
Marcas comerciales 528
Patentes 528
Política de privacidad 528
Código abierto 528

Documentación de Trident 25.06

Notas de la versión

Qué hay de nuevo

Las notas de la versión proporcionan información sobre las nuevas características, mejoras y correcciones de errores en la última versión de NetApp Trident.

> El `tridentctl` El binario para Linux que se proporciona en el archivo zip del instalador es la versión probada y compatible. Tenga en cuenta que el `macos` binario proporcionado en el `/extras` Parte del archivo zip no ha sido probado ni cuenta con soporte.

Novedades de la versión 25.06.2

El resumen de novedades proporciona detalles sobre mejoras, correcciones y desuso de las versiones Trident y Trident Protect.

Trident

Correcciones

- **Kubernetes**: Se solucionó un problema crítico en el que se detectaban dispositivos iSCSI incorrectos al desconectar volúmenes de los nodos de Kubernetes.

Cambios en 25.06.1

Trident

> Para los clientes que usan SolidFire, no actualicen a la versión 25.06.1 debido a un problema conocido al anular la publicación de volúmenes. Próximamente se lanzará la versión 25.06.2 para solucionar este problema.

Correcciones

- **Kubernetes**:
  - Se solucionó un problema en el cual los NQN no se verificaban antes de desasignarlos de los subsistemas.
  - Se solucionó un problema en el que varios intentos de cerrar un dispositivo LUKS provocaban fallas en la separación de volúmenes.
  - Se solucionó el problema del volumen iSCSI sin preparación cuando la ruta del dispositivo cambió desde su creación.
  - Clonación de bloques de volúmenes en distintas clases de almacenamiento.
- **OpenShift**: se solucionó un problema en el que la preparación del nodo iSCSI fallaba con OCP 4.19.
- Se aumentó el tiempo de espera al clonar un volumen utilizando backends de SolidFire ("Número 1008").

Cambios en 25.06

Trident

Mejoras

- **Kubernetes**:
  - Se agregó compatibilidad con instantáneas de grupos de volúmenes CSI con `v1beta1` API de Kubernetes para instantáneas de grupos de volúmenes para el controlador iSCSI ONTAP-SAN. Ver"Trabajar con instantáneas de grupos de volúmenes" .

    > VolumeGroupSnapshot es una función beta en Kubernetes con API beta. Kubernetes 1.32 es la versión mínima requerida para VolumeGroupSnapshot.

  - Se ha añadido compatibilidad con ONTAP ASA r2 para NVMe/TCP además de iSCSI. Verlink:"Opciones y ejemplos de configuración de SAN de ONTAP" .
  - Se ha añadido compatibilidad segura con SMB para los volúmenes ONTAP-NAS y ONTAP-NAS-Economy. Ahora es posible utilizar usuarios y grupos de Active Directory con volúmenes SMB para mejorar la seguridad. Ver"Habilitar SMB seguro" .
  - Concurrencia de nodos Trident mejorada para una mayor escalabilidad en las operaciones de nodos para volúmenes iSCSI.
  - Agregado `--allow-discards` al abrir volúmenes LUKS para permitir comandos de descarte/TRIM para la recuperación de espacio.
  - Rendimiento mejorado al formatear volúmenes cifrados con LUKS.
  - Limpieza mejorada de LUKS para dispositivos LUKS fallidos pero parcialmente formateados.
  - Idempotencia mejorada del nodo Trident para la conexión y desconexión de volúmenes NVMe.
  - Agregado `internalID` campo a la configuración de volumen Trident para el controlador ONTAP-SAN-Economy.
  - Se ha añadido compatibilidad con la replicación de volúmenes con SnapMirror para backends NVMe. Ver"Replicar volúmenes usando SnapMirror" .

Mejoras experimentales

> No apto para su uso en entornos de producción.

- [Vista previa técnica] Se habilitaron operaciones simultáneas del controlador Trident a través de `--enable-concurrency` bandera de característica. Esto permite que las operaciones del controlador se ejecuten en paralelo, mejorando el rendimiento en entornos con mucha actividad o de gran tamaño.

  > Esta función es experimental y actualmente admite flujos de trabajo paralelos limitados con el controlador ONTAP-SAN (protocolos iSCSI y FCP).

- [Vista previa técnica] Se ha añadido compatibilidad con QoS manual con el controlador ANF.

Correcciones

- **Kubernetes**:
  - Se solucionó un problema con CSI NodeExpandVolume por el cual los dispositivos de múltiples rutas podían quedar con tamaños incongruentes cuando los discos SCSI subyacentes no estaban disponibles.

- Se corrigió un error al limpiar las políticas de exportación duplicadas para los controladores ONTAP-NAS y ONTAP-NAS-Economy.
- Se corrigió el problema de los volúmenes GCNV que usaban NFSv3 por defecto. `nfsMountOptions` no está configurado; ahora se admiten los protocolos NFSv3 y NFSv4. Si `nfsMountOptions` Si no se proporciona, se utilizará la versión NFS predeterminada del host (NFSv3 o NFSv4).
- Se solucionó un problema de implementación al instalar Trident usando Kustomize ("Número 831").
- Se corrigieron las políticas de exportación faltantes para los PVC creados a partir de instantáneas ("Número 1016").
- Se ha solucionado un problema por el cual los tamaños de volumen ANF no se alineaban automáticamente a incrementos de 1 GiB.
- Se ha solucionado un problema que ocurría al usar NFSv3 con Bottlerocket.

- Se ha corregido el tiempo de espera al clonar un volumen utilizando backends de SolidFire ("Número 1008").
- Se solucionó un problema con los volúmenes ONTAP-NAS-Economy que se expandían hasta 300 TB a pesar de los fallos de redimensionamiento.
- Se solucionó un problema por el cual las operaciones de división de clones se realizaban de forma síncrona al usar la API REST de ONTAP .

Depreciaciones:

- **Kubernetes**: Se actualizó la versión mínima compatible de Kubernetes a la v1.27.

Trident Protect

NetApp Trident Protect proporciona capacidades avanzadas de gestión de datos de aplicaciones que mejoran la funcionalidad y la disponibilidad de las aplicaciones Kubernetes con estado respaldadas por los sistemas de almacenamiento NetApp ONTAP y el aprovisionador de almacenamiento NetApp Trident CSI.

Mejoras

- Mejora de los tiempos de restauración, ofreciendo la opción de realizar copias de seguridad completas con mayor frecuencia.
- Mejora de la granularidad en la definición de aplicaciones y restauración selectiva con filtrado por tipo de versión de grupo (GVK).
- Resincronización y replicación inversa eficientes al usar AppMirrorRelationship (AMR) con NetApp SnapMirror, para evitar la replicación completa de PVC.
- Se ha añadido la capacidad de usar la identidad de pod de EKS para crear buckets de AppVault, eliminando la necesidad de especificar un secreto con las credenciales del bucket para los clústeres de EKS.
- Se ha añadido la posibilidad de omitir la restauración de etiquetas y anotaciones en el espacio de nombres de restauración, si fuera necesario.
- AppMirrorRelationship (AMR) ahora comprobará la expansión del PVC de origen y realizará la expansión apropiada en el PVC de destino según sea necesario.

Correcciones

- Se corrigió un error por el cual los valores de anotación de instantáneas anteriores se aplicaban a instantáneas más recientes. Ahora todas las anotaciones de instantáneas se aplican correctamente.

- Se define un secreto para el cifrado del gestor de datos (Kopia / Restic) por defecto, si no se define…
- Se han añadido validaciones mejoradas y mensajes de error para la creación de appvaults en S3.
- Ahora, AppMirrorRelationship (AMR) solo replica los PV en estado Bound, para evitar intentos fallidos.
- Se solucionó un problema por el cual se mostraban errores al obtener AppVaultContent en un AppVault con una gran cantidad de copias de seguridad.
- Las instantáneas de máquinas virtuales de KubeVirt se excluyen de las operaciones de restauración y conmutación por error para evitar fallos.
- Se solucionó un problema con Kopia por el cual las instantáneas se eliminaban prematuramente debido a que el programa de retención predeterminado de Kopia anulaba el establecido por el usuario.

Cambios en 25.02.1

Trident

Correcciones

- **Kubernetes**:
  - Se solucionó un problema en el operador trident donde los nombres y versiones de las imágenes sidecar se completaban incorrectamente al usar un registro de imágenes no predeterminado ("Número 983").
  - Se solucionó el problema por el cual las sesiones de múltiples rutas no se recuperaban durante una recuperación por error de ONTAP ("Número 961").

Cambios en 25.02

A partir de Trident 25.02, el resumen de novedades proporciona detalles sobre mejoras, correcciones y desuso de las versiones Trident y Trident Protect.

Trident

Mejoras

- **Kubernetes**:
  - Se ha añadido compatibilidad con ONTAP ASA r2 para iSCSI.
  - Se ha añadido compatibilidad con la desconexión forzada de volúmenes ONTAP-NAS durante escenarios de apagado de nodo no controlado. Los nuevos volúmenes ONTAP-NAS ahora utilizarán políticas de exportación por volumen gestionadas por Trident. Se proporcionó una ruta de actualización para que los volúmenes existentes pudieran migrar al nuevo modelo de política de exportación al anular la publicación sin afectar las cargas de trabajo activas.
  - Se agregó la anotación cloneFromSnapshot.
  - Se ha añadido compatibilidad con la clonación de volúmenes entre espacios de nombres.
  - Mejoras en las correcciones de escaneo de autorreparación de iSCSI para iniciar nuevos escaneos por ID de host, canal, destino y LUN exactos.
  - Se ha añadido compatibilidad con Kubernetes 1.32.
- **OpenShift**:
  - Se ha añadido compatibilidad con la preparación automática de nodos iSCSI para RHCOS en clústeres ROSA.

    - Se ha añadido compatibilidad con la virtualización de OpenShift para los controladores ONTAP .
- Se ha añadido compatibilidad con Fibre Channel en el controlador ONTAP-SAN.
- Se ha añadido compatibilidad con NVMe LUKS.
- Se cambió a una imagen de prueba para todas las imágenes base.
- Se agregó la detección y el registro del estado de la conexión iSCSI cuando las sesiones iSCSI deberían estar iniciadas pero no lo están ("Número 961").
- Se ha añadido compatibilidad con volúmenes SMB mediante el controlador google-cloud-netapp-volumes.
- Se ha añadido compatibilidad para permitir que los volúmenes ONTAP omitan la cola de recuperación al eliminarse.
- Se ha añadido compatibilidad para sustituir las imágenes predeterminadas mediante SHA en lugar de etiquetas.
- Se agregó el flag image-pull-secrets al instalador tridentctl.

Correcciones

- **Kubernetes**:
    - Se corrigieron las direcciones IP de nodos faltantes en las políticas de exportación automática ("Número 965").
    - Se corrigió el cambio prematuro de las políticas de exportación automática a la política por volumen para ONTAP-NAS-Economy.
    - Se corrigieron las credenciales de configuración del backend para admitir todas las particiones ARN de AWS disponibles ("Número 913").
    - Se ha añadido la opción de desactivar la reconciliación automática del configurador en el operador Trident ("Número 924").
    - Se agregó securityContext para el contenedor csi-resizer ("Número 976").

Trident Protect

NetApp Trident Protect proporciona capacidades avanzadas de gestión de datos de aplicaciones que mejoran la funcionalidad y la disponibilidad de las aplicaciones Kubernetes con estado respaldadas por los sistemas de almacenamiento NetApp ONTAP y el aprovisionador de almacenamiento NetApp Trident CSI.

Mejoras

- Se ha añadido compatibilidad con copias de seguridad y restauración para máquinas virtuales de virtualización KubeVirt / OpenShift tanto para almacenamiento volumeMode: File como volumeMode: Block (dispositivo sin formato). Este soporte es compatible con todos los controladores Trident y mejora las funciones de protección existentes al replicar almacenamiento utilizando NetApp SnapMirror con Trident Protect.
- Se ha añadido la capacidad de controlar el comportamiento de congelación a nivel de aplicación para entornos Kubevirt.
- Se ha añadido compatibilidad para configurar las conexiones proxy de AutoSupport .
- Se agregó la capacidad de definir un secreto para el cifrado del movimiento de datos (Kopia / Restic).
- Se ha añadido la capacidad de ejecutar manualmente un gancho de ejecución.
- Se agregó la capacidad de configurar restricciones de contexto de seguridad (SCC) durante la instalación de Trident Protect.

- Se agregó soporte para configurar nodeSelector durante la instalación de Trident Protect.
- Se ha añadido compatibilidad con proxy de salida HTTP/HTTPS para objetos de AppVault.
- Se ha extendido el filtro de recursos para permitir la exclusión de recursos con ámbito de clúster.
- Se ha añadido compatibilidad con el token de sesión de AWS en las credenciales de S3 AppVault.
- Se ha añadido compatibilidad con la recopilación de recursos después de los ganchos de ejecución previos a la instantánea.

Correcciones

- Se mejoró la gestión de volúmenes temporales para omitir la cola de recuperación de volúmenes ONTAP .
- Las anotaciones SCC han sido restauradas a sus valores originales.
- Mejora de la eficiencia de restauración gracias al soporte para operaciones en paralelo.
- Soporte mejorado para tiempos de espera de ganchos de ejecución para aplicaciones de mayor tamaño.

Cambios en 24.10.1

Mejoras

- **Kubernetes**: Se agregó soporte para Kubernetes 1.32.
- Se agregó la detección y el registro del estado de la conexión iSCSI cuando las sesiones iSCSI deberían estar iniciadas pero no lo están ("Número 961").

Correcciones

- Se corrigieron las direcciones IP de nodos faltantes en las políticas de exportación automática ("Número 965").
- Se corrigió el cambio prematuro de las políticas de exportación automática a la política por volumen para ONTAP-NAS-Economy.
- Se actualizaron las dependencias de Trident y Trident-ASUP para abordar CVE-2024-45337 y CVE-2024-45310.
- Se eliminaron los cierres de sesión para portales no CHAP intermitentemente defectuosos durante la autorreparación de iSCSI ("Número 961").

Cambios en 24.10

Mejoras

- El controlador Google Cloud NetApp Volumes ya está disponible de forma general para volúmenes NFS y admite el aprovisionamiento con reconocimiento de zonas.
- La identidad de carga de trabajo de GCP se utilizará como identidad en la nube para los Google Cloud NetApp Volumes con GKE.
- Agregado `formatOptions` Parámetro de configuración para los controladores ONTAP-SAN y ONTAP-SAN-Economy para permitir a los usuarios especificar opciones de formato LUN.
- Se ha reducido el tamaño mínimo del volumen de Azure NetApp Files a 50 GiB. Se espera que el nuevo tamaño mínimo de Azure esté disponible de forma generalizada en noviembre.
- Agregado `denyNewVolumePools` Parámetro de configuración para restringir los controladores ONTAP-NAS-Economy y ONTAP-SAN-Economy a los grupos Flexvol preexistentes.

- Se ha añadido detección para la adición, eliminación o cambio de nombre de agregados del SVM en todos los controladores ONTAP .
- Se agregaron 18 MiB de sobrecarga a los LUN LUKS para garantizar que el tamaño de PVC informado sea utilizable.
- Se ha mejorado el manejo de errores de preparación y despreparación de nodos ONTAP-SAN y ONTAP-SAN-Economy para permitir la despreparación de dispositivos después de una preparación fallida.
- Se ha añadido un generador de roles personalizado que permite a los clientes crear un rol minimalista para Trident en ONTAP.
- Se agregó registro adicional para la resolución de problemas. `lsscsi` ("Número 792").

Kubernetes

- Se han añadido nuevas funciones de Trident para flujos de trabajo nativos de Kubernetes:
  - Protección de datos
  - Migración de datos
  - Recuperación ante desastres
  - Movilidad de la aplicación

  "Obtenga más información sobre Trident Protect".
- Se agregó una nueva bandera `--k8s-api-qps` a los instaladores para que configuren el valor QPS utilizado por Trident para comunicarse con el servidor de la API de Kubernetes.
- Agregado `--node-prep` flag para los instaladores para la gestión automática de las dependencias del protocolo de almacenamiento en los nodos del clúster de Kubernetes. Compatibilidad probada y verificada con el protocolo de almacenamiento iSCSI de Amazon Linux 2023
- Se ha añadido compatibilidad con la desconexión forzada de volúmenes ONTAP-NAS-Economy durante escenarios de apagado de nodo no controlado.
- Los nuevos volúmenes NFS de ONTAP-NAS-Economy utilizarán políticas de exportación por qtree al usar `autoExportPolicy` Opción de backend. Los Qtrees solo se asignarán a políticas de exportación restrictivas de nodos en el momento de la publicación para mejorar el control de acceso y la seguridad. Los qtrees existentes se cambiarán al nuevo modelo de política de exportación cuando Trident despublique el volumen de todos los nodos para hacerlo sin afectar las cargas de trabajo activas.
- Se ha añadido compatibilidad con Kubernetes 1.31.

Mejoras experimentales

- Se ha añadido una vista previa técnica para la compatibilidad con Fibre Channel en el controlador ONTAP-SAN.

Correcciones

- **Kubernetes**:
  - Se corrigió el webhook de admisión de Rancher que impedía las instalaciones de Trident Helm ("Número 839").
  - Clave de afinidad fija en los valores del gráfico de Helm ("Número 898").
  - Se ha corregido el problema con tridentControllerPluginNodeSelector/tridentNodePluginNodeSelector, que no funciona con el valor "true" ("Número 899").

    - Instantáneas efímeras eliminadas creadas durante la clonación ("Número 901").
- Se ha añadido compatibilidad con Windows Server 2019.
- Se corrigió el mod `go tidy` en el repositorio de Trident ("Número 767").

Desusos

- **Kubernetes:**
    - Se actualizó la versión mínima compatible de Kubernetes a 1.25.
    - Se eliminó la compatibilidad con la política de seguridad de POD.

cambio de marca del producto

A partir de la versión 24.10, Astra Trident pasa a llamarse Trident (Netapp Trident). Este cambio de marca no afecta a ninguna de las características, plataformas compatibles ni interoperabilidad de Trident.

Cambios en 24.06

Mejoras

- **IMPORTANTE**: El `limitVolumeSize` Este parámetro ahora limita los tamaños de qtree/LUN en los controladores económicos de ONTAP . Utilice el nuevo `limitVolumePoolSize` parámetro para controlar los tamaños de Flexvol en esos controladores. ("Número 341").
- Se ha añadido la capacidad de autorreparación de iSCSI para iniciar escaneos SCSI por ID de LUN exacto si se están utilizando igroups obsoletos ("Número 883").
- Se ha añadido compatibilidad para permitir operaciones de clonación y redimensionamiento de volúmenes incluso cuando el servidor esté en modo suspendido.
- Se ha añadido la capacidad de que la configuración de registro del controlador Trident , definida por el usuario, se propague a los pods de nodos Trident .
- Se agregó soporte en Trident para usar REST de forma predeterminada en lugar de ONTAPI (ZAPI) para las versiones de ONTAP 9.15.1 y posteriores.
- Se ha añadido compatibilidad con nombres de volumen y metadatos personalizados en los backends de almacenamiento ONTAP para nuevos volúmenes persistentes.
- Mejoró el `azure-netapp-files` Controlador (ANF) para habilitar automáticamente el directorio de instantáneas de forma predeterminada cuando las opciones de montaje NFS estén configuradas para usar la versión 4.x de NFS.
- Se ha añadido compatibilidad con Bottlerocket para volúmenes NFS.
- Se ha añadido compatibilidad con la versión preliminar técnica para Google Cloud NetApp Volumes.

Kubernetes

- Se ha añadido compatibilidad con Kubernetes 1.30.
- Se ha añadido la capacidad para Trident DaemonSet de limpiar montajes zombis y archivos de seguimiento residuales al inicio ("Número 883").
- Anotación de PVC añadida `trident.netapp.io/luksEncryption` para importar dinámicamente volúmenes LUKS ("Número 849").
- Se ha añadido reconocimiento de topología al controlador ANF.

- Se ha añadido compatibilidad con nodos de Windows Server 2022.

Correcciones

- Se han corregido los fallos de instalación de Trident debidos a transacciones obsoletas.
- Se corrigió tridentctl para que ignore los mensajes de advertencia de Kubernetes ("Número 892").
- Controlador Trident modificado `SecurityContextConstraint` prioridad a `0` ("Número 887").
- Los controladores ONTAP ahora aceptan tamaños de volumen inferiores a 20 MiB ("Problema n.º 885").
- Se ha corregido Trident para evitar la reducción de los volúmenes FlexVol durante la operación de cambio de tamaño para el controlador ONTAP-SAN.
- Se corrigió el fallo de importación de volúmenes ANF con NFS v4.1.

Cambios en 24.02

Mejoras

- Se ha añadido compatibilidad con Cloud Identity.
  - AKS con ANF - Azure Workload Identity se utilizará como identidad en la nube.
  - EKS con FSxN: se utilizará el rol de AWS IAM como identidad en la nube.
- Se ha añadido compatibilidad para instalar Trident como complemento en un clúster EKS desde la consola EKS.
- Se ha añadido la capacidad de configurar y deshabilitar la autorreparación de iSCSI ("Número 864").
- Se agregó la funcionalidad de Amazon FSx a los controladores ONTAP para permitir la integración con AWS IAM y SecretsManager, y para permitir que Trident elimine volúmenes FSx con copias de seguridad ("Número 453").

Kubernetes

- Se ha añadido compatibilidad con Kubernetes 1.29.

Correcciones

- Se corrigieron los mensajes de advertencia de ACP cuando ACP no estaba habilitado."Número 866").
- Se agregó una demora de 10 segundos antes de realizar una división de clon durante la eliminación de instantáneas para los controladores ONTAP , cuando un clon está asociado con la instantánea.

Desusos

- Se eliminó el marco de atestación in-toto de los manifiestos de imágenes multiplataforma.

Cambios en 23.10

Correcciones

- Expansión de volumen fija si el nuevo tamaño solicitado es menor que el tamaño total del volumen para los controladores de almacenamiento ontap-nas y ontap-nas-flexgroup ("Número 834").
- Tamaño de volumen fijo para mostrar solo el tamaño utilizable del volumen durante la importación para los controladores de almacenamiento ontap-nas y ontap-nas-flexgroup ("Número 722").

- Se corrigió la conversión de nombres de FlexVol para ONTAP-NAS-Economy.
- Se solucionó un problema de inicialización de Trident en un nodo Windows cuando se reinicia el nodo.

Mejoras

Kubernetes

Se ha añadido compatibilidad con Kubernetes 1.28.

Trident

- Se ha añadido compatibilidad para usar identidades administradas de Azure (AMI) con el controlador de almacenamiento azure-netapp-files.
- Se ha añadido compatibilidad con NVMe sobre TCP para el controlador ONTAP-SAN.
- Se ha añadido la capacidad de pausar el aprovisionamiento de un volumen cuando el usuario configura el backend en estado suspendido ("Número 558").

Cambios en 23.07.1

Kubernetes: Se corrigió la eliminación de daemonset para admitir actualizaciones sin tiempo de inactividad ("Número 740").

Cambios en 23.07

Correcciones

Kubernetes

- Se corrigió la actualización de Trident para que ignore los pods antiguos atascados en estado de terminación ("Número 740").
- Se agregó tolerancia a la definición de "transient-trident-version-pod" ("Número 795").

Trident

- Se corrigieron las solicitudes ONTAPI (ZAPI) para garantizar que se consulten los números de serie de LUN al obtener los atributos de LUN para identificar y corregir dispositivos iSCSI fantasma durante las operaciones de Node Staging.
- Se corrigió el manejo de errores en el código del controlador de almacenamiento ("Número 816").
- Redimensionamiento de cuota fija al usar controladores ONTAP con use-rest=true.
- Se ha corregido la creación de clones LUN en ontap-san-economy.
- Revertir el campo de información de publicación desde `rawDevicePath` a `devicePath` Se agregó lógica para completar y recuperar datos (en algunos casos). `devicePath` campo.

Mejoras

Kubernetes

- Se ha añadido compatibilidad para importar instantáneas preconfiguradas.
- Despliegue minimizado y permisos de daemonset en Linux ("Número 817").

Trident

- Ya no se informa del campo de estado para volúmenes y instantáneas "en línea".
- Actualiza el estado del backend si el backend de ONTAP está fuera de línea ("Número 801" ,"#543").
- El número de serie de LUN siempre se recupera y publica durante el flujo de trabajo ControllerVolumePublish.
- Se agregó lógica adicional para verificar el número de serie y el tamaño del dispositivo iSCSI multipath.
- Verificación adicional para volúmenes iSCSI para garantizar que el dispositivo multipath correcto no esté configurado por etapas.

Mejora experimental

Se ha añadido soporte de vista previa técnica para NVMe sobre TCP para el controlador ONTAP-SAN.

Documentación

Se han realizado numerosas mejoras organizativas y de formato.

Desusos

Kubernetes

- Se eliminó la compatibilidad con las instantáneas v1beta1.
- Se eliminó la compatibilidad con volúmenes y clases de almacenamiento anteriores a CSI.
- Se actualizó la versión mínima compatible de Kubernetes a 1.22.

Cambios en 23.04

> La desconexión forzada de volúmenes ONTAP-SAN-* solo es compatible con versiones de Kubernetes con la puerta de función Non-Graceful Node Shutdown habilitada. La desconexión forzada debe habilitarse durante la instalación mediante `--enable-force-detach` Bandera del instalador de Trident .

Correcciones

- Se ha corregido el operador Trident para que utilice IPv6 localhost para la instalación cuando se especifique en la especificación.
- Se corrigieron los permisos del rol de clúster de operador de Trident para que estuvieran sincronizados con los permisos del paquete ("Número 799").
- Se solucionó un problema al adjuntar volúmenes de bloques sin formato en varios nodos en modo RWX.
- Se ha corregido la compatibilidad con la clonación de FlexGroup y la importación de volúmenes para volúmenes SMB.
- Se solucionó un problema por el cual el controlador Trident no se podía apagar inmediatamente ("Número 811").
- Se agregó una corrección para listar todos los nombres de igroup asociados con un LUN específico aprovisionado con controladores ontap-san-*.
- Se ha añadido una solución para permitir que los procesos externos se ejecuten hasta su finalización.
- Se corrigió el error de compilación para la arquitectura s390 ("Número 537").

- Se corrigió el nivel de registro incorrecto durante las operaciones de montaje de volumen ("Número 781").
- Se corrigió un posible error de aserción de tipo ("Número 802").

Mejoras

- Kubernetes:
  - Se ha añadido compatibilidad con Kubernetes 1.27.
  - Se ha añadido compatibilidad para importar volúmenes LUKS.
  - Se ha añadido compatibilidad con el modo de acceso a PVC ReadWriteOncePod.
  - Se agregó soporte para la desconexión forzada de volúmenes ONTAP-SAN-* durante escenarios de apagado de nodo no ordenado.
  - Todos los volúmenes ONTAP-SAN-* ahora utilizarán igroups por nodo. Las LUN solo se asignarán a igroups mientras estén activamente publicadas en esos nodos para mejorar nuestra postura de seguridad. Los volúmenes existentes se cambiarán oportunamente al nuevo esquema igroup cuando Trident determine que es seguro hacerlo sin afectar las cargas de trabajo activas ("Número 758").
  - Se mejoró la seguridad de Trident limpiando los igroups no utilizados administrados por Trident de los backends ONTAP-SAN-*.
- Se ha añadido compatibilidad con volúmenes SMB con Amazon FSx a los controladores de almacenamiento ontap-nas-economy y ontap-nas-flexgroup.
- Se ha añadido compatibilidad con recursos compartidos SMB con los controladores de almacenamiento ontap-nas, ontap-nas-economy y ontap-nas-flexgroup.
- Se agregó soporte para nodos arm64 ("Número 732").
- Procedimiento de apagado de Trident mejorado mediante la desactivación previa de los servidores API ("Número 811").
- Se agregó compatibilidad con compilación multiplataforma para hosts Windows y arm64 al archivo Makefile; consulte BUILD.md.

Desusos

Kubernetes: Ya no se crearán igroups con ámbito de backend al configurar los controladores ontap-san y ontap-san-economy ("Número 758").

Cambios en 23.01.1

Correcciones

- Se ha corregido el operador Trident para que utilice IPv6 localhost para la instalación cuando se especifique en la especificación.
- Se corrigieron los permisos del rol de clúster de operador de Trident para que estuvieran sincronizados con los permisos del paquete."Número 799" .
- Se ha añadido una solución para permitir que los procesos externos se ejecuten hasta su finalización.
- Se solucionó un problema al adjuntar volúmenes de bloques sin formato en varios nodos en modo RWX.
- Se ha corregido la compatibilidad con la clonación de FlexGroup y la importación de volúmenes para volúmenes SMB.

Cambios en 23.01

Kubernetes 1.27 ahora es compatible con Trident. Por favor, actualice Trident antes de actualizar Kubernetes.

Correcciones

- Kubernetes: Se agregaron opciones para excluir la creación de la Política de Seguridad de Pods para corregir las instalaciones de Trident a través de Helm ("Números #783, #794").

Mejoras

Kubernetes

- Se ha añadido compatibilidad con Kubernetes 1.26.
- Mejora de la utilización general de los recursos del RBAC de Trident ("Número 757").
- Se ha añadido automatización para detectar y corregir sesiones iSCSI rotas o obsoletas en los nodos host.
- Se ha añadido compatibilidad para ampliar volúmenes cifrados con LUKS.
- Kubernetes: Se ha añadido compatibilidad con la rotación de credenciales para volúmenes cifrados con LUKS.

Trident

- Se ha añadido compatibilidad con volúmenes SMB con Amazon FSx for NetApp ONTAP al controlador de almacenamiento ontap-nas.
- Se ha añadido compatibilidad con permisos NTFS al usar volúmenes SMB.
- Se ha añadido compatibilidad con pools de almacenamiento para volúmenes de GCP con nivel de servicio CVS.
- Se ha añadido compatibilidad para el uso opcional de flexgroupAggregateList al crear FlexGroups con el controlador de almacenamiento ontap-nas-flexgroup.
- Mejora del rendimiento del controlador de almacenamiento ontap-nas-economy al gestionar varios volúmenes FlexVol.
- Se habilitaron las actualizaciones dataLIF para todos los controladores de almacenamiento NAS de ONTAP .
- Se actualizó la convención de nomenclatura de Trident Deployment y DaemonSet para reflejar el sistema operativo del nodo host.

Desusos

- Kubernetes: Se actualizó la versión mínima compatible de Kubernetes a la 1.21.
- Ya no se deben especificar los DataLIF al configurar `ontap-san` o `ontap-san-economy` conductores.

Cambios en 22.10

Debe leer la siguiente información crítica antes de actualizar a Trident 22.10.

Información crítica sobre Trident 22.10

- Kubernetes 1.25 ahora es compatible con Trident. Debe actualizar Trident a la versión 22.10 antes de actualizar a Kubernetes 1.25.
- Trident ahora impone estrictamente el uso de la configuración de rutas múltiples en entornos SAN, con un valor recomendado de `find_multipaths: no` en el archivo multipath.conf.

  Uso de una configuración sin rutas múltiples o uso de `find_multipaths: yes` o `find_multipaths: smart` Un valor incorrecto en el archivo multipath.conf provocará fallos de montaje. Trident ha recomendado el uso de `find_multipaths: no` Desde el lanzamiento del 21.07.

Correcciones

- Se solucionó un problema específico del backend de ONTAP creado mediante `credentials` El campo no se conectó durante la actualización 22.07.0 ("Número 759").
- **Docker:** Se solucionó un problema que provocaba que el complemento de volumen de Docker no se iniciara en algunos entornos ("Número 548" y"Número 760").
- Se solucionó un problema de SLM específico de los backends SAN de ONTAP para garantizar que solo se publique un subconjunto de dataLIF pertenecientes a los nodos de informes.
- Se solucionó un problema de rendimiento que provocaba escaneos innecesarios de LUN iSCSI al conectar un volumen.
- Se eliminaron los reintentos granulares dentro del flujo de trabajo iSCSI de Trident para detectar fallos rápidamente y reducir los intervalos de reintento externos.
- Se solucionó un problema por el cual se devolvía un error al vaciar un dispositivo iSCSI cuando el dispositivo multipath correspondiente ya se había vaciado.

Mejoras

- Kubernetes:
  - Se ha añadido compatibilidad con Kubernetes 1.25. Debe actualizar Trident a la versión 22.10 antes de actualizar a Kubernetes 1.25.
  - Se agregó una cuenta de servicio, un rol de clúster y un enlace de rol de clúster independientes para la implementación de Trident y el DaemonSet para permitir futuras mejoras de permisos.
  - Se agregó soporte para"Compartición de volúmenes entre espacios de nombres" .
- Todo Trident `ontap-*` Los controladores de almacenamiento ahora funcionan con la API REST de ONTAP .
- Se agregó un nuevo operador yaml(`bundle_post_1_25.yaml`) sin un `PodSecurityPolicy` para dar soporte a Kubernetes 1.25.
- Agregado"Compatibilidad con volúmenes cifrados con LUKS" para `ontap-san` y `ontap-san-economy` Controladores de almacenamiento.
- Se ha añadido compatibilidad con nodos de Windows Server 2019.
- Agregado"Compatibilidad con volúmenes SMB en nodos Windows" A través de la `azure-netapp-files` Controlador de almacenamiento.
- La detección automática de conmutación por error de MetroCluster para controladores ONTAP ya está

disponible de forma general.

Desusos

- **Kubernetes:** Se actualizó la versión mínima compatible de Kubernetes a 1.20.
- Se eliminó el controlador Astra Data Store (ADS).
- Se eliminó el soporte para `yes` y `smart` opciones para `find_multipaths` al configurar el multipathing de nodos de trabajo para iSCSI.

Cambios en 22.07

Correcciones

Kubernetes

- Se solucionó un problema que impedía manejar valores booleanos y numéricos para el selector de nodos al configurar Trident con Helm o el operador de Trident . ("Problema de GitHub n.º 700")
- Se ha solucionado un problema en el manejo de errores procedentes de rutas no CHAP, de modo que kubelet volverá a intentarlo si falla. "Problema de GitHub n.º 736")

Mejoras

- Transición de k8s.gcr.io a registry.k8s.io como registro predeterminado para imágenes CSI
- Los volúmenes ONTAP-SAN ahora utilizarán igroups por nodo y solo asignarán LUN a igroups mientras estén publicados activamente en esos nodos para mejorar nuestra postura de seguridad. Los volúmenes existentes se cambiarán oportunamente al nuevo esquema igroup cuando Trident determine que es seguro hacerlo sin afectar las cargas de trabajo activas.
- Se incluyó una cuota de recursos con las instalaciones de Trident para garantizar que el DaemonSet de Trident se programe cuando el consumo de PriorityClass esté limitado de forma predeterminada.
- Se ha añadido compatibilidad con las funciones de red al controlador de Azure NetApp Files . ("Problema de GitHub n.º 717")
- Se agregó la detección automática de conmutación por error de MetroCluster en versión preliminar técnica a los controladores ONTAP . ("Problema de GitHub n.º 228")

Desusos

- **Kubernetes:** Se actualizó la versión mínima compatible de Kubernetes a 1.19.
- La configuración del backend ya no permite múltiples tipos de autenticación en una sola configuración.

Mudanzas

- Se ha eliminado el controlador AWS CVS (obsoleto desde la versión 22.04).
- Kubernetes
  - Se eliminó la capacidad innecesaria SYS_ADMIN de los pods de nodos.
  - Reduce nodeprep a información simple del host y detección activa de servicios para realizar una confirmación, en el mejor de los casos, de que los servicios NFS/iSCSI están disponibles en los nodos de trabajo.

Documentación

Un nuevo"Estándares de seguridad de pods" Se ha añadido la sección (PSS) que detalla los permisos habilitados por Trident durante la instalación.

Cambios en 22.04

NetApp mejora y optimiza continuamente sus productos y servicios. Estas son algunas de las últimas características de Trident. Para versiones anteriores, consulte "Versiones anteriores de la documentación" .

> Si está actualizando desde cualquier versión anterior de Trident y utiliza Azure NetApp Files,`location` El parámetro de configuración ahora es un campo obligatorio y único.

Correcciones

- Mejora del análisis sintáctico de los nombres de iniciadores iSCSI. ("Problema de GitHub n.º 681")
- Se ha solucionado un problema por el cual no se permitían los parámetros de la clase de almacenamiento CSI. ("Problema de GitHub n.º 598")
- Se corrigió la declaración de clave duplicada en Trident CRD. ("Problema de GitHub n.º 671")
- Se corrigieron los registros de instantáneas CSI inexactos. ("Problema de GitHub n.º 629"))
- Se solucionó un problema relacionado con la anulación de la publicación de volúmenes en nodos eliminados. ("Problema de GitHub n.º 691")
- Se ha añadido la gestión de inconsistencias del sistema de archivos en dispositivos de bloque. ("Problema de GitHub n.º 656")
- Se solucionó un problema al extraer automáticamente las imágenes de soporte al configurar `imageRegistry` bandera durante la instalación. ("Problema de GitHub n.º 715")
- Se ha solucionado un problema por el cual el controlador Azure NetApp Files no podía clonar un volumen con varias reglas de exportación.

Mejoras

- Las conexiones entrantes a los puntos de conexión seguros de Trident ahora requieren un mínimo de TLS 1.3. ("Problema de GitHub n.º 698")
- Trident ahora agrega encabezados HSTS a las respuestas de sus puntos de conexión seguros.
- Trident ahora intenta habilitar automáticamente la función de permisos Unix de Azure NetApp Files .
- **Kubernetes**: El daemonset Trident ahora se ejecuta en la clase de prioridad system-node-critical. ("Problema de GitHub n.º 694")

Mudanzas

Se ha eliminado el controlador de la serie E (deshabilitado desde la versión 20.07).

Cambios en 22.01.1

Correcciones

- Se solucionó un problema relacionado con la anulación de la publicación de volúmenes en nodos eliminados. ("Problema de GitHub n.º 691")

- Se corrigió el error que se producía al acceder a campos nulos para el espacio agregado en las respuestas de la API de ONTAP .

Cambios en 22.01.0

Correcciones

- **Kubernetes:** Aumentar el tiempo de reintento de registro de nodos para clústeres grandes.
- Se solucionó un problema por el cual el controlador azure-netapp-files podía confundirse con varios recursos con el mismo nombre.
- Los DataLIF IPv6 de ONTAP SAN ahora funcionan si se especifican entre corchetes.
- Se solucionó un problema por el cual, al intentar importar un volumen ya importado, se devolvía EOF y el PVC quedaba en estado pendiente. ("Problema de GitHub n.º 489")
- Se solucionó un problema por el cual el rendimiento de Trident se ralentizaba cuando se creaban más de 32 instantáneas en un volumen SolidFire .
- Se reemplazó SHA-1 con SHA-256 en la creación del certificado SSL.
- Se corrigió el controlador de Azure NetApp Files para permitir nombres de recursos duplicados y limitar las operaciones a una sola ubicación.
- Se corrigió el controlador de Azure NetApp Files para permitir nombres de recursos duplicados y limitar las operaciones a una sola ubicación.

Mejoras

- Mejoras de Kubernetes:
  - Se ha añadido compatibilidad con Kubernetes 1.23.
  - Agregar opciones de programación para los pods de Trident cuando se instalen mediante Trident Operator o Helm. ("Problema de GitHub n.º 651")
- Habilitar volúmenes entre regiones en el controlador de GCP. ("Problema de GitHub n.º 633")
- Se ha añadido compatibilidad con la opción 'unixPermissions' a los volúmenes de Azure NetApp Files . ("Problema de GitHub n.º 666")

Desusos

La interfaz REST de Trident solo puede escuchar y servir en la dirección 127.0.0.1 o en la dirección [::1].

Cambios en 21.10.1

> La versión v21.10.0 tiene un problema que puede hacer que el controlador Trident entre en un estado CrashLoopBackOff cuando se elimina un nodo y luego se vuelve a agregar al clúster de Kubernetes. Este problema se solucionó en la versión 21.10.1 (problema 669 de GitHub).

Correcciones

- Se corrigió una posible condición de carrera al importar un volumen en un backend de GCP CVS que provocaba un fallo en la importación.
- Se solucionó un problema que podía provocar que el controlador Trident entrara en un estado CrashLoopBackOff cuando se eliminaba un nodo y luego se volvía a agregar al clúster de Kubernetes (problema de GitHub 669).

- Se solucionó un problema por el cual las SVM ya no se detectaban si no se especificaba un nombre de SVM (problema de GitHub 612).

Cambios en 21.10.0

Correcciones

- Se solucionó un problema por el cual los clones de volúmenes XFS no se podían montar en el mismo nodo que el volumen de origen (problema 514 de GitHub).
- Se solucionó un problema por el cual Trident registraba un error fatal al apagarse (problema de GitHub 597).
- Correcciones relacionadas con Kubernetes:
  - Devuelve el espacio utilizado de un volumen como el tamaño mínimo de restauración al crear instantáneas con `ontap-nas` y `ontap-nas-flexgroup` controladores (problema 645 de GitHub).
  - Se solucionó un problema donde `Failed to expand filesystem` Se registró un error después de cambiar el tamaño del volumen (problema 560 de GitHub).
  - Se solucionó un problema por el cual un pod podía quedarse atascado. `Terminating` estado (problema 572 de GitHub).
  - Se corrigió el caso en el que un `ontap-san-economy` FlexVol podría estar lleno de LUN de instantáneas (problema 533 de GitHub).
  - Se solucionó un problema con el instalador YAML personalizado que utilizaba una imagen diferente (problema de GitHub 613).
  - Se corrigió el cálculo del tamaño de la instantánea (problema 611 de GitHub).
  - Se solucionó un problema por el cual todos los instaladores de Trident podían identificar Kubernetes simple como OpenShift (problema de GitHub 639).
  - Se corrigió el operador Trident para detener la reconciliación si el servidor de la API de Kubernetes no está disponible (problema 599 de GitHub).

Mejoras

- Se agregó soporte para `unixPermissions` Opción para volúmenes de rendimiento de GCP-CVS.
- Se ha añadido compatibilidad con volúmenes CVS optimizados para escalado en GCP en el rango de 600 GiB a 1 TiB.
- Mejoras relacionadas con Kubernetes:
  - Se ha añadido compatibilidad con Kubernetes 1.22.
  - Se habilitó el operador Trident y el gráfico Helm para que funcionen con Kubernetes 1.22 (problema de GitHub 628).
  - Se agregó la imagen del operador a `tridentctl` Comando de imágenes (problema 570 de GitHub).

Mejoras experimentales

- Se agregó compatibilidad con la replicación de volúmenes en el `ontap-san` conductor.
- Se agregó compatibilidad con REST en versión preliminar para el **segmento REST** `ontap-nas-flexgroup` , `ontap-san` , y `ontap-nas-economy` conductores.

Problemas conocidos

Los problemas conocidos identifican los inconvenientes que podrían impedirle utilizar el producto correctamente.

- 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.
- Trident ahora impone un espacio en blanco `fsType` (`fsType=""`) para volúmenes que no tienen el `fsType` especificado en su StorageClass. Al trabajar con Kubernetes 1.17 o posterior, Trident admite proporcionar un espacio en blanco. `fsType` para volúmenes NFS. Para los volúmenes iSCSI, es necesario configurar el `fsType` en su StorageClass al aplicar un `fsGroup` utilizando un contexto de seguridad.
- Al utilizar un backend en varias instancias de Trident , cada archivo de configuración del backend debe tener un archivo diferente. `storagePrefix` valor para backends de ONTAP o usar uno diferente `TenantName` para backends de SolidFire . Trident no puede detectar volúmenes creados por otras instancias de Trident . El intento de crear un volumen existente en los sistemas backend ONTAP o SolidFire tiene éxito, porque Trident trata la creación de volúmenes como una operación idempotente. Si `storagePrefix` o `TenantName` No difieren, podrían producirse colisiones de nombres para volúmenes creados en el mismo backend.
- Al instalar Trident (usando `tridentctl` o el Operador Trident) y utilizando `tridentctl` Para gestionar Trident, debes asegurarte de que `KUBECONFIG` La variable de entorno está configurada. Esto es necesario para indicar al clúster de Kubernetes que `tridentctl` debería funcionar en contra. Al trabajar con múltiples entornos de Kubernetes, debe asegurarse de que `KUBECONFIG` El archivo proviene de la fuente correcta.
- Para realizar la recuperación de espacio en línea para volúmenes persistentes iSCSI, el sistema operativo subyacente en el nodo de trabajo podría requerir que se pasen opciones de montaje al volumen. Esto es cierto para las instancias de RHEL/Red Hat Enterprise Linux CoreOS (RHCOS), que requieren `discard` "opción de montaje" Asegúrese de que la opción de montaje de descarte esté incluida en su[`StorageClass` ^] para admitir el descarte de bloques en línea.
- Si tiene más de una instancia de Trident por clúster de Kubernetes, Trident no puede comunicarse con otras instancias ni descubrir otros volúmenes que hayan creado, lo que provoca un comportamiento inesperado e incorrecto si se ejecuta más de una instancia dentro de un clúster. Solo debería haber una instancia de Trident por clúster de Kubernetes.
- Si se basa en Trident `StorageClass` Los objetos se eliminan de Kubernetes mientras Trident está fuera de línea; Trident no elimina las clases de almacenamiento correspondientes de su base de datos cuando vuelve a estar en línea. Debe eliminar estas clases de almacenamiento usando `tridentctl` o la API REST.
- Si un usuario elimina un PV aprovisionado por Trident antes de eliminar el PVC correspondiente, Trident no elimina automáticamente el volumen de respaldo. Debe eliminar el volumen mediante `tridentctl` o la API REST.
- ONTAP no puede aprovisionar simultáneamente más de un FlexGroup a la vez a menos que el conjunto de agregados sea único para cada solicitud de aprovisionamiento.
- Al usar Trident sobre IPv6, debe especificar `managementLIF` y `dataLIF` en la definición del backend entre corchetes. Por ejemplo, `[fd20:8b1e:b258:2000:f816:3eff:feec:0]` .

> No puedes especificar `dataLIF` en un backend SAN ONTAP . Trident descubre todas las LIF iSCSI disponibles y las utiliza para establecer la sesión de múltiples rutas.

- Si utiliza el `solidfire-san` En el controlador con OpenShift 4.5, asegúrese de que los nodos de trabajo subyacentes utilicen MD5 como algoritmo de autenticación CHAP. Los algoritmos CHAP seguros compatibles con FIPS SHA1, SHA-256 y SHA3-256 están disponibles con Element 12.7.

Encuentra más información

- "GitHub de Trident"
- "Blogs de Trident"

Versiones anteriores de la documentación

Si no está utilizando Trident 25.06, la documentación de las versiones anteriores está disponible en función de"ciclo de vida del soporte Trident" .

- "Trident 25.02"
- "Trident 24.10"
- "Trident 24.06"
- "Trident 24.02"
- "Trident 23.10"
- "Trident 23.07"
- "Trident 23.04"
- "Trident 23.01"
- "Trident 22.10"

Problemas conocidos

Los problemas conocidos identifican problemas que podrían impedirle utilizar esta versión del producto con éxito.

Los siguientes problemas conocidos afectan a la versión actual:

La restauración de copias de seguridad de archivos grandes de Restic puede fallar.

Al restaurar archivos de 30 GB o más grandes desde una copia de seguridad de Amazon S3 realizada con Restic, la operación de restauración puede fallar. Como solución alternativa, realice una copia de seguridad de los datos utilizando Kopia como gestor de copias de seguridad (Kopia es el gestor de copias de seguridad predeterminado). Referirse a "Proteja las aplicaciones con Trident Protect" para obtener instrucciones.

Empezar

Aprende sobre Trident

Aprende sobre Trident

Trident es un proyecto de código abierto con soporte completo mantenido por NetApp. Ha sido diseñado para ayudarle a satisfacer las demandas de persistencia de su aplicación en contenedores utilizando interfaces estándar de la industria, como la Interfaz de Almacenamiento de Contenedores (CSI).

¿Qué es Trident?

NetApp Trident permite el consumo y la gestión de recursos de almacenamiento en todas las plataformas de almacenamiento NetApp más populares, tanto en la nube pública como en las instalaciones, incluidos los clústeres ONTAP locales (AFF, FAS y ASA), ONTAP Select, Cloud Volumes ONTAP, el software Element (NetApp HCI, SolidFire), Azure NetApp Files, Amazon FSx for NetApp ONTAP y Cloud Volumes Service en Google Cloud.

Trident es un orquestador de almacenamiento dinámico compatible con la interfaz de almacenamiento de contenedores (CSI) que se integra de forma nativa con"Kubernetes" . Trident se ejecuta como un único Pod de controlador más un Pod de nodo en cada nodo de trabajo del clúster. Referirse a "Arquitectura Trident" Para más detalles.

Trident también proporciona integración directa con el ecosistema Docker para las plataformas de almacenamiento NetApp . El complemento NetApp Docker Volume (nDVP) admite el aprovisionamiento y la gestión de recursos de almacenamiento desde la plataforma de almacenamiento a los hosts de Docker. Referirse a "Implementar Trident para Docker" Para más detalles.

> Si esta es la primera vez que usa Kubernetes, debería familiarizarse con…"Conceptos y herramientas de Kubernetes" .

Integración de Kubernetes con productos NetApp

La cartera de productos de almacenamiento de NetApp se integra con muchos aspectos de un clúster de Kubernetes, proporcionando capacidades avanzadas de gestión de datos que mejoran la funcionalidad, la capacidad, el rendimiento y la disponibilidad de la implementación de Kubernetes.

Amazon FSx for NetApp ONTAP

"Amazon FSx for NetApp ONTAP"Es un servicio de AWS totalmente administrado que le permite lanzar y ejecutar sistemas de archivos basados en el sistema operativo de almacenamiento NetApp ONTAP .

Azure NetApp Files

"Azure NetApp Files"Es un servicio de compartición de archivos de Azure de nivel empresarial, impulsado por NetApp. Puede ejecutar sus cargas de trabajo basadas en archivos más exigentes en Azure de forma nativa, con el rendimiento y la rica gestión de datos que espera de NetApp.

Cloud Volumes ONTAP

"Cloud Volumes ONTAP"Es un dispositivo de almacenamiento exclusivamente de software que ejecuta el software de gestión de datos ONTAP en la nube.

Google Cloud NetApp Volumes

"Google Cloud NetApp Volumes"Es un servicio de almacenamiento de archivos totalmente administrado en Google Cloud que proporciona almacenamiento de archivos de alto rendimiento y nivel empresarial.

Software Element

"Elemento"Permite al administrador de almacenamiento consolidar las cargas de trabajo garantizando el rendimiento y posibilitando una infraestructura de almacenamiento simplificada y optimizada.

HCI de NetApp

"HCI de NetApp"Simplifica la gestión y la escalabilidad del centro de datos al automatizar las tareas rutinarias y permitir que los administradores de infraestructura se centren en funciones más importantes.

Trident puede aprovisionar y administrar dispositivos de almacenamiento para aplicaciones en contenedores directamente en la plataforma de almacenamiento NetApp HCI subyacente.

ONTAP de NetApp

"ONTAP de NetApp"es el sistema operativo de almacenamiento unificado y multiprotocolo de NetApp que proporciona capacidades avanzadas de gestión de datos para cualquier aplicación.

Los sistemas ONTAP tienen configuraciones totalmente flash, híbridas o totalmente HDD y ofrecen muchos modelos de implementación diferentes: clústeres FAS, AFA y ASA locales, ONTAP Select y Cloud Volumes ONTAP. Trident admite estos modelos de implementación de ONTAP .

Arquitectura Trident

Trident se ejecuta como un único Pod de controlador más un Pod de nodo en cada nodo de trabajo del clúster. El pod del nodo debe estar ejecutándose en cualquier host donde se desee potencialmente montar un volumen Trident .

Comprender los pods del controlador y los pods de nodo

Trident se despliega como un soloCápsula de controlador Trident y uno o másCápsulas de nodo Trident en el clúster de Kubernetes y utiliza contenedores laterales CSI estándar de Kubernetes para simplificar la implementación de complementos CSI. "Contenedores laterales CSI de Kubernetes" son mantenidas por la comunidad de almacenamiento de Kubernetes.

Kubernetes"selectores de nodos" y"tolerancias y máculas" Se utilizan para restringir un pod a ejecutarse en un nodo específico o preferido. Durante la instalación de Trident , puede configurar los selectores de nodos y las tolerancias para los pods del controlador y los nodos.

- El plugin del controlador gestiona el aprovisionamiento y la administración de volúmenes, como las instantáneas y el cambio de tamaño.
- El plugin del nodo se encarga de conectar el almacenamiento al nodo.

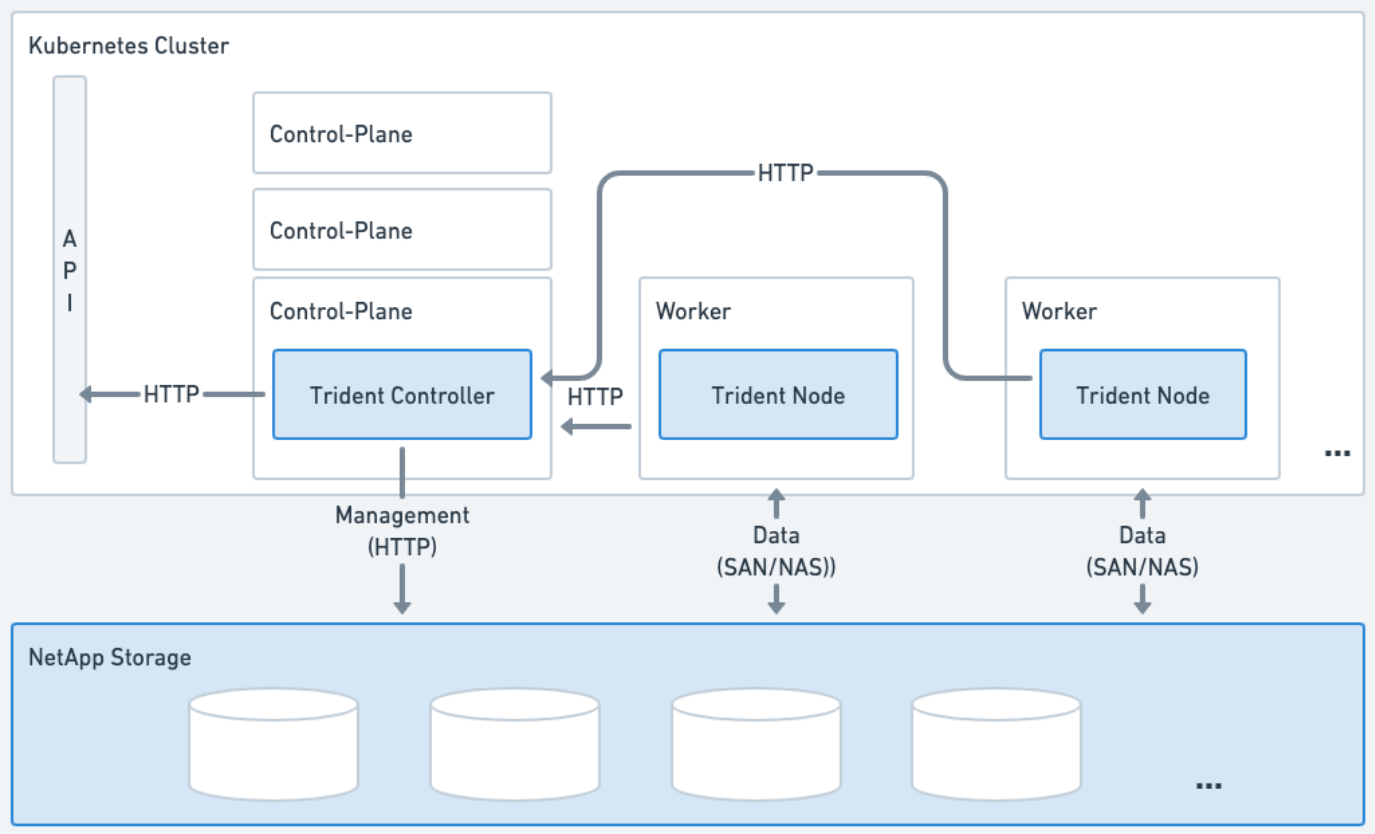


Figura 1. Trident se implementó en el clúster de Kubernetes.

Cápsula de controlador Trident

El Trident Controller Pod es un único Pod que ejecuta el plugin CSI Controller.

- Responsable del aprovisionamiento y la gestión de volúmenes en el almacenamiento NetApp
- Gestionado por una implementación de Kubernetes
- Puede ejecutarse en el plano de control o en los nodos de trabajo, dependiendo de los parámetros de instalación.

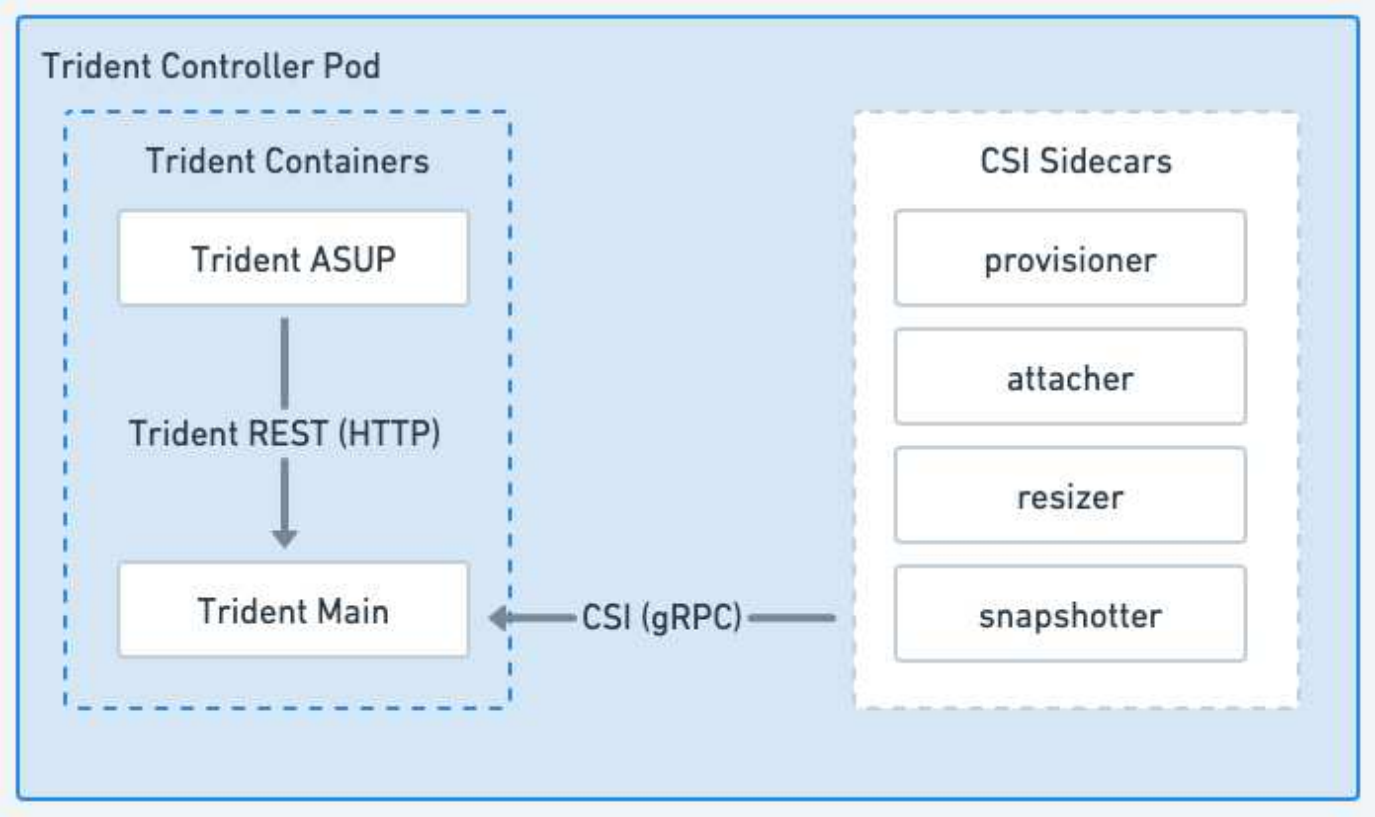


Figura 2. Diagrama del módulo controlador Trident

Cápsulas de nodo Trident

Los pods de nodo Trident son pods privilegiados que ejecutan el complemento de nodo CSI.

- Responsable del montaje y desmontaje del almacenamiento para los Pods que se ejecutan en el host.
- Gestionado por un DaemonSet de Kubernetes
- Debe ejecutarse en cualquier nodo que admita el montaje de almacenamiento NetApp .

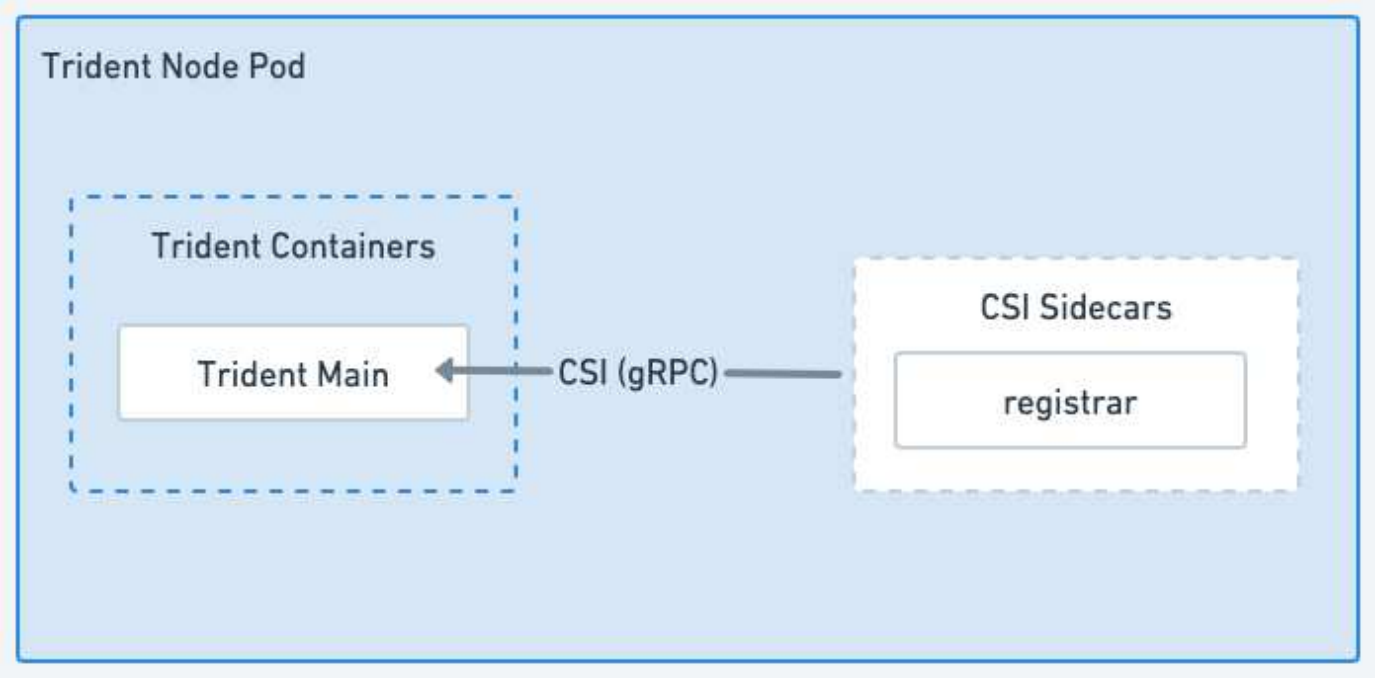


Figura 3. Diagrama del nodo Trident

Arquitecturas de clúster de Kubernetes compatibles

Trident es compatible con las siguientes arquitecturas de Kubernetes:

Arquitecturas de clústeres de Kubernetes	Apoyado	Instalación predeterminada
Maestro único, computador	Sí	Sí
Múltiples maestros, computadores	Sí	Sí
Maestro, `etcd` , calcular	Sí	Sí
Maestro, infraestructura, computación	Sí	Sí

Conceptos

Aprovisionamiento

El aprovisionamiento en Trident tiene dos fases principales. La primera fase asocia una clase de almacenamiento con el conjunto de pools de almacenamiento backend adecuados y se produce como una preparación necesaria antes del aprovisionamiento. La segunda fase incluye la creación del volumen propiamente dicha y requiere elegir un grupo de almacenamiento de entre los asociados a la clase de almacenamiento del volumen pendiente.

Asociación de clase de almacenamiento

La asociación de grupos de almacenamiento de backend con una clase de almacenamiento depende tanto de los atributos solicitados de la clase de almacenamiento como de su `storagePools` ,

`additionalStoragePools` , y `excludeStoragePools` liza. Cuando se crea una clase de almacenamiento, Trident compara los atributos y los grupos que ofrece cada uno de sus backends con los que solicita la clase de almacenamiento. Si los atributos y el nombre de un grupo de almacenamiento coinciden con todos los atributos y nombres de grupo solicitados, Trident agrega ese grupo de almacenamiento al conjunto de grupos de almacenamiento adecuados para esa clase de almacenamiento. Además, Trident agrega todos los grupos de almacenamiento enumerados en el `additionalStoragePools` incluir en esa lista, incluso si sus atributos no cumplen con todos o alguno de los atributos solicitados por la clase de almacenamiento. Debes usar el `excludeStoragePools` lista para anular y eliminar grupos de almacenamiento de uso para una clase de almacenamiento. Trident realiza un proceso similar cada vez que se agrega un nuevo backend, comprobando si sus grupos de almacenamiento satisfacen los de las clases de almacenamiento existentes y eliminando cualquiera que haya sido marcado como excluido.

Creación de volumen

A continuación, Trident utiliza las asociaciones entre clases de almacenamiento y grupos de almacenamiento para determinar dónde aprovisionar volúmenes. Cuando se crea un volumen, Trident primero obtiene el conjunto de grupos de almacenamiento para la clase de almacenamiento de ese volumen y, si se especifica un protocolo para el volumen, Trident elimina aquellos grupos de almacenamiento que no pueden proporcionar el protocolo solicitado (por ejemplo, un backend NetApp HCI/ SolidFire no puede proporcionar un volumen basado en archivos, mientras que un backend ONTAP NAS no puede proporcionar un volumen basado en bloques). Trident aleatoriza el orden de este conjunto resultante para facilitar una distribución uniforme de los volúmenes y, a continuación, itera a través de él, intentando aprovisionar el volumen en cada grupo de almacenamiento por turno. Si tiene éxito en una de las pruebas, devuelve un resultado exitoso, registrando cualquier fallo encontrado en el proceso. Trident devuelve un error **solo** si no puede aprovisionar en **todos** los grupos de almacenamiento disponibles para la clase de almacenamiento y el protocolo solicitados.

Instantáneas de volumen

Obtenga más información sobre cómo Trident gestiona la creación de instantáneas de volumen para sus controladores.

Aprenda sobre la creación de instantáneas de volumen

- Para el `ontap-nas` , `ontap-san` , `gcp-cvs` , y `azure-netapp-files` En cada controlador, cada volumen persistente (PV) se asigna a un FlexVol volume . Como resultado, las instantáneas de volumen se crean como instantáneas de NetApp . La tecnología de instantáneas de NetApp ofrece mayor estabilidad, escalabilidad, capacidad de recuperación y rendimiento que las tecnologías de instantáneas de la competencia. Estas copias instantáneas son extremadamente eficientes tanto en el tiempo necesario para crearlas como en el espacio de almacenamiento.
- Para el `ontap-nas-flexgroup` controlador, cada Volumen Persistente (PV) se asigna a un FlexGroup. Como resultado, las instantáneas de volumen se crean como instantáneas de NetApp FlexGroup . La tecnología de instantáneas de NetApp ofrece mayor estabilidad, escalabilidad, capacidad de recuperación y rendimiento que las tecnologías de instantáneas de la competencia. Estas copias instantáneas son extremadamente eficientes tanto en el tiempo necesario para crearlas como en el espacio de almacenamiento.
- Para el `ontap-san-economy` El controlador asigna los PV a las LUN creadas en volúmenes FlexVol compartidos. Las instantáneas de volumen de los PV se obtienen realizando FlexClones de la LUN asociada. La tecnología ONTAP FlexClone permite crear copias de incluso los conjuntos de datos más grandes de forma casi instantánea. Las copias comparten bloques de datos con sus padres, sin consumir almacenamiento excepto el necesario para los metadatos.
- Para el `solidfire-san` controlador, cada PV se asigna a un LUN creado en el software NetApp Element

/clúster NetApp HCI . Las instantáneas de volumen están representadas por instantáneas de elementos del LUN subyacente. Estas instantáneas son copias puntuales y solo ocupan una pequeña cantidad de recursos del sistema y espacio.

- Al trabajar con el `ontap-nas` y `ontap-san` Los controladores, las instantáneas de ONTAP son copias puntuales del FlexVol y consumen espacio en el propio FlexVol . Esto puede provocar que la cantidad de espacio de escritura en el volumen se reduzca con el tiempo a medida que se crean o programan instantáneas. Una forma sencilla de abordar esto es aumentar el volumen redimensionándolo a través de Kubernetes. Otra opción es eliminar las instantáneas que ya no son necesarias. Cuando se elimina un VolumeSnapshot creado a través de Kubernetes, Trident eliminará la instantánea ONTAP asociada. También se pueden eliminar las instantáneas de ONTAP que no se hayan creado a través de Kubernetes.

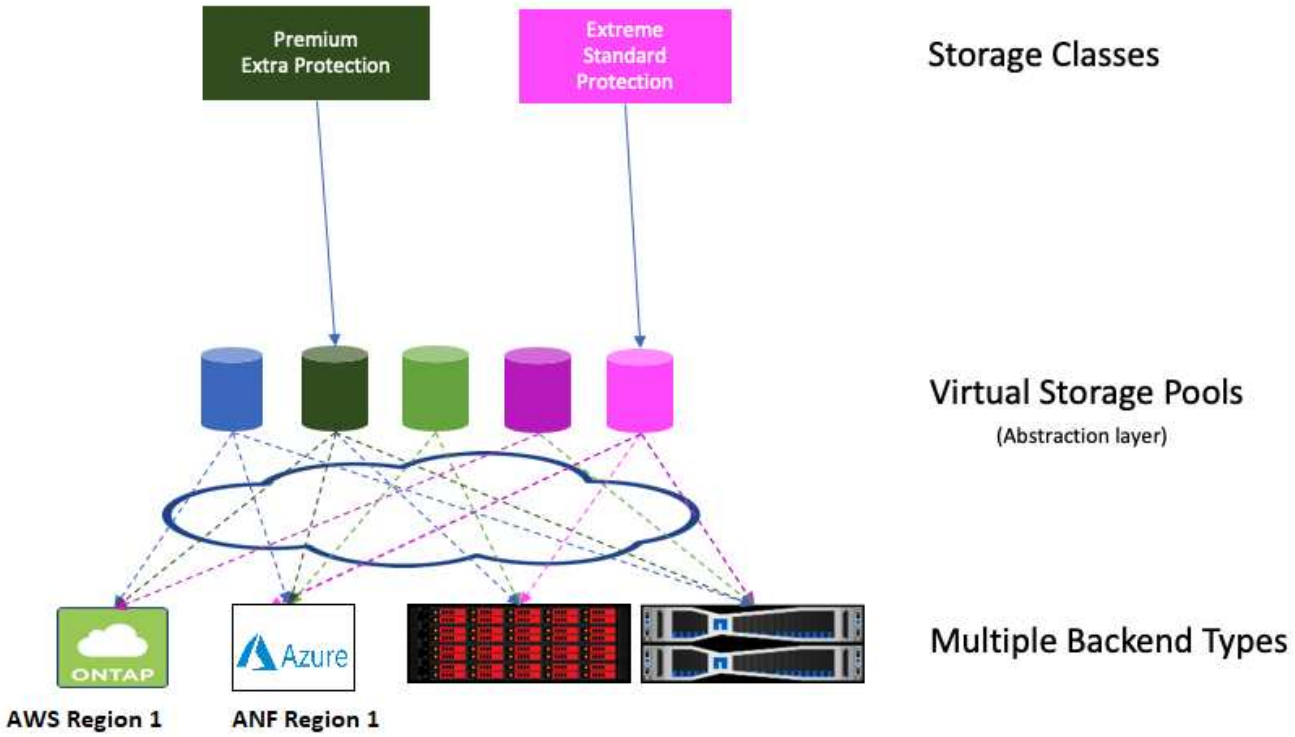
Con Trident, puedes usar VolumeSnapshots para crear nuevos PV a partir de ellos. La creación de PV a partir de estas instantáneas se realiza utilizando la tecnología FlexClone para los backends ONTAP y CVS compatibles. Al crear un PV a partir de una instantánea, el volumen de respaldo es un FlexClone del volumen principal de la instantánea. El `solidfire-san` El controlador utiliza clones de volumen de software Element para crear PV a partir de instantáneas. Aquí se crea un clon a partir de la instantánea del elemento.

piscinas virtuales

Los pools virtuales proporcionan una capa de abstracción entre los backends de almacenamiento de Trident y Kubernetes. `StorageClasses` . Permiten a un administrador definir aspectos, como la ubicación, el rendimiento y la protección de cada backend, de una manera común e independiente del backend, sin necesidad de realizar una configuración específica. `StorageClass` Especifique qué backend físico, grupo de backends o tipo de backend utilizar para cumplir con los criterios deseados.

Aprende sobre piscinas virtuales

El administrador de almacenamiento puede definir grupos virtuales en cualquiera de los backends de Trident en un archivo de definición JSON o YAML.



Cualquier aspecto especificado fuera de la lista de grupos virtuales es global para el backend y se aplicará a todos los grupos virtuales, mientras que cada grupo virtual puede especificar uno o más aspectos individualmente (anulando cualquier aspecto global del backend).

- Al definir grupos virtuales, no intente reorganizar el orden de los grupos virtuales existentes en una definición de backend.
- Desaconsejamos modificar los atributos de un grupo virtual existente. Deberás definir un nuevo grupo virtual para realizar cambios.

La mayoría de los aspectos se especifican en términos específicos del backend. Fundamentalmente, los valores de aspecto no se exponen fuera del controlador del backend y no están disponibles para la comparación en `StorageClasses` En cambio, el administrador define una o más etiquetas para cada grupo virtual. Cada etiqueta es un par clave:valor, y las etiquetas pueden ser comunes en diferentes sistemas backend. Al igual que los aspectos, las etiquetas se pueden especificar por grupo o de forma global para el backend. A diferencia de los aspectos, que tienen nombres y valores predefinidos, el administrador tiene plena discreción para definir las claves y los valores de las etiquetas según sea necesario. Para mayor comodidad, los administradores de almacenamiento pueden definir etiquetas por grupo virtual y agrupar volúmenes por etiqueta.

Las etiquetas del pool virtual se pueden desafiar usando estos caracteres:

- letras mayúsculas `A-Z`
- letras minúsculas `a-z`
- números `0-9`
- subraya _

- guiones –

A `StorageClass` identifica qué pool virtual utilizar haciendo referencia a las etiquetas dentro de un parámetro selector. Los selectores de grupos virtuales admiten los siguientes operadores:

Operador	Ejemplo	El valor de la etiqueta de un pool debe:
`=`	rendimiento=premium	Fósforo
`!=`	rendimiento != extremo	No coinciden
`in`	ubicación en (este, oeste)	Pertenece al conjunto de valores
`notin`	Nota de rendimiento (plata, bronce)	No pertenecer al conjunto de valores
`<key>`	protección	Existe con cualquier valor
`!<key>`	!protección	No existe

Grupos de acceso por volumen

Obtén más información sobre cómo Trident utiliza "grupos de acceso por volumen" .

> Ignore esta sección si está utilizando CHAP, lo cual se recomienda para simplificar la administración y evitar el límite de escalabilidad que se describe a continuación. Además, si está utilizando Trident en modo CSI, puede ignorar esta sección. Trident utiliza CHAP cuando se instala como un aprovisionador CSI mejorado.

Obtenga información sobre los grupos de acceso por volumen.

Trident puede utilizar grupos de acceso a volúmenes para controlar el acceso a los volúmenes que aprovisiona. Si CHAP está deshabilitado, espera encontrar un grupo de acceso llamado `trident` a menos que especifique uno o más ID de grupo de acceso en la configuración.

Si bien Trident asocia los nuevos volúmenes con los grupos de acceso configurados, no crea ni administra los grupos de acceso por sí mismos. Los grupos de acceso deben existir antes de que se agregue el backend de almacenamiento a Trident, y deben contener los IQN iSCSI de cada nodo del clúster de Kubernetes que potencialmente podría montar los volúmenes aprovisionados por ese backend. En la mayoría de las instalaciones, esto incluye todos los nodos de trabajo del clúster.

Para clústeres de Kubernetes con más de 64 nodos, se recomienda utilizar varios grupos de acceso. Cada grupo de acceso puede contener hasta 64 IQN y cada volumen puede pertenecer a cuatro grupos de acceso. Con un máximo de cuatro grupos de acceso configurados, cualquier nodo de un clúster de hasta 256 nodos podrá acceder a cualquier volumen. Para conocer los límites más recientes sobre los grupos de acceso por volumen, consulte "aquí" .

Si estás modificando la configuración de una que utiliza la predeterminada. `trident` Si tiene acceso a un grupo que también utiliza otros, incluya el ID del mismo. `trident` grupo de acceso en la lista.

Guía de inicio rápido para Trident

Puedes instalar Trident y comenzar a administrar los recursos de almacenamiento en pocos pasos. Antes de comenzar, revise"Requisitos de Trident" .

Para Docker, consulte"Trident para Docker" .

Preparar el nodo de trabajo

Todos los nodos de trabajo del clúster de Kubernetes deben poder montar los volúmenes que has aprovisionado para tus pods.

"Preparar el nodo de trabajo"

Instalar Trident

Trident ofrece varios métodos y modos de instalación optimizados para una variedad de entornos y organizaciones.

"Instalar Trident"

Crear un backend

Un backend define la relación entre Trident y un sistema de almacenamiento. Le indica a Trident cómo comunicarse con ese sistema de almacenamiento y cómo Trident aprovisionar volúmenes desde él.

"Configurar un backend"para su sistema de almacenamiento

Crea una clase de almacenamiento de Kubernetes

El objeto StorageClass de Kubernetes especifica Trident como el proveedor y permite crear una clase de almacenamiento para aprovisionar volúmenes con atributos personalizables. Trident crea una clase de almacenamiento coincidente para los objetos de Kubernetes que especifican el aprovisionador de Trident .

"Crear una clase de almacenamiento"

Provisión de un volumen

Un *PersistentVolume* (PV) es un recurso de almacenamiento físico aprovisionado por el administrador del clúster en un clúster de Kubernetes. La *PersistentVolumeClaim* (PVC) es una solicitud de acceso al PersistentVolume en el clúster.

Cree un PersistentVolume (PV) y un PersistentVolumeClaim (PVC) que utilice la StorageClass de Kubernetes configurada para solicitar acceso al PV. Luego puedes montar el panel fotovoltaico en un soporte.

"Provisión de un volumen"

¿Que sigue?

Ahora puede agregar backends adicionales, administrar clases de almacenamiento, administrar backends y realizar operaciones de volumen.

Requisitos

Antes de instalar Trident, debe revisar estos requisitos generales del sistema. Algunos sistemas backend específicos pueden tener requisitos adicionales.

Información crítica sobre Trident

Debe leer la siguiente información crítica sobre Trident.

Información crítica sobre Trident

- Kubernetes 1.34 ahora es compatible con Trident. Actualice Trident antes de actualizar Kubernetes.
- Trident impone estrictamente el uso de la configuración de rutas múltiples en entornos SAN, con un valor recomendado de `find_multipaths: no` en el archivo multipath.conf.

  Uso de una configuración sin rutas múltiples o uso de `find_multipaths: yes` o `find_multipaths: smart` Un valor incorrecto en el archivo multipath.conf provocará fallos de montaje. Trident ha recomendado el uso de `find_multipaths: no` Desde el lanzamiento del 21.07.

Front-ends compatibles (orquestadores)

Trident admite múltiples motores de contenedores y orquestadores, incluidos los siguientes:

- Anthos On-Prem (VMware) y Anthos en hardware 1.16
- Kubernetes 1.27 - 1.34
- OpenShift 4.12, 4.14 - 4.19 (Si planea utilizar la preparación de nodos iSCSI con OpenShift 4.19, la versión mínima de Trident compatible es 25.06.1).

> Trident continúa brindando soporte a versiones anteriores de OpenShift en consonancia con"Ciclo de vida de lanzamiento de Red Hat Extended Update Support (EUS)" , incluso si dependen de versiones de Kubernetes que ya no cuentan con soporte oficial de la versión original. Al instalar Trident en estos casos, puede ignorar con seguridad cualquier mensaje de advertencia sobre la versión de Kubernetes.

- Rancher Kubernetes Engine 2 (RKE2) v1.27.x - 1.34.x

> *Si bien Trident es compatible con Rancher Kubernetes Engine 2 (RKE2) versiones 1.27.x - 1.34.x, Trident actualmente solo está calificado para RKE2 v1.28.5+rke2r1.*

Trident también funciona con una variedad de otras ofertas de Kubernetes totalmente administradas y autoadministradas, incluidas Google Kubernetes Engine (GKE), Amazon Elastic Kubernetes Services (EKS), Azure Kubernetes Service (AKS), Mirantis Kubernetes Engine (MKE) y VMWare Tanzu Portfolio.

Trident y ONTAP pueden utilizarse como proveedor de almacenamiento para"KubeVirt" .

Antes de actualizar un clúster de Kubernetes de la versión 1.25 a la 1.26 o posterior que tenga Trident instalado, consulte"Actualizar una instalación de Helm" .

Sistemas de almacenamiento compatibles

Para usar Trident, necesitas uno o más de los siguientes backends compatibles:

- Amazon FSx for NetApp ONTAP
- Azure NetApp Files
- Cloud Volumes ONTAP
- Google Cloud NetApp Volumes
- NetApp All SAN Array (ASA)
- Versiones de clúster FAS, AFF, Select o ASA r2 (iSCSI y NVMe/TCP) locales con soporte limitado de NetApp. Ver"Compatibilidad con versiones de software" .
- Software NetApp HCI/Element versión 11 o superior

Compatibilidad de Trident con la virtualización de KubeVirt y OpenShift

Controladores de almacenamiento compatibles:

Trident admite los siguientes controladores ONTAP para la virtualización de KubeVirt y OpenShift:

- ontap-nas
- economía ontap-nas
- ontap-san (iSCSI, FCP, NVMe sobre TCP)
- economía ontap-san (solo iSCSI)

Puntos a considerar:

- Actualizar la clase de almacenamiento para que tenga `fsType` parámetro (por ejemplo: `fsType: "ext4"`) en un entorno de virtualización OpenShift. Si es necesario, configure el modo de volumen en bloqueo explícitamente utilizando el `volumeMode=Block` parámetro en el `dataVolumeTemplates` notificar a CDI para que cree volúmenes de datos de bloques.
- *Modo de acceso RWX para controladores de almacenamiento en bloque*: los controladores ontap-san (iSCSI, NVMe/TCP, FC) y ontap-san-economy (iSCSI) solo son compatibles con "volumeMode: Block" (dispositivo sin formato). Para estos conductores, el `fstype` El parámetro no se puede utilizar porque los volúmenes se proporcionan en modo de dispositivo sin procesar.
- Para los flujos de trabajo de migración en vivo en los que se requiere el modo de acceso RWX, se admiten las siguientes combinaciones:
  - NFS + `volumeMode=Filesystem`
  - iSCSI + `volumeMode=Block` (dispositivo sin procesar)
  - NVMe/TCP + `volumeMode=Block` (dispositivo sin procesar)
  - FC + `volumeMode=Block` (dispositivo sin procesar)

Requisitos de las funciones

La tabla siguiente resume las características disponibles en esta versión de Trident y las versiones de Kubernetes que admite.

Característica	Versión de Kubernetes	¿Se requieren puertas de características?
Trident	1,27 - 1,34	No
Instantáneas de volumen	1,27 - 1,34	No
PVC de instantáneas de volumen	1,27 - 1,34	No
redimensionamiento de PV iSCSI	1,27 - 1,34	No
ONTAP CHAP bidireccional	1,27 - 1,34	No
Políticas de exportación dinámicas	1,27 - 1,34	No
Operador Trident	1,27 - 1,34	No
Topología CSI	1,27 - 1,34	No

Sistemas operativos host probados

Aunque Trident no ofrece soporte oficial para sistemas operativos específicos, se sabe que los siguientes funcionan:

- Versiones de Red Hat Enterprise Linux CoreOS (RHCOS) compatibles con OpenShift Container Platform en AMD64 y ARM64
- Red Hat Enterprise Linux (RHEL) 8 o posterior en AMD64 y ARM64

  NVMe/TCP requiere RHEL 9 o posterior.

- Ubuntu 22.04 LTS o posterior en AMD64 y ARM64
- Servidor Windows 2022
- SUSE Linux Enterprise Server (SLES) 15 o posterior

Por defecto, Trident se ejecuta en un contenedor y, por lo tanto, se ejecutará en cualquier nodo Linux. Sin embargo, esos trabajadores deben poder montar los volúmenes que proporciona Trident utilizando el cliente NFS estándar o el iniciador iSCSI, dependiendo de los backends que esté utilizando.

El `tridentctl` Esta utilidad también funciona en cualquiera de estas distribuciones de Linux.

Configuración del host

Todos los nodos de trabajo del clúster de Kubernetes deben poder montar los volúmenes que has

aprovisionado para tus pods. Para preparar los nodos de trabajo, debe instalar las herramientas NFS, iSCSI o NVMe según el controlador que haya seleccionado.

"Preparar el nodo de trabajo"

Configuración del sistema de almacenamiento

Es posible que Trident requiera cambios en un sistema de almacenamiento antes de que una configuración de backend pueda utilizarlo.

"Configurar backends"

Puertos Trident

Trident requiere acceso a puertos específicos para la comunicación.

"Puertos Trident"

Imágenes de contenedores y versiones de Kubernetes correspondientes

Para instalaciones aisladas físicamente, la siguiente lista es una referencia de las imágenes de contenedor necesarias para instalar Trident. Utilice el `tridentctl images` comando para verificar la lista de imágenes de contenedor necesarias.

Imágenes de contenedor necesarias para Trident 25.06.2

Versiones de Kubernetes	Imagen del contenedor
v1.27.0, v1.28.0, v1.29.0, v1.30.0, v1.31.0, v1.32.0, v1.33.0, v1.34.0	• docker.io/netapp/trident:25.06.2 • docker.io/netapp/trident-autosupport:25.06 • registro.k8s.io/sig-storage/csi-provisioner:v5.2.0 • registro.k8s.io/sig-storage/csi-attacher:v4.8.1 • registro.k8s.io/sig-storage/csi-resizer:v1.13.2 • registro.k8s.io/sig-storage/csi-snapshotter:v8.2.1 • registro.k8s.io/sig-storage/csi-node-driver-registrar:v2.13.0 • docker.io/netapp/trident-operator:25.06.2 (opcional)

Imágenes de contenedor necesarias para Trident 25.06

Versiones de Kubernetes	Imagen del contenedor
v1.27.0, v1.28.0, v1.29.0, v1.30.0, v1.31.0, v1.32.0, v1.33.0, v1.34.0	• docker.io/netapp/trident:25.06.0 • docker.io/netapp/trident-autosupport:25.06 • registro.k8s.io/sig-storage/csi-provisioner:v5.2.0 • registro.k8s.io/sig-storage/csi-attacher:v4.8.1 • registro.k8s.io/sig-storage/csi-resizer:v1.13.2 • registro.k8s.io/sig-storage/csi-snapshotter:v8.2.1 • registro.k8s.io/sig-storage/csi-node-driver-registrar:v2.13.0 • docker.io/netapp/trident-operator:25.06.0 (opcional)

Instalar Trident

Instalar usando el operador Trident

Instalar usando tridentctl

Instalar utilizando un operador certificado de OpenShift

Usar Trident

Preparar el nodo de trabajo

Todos los nodos de trabajo del clúster de Kubernetes deben poder montar los volúmenes que has aprovisionado para tus pods. Para preparar los nodos de trabajo, debe instalar las herramientas NFS, iSCSI, NVMe/TCP o FC según el controlador que haya seleccionado.

Seleccionar las herramientas adecuadas

Si utiliza una combinación de controladores, deberá instalar todas las herramientas necesarias para sus controladores. Las versiones recientes de Red Hat Enterprise Linux CoreOS (RHCOS) tienen las herramientas instaladas por defecto.

Herramientas NFS

"Instala las herramientas NFS"Si estás usando: `ontap-nas` , `ontap-nas-economy` , `ontap-nas-flexgroup` , `azure-netapp-files` , `gcp-cvs` .

herramientas iSCSI

"Instala las herramientas iSCSI"Si estás usando: `ontap-san` , `ontap-san-economy` , `solidfire-san` .

Herramientas NVMe

"Instala las herramientas NVMe"Si estas usando `ontap-san` para el protocolo de memoria no volátil express (NVMe) sobre TCP (NVMe/TCP).

NetApp recomienda ONTAP 9.12 o posterior para NVMe/TCP.

Herramientas SCSI sobre FC

Referirse a"Formas de configurar hosts SAN FC y FC-NVMe" Para obtener más información sobre la configuración de sus hosts SAN FC y FC-NVMe.

"Instala las herramientas FC"Si estas usando `ontap-san` con sanType `fcp` (SCSI sobre FC).

Puntos a considerar: * SCSI sobre FC es compatible con entornos OpenShift y KubeVirt. * SCSI sobre FC no es compatible con Docker. * La autorreparación de iSCSI no es aplicable a SCSI sobre FC.

Descubrimiento de servicios de nodo

Trident intenta detectar automáticamente si el nodo puede ejecutar servicios iSCSI o NFS.

La detección de servicios de nodo identifica los servicios descubiertos, pero no garantiza que los servicios estén configurados correctamente. Por el contrario, la ausencia de un servicio detectado no garantiza que el montaje del volumen vaya a fallar.

Revisar eventos

Trident crea eventos para que el nodo identifique los servicios descubiertos. Para revisar estos eventos, ejecute:

```
kubectl get event -A --field-selector involvedObject.name=<Kubernetes node name>
```

Revisión de servicios descubiertos

Trident identifica los servicios habilitados para cada nodo en el nodo CR de Trident . Para ver los servicios detectados, ejecute:

```
tridentctl get node -o wide -n <Trident namespace>
```

Volúmenes de NFS

Instala las herramientas NFS utilizando los comandos correspondientes a tu sistema operativo. Asegúrese de que el servicio NFS se inicie durante el arranque del sistema.

RHEL 8+

```
sudo yum install -y nfs-utils
```

Ubuntu

```
sudo apt-get install -y nfs-common
```

Reinicie sus nodos de trabajo después de instalar las herramientas NFS para evitar fallos al conectar volúmenes a los contenedores.

volúmenes iSCSI

Trident puede establecer automáticamente una sesión iSCSI, escanear LUN, descubrir dispositivos multipath, formatearlos y montarlos en un pod.

capacidades de autorreparación de iSCSI

Para los sistemas ONTAP , Trident ejecuta la autorreparación iSCSI cada cinco minutos para:

1. **Identificar** el estado de sesión iSCSI deseado y el estado de sesión iSCSI actual.
2. **Compara** el estado deseado con el estado actual para identificar las reparaciones necesarias. Trident determina las prioridades de reparación y cuándo anticiparse a las reparaciones.
3. **Realizar las reparaciones** necesarias para devolver el estado actual de la sesión iSCSI al estado deseado.

Los registros de actividad de autocuración se encuentran en el `trident-main` contenedor en el pod Daemonset correspondiente. Para ver los registros, debe haber configurado `debug` a "verdadero" durante la instalación de Trident .

Las capacidades de autorreparación de Trident iSCSI pueden ayudar a prevenir:

- Sesiones iSCSI obsoletas o defectuosas que podrían producirse tras un problema de conectividad de red. En caso de una sesión inactiva, Trident espera siete minutos antes de cerrar sesión para restablecer la conexión con un portal.

> Por ejemplo, si los secretos CHAP se rotaran en el controlador de almacenamiento y la red perdiera la conectividad, los secretos CHAP antiguos (*obsoletos*) podrían persistir. La autocuración puede reconocer esto y restablecer automáticamente la sesión para aplicar los secretos CHAP actualizados.

- Faltan sesiones iSCSI
- LUN faltantes

Puntos a considerar antes de actualizar Trident

- Si solo se utilizan grupos de información por nodo (introducidos en la versión 23.04+), la autocuración de iSCSI iniciará nuevos escaneos SCSI para todos los dispositivos en el bus SCSI.
- Si solo se utilizan igroups con ámbito de backend (obsoletos a partir de la versión 23.04), la autocuración de iSCSI iniciará nuevos escaneos SCSI para los ID de LUN exactos en el bus SCSI.
- Si se utiliza una combinación de igroups por nodo e igroups con ámbito de backend, la autorreparación de iSCSI iniciará nuevos escaneos SCSI para los ID de LUN exactos en el bus SCSI.

Instala las herramientas iSCSI

Instale las herramientas iSCSI utilizando los comandos correspondientes a su sistema operativo.

Antes de empezar

- Cada nodo del clúster de Kubernetes debe tener un IQN único. **Este es un requisito previo necesario**.
- Si utiliza RHCOS versión 4.5 o posterior, u otra distribución de Linux compatible con RHEL, con el `solidfire-san` En el controlador y Element OS 12.5 o anterior, asegúrese de que el algoritmo de autenticación CHAP esté configurado en MD5 en `/etc/iscsi/iscsid.conf` Los algoritmos CHAP seguros y compatibles con FIPS SHA1, SHA-256 y SHA3-256 están disponibles con Element 12.7.

```
sudo sed -i 's/^\(node.session.auth.chap_algs\).*/\1 = MD5/'
/etc/iscsi/iscsid.conf
```

- Cuando utilice nodos de trabajo que ejecuten RHEL/Red Hat Enterprise Linux CoreOS (RHCOS) con volúmenes persistentes iSCSI, especifique el `discard` mountOption en la StorageClass para realizar la recuperación de espacio en línea. Referirse a "Documentación de Red Hat" .
- Asegúrese de haber actualizado a la última versión de `multipath-tools` .

RHEL 8+

1. Instale los siguientes paquetes del sistema:

```
sudo yum install -y lsscsi iscsi-initiator-utils device-mapper-
multipath
```

2. Compruebe que la versión de iscsi-initiator-utils sea la 6.2.0.874-2.el7 o posterior:

```
rpm -q iscsi-initiator-utils
```

3. Configurar el escaneo en modo manual:

```
sudo sed -i 's/^\(node.session.scan\).*/\1 = manual/'
/etc/iscsi/iscsid.conf
```

4. Habilitar rutas múltiples:

```
sudo mpathconf --enable --with_multipathd y --find_multipaths n
```

> Asegurar `/etc/multipath.conf` contiene `find_multipaths no` bajo `defaults` .

5. Asegúrese de que `iscsid` y `multipathd` están en funcionamiento:

```
sudo systemctl enable --now iscsid multipathd
```

6. Habilitar e iniciar `iscsi` :

```
sudo systemctl enable --now iscsi
```

Ubuntu

1. Instale los siguientes paquetes del sistema:

```
sudo apt-get install -y open-iscsi lsscsi sg3-utils multipath-tools
scsitools
```

2. Compruebe que la versión de open-iscsi sea 2.0.874-5ubuntu2.10 o posterior (para bionic) o 2.0.874-7.1ubuntu6.1 o posterior (para focal):

```
dpkg -l open-iscsi
```

3. Configurar el escaneo en modo manual:

```
sudo sed -i 's/^\(node.session.scan\).*/\1 = manual/'
/etc/iscsi/iscsid.conf
```

4. Habilitar rutas múltiples:

```
sudo tee /etc/multipath.conf <<-EOF
defaults {
    user_friendly_names yes
    find_multipaths no
}
EOF
sudo systemctl enable --now multipath-tools.service
sudo service multipath-tools restart
```

> Asegurar `/etc/multipath.conf` contiene `find_multipaths no` bajo `defaults` .

5. Asegúrese de que `open-iscsi` y `multipath-tools` están habilitados y en funcionamiento:

```
sudo systemctl status multipath-tools
sudo systemctl enable --now open-iscsi.service
sudo systemctl status open-iscsi
```

> Para Ubuntu 18.04, debe descubrir los puertos de destino con `iscsiadm` antes de comenzar `open-iscsi` para que se inicie el demonio iSCSI. También puedes modificar el `iscsi` servicio para comenzar `iscsid` automáticamente.

Configurar o deshabilitar la autorreparación de iSCSI

Puede configurar los siguientes ajustes de autorreparación de Trident iSCSI para corregir sesiones obsoletas:

- **Intervalo de autocuración iSCSI**: Determina la frecuencia con la que se invoca la autocuración iSCSI (predeterminado: 5 minutos). Puedes configurarlo para que se ejecute con mayor frecuencia estableciendo un número menor o con menor frecuencia estableciendo un número mayor.

Configurar el intervalo de autocuración de iSCSI en 0 detiene por completo la autocuración de iSCSI. No recomendamos deshabilitar la autorreparación de iSCSI; solo debe deshabilitarse en ciertos escenarios cuando la autorreparación de iSCSI no funciona como se espera o con fines de depuración.

- **Tiempo de espera de autocuración de iSCSI**: Determina la duración que la autocuración de iSCSI espera antes de cerrar la sesión de una sesión defectuosa e intentar iniciar sesión de nuevo (predeterminado: 7 minutos). Puede configurarlo con un número mayor para que las sesiones identificadas como no saludables tengan que esperar más tiempo antes de cerrarse y luego se intente volver a iniciar sesión, o con un número menor para cerrar sesión e iniciarla antes.

Timón

Para configurar o cambiar los ajustes de autorreparación de iSCSI, pase el `iscsiSelfHealingInterval` y `iscsiSelfHealingWaitTime` parámetros durante la instalación o actualización de Helm.

El siguiente ejemplo establece el intervalo de autorreparación de iSCSI en 3 minutos y el tiempo de espera de autorreparación en 6 minutos:

```
helm install trident trident-operator-100.2506.0.tgz --set
iscsiSelfHealingInterval=3m0s --set iscsiSelfHealingWaitTime=6m0s -n
trident
```

tridentctl

Para configurar o cambiar los ajustes de autorreparación de iSCSI, pase el `iscsi-self-healing-interval` y `iscsi-self-healing-wait-time` parámetros durante la instalación o actualización de tridentctl.

El siguiente ejemplo establece el intervalo de autorreparación de iSCSI en 3 minutos y el tiempo de espera de autorreparación en 6 minutos:

```
tridentctl install --iscsi-self-healing-interval=3m0s --iscsi-self
-healing-wait-time=6m0s -n trident
```

volúmenes NVMe/TCP

Instala las herramientas NVMe utilizando los comandos correspondientes a tu sistema operativo.

- NVMe requiere RHEL 9 o posterior.
- Si la versión del kernel de su nodo de Kubernetes es demasiado antigua o si el paquete NVMe no está disponible para su versión del kernel, es posible que deba actualizar la versión del kernel de su nodo a una que incluya el paquete NVMe.

RHEL 9

```
sudo yum install nvme-cli
sudo yum install linux-modules-extra-$(uname -r)
sudo modprobe nvme-tcp
```

Ubuntu

```
sudo apt install nvme-cli
sudo apt -y install linux-modules-extra-$(uname -r)
sudo modprobe nvme-tcp
```

Verificar la instalación

Tras la instalación, verifique que cada nodo del clúster de Kubernetes tenga un NQN único mediante el siguiente comando:

```
cat /etc/nvme/hostnqn
```

Trident modifica el `ctrl_device_tmo` Valor para garantizar que NVMe no abandone la ruta si falla. No cambie esta configuración.

SCSI sobre volúmenes FC

Ahora puede utilizar el protocolo Fibre Channel (FC) con Trident para aprovisionar y administrar recursos de almacenamiento en el sistema ONTAP .

Prerrequisitos

Configure los ajustes de red y nodo necesarios para FC.

Configuración de red

1. Obtén el WWPN de las interfaces de destino. Referirse a "Mostrar interfaz de red" Para más información.
2. Obtenga el WWPN para las interfaces en el iniciador (Host).

   Consulte las utilidades correspondientes del sistema operativo anfitrión.

3. Configure el zonificado en el switch FC utilizando los WWPN del host y del destino.

   Consulte la documentación del proveedor del conmutador correspondiente para obtener información.

   Consulte la siguiente documentación de ONTAP para obtener más detalles:

   - "Descripción general de la zonificación de Fibre Channel y FCoE"
   - "Formas de configurar hosts SAN FC y FC-NVMe"

Instala las herramientas FC

Instala las herramientas FC utilizando los comandos correspondientes a tu sistema operativo.

- Cuando utilice nodos de trabajo que ejecuten RHEL/Red Hat Enterprise Linux CoreOS (RHCOS) con PV FC, especifique el `discard` mountOption en la StorageClass para realizar la recuperación de espacio en línea. Referirse a "Documentación de Red Hat" .

RHEL 8+

1. Instale los siguientes paquetes del sistema:

```
sudo yum install -y lsscsi device-mapper-multipath
```

2. Habilitar rutas múltiples:

```
sudo mpathconf --enable --with_multipathd y --find_multipaths n
```

> Asegurar `/etc/multipath.conf` contiene `find_multipaths no` bajo `defaults` .

3. Asegúrese de que `multipathd` está en funcionamiento:

```
sudo systemctl enable --now multipathd
```

Ubuntu

1. Instale los siguientes paquetes del sistema:

```
sudo apt-get install -y lsscsi sg3-utils multipath-tools scsitools
```

2. Habilitar rutas múltiples:

```
sudo tee /etc/multipath.conf <<-EOF
defaults {
    user_friendly_names yes
    find_multipaths no
}
EOF
sudo systemctl enable --now multipath-tools.service
sudo service multipath-tools restart
```

> Asegurar `/etc/multipath.conf` contiene `find_multipaths no` bajo `defaults` .

3. Asegúrese de que `multipath-tools` está habilitado y en funcionamiento:

```
sudo systemctl status multipath-tools
```

Configurar y administrar backends

Configurar backends

Un backend define la relación entre Trident y un sistema de almacenamiento. Le indica a Trident cómo comunicarse con ese sistema de almacenamiento y cómo Trident aprovisionar volúmenes desde él.

Trident ofrece automáticamente pools de almacenamiento de backends que coinciden con los requisitos definidos por una clase de almacenamiento. Aprenda cómo configurar el backend de su sistema de almacenamiento.

- "Configurar un backend de Azure NetApp Files"
- "Configurar un backend de Google Cloud NetApp Volumes"
- "Configurar un Cloud Volumes Service para el backend de Google Cloud Platform"
- "Configure un backend de NetApp HCI o SolidFire"
- "Configure un backend con ONTAP o Cloud Volumes ONTAP NAS drivers"
- "Configure un backend con controladores SAN ONTAP o Cloud Volumes ONTAP."
- "Utilice Trident con Amazon FSx for NetApp ONTAP"

Azure NetApp Files

Configurar un backend de Azure NetApp Files

Puede configurar Azure NetApp Files como backend para Trident. Puede conectar volúmenes NFS y SMB utilizando un backend de Azure NetApp Files . Trident también admite la administración de credenciales mediante identidades administradas para clústeres de Azure Kubernetes Services (AKS).

Detalles del controlador de Azure NetApp Files

Trident proporciona los siguientes controladores de almacenamiento de Azure NetApp Files para comunicarse con el clúster. Los modos de acceso compatibles son: *ReadWriteOnce* (RWO), *ReadOnlyMany* (ROX), *ReadWriteMany* (RWX), *ReadWriteOncePod* (RWOP).

Conductor	Protocolo	modo de volumen	Modos de acceso compatibles	Sistemas de archivos compatibles
`azure-netapp-files`	NFS SMB	Sistema de archivos	RWO, ROX, RWX, RWOP	`nfs`, `smb`

Consideraciones

- El servicio Azure NetApp Files no admite volúmenes inferiores a 50 GiB. Trident crea automáticamente volúmenes de 50 GiB si se solicita un volumen más pequeño.
- Trident solo admite volúmenes SMB montados en pods que se ejecutan en nodos Windows.

Identidades administradas para AKS

Trident apoya"identidades gestionadas" para clústeres de Azure Kubernetes Services. Para aprovechar la gestión simplificada de credenciales que ofrecen las identidades gestionadas, debe tener:

- Un clúster de Kubernetes implementado mediante AKS
- Identidades administradas configuradas en el clúster de Kubernetes de AKS
- Trident instalado que incluye el `cloudProvider` para especificar `"Azure"` .

Operador de Trident

Para instalar Trident usando el operador Trident , edite `tridentorchestrator_cr.yaml` para establecer `cloudProvider` a `"Azure"` . Por ejemplo:

```
apiVersion: trident.netapp.io/v1
kind: TridentOrchestrator
metadata:
  name: trident
spec:
  debug: true
  namespace: trident
  imagePullPolicy: IfNotPresent
  cloudProvider: "Azure"
```

Timón

El siguiente ejemplo instala conjuntos Trident `cloudProvider` a Azure usando la variable de entorno `$CP` :

```
helm install trident trident-operator-100.2506.0.tgz --create
-namespace --namespace <trident-namespace> --set cloudProvider=$CP
```

<code>tridentctl</code>

El siguiente ejemplo instala Trident y configura `cloudProvider` bandera a `Azure` :

```
tridentctl install --cloud-provider="Azure" -n trident
```

Identidad en la nube para AKS

La identidad en la nube permite que los pods de Kubernetes accedan a los recursos de Azure autenticándose como una identidad de carga de trabajo en lugar de proporcionar credenciales explícitas de Azure.

Para aprovechar la identidad en la nube en Azure, debe tener:

- Un clúster de Kubernetes implementado mediante AKS

- La identidad de la carga de trabajo y el emisor OIDC están configurados en el clúster de Kubernetes de AKS.
- Trident instalado que incluye el `cloudProvider` para especificar `"Azure"` y `cloudIdentity` especificar la identidad de la carga de trabajo

Operador de Trident

Para instalar Trident usando el operador Trident , edite `tridentorchestrator_cr.yaml` para establecer `cloudProvider` a `"Azure"` y establecer `cloudIdentity` a `azure.workload.identity/client-id: xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxx` .

Por ejemplo:

```
apiVersion: trident.netapp.io/v1
kind: TridentOrchestrator
metadata:
  name: trident
spec:
  debug: true
  namespace: trident
  imagePullPolicy: IfNotPresent
  cloudProvider: "Azure"
  cloudIdentity: 'azure.workload.identity/client-id: xxxxxxxx-xxxx-
xxxx-xxxx-xxxxxxxxxxx' # Edit
```

Timón

Establezca los valores para las marcas **cloud-provider (CP)** y **cloud-identity (CI)** utilizando las siguientes variables de entorno:

```
export CP="Azure"
export CI="'azure.workload.identity/client-id: xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxx'"
```

El siguiente ejemplo instala Trident y lo configura. `cloudProvider` a Azure usando la variable de entorno `$CP` y establece el `cloudIdentity` utilizando la variable de entorno `$CI` :

```
helm install trident trident-operator-100.6.0.tgz --set
cloudProvider=$CP --set cloudIdentity="$CI"
```

<code>tridentctl</code>

Configure los valores para las marcas **proveedor de nube** e **identidad de nube** utilizando las siguientes variables de entorno:

```
export CP="Azure"
export CI="azure.workload.identity/client-id: xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxx"
```

El siguiente ejemplo instala Trident y configura `cloud-provider` bandera a `$CP` , y `cloud-identity` a `$CI` :

```
tridentctl install --cloud-provider=$CP --cloud-identity="$CI" -n
trident
```

Prepárese para configurar un backend de Azure NetApp Files.

Antes de poder configurar su backend de Azure NetApp Files , debe asegurarse de que se cumplen los siguientes requisitos.

Requisitos previos para volúmenes NFS y SMB

Si utiliza Azure NetApp Files por primera vez o en una nueva ubicación, se requiere una configuración inicial para configurar Azure NetApp Files y crear un volumen NFS. Referirse a "Azure: Configurar Azure NetApp Files y crear un volumen NFS" .

Para configurar y usar un "Azure NetApp Files" Para el backend, necesitas lo siguiente:

- `subscriptionID`, `tenantID` , `clientID` , `location` , y `clientSecret` son opcionales cuando se utilizan identidades administradas en un clúster de AKS.
- `tenantID`, `clientID` , y `clientSecret` son opcionales cuando se utiliza una identidad en la nube en un clúster de AKS.

- Un fondo común de capacidad. Referirse a"Microsoft: Crear un grupo de capacidad para Azure NetApp Files" .
- Una subred delegada a Azure NetApp Files. Referirse a"Microsoft: Delegar una subred a Azure NetApp Files" .
- `subscriptionID`desde una suscripción de Azure con Azure NetApp Files habilitado.
- `tenantID`, `clientID` , y `clientSecret` de un"Registro de la aplicación" en Azure Active Directory con permisos suficientes para el servicio Azure NetApp Files . El registro de la aplicación debe utilizar una de las siguientes opciones:
  - El rol de propietario o colaborador"predefinido por Azure" .
  - A"rol de colaborador personalizado" a nivel de suscripción(`assignableScopes`) con los siguientes permisos, que se limitan únicamente a lo que Trident requiere. Tras crear el rol personalizado,"Asigna el rol mediante el portal de Azure." .

Rol de colaborador personalizado

```
{
  "id": "/subscriptions/<subscription-
id>/providers/Microsoft.Authorization/roleDefinitions/<role-
definition-id>",
  "properties": {
    "roleName": "custom-role-with-limited-perms",
    "description": "custom role providing limited permissions",
    "assignableScopes": [
      "/subscriptions/<subscription-id>"
    ],
    "permissions": [
      {
        "actions": [
          "Microsoft.NetApp/netAppAccounts/capacityPools/read",
          "Microsoft.NetApp/netAppAccounts/capacityPools/write",

"Microsoft.NetApp/netAppAccounts/capacityPools/volumes/read",

"Microsoft.NetApp/netAppAccounts/capacityPools/volumes/write",

"Microsoft.NetApp/netAppAccounts/capacityPools/volumes/delete",

"Microsoft.NetApp/netAppAccounts/capacityPools/volumes/snapshots/
read",

"Microsoft.NetApp/netAppAccounts/capacityPools/volumes/snapshots/
write",

"Microsoft.NetApp/netAppAccounts/capacityPools/volumes/snapshots/
delete",

"Microsoft.NetApp/netAppAccounts/capacityPools/volumes/MountTarge
ts/read",
          "Microsoft.Network/virtualNetworks/read",
          "Microsoft.Network/virtualNetworks/subnets/read",

"Microsoft.Features/featureProviders/subscriptionFeatureRegistrat
ions/read",

"Microsoft.Features/featureProviders/subscriptionFeatureRegistrat
ions/write",

"Microsoft.Features/featureProviders/subscriptionFeatureRegistrat
ions/delete",
```

```
            "Microsoft.Features/features/read",
            "Microsoft.Features/operations/read",
            "Microsoft.Features/providers/features/read",

"Microsoft.Features/providers/features/register/action",

"Microsoft.Features/providers/features/unregister/action",

"Microsoft.Features/subscriptionFeatureRegistrations/read"
          ],
          "notActions": [],
          "dataActions": [],
          "notDataActions": []
        }
      ]
    }
}
```

- El Azure `location` que contiene al menos uno "subred delegada" . A partir de Trident 22.01, el `location` El parámetro es un campo obligatorio en el nivel superior del archivo de configuración del backend. Los valores de ubicación especificados en los grupos virtuales se ignoran.
- Para usar `Cloud Identity` , obtén el `client ID` de un "identidad gestionada asignada por el usuario" y especifique ese ID en `azure.workload.identity/client-id: xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxx` .

Requisitos adicionales para volúmenes de PYMES

Para crear un volumen SMB, debe tener:

- Active Directory configurado y conectado a Azure NetApp Files. Referirse a"Microsoft: Crear y administrar conexiones de Active Directory para Azure NetApp Files" .
- Un clúster de Kubernetes con un nodo controlador Linux y al menos un nodo de trabajo Windows que ejecuta Windows Server 2022. Trident solo admite volúmenes SMB montados en pods que se ejecutan en nodos Windows.
- Al menos un secreto de Trident que contenga sus credenciales de Active Directory para que Azure NetApp Files pueda autenticarse en Active Directory. Para generar secretos `smbcreds` :

```
kubectl create secret generic smbcreds --from-literal username=user
--from-literal password='password'
```

- Un proxy CSI configurado como servicio de Windows. Para configurar un `csi-proxy` , consulte a"GitHub: Proxy CSI" o"GitHub: Proxy CSI para Windows" para nodos de Kubernetes que se ejecutan en Windows.

Opciones y ejemplos de configuración del backend de Azure NetApp Files

Obtenga información sobre las opciones de configuración de backend NFS y SMB para

Azure NetApp Files y revise los ejemplos de configuración.

Opciones de configuración del backend

Trident utiliza su configuración de backend (subred, red virtual, nivel de servicio y ubicación) para crear volúmenes de Azure NetApp Files en grupos de capacidad que estén disponibles en la ubicación solicitada y que coincidan con el nivel de servicio y la subred solicitados.

* A partir de la versión 25.06 de NetApp Trident , los grupos de capacidad de QoS manuales se admiten como versión preliminar técnica.*

Los backends de Azure NetApp Files proporcionan estas opciones de configuración.

Parámetro	Descripción	Por defecto
`version`		Siempre 1
`storageDriverName`	Nombre del controlador de almacenamiento	"archivos de azure-netapp"
`backendName`	Nombre personalizado o el backend de almacenamiento	Nombre del conductor + "_" + caracteres aleatorios
`subscriptionID`	El ID de suscripción de su suscripción de Azure. Opcional cuando las identidades administradas están habilitadas en un clúster de AKS.	
`tenantID`	El ID de inquilino de un registro de aplicación es opcional cuando se utilizan identidades administradas o identidades en la nube en un clúster de AKS.	
`clientID`	El ID de cliente de un registro de aplicación es opcional cuando se utilizan identidades administradas o identidades en la nube en un clúster de AKS.	
`clientSecret`	El secreto del cliente de un registro de aplicación es opcional cuando se utilizan identidades administradas o identidades en la nube en un clúster de AKS.	
`serviceLevel`	Uno de `Standard` , `Premium` , o `Ultra`	"" (aleatorio)
`location`	Nombre de la ubicación de Azure donde se crearán los nuevos volúmenes. Opcional cuando las identidades administradas están habilitadas en un clúster de AKS.	

Parámetro	Descripción	Por defecto
`resourceGroups`	Lista de grupos de recursos para filtrar los recursos descubiertos	"[]" (sin filtro)
`netappAccounts`	Lista de cuentas de NetApp para filtrar los recursos detectados	"[]" (sin filtro)
`capacityPools`	Lista de grupos de capacidad para filtrar los recursos descubiertos	"[]" (sin filtro, aleatorio)
`virtualNetwork`	Nombre de una red virtual con una subred delegada	""
`subnet`	Nombre de una subred delegada a `Microsoft.Netapp/volumes`	""
`networkFeatures`	Conjunto de características de VNet para un volumen, puede ser `Basic` o `Standard` . La función de red no está disponible en todas las regiones y puede que sea necesario habilitarla mediante una suscripción. Especificar `networkFeatures` Cuando la funcionalidad no está habilitada, se produce un fallo en el aprovisionamiento de volúmenes.	""
`nfsMountOptions`	Control preciso de las opciones de montaje NFS. Se ignora para volúmenes SMB. Para montar volúmenes utilizando NFS versión 4.1, incluya `nfsvers=4` en la lista de opciones de montaje delimitadas por comas, seleccione NFS v4.1. Las opciones de montaje establecidas en la definición de una clase de almacenamiento anulan las opciones de montaje establecidas en la configuración del backend.	"nfsvers=3"
`limitVolumeSize`	Fallará el aprovisionamiento si el tamaño del volumen solicitado supera este valor.	" (no se aplica por defecto)
`debugTraceFlags`	Indicadores de depuración para usar al solucionar problemas. Ejemplo, `\{"api": false, "method": true, "discovery": true}` . No utilice esta función a menos que esté solucionando problemas y necesite un registro detallado.	nulo

Parámetro	Descripción	Por defecto
`nasType`	Configure la creación de volúmenes NFS o SMB. Las opciones son `nfs` , `smb` o nulo. Si se establece en nulo, se utilizarán volúmenes NFS por defecto.	`nfs`
`supportedTopologies`	Representa una lista de regiones y zonas compatibles con este backend. Para obtener más información, consulte"Utilizar la topología CSI" .	
`qosType`	Representa el tipo de QoS: Automático o Manual. **Vista previa técnica de Trident 25.06**	Auto
`maxThroughput`	Establece el rendimiento máximo permitido en MiB/seg. Compatible solo con grupos de capacidad de QoS manuales. **Vista previa técnica de Trident 25.06**	`4 MiB/sec`

Para obtener más información sobre las funciones de red, consulte"Configurar las características de red para un volumen de Azure NetApp Files" .

Permisos y recursos necesarios

Si recibe un error de "No se encontraron grupos de capacidad" al crear un PVC, es probable que el registro de su aplicación no tenga asociados los permisos y recursos necesarios (subred, red virtual, grupo de capacidad). Si la depuración está habilitada, Trident registrará los recursos de Azure detectados cuando se cree el backend. Verifique que se esté utilizando el rol adecuado.

Los valores para `resourceGroups` , `netappAccounts` , `capacityPools` , `virtualNetwork` , y `subnet` se pueden especificar utilizando nombres cortos o nombres completos. En la mayoría de los casos se recomienda utilizar nombres completos, ya que los nombres cortos pueden coincidir con varios recursos que tengan el mismo nombre.

El `resourceGroups` , `netappAccounts` , y `capacityPools` Los valores son filtros que restringen el conjunto de recursos descubiertos a aquellos disponibles para este backend de almacenamiento y pueden especificarse en cualquier combinación. Los nombres completos siguen este formato:

Tipo	Formato
Grupo de recursos	<grupo de recursos>
cuenta de NetApp	<grupo de recursos>/<cuenta de NetApp>
reserva de capacidad	<grupo de recursos>/<cuenta de NetApp>/<grupo de capacidad>
Red virtual	<grupo de recursos>/<red virtual>
Subred	<grupo de recursos>/<red virtual>/<subred>

Aprovisionamiento de volumen

Puede controlar el aprovisionamiento de volúmenes predeterminado especificando las siguientes opciones en una sección especial del archivo de configuración. Referirse a Configuraciones de ejemplo Para más detalles.

Parámetro	Descripción	Por defecto
`exportRule`	Reglas de exportación para nuevos volúmenes. `exportRule` Debe ser una lista separada por comas de cualquier combinación de direcciones IPv4 o subredes IPv4 en notación CIDR. Se ignora para volúmenes SMB.	"0.0.0.0/0"
`snapshotDir`	Controla la visibilidad del directorio .snapshot	"verdadero" para NFSv4, "falso" para NFSv3
`size`	El tamaño predeterminado de los nuevos volúmenes	"100G"
`unixPermissions`	Los permisos Unix de los nuevos volúmenes (4 dígitos octales). Se ignora para volúmenes SMB.	" (función de vista previa, requiere inclusión en la lista blanca de la suscripción)

Configuraciones de ejemplo

Los siguientes ejemplos muestran configuraciones básicas que dejan la mayoría de los parámetros con sus valores predeterminados. Esta es la forma más sencilla de definir un backend.

Configuración mínima

Esta es la configuración mínima absoluta del backend. Con esta configuración, Trident descubre todas sus cuentas de NetApp , grupos de capacidad y subredes delegadas a Azure NetApp Files en la ubicación configurada, y coloca nuevos volúmenes en uno de esos grupos y subredes de forma aleatoria. Porque `nasType` se omite, el `nfs` Se aplicará la configuración predeterminada y el backend aprovisionará los volúmenes NFS.

Esta configuración es ideal cuando estás empezando a usar Azure NetApp Files y haciendo pruebas, pero en la práctica querrás proporcionar un alcance adicional para los volúmenes que aprovisiones.

```
---
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-anf-1
  namespace: trident
spec:
  version: 1
  storageDriverName: azure-netapp-files
  subscriptionID: 9f87c765-4774-fake-ae98-a721add45451
  tenantID: 68e4f836-edc1-fake-bff9-b2d865ee56cf
  clientID: dd043f63-bf8e-fake-8076-8de91e5713aa
  clientSecret: SECRET
  location: eastus
```

Identidades administradas para AKS

Esta configuración de backend omite `subscriptionID`, `tenantID`, `clientID`, y `clientSecret`, que son opcionales cuando se utilizan identidades administradas.

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-anf-1
  namespace: trident
spec:
  version: 1
  storageDriverName: azure-netapp-files
  capacityPools:
    - ultra-pool
  resourceGroups:
    - aks-ami-eastus-rg
  netappAccounts:
    - smb-na
  virtualNetwork: eastus-prod-vnet
  subnet: eastus-anf-subnet
```

Identidad en la nube para AKS

Esta configuración de backend omite `tenantID` , `clientID` , y `clientSecret` , que son opcionales al usar una identidad en la nube.

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-anf-1
  namespace: trident
spec:
  version: 1
  storageDriverName: azure-netapp-files
  capacityPools:
    - ultra-pool
  resourceGroups:
    - aks-ami-eastus-rg
  netappAccounts:
    - smb-na
  virtualNetwork: eastus-prod-vnet
  subnet: eastus-anf-subnet
  location: eastus
  subscriptionID: 9f87c765-4774-fake-ae98-a721add45451
```

Configuración específica del nivel de servicio con filtros de capacidad.

Esta configuración de backend coloca volúmenes en Azure `eastus` ubicación en un `Ultra` reserva de capacidad. Trident descubre automáticamente todas las subredes delegadas a Azure NetApp Files en esa ubicación y coloca un nuevo volumen en una de ellas de forma aleatoria.

```
---
version: 1
storageDriverName: azure-netapp-files
subscriptionID: 9f87c765-4774-fake-ae98-a721add45451
tenantID: 68e4f836-edc1-fake-bff9-b2d865ee56cf
clientID: dd043f63-bf8e-fake-8076-8de91e5713aa
clientSecret: SECRET
location: eastus
serviceLevel: Ultra
capacityPools:
  - application-group-1/account-1/ultra-1
  - application-group-1/account-1/ultra-2
```

Ejemplo de backend con grupos de capacidad de QoS manuales

Esta configuración de backend coloca volúmenes en Azure `eastus` Ubicación con pools de capacidad QoS manuales. **Vista previa técnica en NetApp Trident 25.06**.

```
---
version: 1
storageDriverName: azure-netapp-files
backendName: anf1
location: eastus
labels:
  clusterName: test-cluster-1
  cloud: anf
  nasType: nfs
defaults:
  qosType: Manual
storage:
  - serviceLevel: Ultra
    labels:
      performance: gold
    defaults:
      maxThroughput: 10
  - serviceLevel: Premium
    labels:
      performance: silver
    defaults:
      maxThroughput: 5
  - serviceLevel: Standard
    labels:
      performance: bronze
    defaults:
      maxThroughput: 3
```

Configuración avanzada

Esta configuración de backend reduce aún más el alcance de la ubicación de volúmenes a una sola subred y también modifica algunos valores predeterminados de aprovisionamiento de volúmenes.

```
---
version: 1
storageDriverName: azure-netapp-files
subscriptionID: 9f87c765-4774-fake-ae98-a721add45451
tenantID: 68e4f836-edc1-fake-bff9-b2d865ee56cf
clientID: dd043f63-bf8e-fake-8076-8de91e5713aa
clientSecret: SECRET
location: eastus
serviceLevel: Ultra
capacityPools:
  - application-group-1/account-1/ultra-1
  - application-group-1/account-1/ultra-2
virtualNetwork: my-virtual-network
subnet: my-subnet
networkFeatures: Standard
nfsMountOptions: vers=3,proto=tcp,timeo=600
limitVolumeSize: 500Gi
defaults:
  exportRule: 10.0.0.0/24,10.0.1.0/24,10.0.2.100
  snapshotDir: "true"
  size: 200Gi
  unixPermissions: "0777"
```

Configuración de grupo virtual

Esta configuración de backend define múltiples grupos de almacenamiento en un solo archivo. Esto resulta útil cuando se tienen varios grupos de capacidad que admiten diferentes niveles de servicio y se desea crear clases de almacenamiento en Kubernetes que los representen. Se utilizaron etiquetas de piscinas virtuales para diferenciar las piscinas en función de `performance` .

```
---
version: 1
storageDriverName: azure-netapp-files
subscriptionID: 9f87c765-4774-fake-ae98-a721add45451
tenantID: 68e4f836-edc1-fake-bff9-b2d865ee56cf
clientID: dd043f63-bf8e-fake-8076-8de91e5713aa
clientSecret: SECRET
location: eastus
resourceGroups:
  - application-group-1
networkFeatures: Basic
nfsMountOptions: vers=3,proto=tcp,timeo=600
labels:
  cloud: azure
storage:
  - labels:
      performance: gold
    serviceLevel: Ultra
    capacityPools:
      - ultra-1
      - ultra-2
    networkFeatures: Standard
  - labels:
      performance: silver
    serviceLevel: Premium
    capacityPools:
      - premium-1
  - labels:
      performance: bronze
    serviceLevel: Standard
    capacityPools:
      - standard-1
      - standard-2
```

Configuración de topologías admitidas

Trident facilita el aprovisionamiento de volúmenes para cargas de trabajo en función de regiones y zonas de disponibilidad. El `supportedTopologies` Este bloque en la configuración del backend se utiliza para proporcionar una lista de regiones y zonas por backend. Los valores de región y zona especificados aquí deben coincidir con los valores de región y zona de las etiquetas de cada nodo del clúster de Kubernetes. Estas regiones y zonas representan la lista de valores permitidos que se pueden proporcionar en una clase de almacenamiento. Para las clases de almacenamiento que contienen un subconjunto de las regiones y zonas proporcionadas en un backend, Trident crea volúmenes en la región y zona mencionadas. Para obtener más información, consulte"Utilizar la topología CSI" .

```
---
version: 1
storageDriverName: azure-netapp-files
subscriptionID: 9f87c765-4774-fake-ae98-a721add45451
tenantID: 68e4f836-edc1-fake-bff9-b2d865ee56cf
clientID: dd043f63-bf8e-fake-8076-8de91e5713aa
clientSecret: SECRET
location: eastus
serviceLevel: Ultra
capacityPools:
  - application-group-1/account-1/ultra-1
  - application-group-1/account-1/ultra-2
supportedTopologies:
  - topology.kubernetes.io/region: eastus
    topology.kubernetes.io/zone: eastus-1
  - topology.kubernetes.io/region: eastus
    topology.kubernetes.io/zone: eastus-2
```

Definiciones de clases de almacenamiento

La siguiente `StorageClass` Las definiciones se refieren a los grupos de almacenamiento mencionados anteriormente.

Definiciones de ejemplo que utilizan `parameter.selector` campo

Usando `parameter.selector` Puedes especificarlo para cada uno. `StorageClass` el grupo virtual que se utiliza para alojar un volumen. El volumen tendrá los aspectos definidos en el pool elegido.

```
---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: gold
provisioner: csi.trident.netapp.io
parameters:
  selector: performance=gold
allowVolumeExpansion: true

---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: silver
provisioner: csi.trident.netapp.io
parameters:
  selector: performance=silver
allowVolumeExpansion: true

---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: bronze
provisioner: csi.trident.netapp.io
parameters:
  selector: performance=bronze
allowVolumeExpansion: true
```

Definiciones de ejemplo para volúmenes SMB

Usando `nasType` , `node-stage-secret-name` , y `node-stage-secret-namespace` , puede especificar un volumen SMB y proporcionar las credenciales de Active Directory necesarias.

Configuración básica en el espacio de nombres predeterminado

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: anf-sc-smb
provisioner: csi.trident.netapp.io
parameters:
  backendType: "azure-netapp-files"
  trident.netapp.io/nasType: "smb"
  csi.storage.k8s.io/node-stage-secret-name: "smbcreds"
  csi.storage.k8s.io/node-stage-secret-namespace: "default"
```

Utilizar diferentes secretos por espacio de nombres

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: anf-sc-smb
provisioner: csi.trident.netapp.io
parameters:
  backendType: "azure-netapp-files"
  trident.netapp.io/nasType: "smb"
  csi.storage.k8s.io/node-stage-secret-name: "smbcreds"
  csi.storage.k8s.io/node-stage-secret-namespace: ${pvc.namespace}
```

Utilizando diferentes secretos por volumen

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: anf-sc-smb
provisioner: csi.trident.netapp.io
parameters:
  backendType: "azure-netapp-files"
  trident.netapp.io/nasType: "smb"
  csi.storage.k8s.io/node-stage-secret-name: ${pvc.name}
  csi.storage.k8s.io/node-stage-secret-namespace: ${pvc.namespace}
```

`nasType: smb`Filtros para pools que admiten volúmenes SMB. `nasType: nfs` o `nasType: null` Filtros para pools NFS.

Crea el backend

Después de crear el archivo de configuración del backend, ejecute el siguiente comando:

```
tridentctl create backend -f <backend-file>
```

Si falla la creación del backend, algo falla en la configuración del backend. Puedes consultar los registros para determinar la causa ejecutando el siguiente comando:

```
tridentctl logs
```

Después de identificar y corregir el problema con el archivo de configuración, puede volver a ejecutar el comando de creación.

Google Cloud NetApp Volumes

Configurar un backend de Google Cloud NetApp Volumes

Ahora puedes configurar Google Cloud NetApp Volumes como backend para Trident. Puedes conectar volúmenes NFS y SMB utilizando un backend de Google Cloud NetApp Volumes .

Detalles del controlador de Google Cloud NetApp Volumes

Trident proporciona el `google-cloud-netapp-volumes` controlador para comunicarse con el clúster. Los modos de acceso compatibles son: *ReadWriteOnce* (RWO), *ReadOnlyMany* (ROX), *ReadWriteMany* (RWX), *ReadWriteOncePod* (RWOP).

Conductor	Protocolo	modo de volumen	Modos de acceso compatibles	Sistemas de archivos compatibles
`google-cloud-netapp-volumes`	NFS SMB	Sistema de archivos	RWO, ROX, RWX, RWOP	`nfs`, `smb`

Identidad en la nube para GKE

La identidad en la nube permite que los pods de Kubernetes accedan a los recursos de Google Cloud autenticándose como una identidad de carga de trabajo en lugar de proporcionar credenciales explícitas de Google Cloud.

Para aprovechar la identidad en la nube de Google Cloud, debes tener:

- Un clúster de Kubernetes desplegado utilizando GKE.
- La identidad de la carga de trabajo está configurada en el clúster de GKE y el servidor de metadatos de GKE está configurado en los grupos de nodos.

- Una cuenta de servicio de GCP con el rol de administrador de Google Cloud NetApp Volumes (roles/netapp.admin) o un rol personalizado.
- Trident instalado incluye cloudProvider para especificar "GCP" y cloudIdentity para especificar la nueva cuenta de servicio de GCP. A continuación se muestra un ejemplo.

Operador de Trident

Para instalar Trident usando el operador Trident , edite `tridentorchestrator_cr.yaml` para establecer `cloudProvider` a `"GCP"` y establecer `cloudIdentity` a `iam.gke.io/gcp-service-account: cloudvolumes-admin-sa@mygcpproject.iam.gserviceaccount.com` .

Por ejemplo:

```
apiVersion: trident.netapp.io/v1
kind: TridentOrchestrator
metadata:
  name: trident
spec:
  debug: true
  namespace: trident
  imagePullPolicy: IfNotPresent
  cloudProvider: "GCP"
  cloudIdentity: 'iam.gke.io/gcp-service-account: cloudvolumes-
admin-sa@mygcpproject.iam.gserviceaccount.com'
```

Timón

Establezca los valores para las marcas **cloud-provider (CP)** y **cloud-identity (CI)** utilizando las siguientes variables de entorno:

```
export CP="GCP"
export ANNOTATION="'iam.gke.io/gcp-service-account: cloudvolumes-admin-
sa@mygcpproject.iam.gserviceaccount.com'"
```

El siguiente ejemplo instala Trident y lo configura. `cloudProvider` a GCP utilizando la variable de entorno `$CP` y establece el `cloudIdentity` utilizando la variable de entorno `$ANNOTATION` :

```
helm install trident trident-operator-100.6.0.tgz --set
cloudProvider=$CP --set cloudIdentity="$ANNOTATION"
```

<code>tridentctl</code>

Configure los valores para las marcas **proveedor de nube** e **identidad de nube** utilizando las siguientes variables de entorno:

```
export CP="GCP"
export ANNOTATION="'iam.gke.io/gcp-service-account: cloudvolumes-admin-
sa@mygcpproject.iam.gserviceaccount.com'"
```

El siguiente ejemplo instala Trident y configura `cloud-provider` bandera a `$CP` , y `cloud-identity` a `$ANNOTATION` :

```
tridentctl install --cloud-provider=$CP --cloud
-identity="$ANNOTATION" -n trident
```

Prepárate para configurar un backend de Google Cloud NetApp Volumes.

Antes de poder configurar su backend de Google Cloud NetApp Volumes , debe asegurarse de que se cumplen los siguientes requisitos.

Requisitos previos para volúmenes NFS

Si utiliza Google Cloud NetApp Volumes por primera vez o en una nueva ubicación, se requiere una configuración inicial para configurar Google Cloud NetApp Volumes y crear un volumen NFS. Referirse a"Antes de empezar" .

Asegúrese de tener lo siguiente antes de configurar el backend de Google Cloud NetApp Volumes :

- Una cuenta de Google Cloud configurada con el servicio Google Cloud NetApp Volumes . Referirse a"Google Cloud NetApp Volumes" .
- Número de proyecto de tu cuenta de Google Cloud. Referirse a"Identificación de proyectos" .
- Una cuenta de servicio de Google Cloud con el administrador de volúmenes de NetApp(`roles/netapp.admin`) role. Referirse a"Roles y permisos de gestión de identidades y accesos" .
- Archivo de clave API para su cuenta de GCNV. Referirse a"Cree una clave de cuenta de servicio"
- Un grupo de almacenamiento. Referirse a"Descripción general de los grupos de almacenamiento" .

Para obtener más información sobre cómo configurar el acceso a Google Cloud NetApp Volumes, consulte"Configurar el acceso a Google Cloud NetApp Volumes" .

Opciones y ejemplos de configuración del backend de Google Cloud NetApp Volumes

Aprenda sobre las opciones de configuración de backend para Google Cloud NetApp Volumes y revise ejemplos de configuración.

Opciones de configuración del backend

Cada backend aprovisiona volúmenes en una única región de Google Cloud. Para crear volúmenes en otras regiones, puede definir backends adicionales.

Parámetro	Descripción	Por defecto
`version`		Siempre 1
`storageDriverName`	Nombre del controlador de almacenamiento	El valor de `storageDriverName` debe especificarse como "google-cloud-netapp-volumes".

Parámetro	Descripción	Por defecto
`backendName`	(Opcional) Nombre personalizado del backend de almacenamiento	Nombre del controlador + "_" + parte de la clave API
`storagePools`	Parámetro opcional utilizado para especificar los grupos de almacenamiento para la creación de volúmenes.	
`projectNumber`	Número de proyecto de la cuenta de Google Cloud. El valor se encuentra en la página principal del portal de Google Cloud.	
`location`	La ubicación de Google Cloud donde Trident crea volúmenes GCNV. Al crear clústeres de Kubernetes entre regiones, los volúmenes creados en un `location` Puede utilizarse en cargas de trabajo programadas en nodos de varias regiones de Google Cloud. El tráfico interregional conlleva un coste adicional.	
`apiKey`	clave API para la cuenta de servicio de Google Cloud con la `netapp.admin` role. Incluye el contenido en formato JSON del archivo de clave privada de una cuenta de servicio de Google Cloud (copiado textualmente en el archivo de configuración del backend). El `apiKey` Debe incluir pares clave-valor para las siguientes claves: `type` , `project_id` , `client_email` , `client_id` , `auth_uri` , `token_uri` , `auth_provider_x509_cert_url` , y `client_x509_cert_url` .	
`nfsMountOptions`	Control preciso de las opciones de montaje NFS.	"nfsvers=3"
`limitVolumeSize`	Fallará el aprovisionamiento si el tamaño de volumen solicitado supera este valor.	" (no se aplica por defecto)
`serviceLevel`	El nivel de servicio de un grupo de almacenamiento y sus volúmenes. Los valores son `flex` , `standard` , `premium` , o `extreme` .	
`labels`	Conjunto de etiquetas arbitrarias con formato JSON para aplicar a los volúmenes	""
`network`	Red de Google Cloud utilizada para los volúmenes de GCNV.	
`debugTraceFlags`	Indicadores de depuración para usar al solucionar problemas. Ejemplo, `{"api":false, "method":true}` . No utilice esta función a menos que esté solucionando problemas y necesite un registro detallado.	nulo
`nasType`	Configure la creación de volúmenes NFS o SMB. Las opciones son `nfs` , `smb` o nulo. Si se establece en nulo, se utilizarán volúmenes NFS por defecto.	`nfs`

Parámetro	Descripción	Por defecto
`supportedTopologies`	Representa una lista de regiones y zonas compatibles con este backend. Para obtener más información, consulte"Utilizar la topología CSI" . Por ejemplo: `supportedTopologies: - topology.kubernetes.io/region: asia-east1 topology.kubernetes.io/zone: asia-east1-a`	

Opciones de aprovisionamiento de volumen

Puedes controlar el aprovisionamiento de volúmenes predeterminado en el `defaults` sección del archivo de configuración.

Parámetro	Descripción	Por defecto
`exportRule`	Las reglas de exportación para nuevos volúmenes. Debe ser una lista separada por comas de cualquier combinación de direcciones IPv4.	"0.0.0.0/0"
`snapshotDir`	Acceso a la `.snapshot` directorio	"verdadero" para NFSv4, "falso" para NFSv3
`snapshotReserve`	Porcentaje de volumen reservado para instantáneas	"" (aceptar valor predeterminado de 0)
`unixPermissions`	Los permisos Unix de los nuevos volúmenes (4 dígitos octales).	""

Configuraciones de ejemplo

Los siguientes ejemplos muestran configuraciones básicas que dejan la mayoría de los parámetros con sus valores predeterminados. Esta es la forma más sencilla de definir un backend.

Configuración mínima

Esta es la configuración mínima absoluta del backend. Con esta configuración, Trident descubre todos los grupos de almacenamiento delegados a Google Cloud NetApp Volumes en la ubicación configurada y coloca nuevos volúmenes en uno de esos grupos de forma aleatoria. Porque `nasType` se omite, el `nfs` Se aplicará la configuración predeterminada y el backend aprovisionará los volúmenes NFS.

Esta configuración es ideal cuando estás empezando a usar Google Cloud NetApp Volumes y haciendo pruebas, pero en la práctica lo más probable es que necesites proporcionar un alcance adicional para los volúmenes que aprovisiones.

```
---
apiVersion: v1
kind: Secret
metadata:
  name: backend-tbc-gcnv-secret
type: Opaque
stringData:
  private_key_id: f2cb6ed6d7cc10c453f7d3406fc700c5df0ab9ec
  private_key: |
    -----BEGIN PRIVATE KEY-----\n
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m\n
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m\n
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m\n
    XsYg6gyxy4zq7OlwWgLwGa==\n
    -----END PRIVATE KEY-----\n

---
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-gcnv
spec:
  version: 1
  storageDriverName: google-cloud-netapp-volumes
  projectNumber: "123455380079"
  location: europe-west6
  serviceLevel: premium
  apiKey:
    type: service_account
    project_id: my-gcnv-project
    client_email: myproject-prod@my-gcnv-
project.iam.gserviceaccount.com
    client_id: "103346282737811234567"
    auth_uri: https://accounts.google.com/o/oauth2/auth
    token_uri: https://oauth2.googleapis.com/token
    auth_provider_x509_cert_url:
https://www.googleapis.com/oauth2/v1/certs
    client_x509_cert_url:
https://www.googleapis.com/robot/v1/metadata/x509/myproject-prod%40my-
gcnv-project.iam.gserviceaccount.com
  credentials:
    name: backend-tbc-gcnv-secret
```

Configuración para volúmenes SMB

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-gcnv1
  namespace: trident
spec:
  version: 1
  storageDriverName: google-cloud-netapp-volumes
  projectNumber: "123456789"
  location: asia-east1
  serviceLevel: flex
  nasType: smb
  apiKey:
    type: service_account
    project_id: cloud-native-data
    client_email: trident-sample@cloud-native-
data.iam.gserviceaccount.com
    client_id: "123456789737813416734"
    auth_uri: https://accounts.google.com/o/oauth2/auth
    token_uri: https://oauth2.googleapis.com/token
    auth_provider_x509_cert_url:
https://www.googleapis.com/oauth2/v1/certs
    client_x509_cert_url:
https://www.googleapis.com/robot/v1/metadata/x509/trident-
sample%40cloud-native-data.iam.gserviceaccount.com
  credentials:
    name: backend-tbc-gcnv-secret
```

Configuración con filtro StoragePools

```
---
apiVersion: v1
kind: Secret
metadata:
  name: backend-tbc-gcnv-secret
type: Opaque
stringData:
  private_key_id: f2cb6ed6d7cc10c453f7d3406fc700c5df0ab9ec
  private_key: |
    -----BEGIN PRIVATE KEY-----
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    XsYg6gyxy4zq7OlwWgLwGa==
    -----END PRIVATE KEY-----

---
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-gcnv
spec:
  version: 1
  storageDriverName: google-cloud-netapp-volumes
  projectNumber: "123455380079"
  location: europe-west6
  serviceLevel: premium
  storagePools:
    - premium-pool1-europe-west6
    - premium-pool2-europe-west6
  apiKey:
    type: service_account
    project_id: my-gcnv-project
    client_email: myproject-prod@my-gcnv-
project.iam.gserviceaccount.com
    client_id: "103346282737811234567"
    auth_uri: https://accounts.google.com/o/oauth2/auth
    token_uri: https://oauth2.googleapis.com/token
    auth_provider_x509_cert_url:
https://www.googleapis.com/oauth2/v1/certs
    client_x509_cert_url:
https://www.googleapis.com/robot/v1/metadata/x509/myproject-prod%40my-
gcnv-project.iam.gserviceaccount.com
  credentials:
    name: backend-tbc-gcnv-secret
```

Configuración de grupo virtual

Esta configuración de backend define múltiples pools virtuales en un solo archivo. Los grupos virtuales se definen en el `storage` sección. Son útiles cuando se tienen varios grupos de almacenamiento que admiten diferentes niveles de servicio y se desea crear clases de almacenamiento en Kubernetes que los representen. Las etiquetas de los grupos virtuales se utilizan para diferenciar los grupos. Por ejemplo, en el siguiente ejemplo. `performance` etiqueta y `serviceLevel` El tipo se utiliza para diferenciar los grupos virtuales.

También puede establecer algunos valores predeterminados que se aplicarán a todos los grupos virtuales y sobrescribir los valores predeterminados para grupos virtuales individuales. En el siguiente ejemplo, `snapshotReserve` y `exportRule` sirven como valores predeterminados para todos los grupos virtuales.

Para obtener más información, consulte"piscinas virtuales" .

```
---
apiVersion: v1
kind: Secret
metadata:
  name: backend-tbc-gcnv-secret
type: Opaque
stringData:
  private_key_id: f2cb6ed6d7cc10c453f7d3406fc700c5df0ab9ec
  private_key: |
    -----BEGIN PRIVATE KEY-----
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzZE4jK3bl/qp8B4Kws8zX5ojY9m
    XsYg6gyxy4zq7OlwWgLwGa==
    -----END PRIVATE KEY-----

---
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-gcnv
spec:
  version: 1
  storageDriverName: google-cloud-netapp-volumes
  projectNumber: "123455380079"
  location: europe-west6
  apiKey:
    type: service_account
    project_id: my-gcnv-project
    client_email: myproject-prod@my-gcnv-
project.iam.gserviceaccount.com
```

```
    client_id: "103346282737811234567"
    auth_uri: https://accounts.google.com/o/oauth2/auth
    token_uri: https://oauth2.googleapis.com/token
    auth_provider_x509_cert_url:
https://www.googleapis.com/oauth2/v1/certs
    client_x509_cert_url:
https://www.googleapis.com/robot/v1/metadata/x509/myproject-prod%40my-
gcnv-project.iam.gserviceaccount.com
  credentials:
    name: backend-tbc-gcnv-secret
  defaults:
    snapshotReserve: "10"
    exportRule: 10.0.0.0/24
  storage:
    - labels:
        performance: extreme
      serviceLevel: extreme
      defaults:
        snapshotReserve: "5"
        exportRule: 0.0.0.0/0
    - labels:
        performance: premium
      serviceLevel: premium
    - labels:
        performance: standard
      serviceLevel: standard
```

Identidad en la nube para GKE

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-gcp-gcnv
spec:
  version: 1
  storageDriverName: google-cloud-netapp-volumes
  projectNumber: '012345678901'
  network: gcnv-network
  location: us-west2
  serviceLevel: Premium
  storagePool: pool-premium1
```

Configuración de topologías admitidas

Trident facilita el aprovisionamiento de volúmenes para cargas de trabajo en función de regiones y zonas de disponibilidad. El `supportedTopologies` Este bloque en la configuración del backend se utiliza para proporcionar una lista de regiones y zonas por backend. Los valores de región y zona especificados aquí deben coincidir con los valores de región y zona de las etiquetas de cada nodo del clúster de Kubernetes. Estas regiones y zonas representan la lista de valores permitidos que se pueden proporcionar en una clase de almacenamiento. Para las clases de almacenamiento que contienen un subconjunto de las regiones y zonas proporcionadas en un backend, Trident crea volúmenes en la región y zona mencionadas. Para obtener más información, consulte"Utilizar la topología CSI" .

```
---
version: 1
storageDriverName: google-cloud-netapp-volumes
subscriptionID: 9f87c765-4774-fake-ae98-a721add45451
tenantID: 68e4f836-edc1-fake-bff9-b2d865ee56cf
clientID: dd043f63-bf8e-fake-8076-8de91e5713aa
clientSecret: SECRET
location: asia-east1
serviceLevel: flex
supportedTopologies:
  - topology.kubernetes.io/region: asia-east1
    topology.kubernetes.io/zone: asia-east1-a
  - topology.kubernetes.io/region: asia-east1
    topology.kubernetes.io/zone: asia-east1-b
```

¿Que sigue?

Después de crear el archivo de configuración del backend, ejecute el siguiente comando:

```
kubectl create -f <backend-file>
```

Para verificar que el backend se ha creado correctamente, ejecute el siguiente comando:

```
kubectl get tridentbackendconfig

NAME                BACKEND NAME        BACKEND UUID
PHASE   STATUS
backend-tbc-gcnv    backend-tbc-gcnv    b2fd1ff9-b234-477e-88fd-713913294f65
Bound   Success
```

Si falla la creación del backend, algo falla en la configuración del backend. Puedes describir el backend usando el `kubectl get tridentbackendconfig <backend-name>` Ejecute el siguiente comando o consulte los registros para determinar la causa:

```
tridentctl logs
```

Una vez que haya identificado y corregido el problema con el archivo de configuración, puede eliminar el backend y volver a ejecutar el comando de creación.

Definiciones de clases de almacenamiento

Lo siguiente es un esquema básico `StorageClass` Definición que hace referencia al backend mencionado anteriormente.

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: gcnv-nfs-sc
provisioner: csi.trident.netapp.io
parameters:
  backendType: "google-cloud-netapp-volumes"
```

Ejemplos de definiciones utilizando el `parameter.selector` campo:

Usando `parameter.selector` Puedes especificarlo para cada uno. `StorageClass` el"piscina virtual" que se utiliza para alojar un volumen. El volumen tendrá los aspectos definidos en el pool elegido.

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: extreme-sc
provisioner: csi.trident.netapp.io
parameters:
  selector: performance=extreme
  backendType: google-cloud-netapp-volumes

---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: premium-sc
provisioner: csi.trident.netapp.io
parameters:
  selector: performance=premium
  backendType: google-cloud-netapp-volumes

---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: standard-sc
provisioner: csi.trident.netapp.io
parameters:
  selector: performance=standard
  backendType: google-cloud-netapp-volumes
```

Para obtener más detalles sobre las clases de almacenamiento, consulte"Crear una clase de almacenamiento" .

Definiciones de ejemplo para volúmenes SMB

Usando `nasType` , `node-stage-secret-name` , y `node-stage-secret-namespace` , puede especificar un volumen SMB y proporcionar las credenciales de Active Directory necesarias. Se puede utilizar cualquier usuario/contraseña de Active Directory con cualquier permiso o sin permisos para el secreto de etapa del nodo.

Configuración básica en el espacio de nombres predeterminado

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: gcnv-sc-smb
provisioner: csi.trident.netapp.io
parameters:
  backendType: "google-cloud-netapp-volumes"
  trident.netapp.io/nasType: "smb"
  csi.storage.k8s.io/node-stage-secret-name: "smbcreds"
  csi.storage.k8s.io/node-stage-secret-namespace: "default"
```

Utilizar diferentes secretos por espacio de nombres

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: gcnv-sc-smb
provisioner: csi.trident.netapp.io
parameters:
  backendType: "google-cloud-netapp-volumes"
  trident.netapp.io/nasType: "smb"
  csi.storage.k8s.io/node-stage-secret-name: "smbcreds"
  csi.storage.k8s.io/node-stage-secret-namespace: ${pvc.namespace}
```

Utilizando diferentes secretos por volumen

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: gcnv-sc-smb
provisioner: csi.trident.netapp.io
parameters:
  backendType: "google-cloud-netapp-volumes"
  trident.netapp.io/nasType: "smb"
  csi.storage.k8s.io/node-stage-secret-name: ${pvc.name}
  csi.storage.k8s.io/node-stage-secret-namespace: ${pvc.namespace}
```

`nasType: smb`Filtros para pools que admiten volúmenes SMB. `nasType: nfs` o `nasType: null` Filtros para pools NFS.

Ejemplo de definición de PVC

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: gcnv-nfs-pvc
spec:
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 100Gi
  storageClassName: gcnv-nfs-sc
```

Para verificar si el PVC está enlazado, ejecute el siguiente comando:

```
kubectl get pvc gcnv-nfs-pvc

NAME          STATUS   VOLUME                                     CAPACITY
ACCESS MODES   STORAGECLASS AGE
gcnv-nfs-pvc  Bound    pvc-b00f2414-e229-40e6-9b16-ee03eb79a213  100Gi
RWX            gcnv-nfs-sc  1m
```

Configurar un Cloud Volumes Service para el backend de Google Cloud

Aprenda a configurar NetApp Cloud Volumes Service para Google Cloud como backend para su instalación de Trident utilizando las configuraciones de ejemplo proporcionadas.

Detalles del controlador de Google Cloud

Trident proporciona el `gcp-cvs` controlador para comunicarse con el clúster. Los modos de acceso compatibles son: *ReadWriteOnce* (RWO), *ReadOnlyMany* (ROX), *ReadWriteMany* (RWX), *ReadWriteOncePod* (RWOP).

Conductor	Protocolo	modo de volumen	Modos de acceso compatibles	Sistemas de archivos compatibles
`gcp-cvs`	Sistema Nacional de Archivos	Sistema de archivos	RWO, ROX, RWX, RWOP	`nfs`

Obtén más información sobre la compatibilidad de Trident con Cloud Volumes Service para Google Cloud.

Trident puede crear volúmenes de Cloud Volumes Service en uno de dos"tipos de servicio" :

- **CVS-Performance**: El tipo de servicio Trident predeterminado. Este tipo de servicio optimizado para el rendimiento es el más adecuado para cargas de trabajo de producción que valoran el rendimiento. El tipo de servicio CVS-Performance es una opción de hardware que admite volúmenes con un tamaño mínimo de 100 GiB. Puedes elegir uno de"tres niveles de servicio" :
  - `standard`
  - `premium`
  - `extreme`
- **CVS**: El tipo de servicio CVS proporciona una alta disponibilidad zonal con niveles de rendimiento limitados a moderados. El tipo de servicio CVS es una opción de software que utiliza grupos de almacenamiento para admitir volúmenes de tan solo 1 GiB. El grupo de almacenamiento puede contener hasta 50 volúmenes, donde todos los volúmenes comparten la capacidad y el rendimiento del grupo. Puedes elegir uno de"dos niveles de servicio" :
  - `standardsw`
  - `zoneredundantstandardsw`

Lo que necesitarás

Para configurar y usar el "Cloud Volumes Service para Google Cloud" Para el backend, necesitas lo siguiente:

- Una cuenta de Google Cloud configurada con el servicio NetApp Cloud Volumes Service.
- Número de proyecto de tu cuenta de Google Cloud
- cuenta de servicio de Google Cloud con la `netappcloudvolumes.admin` role
- Archivo de clave API para su cuenta de Cloud Volumes Service

Opciones de configuración del backend

Cada backend aprovisiona volúmenes en una única región de Google Cloud. Para crear volúmenes en otras regiones, puede definir backends adicionales.

Parámetro	Descripción	Por defecto
`version`		Siempre 1
`storageDriverName`	Nombre del controlador de almacenamiento	"gcp-cvs"
`backendName`	Nombre personalizado o el backend de almacenamiento	Nombre del controlador + "_" + parte de la clave API
`storageClass`	Parámetro opcional utilizado para especificar el tipo de servicio CVS. Usar `software` para seleccionar el tipo de servicio CVS. De lo contrario, Trident asume el tipo de servicio CVS-Performance.(`hardware`).	
`storagePools`	Solo servicio de tipo CVS. Parámetro opcional utilizado para especificar los grupos de almacenamiento para la creación de volúmenes.	

Parámetro	Descripción	Por defecto
`projectNumber`	Número de proyecto de la cuenta de Google Cloud. El valor se encuentra en la página principal del portal de Google Cloud.	
`hostProjectNumber`	Requerido si se utiliza una red VPC compartida. En este escenario, `projectNumber` es el proyecto de servicio, y `hostProjectNumber` es el proyecto anfitrión.	
`apiRegion`	La región de Google Cloud donde Trident crea volúmenes de Cloud Volumes Service . Al crear clústeres de Kubernetes entre regiones, los volúmenes creados en un `apiRegion` Puede utilizarse en cargas de trabajo programadas en nodos de varias regiones de Google Cloud. El tráfico interregional conlleva un coste adicional.	
`apiKey`	clave API para la cuenta de servicio de Google Cloud con la `netappcloudvolumes.admin` role. Incluye el contenido en formato JSON del archivo de clave privada de una cuenta de servicio de Google Cloud (copiado textualmente en el archivo de configuración del backend).	
`proxyURL`	URL del proxy si se requiere un servidor proxy para conectarse a la cuenta de CVS. El servidor proxy puede ser un proxy HTTP o un proxy HTTPS. En el caso de un proxy HTTPS, se omite la validación del certificado para permitir el uso de certificados autofirmados en el servidor proxy. No se admiten servidores proxy con autenticación habilitada.	
`nfsMountOptions`	Control preciso de las opciones de montaje NFS.	"nfsvers=3"
`limitVolumeSize`	Fallará el aprovisionamiento si el tamaño de volumen solicitado supera este valor.	" (no se aplica por defecto)
`serviceLevel`	El nivel de servicio CVS-Performance o CVS para nuevos volúmenes. Los valores de rendimiento de CVS son `standard` , `premium` , o `extreme` . Los valores CVS son `standardsw` o `zoneredundantstandardsw` .	El valor predeterminado de CVS-Performance es "estándar". El valor predeterminado de CVS es "standardsw".
`network`	Red de Google Cloud utilizada para los volúmenes del Cloud Volumes Service .	"por defecto"
`debugTraceFlags`	Indicadores de depuración para usar al solucionar problemas. Ejemplo, `\{"api":false, "method":true}` . No utilice esta función a menos que esté solucionando problemas y necesite un registro detallado.	nulo

Parámetro	Descripción	Por defecto

`allowedTopologies`	Para habilitar el acceso entre regiones, su definición de StorageClass para `allowedTopologies` Debe incluir todas las regiones. Por ejemplo:
	`- key: topology.kubernetes.io/region`
	`values:`
	`- us-east1`
	`- europe-west1`	

Opciones de aprovisionamiento de volumen

Puedes controlar el aprovisionamiento de volúmenes predeterminado en el `defaults` sección del archivo de configuración.

Parámetro	Descripción	Por defecto
`exportRule`	Las reglas de exportación para nuevos volúmenes. Debe ser una lista separada por comas de cualquier combinación de direcciones IPv4 o subredes IPv4 en notación CIDR.	"0.0.0.0/0"
`snapshotDir`	Acceso a la `.snapshot` directorio	"FALSO"
`snapshotReserve`	Porcentaje de volumen reservado para instantáneas	"" (aceptar el valor predeterminado de CVS de 0)
`size`	El tamaño de los nuevos volúmenes. El rendimiento mínimo de CVS es de 100 GiB. El tamaño mínimo de CVS es de 1 GiB.	El tipo de servicio CVS-Performance tiene como valor predeterminado "100 GiB". El tipo de servicio CVS no establece un valor predeterminado, pero requiere un mínimo de 1 GiB.

Ejemplos de tipos de servicio de CVS-Performance

Los siguientes ejemplos proporcionan configuraciones de muestra para el tipo de servicio CVS-Performance.

Ejemplo 1: Configuración mínima

Esta es la configuración mínima de backend utilizando el tipo de servicio CVS-Performance predeterminado con el nivel de servicio "estándar" predeterminado.

```
---
version: 1
storageDriverName: gcp-cvs
projectNumber: "012345678901"
apiRegion: us-west2
apiKey:
  type: service_account
  project_id: my-gcp-project
  private_key_id: <id_value>
  private_key: |
    -----BEGIN PRIVATE KEY-----
    <key_value>
    -----END PRIVATE KEY-----
  client_email: cloudvolumes-admin-sa@my-gcp-
project.iam.gserviceaccount.com
  client_id: "123456789012345678901"
  auth_uri: https://accounts.google.com/o/oauth2/auth
  token_uri: https://oauth2.googleapis.com/token
  auth_provider_x509_cert_url:
https://www.googleapis.com/oauth2/v1/certs
  client_x509_cert_url:
https://www.googleapis.com/robot/v1/metadata/x509/cloudvolumes-admin-
sa%40my-gcp-project.iam.gserviceaccount.com
```

Ejemplo 2: Configuración del nivel de servicio

Este ejemplo ilustra las opciones de configuración del backend, incluyendo el nivel de servicio y los valores predeterminados de volumen.

```
---
version: 1
storageDriverName: gcp-cvs
projectNumber: '012345678901'
apiRegion: us-west2
apiKey:
  type: service_account
  project_id: my-gcp-project
  private_key_id: "<id_value>"
  private_key: |
    -----BEGIN PRIVATE KEY-----
    <key_value>
    -----END PRIVATE KEY-----
  client_email: cloudvolumes-admin-sa@my-gcp-
project.iam.gserviceaccount.com
  client_id: '123456789012345678901'
  auth_uri: https://accounts.google.com/o/oauth2/auth
  token_uri: https://oauth2.googleapis.com/token
  auth_provider_x509_cert_url:
https://www.googleapis.com/oauth2/v1/certs
  client_x509_cert_url:
https://www.googleapis.com/robot/v1/metadata/x509/cloudvolumes-admin-
sa%40my-gcp-project.iam.gserviceaccount.com
proxyURL: http://proxy-server-hostname/
nfsMountOptions: vers=3,proto=tcp,timeo=600
limitVolumeSize: 10Ti
serviceLevel: premium
defaults:
  snapshotDir: 'true'
  snapshotReserve: '5'
  exportRule: 10.0.0.0/24,10.0.1.0/24,10.0.2.100
  size: 5Ti
```

Ejemplo 3: Configuración de grupo virtual

Esta muestra utiliza `storage` para configurar grupos virtuales y el `StorageClasses` que hacen referencia a ellos. Referirse aDefiniciones de clases de almacenamiento para ver cómo se definieron las clases de almacenamiento.

Aquí se establecen valores predeterminados específicos para todos los grupos virtuales, que definen `snapshotReserve` al 5% y el `exportRule` a 0.0.0.0/0. Los grupos virtuales se definen en el `storage` sección. Cada grupo virtual individual define su propio `serviceLevel` , y algunos grupos sobrescriben los valores predeterminados. Se utilizaron etiquetas de piscinas virtuales para diferenciar las piscinas en función de `performance` y `protection` .

```
---
version: 1
storageDriverName: gcp-cvs
projectNumber: '012345678901'
apiRegion: us-west2
apiKey:
  type: service_account
  project_id: my-gcp-project
  private_key_id: "<id_value>"
  private_key: |
    -----BEGIN PRIVATE KEY-----
    <key_value>
    -----END PRIVATE KEY-----
  client_email: cloudvolumes-admin-sa@my-gcp-
project.iam.gserviceaccount.com
  client_id: '123456789012345678901'
  auth_uri: https://accounts.google.com/o/oauth2/auth
  token_uri: https://oauth2.googleapis.com/token
  auth_provider_x509_cert_url:
https://www.googleapis.com/oauth2/v1/certs
  client_x509_cert_url:
https://www.googleapis.com/robot/v1/metadata/x509/cloudvolumes-admin-
sa%40my-gcp-project.iam.gserviceaccount.com
nfsMountOptions: vers=3,proto=tcp,timeo=600
defaults:
  snapshotReserve: '5'
  exportRule: 0.0.0.0/0
labels:
  cloud: gcp
region: us-west2
storage:
- labels:
    performance: extreme
    protection: extra
  serviceLevel: extreme
```

```
  defaults:
    snapshotDir: 'true'
    snapshotReserve: '10'
    exportRule: 10.0.0.0/24
- labels:
    performance: extreme
    protection: standard
  serviceLevel: extreme
- labels:
    performance: premium
    protection: extra
  serviceLevel: premium
  defaults:
    snapshotDir: 'true'
    snapshotReserve: '10'
- labels:
    performance: premium
    protection: standard
  serviceLevel: premium
- labels:
    performance: standard
  serviceLevel: standard
```

Definiciones de clases de almacenamiento

Las siguientes definiciones de StorageClass se aplican al ejemplo de configuración de grupo virtual. Usando `parameters.selector` , puede especificar para cada StorageClass el grupo virtual utilizado para alojar un volumen. El volumen tendrá los aspectos definidos en el pool elegido.

Ejemplo de clase de almacenamiento

```
---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: cvs-extreme-extra-protection
provisioner: csi.trident.netapp.io
parameters:
  selector: performance=extreme; protection=extra
allowVolumeExpansion: true
---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: cvs-extreme-standard-protection
provisioner: csi.trident.netapp.io
parameters:
  selector: performance=premium; protection=standard
allowVolumeExpansion: true
---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: cvs-premium-extra-protection
provisioner: csi.trident.netapp.io
parameters:
  selector: performance=premium; protection=extra
allowVolumeExpansion: true
---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: cvs-premium
provisioner: csi.trident.netapp.io
parameters:
  selector: performance=premium; protection=standard
allowVolumeExpansion: true
---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: cvs-standard
provisioner: csi.trident.netapp.io
parameters:
  selector: performance=standard
```

```
allowVolumeExpansion: true
---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: cvs-extra-protection
provisioner: csi.trident.netapp.io
parameters:
  selector: protection=extra
allowVolumeExpansion: true
```

- La primera clase de almacenamiento(`cvs-extreme-extra-protection`) se asigna al primer grupo virtual. Esta es la única piscina que ofrece un rendimiento extremo con una reserva instantánea del 10%.
- La última clase de almacenamiento(`cvs-extra-protection`) menciona cualquier grupo de almacenamiento que proporcione una reserva de instantáneas del 10%. Trident decide qué grupo virtual se selecciona y garantiza que se cumpla el requisito de reserva de instantáneas.

Ejemplos de tipos de servicio de CVS

Los siguientes ejemplos proporcionan configuraciones de muestra para el tipo de servicio CVS.

Ejemplo 1: Configuración mínima

Esta es la configuración mínima de backend que utiliza `storageClass` para especificar el tipo de servicio CVS y el valor predeterminado `standardsw` nivel de servicio.

```
---
version: 1
storageDriverName: gcp-cvs
projectNumber: '012345678901'
storageClass: software
apiRegion: us-east4
apiKey:
  type: service_account
  project_id: my-gcp-project
  private_key_id: "<id_value>"
  private_key: |
    -----BEGIN PRIVATE KEY-----
    <key_value>
    -----END PRIVATE KEY-----
  client_email: cloudvolumes-admin-sa@my-gcp-
project.iam.gserviceaccount.com
  client_id: '123456789012345678901'
  auth_uri: https://accounts.google.com/o/oauth2/auth
  token_uri: https://oauth2.googleapis.com/token
  auth_provider_x509_cert_url:
https://www.googleapis.com/oauth2/v1/certs
  client_x509_cert_url:
https://www.googleapis.com/robot/v1/metadata/x509/cloudvolumes-admin-
sa%40my-gcp-project.iam.gserviceaccount.com
serviceLevel: standardsw
```

Ejemplo 2: Configuración del grupo de almacenamiento

Esta configuración de backend de ejemplo utiliza `storagePools` para configurar un grupo de almacenamiento.

```
---
version: 1
storageDriverName: gcp-cvs
backendName: gcp-std-so-with-pool
projectNumber: '531265380079'
apiRegion: europe-west1
apiKey:
  type: service_account
  project_id: cloud-native-data
  private_key_id: "<id_value>"
  private_key: |-
    -----BEGIN PRIVATE KEY-----
    <key_value>
    -----END PRIVATE KEY-----
  client_email: cloudvolumes-admin-sa@cloud-native-
data.iam.gserviceaccount.com
  client_id: '107071413297115343396'
  auth_uri: https://accounts.google.com/o/oauth2/auth
  token_uri: https://oauth2.googleapis.com/token
  auth_provider_x509_cert_url:
https://www.googleapis.com/oauth2/v1/certs
  client_x509_cert_url:
https://www.googleapis.com/robot/v1/metadata/x509/cloudvolumes-admin-
sa%40cloud-native-data.iam.gserviceaccount.com
storageClass: software
zone: europe-west1-b
network: default
storagePools:
- 1bc7f380-3314-6005-45e9-c7dc8c2d7509
serviceLevel: Standardsw
```

¿Que sigue?

Después de crear el archivo de configuración del backend, ejecute el siguiente comando:

```
tridentctl create backend -f <backend-file>
```

Si falla la creación del backend, algo falla en la configuración del backend. Puedes consultar los registros para determinar la causa ejecutando el siguiente comando:

```
tridentctl logs
```

Después de identificar y corregir el problema con el archivo de configuración, puede volver a ejecutar el comando de creación.

Configure un backend de NetApp HCI o SolidFire

Aprende cómo crear y usar un backend de Element con tu instalación de Trident .

Detalles del controlador Element

Trident proporciona el `solidfire-san` Controlador de almacenamiento para comunicarse con el clúster. Los modos de acceso compatibles son: *ReadWriteOnce* (RWO), *ReadOnlyMany* (ROX), *ReadWriteMany* (RWX), *ReadWriteOncePod* (RWOP).

El `solidfire-san` El controlador de almacenamiento admite los modos de volumen *archivo* y *bloque*. Para el `Filesystem` En volumeMode, Trident crea un volumen y un sistema de archivos. El tipo de sistema de archivos se especifica mediante la StorageClass.

Conductor	Protocolo	Modo de volumen	Modos de acceso compatibles	Sistemas de archivos compatibles
`solidfire-san`	iSCSI	Bloquear	RWO, ROX, RWX, RWOP	Sin sistema de archivos. Dispositivo de bloque sin procesar.
`solidfire-san`	iSCSI	Sistema de archivos	RWO, RWOP	`xfs`, `ext3` , `ext4`

Antes de empezar

Necesitarás lo siguiente antes de crear un backend de Element.

- Un sistema de almacenamiento compatible que ejecuta el software Element.
- Credenciales de un administrador de clúster NetApp HCI/ SolidFire o de un usuario inquilino que pueda administrar volúmenes.
- Todos tus nodos de trabajo de Kubernetes deben tener instaladas las herramientas iSCSI apropiadas. Referirse a"Información de preparación del nodo de trabajo" .

Opciones de configuración del backend

Consulte la siguiente tabla para ver las opciones de configuración del backend:

Parámetro	Descripción	Por defecto
`version`		Siempre 1
`storageDriverName`	Nombre del controlador de almacenamiento	Siempre "solidfire-san"

Parámetro	Descripción	Por defecto
`backendName`	Nombre personalizado o el backend de almacenamiento	"solidfire_" + dirección IP de almacenamiento (iSCSI)
`Endpoint`	MVIP para el clúster SolidFire con credenciales de inquilino	
`SVIP`	Dirección IP y puerto de almacenamiento (iSCSI)	
`labels`	Conjunto de etiquetas arbitrarias con formato JSON para aplicar a los volúmenes.	""
`TenantName`	Nombre del inquilino a utilizar (se creará si no se encuentra)	
`InitiatorIFace`	Restringir el tráfico iSCSI a una interfaz de host específica	"por defecto"
`UseCHAP`	Utilice CHAP para autenticar iSCSI. Trident utiliza CHAP.	verdadero
`AccessGroups`	Lista de identificadores de grupos de acceso para usar	Encuentra el ID de un grupo de acceso llamado "trident".
`Types`	Especificaciones de QoS	
`limitVolumeSize`	Fallará el aprovisionamiento si el tamaño del volumen solicitado supera este valor.	" (no se aplica por defecto)
`debugTraceFlags`	Indicadores de depuración para usar al solucionar problemas. Ejemplo: {"api":false, "method":true}	nulo

No utilizar `debugTraceFlags` a menos que esté solucionando problemas y necesite un registro detallado.

Ejemplo 1: Configuración del backend para `solidfire-san` Controlador con tres tipos de volumen

Este ejemplo muestra un archivo de backend que utiliza la autenticación CHAP y modela tres tipos de volúmenes con garantías de QoS específicas. Lo más probable es que luego definas clases de almacenamiento para consumir cada una de ellas utilizando el `IOPS` Parámetro de clase de almacenamiento.

```
---
version: 1
storageDriverName: solidfire-san
Endpoint: https://<user>:<password>@<mvip>/json-rpc/8.0
SVIP: <svip>:3260
TenantName: <tenant>
labels:
  k8scluster: dev1
  backend: dev1-element-cluster
UseCHAP: true
Types:
  - Type: Bronze
    Qos:
      minIOPS: 1000
      maxIOPS: 2000
      burstIOPS: 4000
  - Type: Silver
    Qos:
      minIOPS: 4000
      maxIOPS: 6000
      burstIOPS: 8000
  - Type: Gold
    Qos:
      minIOPS: 6000
      maxIOPS: 8000
      burstIOPS: 10000
```

Ejemplo 2: Configuración de la clase de almacenamiento y del backend para `solidfire-san` controlador con grupos virtuales

Este ejemplo muestra el archivo de definición del backend configurado con pools virtuales junto con las StorageClasses que hacen referencia a ellos.

Trident copia las etiquetas presentes en un grupo de almacenamiento al LUN de almacenamiento backend durante el aprovisionamiento. Para mayor comodidad, los administradores de almacenamiento pueden definir etiquetas por grupo virtual y agrupar volúmenes por etiqueta.

En el archivo de definición de backend de ejemplo que se muestra a continuación, se establecen valores predeterminados específicos para todos los grupos de almacenamiento, que establecen el `type` en Plata. Los grupos virtuales se definen en el `storage` sección. En este ejemplo, algunos de los grupos de almacenamiento establecen su propio tipo, y algunos grupos anulan los valores predeterminados establecidos anteriormente.

```
---
version: 1
storageDriverName: solidfire-san
Endpoint: https://<user>:<password>@<mvip>/json-rpc/8.0
```

```
SVIP: <svip>:3260
TenantName: <tenant>
UseCHAP: true
Types:
  - Type: Bronze
    Qos:
      minIOPS: 1000
      maxIOPS: 2000
      burstIOPS: 4000
  - Type: Silver
    Qos:
      minIOPS: 4000
      maxIOPS: 6000
      burstIOPS: 8000
  - Type: Gold
    Qos:
      minIOPS: 6000
      maxIOPS: 8000
      burstIOPS: 10000
type: Silver
labels:
  store: solidfire
  k8scluster: dev-1-cluster
region: us-east-1
storage:
  - labels:
      performance: gold
      cost: "4"
    zone: us-east-1a
    type: Gold
  - labels:
      performance: silver
      cost: "3"
    zone: us-east-1b
    type: Silver
  - labels:
      performance: bronze
      cost: "2"
    zone: us-east-1c
    type: Bronze
  - labels:
      performance: silver
      cost: "1"
    zone: us-east-1d
```

Las siguientes definiciones de StorageClass hacen referencia a los grupos virtuales mencionados

anteriormente. Utilizando el `parameters.selector` En cada campo, cada StorageClass especifica qué grupo o grupos virtuales se pueden usar para alojar un volumen. El volumen tendrá los aspectos definidos en el pool virtual elegido.

La primera clase de almacenamiento(`solidfire-gold-four`) se asignará al primer grupo virtual. Esta es la única piscina que ofrece rendimiento oro con un `Volume Type QoS` de oro. La última clase de almacenamiento(`solidfire-silver`) menciona cualquier grupo de almacenamiento que ofrezca un rendimiento plateado. Trident decidirá qué grupo virtual se selecciona y garantiza que se cumplan los requisitos de almacenamiento.

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: solidfire-gold-four
provisioner: csi.trident.netapp.io
parameters:
  selector: performance=gold; cost=4
  fsType: ext4

---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: solidfire-silver-three
provisioner: csi.trident.netapp.io
parameters:
  selector: performance=silver; cost=3
  fsType: ext4

---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: solidfire-bronze-two
provisioner: csi.trident.netapp.io
parameters:
  selector: performance=bronze; cost=2
  fsType: ext4

---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: solidfire-silver-one
provisioner: csi.trident.netapp.io
parameters:
  selector: performance=silver; cost=1
```

```
  fsType: ext4

---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: solidfire-silver
provisioner: csi.trident.netapp.io
parameters:
  selector: performance=silver
  fsType: ext4
```

Encuentra más información

- "Grupos de acceso por volumen"

Controladores SAN de ONTAP

Descripción general del controlador SAN de ONTAP

Aprenda a configurar un backend de ONTAP con ONTAP y los controladores SAN de Cloud Volumes ONTAP .

Detalles del controlador SAN de ONTAP

Trident proporciona los siguientes controladores de almacenamiento SAN para comunicarse con el clúster ONTAP . Los modos de acceso compatibles son: *ReadWriteOnce* (RWO), *ReadOnlyMany* (ROX), *ReadWriteMany* (RWX), *ReadWriteOncePod* (RWOP).

Conductor	Protocolo	modo de volumen	Modos de acceso compatibles	Sistemas de archivos compatibles
`ontap-san`	iSCSI SCSI sobre FC	Bloquear	RWO, ROX, RWX, RWOP	Sin sistema de archivos; dispositivo de bloques sin formato
`ontap-san`	iSCSI SCSI sobre FC	Sistema de archivos	RWO, RWOP Los parámetros ROX y RWX no están disponibles en el modo de volumen del sistema de archivos.	`xfs`, `ext3` , `ext4`

Conductor	Protocolo	modo de volumen	Modos de acceso compatibles	Sistemas de archivos compatibles
`ontap-san`	NVMe/TCP Referirse aConsideraciones adicionales para NVMe/TCP.	Bloquear	RWO, ROX, RWX, RWOP	Sin sistema de archivos; dispositivo de bloques sin formato
`ontap-san`	NVMe/TCP Referirse aConsideraciones adicionales para NVMe/TCP.	Sistema de archivos	RWO, RWOP Los parámetros ROX y RWX no están disponibles en el modo de volumen del sistema de archivos.	`xfs`, `ext3` , `ext4`
`ontap-san-economy`	iSCSI	Bloquear	RWO, ROX, RWX, RWOP	Sin sistema de archivos; dispositivo de bloques sin formato
`ontap-san-economy`	iSCSI	Sistema de archivos	RWO, RWOP Los parámetros ROX y RWX no están disponibles en el modo de volumen del sistema de archivos.	`xfs`, `ext3` , `ext4`

- Usar `ontap-san-economy` solo si se espera que el recuento de uso de volumen persistente sea superior a"límites de volumen ONTAP compatibles" .
- Usar `ontap-nas-economy` solo si se espera que el recuento de uso de volumen persistente sea superior a"límites de volumen ONTAP compatibles" y el `ontap-san-economy` El controlador no se puede utilizar.
- No usar `ontap-nas-economy` Si prevé la necesidad de protección de datos, recuperación ante desastres o movilidad.
- NetApp no recomienda usar el crecimiento automático de Flexvol en todos los controladores ONTAP , excepto en ontap-san. Como solución alternativa, Trident admite el uso de reserva de instantáneas y escala los volúmenes Flexvol en consecuencia.

Permisos de usuario

Trident espera ejecutarse como administrador de ONTAP o SVM, normalmente utilizando el `admin` usuario del clúster o un `vsadmin` Usuario de SVM, o un usuario con un nombre diferente que tenga la misma función.

Para implementaciones de Amazon FSx for NetApp ONTAP , Trident requiere ejecutarse como administrador de ONTAP o SVM, utilizando el clúster. `fsxadmin` usuario o un `vsadmin` Usuario de SVM, o un usuario con un nombre diferente que tenga la misma función. El `fsxadmin` El usuario es un reemplazo limitado para el usuario administrador del clúster.

> Si utiliza el `limitAggregateUsage` Se requieren permisos de administrador de clúster para este parámetro. Al utilizar Amazon FSx for NetApp ONTAP con Trident, `limitAggregateUsage` El parámetro no funcionará con el `vsadmin` y `fsxadmin` cuentas de usuario. La operación de configuración fallará si especifica este parámetro.

Si bien es posible crear un rol más restrictivo dentro de ONTAP que pueda usar un controlador Trident , no lo recomendamos. La mayoría de las nuevas versiones de Trident utilizarán API adicionales que habría que tener en cuenta, lo que dificultaría las actualizaciones y las haría propensas a errores.

Consideraciones adicionales para NVMe/TCP

Trident admite el protocolo de memoria no volátil express (NVMe) mediante el `ontap-san` controlador que incluye:

- IPv6
- Instantáneas y clones de volúmenes NVMe
- Redimensionar un volumen NVMe
- Importar un volumen NVMe creado fuera de Trident para que su ciclo de vida pueda ser gestionado por Trident.
- Multiruta nativa de NVMe
- Apagado correcto o incorrecto de los nodos K8s (24.06)

Trident no admite:

- DH-HMAC-CHAP compatible de forma nativa con NVMe
- Multiruta del mapeador de dispositivos (DM)
- Cifrado LUKS

> NVMe solo es compatible con las API REST de ONTAP y no con ONTAPI (ZAPI).

Prepárese para configurar el backend con los controladores SAN de ONTAP.

Comprenda los requisitos y las opciones de autenticación para configurar un backend ONTAP con controladores ONTAP SAN.

Requisitos

Para todos los backends de ONTAP , Trident requiere que se asigne al menos un agregado al SVM.

> "Sistemas ASA r2"Se diferencian de otros sistemas ONTAP (ASA, AFF y FAS) en la implementación de su capa de almacenamiento. En los sistemas ASA r2, se utilizan zonas de disponibilidad de almacenamiento en lugar de agregados. Referirse a"este" Artículo de la base de conocimientos sobre cómo asignar agregados a SVM en sistemas ASA r2.

Recuerda que también puedes ejecutar más de un controlador y crear clases de almacenamiento que apunten a uno u otro. Por ejemplo, podrías configurar un `san-dev` clase que utiliza la `ontap-san` conductor y un `san-default` clase que utiliza la `ontap-san-economy` uno.

Todos los nodos de trabajo de Kubernetes deben tener instaladas las herramientas iSCSI adecuadas. Referirse a "Preparar el nodo de trabajo" Para más detalles.

Autenticar el backend de ONTAP

Trident ofrece dos modos de autenticación de un backend ONTAP .

- Basado en credenciales: El nombre de usuario y la contraseña de un usuario de ONTAP con los permisos necesarios. Se recomienda utilizar un rol de inicio de sesión de seguridad predefinido, como por ejemplo: `admin` o `vsadmin` para garantizar la máxima compatibilidad con las versiones de ONTAP .
- Basado en certificados: Trident también puede comunicarse con un clúster ONTAP utilizando un certificado instalado en el backend. Aquí, la definición del backend debe contener valores codificados en Base64 del certificado del cliente, la clave y el certificado de CA de confianza si se utiliza (recomendado).

Puedes actualizar los sistemas backend existentes para alternar entre métodos basados en credenciales y métodos basados en certificados. Sin embargo, solo se admite un método de autenticación a la vez. Para cambiar a un método de autenticación diferente, debe eliminar el método existente de la configuración del backend.

> Si intenta proporcionar **tanto credenciales como certificados**, la creación del backend fallará con un error que indica que se proporcionó más de un método de autenticación en el archivo de configuración.

Habilitar la autenticación basada en credenciales

Trident requiere las credenciales de un administrador con ámbito SVM/ámbito de clúster para comunicarse con el backend de ONTAP . Se recomienda utilizar roles estándar predefinidos, tales como: `admin` o `vsadmin` . Esto garantiza la compatibilidad con versiones futuras de ONTAP que podrían exponer API de funciones para ser utilizadas por futuras versiones de Trident . Se puede crear y usar un rol de inicio de sesión de seguridad personalizado con Trident, pero no se recomienda.

Un ejemplo de definición de backend se vería así:

YAML

```yaml
---
version: 1
backendName: ExampleBackend
storageDriverName: ontap-san
managementLIF: 10.0.0.1
svm: svm_nfs
username: vsadmin
password: password
```

JSON

```json
{
  "version": 1,
  "backendName": "ExampleBackend",
  "storageDriverName": "ontap-san",
  "managementLIF": "10.0.0.1",
  "svm": "svm_nfs",
  "username": "vsadmin",
  "password": "password"
}
```

Tenga en cuenta que la definición del backend es el único lugar donde las credenciales se almacenan en texto plano. Una vez creado el backend, los nombres de usuario y las contraseñas se codifican con Base64 y se almacenan como secretos de Kubernetes. La creación o actualización de un backend es el único paso que requiere conocer las credenciales. Por lo tanto, se trata de una operación exclusiva para administradores, que debe ser realizada por el administrador de Kubernetes/almacenamiento.

Habilitar la autenticación basada en certificados

Los backends nuevos y existentes pueden usar un certificado y comunicarse con el backend de ONTAP . Se requieren tres parámetros en la definición del backend.

- clientCertificate: Valor codificado en Base64 del certificado del cliente.
- clientPrivateKey: Valor codificado en Base64 de la clave privada asociada.
- trustedCACertificate: Valor codificado en Base64 del certificado de CA de confianza. Si se utiliza una CA de confianza, este parámetro debe proporcionarse. Esto puede ignorarse si no se utiliza ninguna CA de confianza.

Un flujo de trabajo típico comprende los siguientes pasos.

Pasos

1. Generar un certificado y una clave de cliente. Al generar, configure el Nombre Común (CN) con el usuario ONTAP con el que se autenticará.

```
openssl req -x509 -nodes -days 1095 -newkey rsa:2048 -keyout k8senv.key
-out k8senv.pem -subj "/C=US/ST=NC/L=RTP/O=NetApp/CN=admin"
```

2. Agregar certificado de CA de confianza al clúster ONTAP . Es posible que esto ya lo gestione el administrador de almacenamiento. Ignorar si no se utiliza ninguna CA de confianza.

```
security certificate install -type server -cert-name <trusted-ca-cert-
name> -vserver <vserver-name>
ssl modify -vserver <vserver-name> -server-enabled true -client-enabled
true -common-name <common-name> -serial <SN-from-trusted-CA-cert> -ca
<cert-authority>
```

3. Instale el certificado y la clave del cliente (del paso 1) en el clúster ONTAP .

```
security certificate install -type client-ca -cert-name <certificate-
name> -vserver <vserver-name>
security ssl modify -vserver <vserver-name> -client-enabled true
```

4. Confirme que el rol de inicio de sesión de seguridad de ONTAP es compatible. `cert` Método de autenticación.

```
security login create -user-or-group-name admin -application ontapi
-authentication-method cert
security login create -user-or-group-name admin -application http
-authentication-method cert
```

5. Prueba de autenticación utilizando el certificado generado. Reemplace < ONTAP Management LIF> y <vserver name> con la IP de Management LIF y el nombre de SVM.

```
curl -X POST -Lk https://<ONTAP-Management-
LIF>/servlets/netapp.servlets.admin.XMLrequest_filer --key k8senv.key
--cert ~/k8senv.pem -d '<?xml version="1.0" encoding="UTF-8"?><netapp
xmlns="http://www.netapp.com/filer/admin" version="1.21"
vfiler="<vserver-name>"><vserver-get></vserver-get></netapp>'
```

6. Codifique el certificado, la clave y el certificado de CA de confianza con Base64.

```
base64 -w 0 k8senv.pem >> cert_base64
base64 -w 0 k8senv.key >> key_base64
base64 -w 0 trustedca.pem >> trustedca_base64
```

7. Crea el backend utilizando los valores obtenidos en el paso anterior.

```
cat cert-backend.json
{
"version": 1,
"storageDriverName": "ontap-san",
"backendName": "SanBackend",
"managementLIF": "1.2.3.4",
"svm": "vserver_test",
"clientCertificate": "Faaaakkkkeeee...Vaaalllluuuueeee",
"clientPrivateKey": "LS0tFaKE...0VaLuES0tLS0K",
"trustedCACertificate": "QNFinfO...SiqOyN",
"storagePrefix": "myPrefix_"
}

tridentctl create backend -f cert-backend.json -n trident
+------------+----------------+--------------------------------------
+--------+---------+
|    NAME    | STORAGE DRIVER |                 UUID                 |
STATE  | VOLUMES |
+------------+----------------+--------------------------------------
+--------+---------+
| SanBackend | ontap-san      | 586b1cd5-8cf8-428d-a76c-2872713612c1 |
online |       0 |
+------------+----------------+--------------------------------------
+--------+---------+
```

Actualizar los métodos de autenticación o rotar las credenciales

Puedes actualizar un backend existente para usar un método de autenticación diferente o para rotar sus credenciales. Esto funciona en ambos sentidos: los sistemas de gestión de backends que utilizan nombre de usuario/contraseña pueden actualizarse para usar certificados; los sistemas de gestión de backends que utilizan certificados pueden actualizarse para basarse en nombre de usuario/contraseña. Para ello, debe eliminar el método de autenticación existente y agregar el nuevo método de autenticación. A continuación, utilice el archivo backend.json actualizado que contiene los parámetros necesarios para ejecutar `tridentctl backend update`.

```
cat cert-backend-updated.json
{
"version": 1,
"storageDriverName": "ontap-san",
"backendName": "SanBackend",
"managementLIF": "1.2.3.4",
"svm": "vserver_test",
"username": "vsadmin",
"password": "password",
"storagePrefix": "myPrefix_"
}

#Update backend with tridentctl
tridentctl update backend SanBackend -f cert-backend-updated.json -n
trident
+------------+----------------+--------------------------------------
+--------+---------+
|    NAME    | STORAGE DRIVER |                 UUID                 |
STATE   | VOLUMES |
+------------+----------------+--------------------------------------
+--------+---------+
| SanBackend | ontap-san      | 586b1cd5-8cf8-428d-a76c-2872713612c1 |
online |       9 |
+------------+----------------+--------------------------------------
+--------+---------+
```

Al rotar las contraseñas, el administrador de almacenamiento primero debe actualizar la contraseña del usuario en ONTAP. A continuación se realiza una actualización del servidor. Al rotar los certificados, se pueden agregar varios certificados al usuario. Posteriormente, se actualiza el sistema backend para utilizar el nuevo certificado, tras lo cual se puede eliminar el certificado antiguo del clúster ONTAP .

La actualización de un backend no interrumpe el acceso a los volúmenes que ya se han creado, ni afecta a las conexiones de volumen realizadas posteriormente. Una actualización exitosa del backend indica que Trident puede comunicarse con el backend de ONTAP y gestionar futuras operaciones de volumen.

Cree un rol ONTAP personalizado para Trident.

Puede crear un rol de clúster ONTAP con privilegios mínimos para que no tenga que usar el rol de administrador de ONTAP para realizar operaciones en Trident. Cuando incluyes el nombre de usuario en una configuración de backend de Trident , Trident utiliza el rol de clúster ONTAP que creaste para realizar las operaciones.

Referirse a"Generador de roles personalizados de Trident" Para obtener más información sobre la creación de roles personalizados de Trident .

Uso de la CLI de ONTAP

1. Crea un nuevo rol utilizando el siguiente comando:

```
security login role create <role_name\> -cmddirname "command" -access all
–vserver <svm_name\>
```

2. Crea un nombre de usuario para el usuario de Trident :

```
security login create -username <user_name\> -application ontapi
-authmethod <password\> -role <name_of_role_in_step_1\> –vserver
<svm_name\> -comment "user_description"
```

3. Asigna el rol al usuario:

```
security login modify username <user_name\> –vserver <svm_name\> -role
<role_name\> -application ontapi -application console -authmethod
<password\>
```

Usando el Administrador del sistema

Realice los siguientes pasos en ONTAP System Manager:

1. **Crea un rol personalizado**:
   a. Para crear un rol personalizado a nivel de clúster, seleccione **Clúster > Configuración**.

      (O) Para crear un rol personalizado a nivel de SVM, seleccione **Almacenamiento > Máquinas virtuales de almacenamiento > `required SVM` > Configuración > Usuarios y roles**.

   b. Seleccione el icono de flecha (→) junto a **Usuarios y roles**.
   c. Seleccione ****Agregar** en **Roles**.
   d. Define las reglas para el rol y haz clic en **Guardar**.
2. **Asigna el rol al usuario de Trident *: + Realiza los siguientes pasos en la página *Usuarios y roles**:
   a. Seleccione el icono Agregar ***+** debajo de **Usuarios**.
   b. Seleccione el nombre de usuario requerido y seleccione un rol en el menú desplegable para **Rol**.
   c. Haga clic en **Guardar**.

Para obtener más información, consulte las siguientes páginas:

- "Roles personalizados para la administración de ONTAP"o"Definir roles personalizados"
- "Trabajar con roles y usuarios"

Autenticar conexiones con CHAP bidireccional

Trident puede autenticar sesiones iSCSI con CHAP bidireccional para la `ontap-san` y `ontap-san-economy` conductores. Esto requiere habilitar el `useCHAP` opción en la definición de tu backend. Cuando se configura para `true` Trident configura la seguridad del iniciador predeterminado de la SVM en CHAP bidireccional y establece el nombre de usuario y los secretos desde el archivo de backend. NetApp recomienda utilizar CHAP bidireccional para autenticar las conexiones. Consulte la siguiente configuración de ejemplo:

```
---
version: 1
storageDriverName: ontap-san
backendName: ontap_san_chap
managementLIF: 192.168.0.135
svm: ontap_iscsi_svm
useCHAP: true
username: vsadmin
password: password
chapInitiatorSecret: cl9qxIm36DKyawxy
chapTargetInitiatorSecret: rqxigXgkesIpwxyz
chapTargetUsername: iJF4heBRT0TCwxyz
chapUsername: uh2aNCLSd6cNwxyz
```

El `useCHAP` El parámetro es una opción booleana que solo se puede configurar una vez. Está configurado como falso por defecto. Una vez que lo hayas configurado como verdadero, no podrás configurarlo como falso.

Además de `useCHAP=true` , el `chapInitiatorSecret` , `chapTargetInitiatorSecret` , `chapTargetUsername` , y `chapUsername` Los campos deben incluirse en la definición del backend. Los secretos se pueden cambiar después de crear un backend ejecutando `tridentctl update` .

Cómo funciona

Mediante la configuración `useCHAP` Para que sea verdadero, el administrador de almacenamiento le indica a Trident que configure CHAP en el backend de almacenamiento. Esto incluye lo siguiente:

- Configuración de CHAP en la SVM:
  - Si el tipo de seguridad del iniciador predeterminado de la SVM es ninguno (establecido por defecto) **y** no hay LUN preexistentes en el volumen, Trident establecerá el tipo de seguridad predeterminado en `CHAP` y proceda a configurar el iniciador CHAP y el nombre de usuario y las claves de destino.
  - Si la SVM contiene LUN, Trident no habilitará CHAP en la SVM. Esto garantiza que el acceso a las LUN que ya están presentes en la SVM no se vea restringido.
- Configurar el nombre de usuario y las claves secretas del iniciador y del destino CHAP; estas opciones deben especificarse en la configuración del backend (como se muestra arriba).

Una vez creado el backend, Trident crea un correspondiente `tridentbackend` CRD y almacena los secretos CHAP y los nombres de usuario como secretos de Kubernetes. Todos los PV que Trident cree en este backend se montarán y conectarán a través de CHAP.

Rotar credenciales y actualizar backends

Puede actualizar las credenciales CHAP actualizando los parámetros CHAP en el archivo `backend.json` archivo. Esto requerirá actualizar los secretos CHAP y utilizar el `tridentctl update` orden para reflejar estos cambios.

Al actualizar los secretos CHAP para un backend, debe usar `tridentctl` para actualizar el backend. No actualice las credenciales en el clúster de almacenamiento mediante ONTAP CLI o ONTAP System Manager, ya que Trident no podrá detectar estos cambios.

```
cat backend-san.json
{
    "version": 1,
    "storageDriverName": "ontap-san",
    "backendName": "ontap_san_chap",
    "managementLIF": "192.168.0.135",
    "svm": "ontap_iscsi_svm",
    "useCHAP": true,
    "username": "vsadmin",
    "password": "password",
    "chapInitiatorSecret": "cl9qxUpDaTeD",
    "chapTargetInitiatorSecret": "rqxigXgkeUpDaTeD",
    "chapTargetUsername": "iJF4heBRT0TCwxyz",
    "chapUsername": "uh2aNCLSd6cNwxyz",
}

./tridentctl update backend ontap_san_chap -f backend-san.json -n trident
+----------------+----------------+--------------------------------------
+--------+---------+
|   NAME         | STORAGE DRIVER |                 UUID                 |
STATE   | VOLUMES |
+----------------+----------------+--------------------------------------
+--------+---------+
| ontap_san_chap | ontap-san      | aa458f3b-ad2d-4378-8a33-1a472ffbeb5c |
online |       7 |
+----------------+----------------+--------------------------------------
+--------+---------+
```

Las conexiones existentes no se verán afectadas; seguirán activas si Trident actualiza las credenciales en la SVM. Las nuevas conexiones utilizan las credenciales actualizadas y las conexiones existentes permanecen activas. Desconectar y volver a conectar los PV antiguos hará que utilicen las credenciales actualizadas.

Opciones y ejemplos de configuración de SAN de ONTAP

Aprenda cómo crear y utilizar controladores ONTAP SAN con su instalación de Trident . Esta sección proporciona ejemplos de configuración de backend y detalles para mapear backends a StorageClasses.

"Sistemas ASA r2"Se diferencia de otros sistemas ONTAP (ASA, AFF y FAS) en la implementación de su capa de almacenamiento. Estas variaciones afectan al uso de ciertos parámetros, tal como se indica. "Obtenga más información sobre las diferencias entre los sistemas ASA r2 y otros sistemas ONTAP .".

Sólo el `ontap-san` El controlador (con protocolos iSCSI y NVMe/TCP) es compatible con los sistemas ASA r2.

En la configuración del backend de Trident , no es necesario especificar que su sistema sea ASA r2. Cuando seleccionas `ontap-san` como el `storageDriverName` Trident detecta automáticamente el ASA r2 o el sistema ONTAP tradicional. Algunos parámetros de configuración del backend no son aplicables a los sistemas ASA r2, como se indica en la tabla siguiente.

Opciones de configuración del backend

Consulte la siguiente tabla para ver las opciones de configuración del backend:

Parámetro	Descripción	Por defecto
`version`		Siempre 1
`storageDriverName`	Nombre del controlador de almacenamiento	`ontap-san`o `ontap-san-economy`
`backendName`	Nombre personalizado o el backend de almacenamiento	Nombre del controlador + "_" + dataLIF
`managementLIF`	Dirección IP de un clúster o LIF de gestión de SVM. Se puede especificar un nombre de dominio completo (FQDN). Se puede configurar para usar direcciones IPv6 si Trident se instaló usando la bandera IPv6. Las direcciones IPv6 deben definirse entre corchetes, como por ejemplo: `[28e8:d9fb:a825:b7bf:69a8:d02f:9e7b:3555]` . Para una transición fluida a MetroCluster , consulteEjemplo de MetroCluster . Si está utilizando credenciales "vsadmin", `managementLIF` debe ser la del SVM; si se utilizan credenciales de "administrador", `managementLIF` debe ser el del grupo.	"10.0.0.1", "[2001:1234:abcd::fefe]"
`dataLIF`	Dirección IP del protocolo LIF. Se puede configurar para usar direcciones IPv6 si Trident se instaló usando la bandera IPv6. Las direcciones IPv6 deben definirse entre corchetes, como por ejemplo: `[28e8:d9fb:a825:b7bf:69a8:d02f:9e7b:3555]` . **No especificar para iSCSI.** Trident utiliza"Mapa selectivo de LUN de ONTAP" para descubrir los LIF iSCSI necesarios para establecer una sesión de múltiples rutas. Se genera una advertencia si `dataLIF` está definido explícitamente. **Omitir para Metrocluster.** Ver elEjemplo de MetroCluster .	Derivado por la SVM

Parámetro	Descripción	Por defecto
`svm`	Máquina virtual de almacenamiento a utilizar **Omitir para Metrocluster.** Ver elEjemplo de MetroCluster .	Derivado si se trata de una SVM `managementLIF` se especifica
`useCHAP`	Utilice CHAP para autenticar iSCSI para controladores SAN ONTAP [Booleano]. Empezar a `true` para que Trident configure y utilice CHAP bidireccional como autenticación predeterminada para el SVM proporcionado en el backend. Referirse a "Prepárese para configurar el backend con los controladores SAN de ONTAP." Para más detalles. **No compatible con FCP ni NVMe/TCP.**	`false`
`chapInitiatorSecret`	Secreto del iniciador de CHAP. Requerido si `useCHAP=true`	""
`labels`	Conjunto de etiquetas arbitrarias con formato JSON para aplicar a los volúmenes	""
`chapTargetInitiatorSecret`	Secreto del iniciador del objetivo CHAP. Requerido si `useCHAP=true`	""
`chapUsername`	Nombre de usuario entrante. Requerido si `useCHAP=true`	""
`chapTargetUsername`	Nombre de usuario objetivo. Requerido si `useCHAP=true`	""
`clientCertificate`	Valor codificado en Base64 del certificado del cliente. Se utiliza para la autenticación basada en certificados.	""
`clientPrivateKey`	Valor codificado en Base64 de la clave privada del cliente. Se utiliza para la autenticación basada en certificados.	""
`trustedCACertificate`	Valor codificado en Base64 del certificado de CA de confianza. Opcional. Se utiliza para la autenticación basada en certificados.	""
`username`	Nombre de usuario necesario para comunicarse con el clúster ONTAP . Se utiliza para la autenticación basada en credenciales. Para la autenticación de Active Directory, consulte "Autenticar Trident en un SVM backend mediante credenciales de Active Directory".	""
`password`	Contraseña necesaria para comunicarse con el clúster ONTAP . Se utiliza para la autenticación basada en credenciales. Para la autenticación de Active Directory, consulte "Autenticar Trident en un SVM backend mediante credenciales de Active Directory".	""
`svm`	máquina virtual de almacenamiento a utilizar	Derivado si se trata de una SVM `managementLIF` se especifica

Parámetro	Descripción	Por defecto
`storagePrefix`	Prefijo utilizado al aprovisionar nuevos volúmenes en la SVM. No se puede modificar posteriormente. Para actualizar este parámetro, deberá crear un nuevo backend.	`trident`
`aggregate`	Agregado para aprovisionamiento (opcional; si se establece, debe asignarse a la SVM). Para el `ontap-nas-flexgroup` conductor, esta opción se ignora. Si no se asigna, cualquiera de los agregados disponibles se puede utilizar para aprovisionar un volumen FlexGroup . Cuando se actualiza el agregado en SVM, se actualiza automáticamente en Trident mediante sondeos a SVM sin necesidad de reiniciar el controlador Trident . Cuando se ha configurado un agregado específico en Trident para aprovisionar volúmenes, si el agregado se renombra o se mueve fuera del SVM, el backend pasará a un estado de error en Trident mientras consulta el agregado del SVM. Debe cambiar el agregado por uno que esté presente en la SVM o eliminarlo por completo para volver a poner en línea el backend. **No especificar para sistemas ASA r2**.	""
`limitAggregateUsage`	Fallará el aprovisionamiento si el uso supera este porcentaje. Si utiliza un backend de Amazon FSx for NetApp ONTAP , no especifique `limitAggregateUsage` . El proporcionado `fsxadmin` y `vsadmin` No contienen los permisos necesarios para recuperar el uso agregado y limitarlo mediante Trident. **No especificar para sistemas ASA r2**.	" (no se aplica por defecto)
`limitVolumeSize`	Fallará el aprovisionamiento si el tamaño de volumen solicitado supera este valor. También restringe el tamaño máximo de los volúmenes que administra para las LUN.	" (no se aplica por defecto)
`lunsPerFlexvol`	Número máximo de LUN por Flexvol, debe estar en el rango [50, 200]	`100`
`debugTraceFlags`	Indicadores de depuración para usar al solucionar problemas. Ejemplo: {"api":false, "method":true} No lo utilice a menos que esté solucionando problemas y necesite un volcado de registro detallado.	`null`

Parámetro	Descripción	Por defecto
`useREST`	Parámetro booleano para utilizar las API REST de ONTAP . `useREST`Cuando se configura para `true` Trident utiliza las API REST de ONTAP para comunicarse con el backend; cuando se configura en `false` Trident utiliza llamadas ONTAPI (ZAPI) para comunicarse con el backend. Esta función requiere ONTAP 9.11.1 y versiones posteriores. Además, el rol de inicio de sesión de ONTAP utilizado debe tener acceso a `ontapi` solicitud. Esto se satisface mediante lo predefinido. `vsadmin` y `cluster-admin` roles. A partir de la versión Trident 24.06 y ONTAP 9.15.1 o posterior, `useREST` está configurado para `true` por defecto; cambiar `useREST` a `false` para utilizar llamadas ONTAPI (ZAPI). `useREST`está totalmente cualificado para NVMe/TCP. NVMe solo es compatible con las API REST de ONTAP y no con ONTAPI (ZAPI). **Si se especifica, siempre se establecerá en `true` para sistemas ASA r2**.	`true`para ONTAP 9.15.1 o posterior, de lo contrario `false`.
`sanType`	Utilice para seleccionar `iscsi` para iSCSI, `nvme` para NVMe/TCP o `fcp` para SCSI sobre Fibre Channel (FC).	`iscsi`si está en blanco

Parámetro	Descripción	Por defecto
`formatOptions`	Usar `formatOptions` para especificar los argumentos de la línea de comandos para el `mkfs` comando que se aplicará siempre que se formatee un volumen. Esto le permite formatear el volumen según sus preferencias. Asegúrese de especificar las opciones de formato de forma similar a las opciones del comando mkfs, excluyendo la ruta del dispositivo. Ejemplo: "-E no descartar" **Compatible con `ontap-san` y `ontap-san-economy` controladores con protocolo iSCSI. Además, se admite para sistemas ASA r2 cuando se utilizan los protocolos iSCSI y NVMe/TCP.**	
`limitVolumePoolSize`	Tamaño máximo de FlexVol solicitable al usar LUN en el backend ontap-san-economy.	" (no se aplica por defecto)
`denyNewVolumePools`	Restringe `ontap-san-economy` Los sistemas backend crean nuevos volúmenes FlexVol para contener sus LUN. Solo se utilizan Flexvols preexistentes para el aprovisionamiento de nuevos PV.	

Recomendaciones para el uso de formatOptions

Trident recomienda la siguiente opción para agilizar el proceso de formateo:

-E no descartar:

- Conservar, no intentar descartar bloques en el momento de mkfs (descartar bloques inicialmente es útil en dispositivos de estado sólido y almacenamiento disperso/de aprovisionamiento ligero). Esto reemplaza la opción obsoleta "-K" y es aplicable a todos los sistemas de archivos (xfs, ext3 y ext4).

Autenticar Trident en un SVM backend mediante credenciales de Active Directory

Puede configurar Trident para autenticarse en un SVM de backend usando credenciales de Active Directory (AD). Antes de que una cuenta de AD pueda acceder a la SVM, debe configurar el acceso del controlador de dominio de AD al clúster o SVM. Para la administración de un clúster con una cuenta de AD, debe crear un túnel de dominio. Referirse a "Configurar el acceso al controlador de dominio de Active Directory en ONTAP" Para más detalles.

pasos

1. Configurar los ajustes del Sistema de nombres de dominio (DNS) para un SVM de backend:

```
vserver services dns create -vserver <svm_name> -dns-servers
<dns_server_ip1>,<dns_server_ip2>
```

2. Ejecute el siguiente comando para crear una cuenta de computadora para la SVM en Active Directory:

```
vserver active-directory create -vserver DataSVM -account-name ADSERVER1
-domain demo.netapp.com
```

3. Utilice este comando para crear un usuario o grupo de AD para administrar el clúster o SVM

```
security login create -vserver <svm_name> -user-or-group-name
<ad_user_or_group> -application <application> -authentication-method domain
-role vsadmin
```

4. En el archivo de configuración del backend de Trident , configure el `username` y `password` parámetros al nombre de usuario o grupo de AD y la contraseña, respectivamente.

Opciones de configuración de backend para el aprovisionamiento de volúmenes

Puedes controlar el aprovisionamiento predeterminado utilizando estas opciones en el `defaults` sección de la configuración. Para ver un ejemplo, consulte los ejemplos de configuración a continuación.

Parámetro	Descripción	Por defecto
`spaceAllocation`	Asignación de espacio para LUN	"verdadero" **Si se especifica, establecer en `true` para sistemas ASA r2**.
`spaceReserve`	Modo de reserva de espacio: "ninguno" (delgado) o "volumen" (grueso). **Empezar a `none` para sistemas ASA r2**.	"ninguno"
`snapshotPolicy`	Política de instantáneas a utilizar. **Empezar a `none` para sistemas ASA r2**.	"ninguno"
`qosPolicy`	Grupo de políticas QoS que se asignará a los volúmenes creados. Elija una de las opciones qosPolicy o adaptiveQosPolicy por grupo de almacenamiento/backend. El uso de grupos de políticas QoS con Trident requiere ONTAP 9.8 o posterior. Debe utilizar un grupo de políticas QoS no compartido y asegurarse de que el grupo de políticas se aplique a cada componente individualmente. Un grupo de políticas QoS compartidas impone un límite máximo al rendimiento total de todas las cargas de trabajo.	""
`adaptiveQosPolicy`	Grupo de políticas QoS adaptativas para asignar a los volúmenes creados. Elija una de las opciones qosPolicy o adaptiveQosPolicy por grupo de almacenamiento/backend.	""
`snapshotReserve`	Porcentaje de volumen reservado para instantáneas. **No especificar para sistemas ASA r2**.	"0" si `snapshotPolicy` es "ninguno", de lo contrario ""
`splitOnClone`	Separar un clon de su progenitor al crearlo	"FALSO"
`encryption`	Habilite el cifrado de volumen de NetApp (NVE) en el nuevo volumen; el valor predeterminado es `false` . Para utilizar esta opción, NVE debe estar licenciado y habilitado en el clúster. Si NAE está habilitado en el backend, cualquier volumen aprovisionado en Trident tendrá NAE habilitado. Para obtener más información, consulte:"Cómo funciona Trident con NVE y NAE" .	"falso" **Si se especifica, establecer en `true` para sistemas ASA r2**.

Parámetro	Descripción	Por defecto
`luksEncryption`	Habilitar el cifrado LUKS. Referirse a"Utilice la configuración de clave unificada de Linux (LUKS)." .	" **Configurado a `false` para sistemas ASA r2**.
`tieringPolicy`	Política de jerarquización para usar "ninguna" **No especificar para sistemas ASA r2** .	
`nameTemplate`	Plantilla para crear nombres de volumen personalizados.	""

Ejemplos de aprovisionamiento por volumen

Aquí tenéis un ejemplo con valores predeterminados definidos:

```
---
version: 1
storageDriverName: ontap-san
managementLIF: 10.0.0.1
svm: trident_svm
username: admin
password: <password>
labels:
  k8scluster: dev2
  backend: dev2-sanbackend
storagePrefix: alternate-trident
debugTraceFlags:
  api: false
  method: true
defaults:
  spaceReserve: volume
  qosPolicy: standard
  spaceAllocation: 'false'
  snapshotPolicy: default
  snapshotReserve: '10'
```

> Para todos los volúmenes creados utilizando el `ontap-san` El controlador Trident agrega un 10 por ciento de capacidad adicional al FlexVol para dar cabida a los metadatos de LUN. La LUN se aprovisionará con el tamaño exacto que el usuario solicite en el PVC. Trident añade un 10 por ciento al FlexVol (se muestra como Tamaño disponible en ONTAP). Los usuarios ahora obtendrán la cantidad de capacidad utilizable que solicitaron. Este cambio también evita que las LUN se conviertan en de solo lectura a menos que se utilice todo el espacio disponible. Esto no se aplica a ontap-san-economy.

Para backends que definen `snapshotReserve` Trident calcula el tamaño de los volúmenes de la siguiente manera:

```
Total volume size = [(PVC requested size) / (1 - (snapshotReserve
percentage) / 100)] * 1.1
```

El 1.1 es el 10 por ciento adicional que Trident agrega al FlexVol para acomodar los metadatos del LUN. Para `snapshotReserve` = 5%, y solicitud de PVC = 5 GiB, el tamaño total del volumen es 5,79 GiB y el tamaño disponible es 5,5 GiB. El `volume show` El comando debería mostrar resultados similares a este ejemplo:

Actualmente, la única forma de utilizar el nuevo cálculo para un volumen existente es redimensionándolo.

Ejemplos de configuración mínima

Los siguientes ejemplos muestran configuraciones básicas que dejan la mayoría de los parámetros con sus valores predeterminados. Esta es la forma más sencilla de definir un backend.

> Si utiliza Amazon FSx en NetApp ONTAP con Trident, NetApp recomienda que especifique nombres DNS para las LIF en lugar de direcciones IP.

Ejemplo de SAN de ONTAP

Esta es una configuración básica que utiliza `ontap-san` conductor.

```
---
version: 1
storageDriverName: ontap-san
managementLIF: 10.0.0.1
svm: svm_iscsi
labels:
  k8scluster: test-cluster-1
  backend: testcluster1-sanbackend
username: vsadmin
password: <password>
```

Ejemplo de MetroCluster

Puede configurar el backend para evitar tener que actualizar manualmente la definición del backend después del cambio de estado y el cambio de estado durante"replicación y recuperación de SVM" .

Para una conmutación y recuperación sin interrupciones, especifique el SVM utilizando `managementLIF` y omitir el `svm` parámetros. Por ejemplo:

```
version: 1
storageDriverName: ontap-san
managementLIF: 192.168.1.66
username: vsadmin
password: password
```

Ejemplo económico de ONTAP SAN

```
version: 1
storageDriverName: ontap-san-economy
managementLIF: 10.0.0.1
svm: svm_iscsi_eco
username: vsadmin
password: <password>
```

Ejemplo de autenticación basada en certificados

En este ejemplo de configuración básica `clientCertificate` , `clientPrivateKey` , y `trustedCACertificate` (opcional, si se utiliza una CA de confianza) se rellenan en `backend.json` y tome los valores codificados en base64 del certificado del cliente, la clave privada y el certificado de CA de confianza, respectivamente.

```
---
version: 1
storageDriverName: ontap-san
backendName: DefaultSANBackend
managementLIF: 10.0.0.1
svm: svm_iscsi
useCHAP: true
chapInitiatorSecret: cl9qxIm36DKyawxy
chapTargetInitiatorSecret: rqxigXgkesIpwxyz
chapTargetUsername: iJF4heBRT0TCwxyz
chapUsername: uh2aNCLSd6cNwxyz
clientCertificate: ZXR0ZXJwYXB...ICMgJ3BhcGVyc2
clientPrivateKey: vciwKIyAgZG...0cnksIGRlc2NyaX
trustedCACertificate: zcyBbaG...b3Igb3duIGNsYXNz
```

Ejemplos de CHAP bidireccionales

Estos ejemplos crean un backend con `useCHAP` empezar a `true` .

Ejemplo de ONTAP SAN CHAP

```
---
version: 1
storageDriverName: ontap-san
managementLIF: 10.0.0.1
svm: svm_iscsi
labels:
  k8scluster: test-cluster-1
  backend: testcluster1-sanbackend
useCHAP: true
chapInitiatorSecret: cl9qxIm36DKyawxy
chapTargetInitiatorSecret: rqxigXgkesIpwxyz
chapTargetUsername: iJF4heBRT0TCwxyz
chapUsername: uh2aNCLSd6cNwxyz
username: vsadmin
password: <password>
```

Ejemplo de economía ONTAP SAN CHAP

```
---
version: 1
storageDriverName: ontap-san-economy
managementLIF: 10.0.0.1
svm: svm_iscsi_eco
useCHAP: true
chapInitiatorSecret: cl9qxIm36DKyawxy
chapTargetInitiatorSecret: rqxigXgkesIpwxyz
chapTargetUsername: iJF4heBRT0TCwxyz
chapUsername: uh2aNCLSd6cNwxyz
username: vsadmin
password: <password>
```

Ejemplo de NVMe/TCP

Debe tener una SVM configurada con NVMe en su backend ONTAP . Esta es una configuración básica de backend para NVMe/TCP.

```
---
version: 1
backendName: NVMeBackend
storageDriverName: ontap-san
managementLIF: 10.0.0.1
svm: svm_nvme
username: vsadmin
password: password
sanType: nvme
useREST: true
```

Ejemplo de SCSI sobre FC (FCP)

Debe tener un SVM configurado con FC en su backend ONTAP . Esta es una configuración básica de backend para FC.

```
---
version: 1
backendName: fcp-backend
storageDriverName: ontap-san
managementLIF: 10.0.0.1
svm: svm_fc
username: vsadmin
password: password
sanType: fcp
useREST: true
```

Ejemplo de configuración de backend con plantilla de nombre

```
---
version: 1
storageDriverName: ontap-san
backendName: ontap-san-backend
managementLIF: <ip address>
svm: svm0
username: <admin>
password: <password>
defaults:
  nameTemplate:
"{{.volume.Name}}_{{.labels.cluster}}_{{.volume.Namespace}}_{{.vo\
    lume.RequestName}}"
labels:
  cluster: ClusterA
  PVC: "{{.volume.Namespace}}_{{.volume.RequestName}}"
```

Ejemplo de opciones de formato para el controlador ontap-san-economy

```
---
version: 1
storageDriverName: ontap-san-economy
managementLIF: ""
svm: svm1
username: ""
password: "!"
storagePrefix: whelk_
debugTraceFlags:
  method: true
  api: true
defaults:
  formatOptions: -E nodiscard
```

Ejemplos de backends con pools virtuales

En estos archivos de definición de backend de ejemplo, se establecen valores predeterminados específicos para todos los grupos de almacenamiento, tales como: `spaceReserve` en ninguno, `spaceAllocation` en falso, y `encryption` en falso. Los grupos virtuales se definen en la sección de almacenamiento.

Trident establece las etiquetas de aprovisionamiento en el campo "Comentarios". Los comentarios se configuran en el FlexVol volume. Trident copia todas las etiquetas presentes en un grupo virtual al volumen de almacenamiento durante el aprovisionamiento. Para mayor comodidad, los administradores de almacenamiento pueden definir etiquetas por grupo virtual y agrupar volúmenes por etiqueta.

En estos ejemplos, algunos de los grupos de almacenamiento establecen sus propias configuraciones. `spaceReserve` , `spaceAllocation` , y `encryption` valores, y algunos pools anulan los valores predeterminados.

Ejemplo de SAN de ONTAP

```yaml
---
version: 1
storageDriverName: ontap-san
managementLIF: 10.0.0.1
svm: svm_iscsi
useCHAP: true
chapInitiatorSecret: cl9qxIm36DKyawxy
chapTargetInitiatorSecret: rqxigXgkesIpwxyz
chapTargetUsername: iJF4heBRT0TCwxyz
chapUsername: uh2aNCLSd6cNwxyz
username: vsadmin
password: <password>
defaults:
  spaceAllocation: "false"
  encryption: "false"
  qosPolicy: standard
labels:
  store: san_store
  kubernetes-cluster: prod-cluster-1
region: us_east_1
storage:
  - labels:
      protection: gold
      creditpoints: "40000"
    zone: us_east_1a
    defaults:
      spaceAllocation: "true"
      encryption: "true"
      adaptiveQosPolicy: adaptive-extreme
  - labels:
      protection: silver
      creditpoints: "20000"
    zone: us_east_1b
    defaults:
      spaceAllocation: "false"
      encryption: "true"
      qosPolicy: premium
  - labels:
      protection: bronze
      creditpoints: "5000"
    zone: us_east_1c
    defaults:
      spaceAllocation: "true"
      encryption: "false"
```

Ejemplo económico de ONTAP SAN

```
---
version: 1
storageDriverName: ontap-san-economy
managementLIF: 10.0.0.1
svm: svm_iscsi_eco
useCHAP: true
chapInitiatorSecret: cl9qxIm36DKyawxy
chapTargetInitiatorSecret: rqxigXgkesIpwxyz
chapTargetUsername: iJF4heBRT0TCwxyz
chapUsername: uh2aNCLSd6cNwxyz
username: vsadmin
password: <password>
defaults:
  spaceAllocation: "false"
  encryption: "false"
labels:
  store: san_economy_store
region: us_east_1
storage:
  - labels:
      app: oracledb
      cost: "30"
    zone: us_east_1a
    defaults:
      spaceAllocation: "true"
      encryption: "true"
  - labels:
      app: postgresdb
      cost: "20"
    zone: us_east_1b
    defaults:
      spaceAllocation: "false"
      encryption: "true"
  - labels:
      app: mysqldb
      cost: "10"
    zone: us_east_1c
    defaults:
      spaceAllocation: "true"
      encryption: "false"
  - labels:
      department: legal
      creditpoints: "5000"
    zone: us_east_1c
```

```
    defaults:
      spaceAllocation: "true"
      encryption: "false"
```

Ejemplo de NVMe/TCP

```
---
version: 1
storageDriverName: ontap-san
sanType: nvme
managementLIF: 10.0.0.1
svm: nvme_svm
username: vsadmin
password: <password>
useREST: true
defaults:
  spaceAllocation: "false"
  encryption: "true"
storage:
  - labels:
      app: testApp
      cost: "20"
    defaults:
      spaceAllocation: "false"
      encryption: "false"
```

Asignar backends a StorageClasses

Las siguientes definiciones de StorageClass hacen referencia aEjemplos de backends con pools virtuales . Utilizando el `parameters.selector` En cada campo, cada StorageClass especifica qué grupos virtuales se pueden usar para alojar un volumen. El volumen tendrá los aspectos definidos en el pool virtual elegido.

- El `protection-gold` StorageClass se asignará al primer grupo virtual en el `ontap-san` backend. Esta es la única piscina que ofrece protección de nivel oro.

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: protection-gold
provisioner: csi.trident.netapp.io
parameters:
  selector: "protection=gold"
  fsType: "ext4"
```

- El `protection-not-gold` StorageClass se asignará al segundo y tercer grupo virtual en `ontap-san` backend. Estas son las únicas piscinas que ofrecen un nivel de protección distinto al oro.

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: protection-not-gold
provisioner: csi.trident.netapp.io
parameters:
  selector: "protection!=gold"
  fsType: "ext4"
```

- El `app-mysqldb` StorageClass se asignará al tercer grupo virtual en `ontap-san-economy` backend. Este es el único pool que ofrece configuración de pool de almacenamiento para aplicaciones de tipo mysqldb.

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: app-mysqldb
provisioner: csi.trident.netapp.io
parameters:
  selector: "app=mysqldb"
  fsType: "ext4"
```

- El `protection-silver-creditpoints-20k` StorageClass se asignará al segundo grupo virtual en `ontap-san` backend. Este es el único fondo que ofrece protección de nivel plata y 20000 puntos de crédito.

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: protection-silver-creditpoints-20k
provisioner: csi.trident.netapp.io
parameters:
  selector: "protection=silver; creditpoints=20000"
  fsType: "ext4"
```

- El `creditpoints-5k` StorageClass se asignará al tercer grupo virtual en `ontap-san` backend y el cuarto grupo virtual en el `ontap-san-economy` backend. Estas son las únicas ofertas de pool con 5000 puntos de crédito.

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: creditpoints-5k
provisioner: csi.trident.netapp.io
parameters:
  selector: "creditpoints=5000"
  fsType: "ext4"
```

- El `my-test-app-sc` StorageClass se asignará a `testAPP` piscina virtual en la `ontap-san` conductor con `sanType: nvme` . Esta es la única oferta de piscina `testApp` .

```
---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: my-test-app-sc
provisioner: csi.trident.netapp.io
parameters:
  selector: "app=testApp"
  fsType: "ext4"
```

Trident decidirá qué grupo virtual se selecciona y garantiza que se cumplan los requisitos de almacenamiento.

Controladores NAS ONTAP

Descripción general del controlador NAS de ONTAP

Aprenda a configurar un backend de ONTAP con ONTAP y los controladores NAS de Cloud Volumes ONTAP .

Detalles del controlador NAS de ONTAP

Trident proporciona los siguientes controladores de almacenamiento NAS para comunicarse con el clúster ONTAP . Los modos de acceso compatibles son: *ReadWriteOnce* (RWO), *ReadOnlyMany* (ROX), *ReadWriteMany* (RWX), *ReadWriteOncePod* (RWOP).

Conductor	Protocolo	modo de volumen	Modos de acceso compatibles	Sistemas de archivos compatibles
`ontap-nas`	NFS SMB	Sistema de archivos	RWO, ROX, RWX, RWOP	"", `nfs` , `smb`
`ontap-nas-economy`	NFS SMB	Sistema de archivos	RWO, ROX, RWX, RWOP	"", `nfs` , `smb`

Conductor	Protocolo	modo de volumen	Modos de acceso compatibles	Sistemas de archivos compatibles
`ontap-nas-flexgroup`	NFS SMB	Sistema de archivos	RWO, ROX, RWX, RWOP	"", `nfs` , `smb`

- Usar `ontap-san-economy` solo si se espera que el recuento de uso de volumen persistente sea superior a"límites de volumen ONTAP compatibles" .
- Usar `ontap-nas-economy` solo si se espera que el recuento de uso de volumen persistente sea superior a"límites de volumen ONTAP compatibles" y el `ontap-san-economy` El controlador no se puede utilizar.
- No usar `ontap-nas-economy` Si prevé la necesidad de protección de datos, recuperación ante desastres o movilidad.
- NetApp no recomienda usar el crecimiento automático de Flexvol en todos los controladores ONTAP , excepto en ontap-san. Como solución alternativa, Trident admite el uso de reserva de instantáneas y escala los volúmenes Flexvol en consecuencia.

Permisos de usuario

Trident espera ejecutarse como administrador de ONTAP o SVM, normalmente utilizando el `admin` usuario del clúster o un `vsadmin` Usuario de SVM, o un usuario con un nombre diferente que tenga la misma función.

Para implementaciones de Amazon FSx for NetApp ONTAP , Trident requiere ejecutarse como administrador de ONTAP o SVM, utilizando el clúster. `fsxadmin` usuario o un `vsadmin` Usuario de SVM, o un usuario con un nombre diferente que tenga la misma función. El `fsxadmin` El usuario es un reemplazo limitado para el usuario administrador del clúster.

Si utiliza el `limitAggregateUsage` Se requieren permisos de administrador de clúster para este parámetro. Al utilizar Amazon FSx for NetApp ONTAP con Trident, `limitAggregateUsage` El parámetro no funcionará con el `vsadmin` y `fsxadmin` cuentas de usuario. La operación de configuración fallará si especifica este parámetro.

Si bien es posible crear un rol más restrictivo dentro de ONTAP que pueda usar un controlador Trident , no lo recomendamos. La mayoría de las nuevas versiones de Trident utilizarán API adicionales que habría que tener en cuenta, lo que dificultaría las actualizaciones y las haría propensas a errores.

Prepárese para configurar un backend con controladores NAS ONTAP .

Comprenda los requisitos, las opciones de autenticación y las políticas de exportación para configurar un backend ONTAP con controladores ONTAP NAS.

Requisitos

- Para todos los backends de ONTAP , Trident requiere que se asigne al menos un agregado al SVM.
- Puedes ejecutar más de un controlador y crear clases de almacenamiento que apunten a uno u otro. Por ejemplo, podrías configurar una clase Gold que utilice la `ontap-nas` conductor y una clase Bronce que utiliza el `ontap-nas-economy` uno.
- Todos tus nodos de trabajo de Kubernetes deben tener instaladas las herramientas NFS apropiadas. Referirse a"aquí" Para más detalles.

- Trident solo admite volúmenes SMB montados en pods que se ejecutan en nodos Windows. Referirse a Preparar el aprovisionamiento de volúmenes SMB Para más detalles.

Autenticar el backend de ONTAP

Trident ofrece dos modos de autenticación de un backend ONTAP .

- Basado en credenciales: Este modo requiere permisos suficientes para el backend de ONTAP . Se recomienda utilizar una cuenta asociada a un rol de inicio de sesión de seguridad predefinido, como por ejemplo: `admin` o `vsadmin` para garantizar la máxima compatibilidad con las versiones de ONTAP .
- Basado en certificados: Este modo requiere un certificado instalado en el backend para que Trident se comunique con un clúster ONTAP . Aquí, la definición del backend debe contener valores codificados en Base64 del certificado del cliente, la clave y el certificado de CA de confianza si se utiliza (recomendado).

Puedes actualizar los sistemas backend existentes para alternar entre métodos basados en credenciales y métodos basados en certificados. Sin embargo, solo se admite un método de autenticación a la vez. Para cambiar a un método de autenticación diferente, debe eliminar el método existente de la configuración del backend.

> Si intenta proporcionar **tanto credenciales como certificados**, la creación del backend fallará con un error que indica que se proporcionó más de un método de autenticación en el archivo de configuración.

Habilitar la autenticación basada en credenciales

Trident requiere las credenciales de un administrador con ámbito SVM/ámbito de clúster para comunicarse con el backend de ONTAP . Se recomienda utilizar roles estándar predefinidos, tales como: `admin` o `vsadmin` . Esto garantiza la compatibilidad con versiones futuras de ONTAP que podrían exponer API de funciones para ser utilizadas por futuras versiones de Trident . Se puede crear y usar un rol de inicio de sesión de seguridad personalizado con Trident, pero no se recomienda.

Un ejemplo de definición de backend se vería así:

YAML

```yaml
---
version: 1
backendName: ExampleBackend
storageDriverName: ontap-nas
managementLIF: 10.0.0.1
dataLIF: 10.0.0.2
svm: svm_nfs
credentials:
  name: secret-backend-creds
```

JSON

```json
{
  "version": 1,
  "backendName": "ExampleBackend",
  "storageDriverName": "ontap-nas",
  "managementLIF": "10.0.0.1",
  "dataLIF": "10.0.0.2",
  "svm": "svm_nfs",
  "credentials": {
        "name": "secret-backend-creds"
    }
}
```

Tenga en cuenta que la definición del backend es el único lugar donde las credenciales se almacenan en texto plano. Una vez creado el backend, los nombres de usuario y las contraseñas se codifican con Base64 y se almacenan como secretos de Kubernetes. La creación/actualización de un backend es el único paso que requiere conocer las credenciales. Por lo tanto, se trata de una operación exclusiva para administradores, que debe ser realizada por el administrador de Kubernetes/almacenamiento.

Habilitar la autenticación basada en certificados

Los backends nuevos y existentes pueden usar un certificado y comunicarse con el backend de ONTAP . Se requieren tres parámetros en la definición del backend.

- clientCertificate: Valor codificado en Base64 del certificado del cliente.
- clientPrivateKey: Valor codificado en Base64 de la clave privada asociada.
- trustedCACertificate: Valor codificado en Base64 del certificado de CA de confianza. Si se utiliza una CA de confianza, este parámetro debe proporcionarse. Esto puede ignorarse si no se utiliza ninguna CA de confianza.

Un flujo de trabajo típico comprende los siguientes pasos.

Pasos

1. Generar un certificado y una clave de cliente. Al generar, configure el Nombre Común (CN) con el usuario ONTAP con el que se autenticará.

```
openssl req -x509 -nodes -days 1095 -newkey rsa:2048 -keyout k8senv.key
-out k8senv.pem -subj "/C=US/ST=NC/L=RTP/O=NetApp/CN=vsadmin"
```

2. Agregar certificado de CA de confianza al clúster ONTAP . Es posible que esto ya lo gestione el administrador de almacenamiento. Ignorar si no se utiliza ninguna CA de confianza.

```
security certificate install -type server -cert-name <trusted-ca-cert-
name> -vserver <vserver-name>
ssl modify -vserver <vserver-name> -server-enabled true -client-enabled
true -common-name <common-name> -serial <SN-from-trusted-CA-cert> -ca
<cert-authority>
```

3. Instale el certificado y la clave del cliente (del paso 1) en el clúster ONTAP .

```
security certificate install -type client-ca -cert-name <certificate-
name> -vserver <vserver-name>
security ssl modify -vserver <vserver-name> -client-enabled true
```

4. Confirme que el rol de inicio de sesión de seguridad de ONTAP es compatible. `cert` Método de autenticación.

```
security login create -user-or-group-name vsadmin -application ontapi
-authentication-method cert -vserver <vserver-name>
security login create -user-or-group-name vsadmin -application http
-authentication-method cert -vserver <vserver-name>
```

5. Prueba de autenticación utilizando el certificado generado. Reemplace < ONTAP Management LIF> y <vserver name> con la IP de Management LIF y el nombre de SVM. Debe asegurarse de que la LIF tenga configurada su política de servicio para `default-data-management` .

```
curl -X POST -Lk https://<ONTAP-Management-
LIF>/servlets/netapp.servlets.admin.XMLrequest_filer --key k8senv.key
--cert ~/k8senv.pem -d '<?xml version="1.0" encoding="UTF-8"?><netapp
xmlns="http://www.netapp.com/filer/admin" version="1.21"
vfiler="<vserver-name>"><vserver-get></vserver-get></netapp>'
```

6. Codifique el certificado, la clave y el certificado de CA de confianza con Base64.

```
base64 -w 0 k8senv.pem >> cert_base64
base64 -w 0 k8senv.key >> key_base64
base64 -w 0 trustedca.pem >> trustedca_base64
```

7. Crea el backend utilizando los valores obtenidos en el paso anterior.

```
cat cert-backend-updated.json
{
"version": 1,
"storageDriverName": "ontap-nas",
"backendName": "NasBackend",
"managementLIF": "1.2.3.4",
"dataLIF": "1.2.3.8",
"svm": "vserver_test",
"clientCertificate": "Faaaakkkkeeee...Vaaalllluuuueeee",
"clientPrivateKey": "LS0tFaKE...0VaLuES0tLS0K",
"storagePrefix": "myPrefix_"
}

#Update backend with tridentctl
tridentctl update backend NasBackend -f cert-backend-updated.json -n
trident
+------------+----------------+--------------------------------------
+--------+---------+
|    NAME    | STORAGE DRIVER |                 UUID                 |
STATE  | VOLUMES |
+------------+----------------+--------------------------------------
+--------+---------+
| NasBackend | ontap-nas      | 98e19b74-aec7-4a3d-8dcf-128e5033b214 |
online |       9 |
+------------+----------------+--------------------------------------
+--------+---------+
```

Actualizar los métodos de autenticación o rotar las credenciales

Puedes actualizar un backend existente para usar un método de autenticación diferente o para rotar sus credenciales. Esto funciona en ambos sentidos: los sistemas de gestión de backends que utilizan nombre de usuario/contraseña pueden actualizarse para usar certificados; los sistemas de gestión de backends que utilizan certificados pueden actualizarse para basarse en nombre de usuario/contraseña. Para ello, debe eliminar el método de autenticación existente y agregar el nuevo método de autenticación. A continuación, utilice el archivo backend.json actualizado que contiene los parámetros necesarios para ejecutar `tridentctl update backend`.

```
cat cert-backend-updated.json
```

```
{
"version": 1,
"storageDriverName": "ontap-nas",
"backendName": "NasBackend",
"managementLIF": "1.2.3.4",
"dataLIF": "1.2.3.8",
"svm": "vserver_test",
"username": "vsadmin",
"password": "password",
"storagePrefix": "myPrefix_"
}
```

```
#Update backend with tridentctl
tridentctl update backend NasBackend -f cert-backend-updated.json -n
trident
+------------+----------------+--------------------------------------
+--------+---------+
|    NAME    | STORAGE DRIVER |                 UUID                 |
STATE  | VOLUMES |
+------------+----------------+--------------------------------------
+--------+---------+
| NasBackend | ontap-nas      | 98e19b74-aec7-4a3d-8dcf-128e5033b214 |
online |       9 |
+------------+----------------+--------------------------------------
+--------+---------+
```

Al rotar las contraseñas, el administrador de almacenamiento primero debe actualizar la contraseña del usuario en ONTAP. A continuación se realiza una actualización del servidor. Al rotar los certificados, se pueden agregar varios certificados al usuario. Posteriormente, se actualiza el sistema backend para utilizar el nuevo certificado, tras lo cual se puede eliminar el certificado antiguo del clúster ONTAP .

La actualización de un backend no interrumpe el acceso a los volúmenes que ya se han creado, ni afecta a las conexiones de volumen realizadas posteriormente. Una actualización exitosa del backend indica que Trident puede comunicarse con el backend de ONTAP y gestionar futuras operaciones de volumen.

Cree un rol ONTAP personalizado para Trident.

Puede crear un rol de clúster ONTAP con privilegios mínimos para que no tenga que usar el rol de administrador de ONTAP para realizar operaciones en Trident. Cuando incluyes el nombre de usuario en una configuración de backend de Trident , Trident utiliza el rol de clúster ONTAP que creaste para realizar las operaciones.

Referirse a"Generador de roles personalizados de Trident" Para obtener más información sobre la creación de roles personalizados de Trident .

Uso de la CLI de ONTAP

1. Crea un nuevo rol utilizando el siguiente comando:

```
security login role create <role_name\> -cmddirname "command" -access all
–vserver <svm_name\>
```

2. Crea un nombre de usuario para el usuario de Trident :

```
security login create -username <user_name\> -application ontapi
-authmethod <password\> -role <name_of_role_in_step_1\> –vserver
<svm_name\> -comment "user_description"
```

3. Asigna el rol al usuario:

```
security login modify username <user_name\> –vserver <svm_name\> -role
<role_name\> -application ontapi -application console -authmethod
<password\>
```

Usando el Administrador del sistema

Realice los siguientes pasos en ONTAP System Manager:

1. **Crea un rol personalizado**:
   a. Para crear un rol personalizado a nivel de clúster, seleccione **Clúster > Configuración**.

      (O) Para crear un rol personalizado a nivel de SVM, seleccione **Almacenamiento > Máquinas virtuales de almacenamiento > `required SVM` > Configuración > Usuarios y roles**.

   b. Seleccione el icono de flecha (→) junto a **Usuarios y roles**.
   c. Seleccione ****Agregar** en **Roles**.
   d. Define las reglas para el rol y haz clic en **Guardar**.
2. **Asigna el rol al usuario de Trident *: + Realiza los siguientes pasos en la página *Usuarios y roles**:
   a. Seleccione el icono Agregar ***+** debajo de **Usuarios**.
   b. Seleccione el nombre de usuario requerido y seleccione un rol en el menú desplegable para **Rol**.
   c. Haga clic en **Guardar**.

Para obtener más información, consulte las siguientes páginas:

- "Roles personalizados para la administración de ONTAP"o"Definir roles personalizados"
- "Trabajar con roles y usuarios"

Gestionar las políticas de exportación NFS

Trident utiliza políticas de exportación NFS para controlar el acceso a los volúmenes que aprovisiona.

Trident ofrece dos opciones al trabajar con políticas de exportación:

- Trident puede gestionar dinámicamente la propia política de exportación; en este modo de funcionamiento, el administrador de almacenamiento especifica una lista de bloques CIDR que representan direcciones IP admisibles. Trident agrega automáticamente a la política de exportación, en el momento de la publicación, las direcciones IP de los nodos aplicables que se encuentren dentro de estos rangos. Alternativamente, cuando no se especifican CIDR, todas las IP de unidifusión de ámbito global que se encuentren en el nodo al que se publica el volumen se agregarán a la política de exportación.
- Los administradores de almacenamiento pueden crear una política de exportación y agregar reglas manualmente. Trident utiliza la política de exportación predeterminada a menos que se especifique un nombre de política de exportación diferente en la configuración.

Gestionar dinámicamente las políticas de exportación

Trident ofrece la capacidad de gestionar dinámicamente las políticas de exportación para los sistemas backend de ONTAP . Esto proporciona al administrador de almacenamiento la capacidad de especificar un espacio de direcciones permitido para las IP de los nodos de trabajo, en lugar de definir reglas explícitas manualmente. Simplifica enormemente la gestión de la política de exportación; las modificaciones a la política de exportación ya no requieren intervención manual en el clúster de almacenamiento. Además, esto ayuda a restringir el acceso al clúster de almacenamiento únicamente a los nodos de trabajo que están montando volúmenes y tienen direcciones IP dentro del rango especificado, lo que permite una gestión automatizada y de grano fino.

> No utilice la traducción de direcciones de red (NAT) cuando utilice políticas de exportación dinámicas. Con NAT, el controlador de almacenamiento ve la dirección NAT de front-end y no la dirección IP real del host, por lo que el acceso se denegará cuando no se encuentre ninguna coincidencia en las reglas de exportación.

Ejemplo

Existen dos opciones de configuración que deben utilizarse. Aquí tenéis un ejemplo de definición de backend:

```
---
version: 1
storageDriverName: ontap-nas-economy
backendName: ontap_nas_auto_export
managementLIF: 192.168.0.135
svm: svm1
username: vsadmin
password: password
autoExportCIDRs:
  - 192.168.0.0/24
autoExportPolicy: true
```

> Al utilizar esta función, debe asegurarse de que la unión raíz en su SVM tenga una política de exportación creada previamente con una regla de exportación que permita el bloque CIDR del nodo (como la política de exportación predeterminada). Siga siempre las mejores prácticas recomendadas NetApp para dedicar una SVM a Trident.

Aquí tienes una explicación de cómo funciona esta función utilizando el ejemplo anterior:

- `autoExportPolicy`está configurado para `true`. Esto indica que Trident crea una política de exportación para cada volumen aprovisionado con este backend para el `svm1` SVM y gestionar la adición y eliminación de reglas utilizando `autoexportCIDRs` bloques de direcciones. Hasta que un volumen se conecta a un nodo, el volumen utiliza una política de exportación vacía sin reglas para evitar el acceso no deseado a ese volumen. Cuando se publica un volumen en un nodo, Trident crea una política de exportación con el mismo nombre que el qtree subyacente que contiene la IP del nodo dentro del bloque CIDR especificado. Estas direcciones IP también se añadirán a la política de exportación utilizada por el FlexVol volume principal.
  - Por ejemplo:
    - UUID del backend 403b5326-8482-40db-96d0-d83fb3f4daec
    - `autoExportPolicy`empezar a `true`
    - prefijo de almacenamiento `trident`
    - UUID de PVC a79bcf5f-7b6d-4a40-9876-e2551f159c1c
    - El qtree denominado trident_pvc_a79bcf5f_7b6d_4a40_9876_e2551f159c1c crea una política de exportación para el FlexVol denominado `trident-403b5326-8482-40db96d0-d83fb3f4daec`, una política de exportación para el qtree llamado `trident_pvc_a79bcf5f_7b6d_4a40_9876_e2551f159c1c` y una política de exportación vacía llamada `trident_empty` en la SVM. Las reglas para la política de exportación de FlexVol serán un superconjunto de cualquier regla contenida en las políticas de exportación de qtree. La política de exportación vacía será reutilizada por cualquier volumen que no esté adjunto.
- `autoExportCIDRs`Contiene una lista de bloques de direcciones. Este campo es opcional y por defecto es ["0.0.0.0/0", "::/0"]. Si no se define, Trident agrega todas las direcciones unicast de ámbito global que se encuentren en los nodos de trabajo con publicaciones.

En este ejemplo, el `192.168.0.0/24` Se proporciona espacio de direcciones. Esto indica que las direcciones IP de los nodos de Kubernetes que se encuentren dentro de este rango de direcciones con publicaciones se agregarán a la política de exportación que crea Trident . Cuando Trident registra un nodo en el que se ejecuta, recupera las direcciones IP del nodo y las compara con los bloques de direcciones proporcionados en `autoExportCIDRs` En el momento de la publicación, después de filtrar las direcciones IP, Trident crea las reglas de política de exportación para las direcciones IP de los clientes del nodo al que está publicando.

Puedes actualizar `autoExportPolicy` y `autoExportCIDRs` para los backends después de crearlos. Puede agregar nuevos CIDR para un backend que se administra automáticamente o eliminar los CIDR existentes. Tenga cuidado al eliminar CIDR para asegurarse de que no se pierdan las conexiones existentes. También puedes optar por desactivar `autoExportPolicy` para un backend y recurrir a una política de exportación creada manualmente. Esto requerirá configurar el `exportPolicy` parámetro en la configuración de tu backend.

Después de que Trident crea o actualiza un backend, puede comprobar el backend mediante `tridentctl` o el correspondiente `tridentbackend` CRD:

```
./tridentctl get backends ontap_nas_auto_export -n trident -o yaml
items:
- backendUUID: 403b5326-8482-40db-96d0-d83fb3f4daec
  config:
    aggregate: ""
    autoExportCIDRs:
    - 192.168.0.0/24
    autoExportPolicy: true
    backendName: ontap_nas_auto_export
    chapInitiatorSecret: ""
    chapTargetInitiatorSecret: ""
    chapTargetUsername: ""
    chapUsername: ""
    dataLIF: 192.168.0.135
    debug: false
    debugTraceFlags: null
    defaults:
      encryption: "false"
      exportPolicy: <automatic>
      fileSystemType: ext4
```

Cuando se elimina un nodo, Trident revisa todas las políticas de exportación para eliminar las reglas de acceso correspondientes al nodo. Al eliminar la IP de este nodo de las políticas de exportación de los backends administrados, Trident evita montajes no autorizados, a menos que esta IP sea reutilizada por un nuevo nodo en el clúster.

Para los backends existentes, actualizar el backend con `tridentctl update backend` garantiza que Trident gestione automáticamente las políticas de exportación. Esto crea dos nuevas políticas de exportación que reciben el nombre del UUID del backend y del nombre del qtree cuando sea necesario. Los volúmenes que se encuentren en el backend utilizarán las políticas de exportación recién creadas después de desmontarlos y volverlos a montar.

> Eliminar un backend con políticas de exportación autogestionadas eliminará la política de exportación creada dinámicamente. Si se vuelve a crear el backend, se tratará como un backend nuevo y dará lugar a la creación de una nueva política de exportación.

Si se actualiza la dirección IP de un nodo activo, debe reiniciar el pod de Trident en el nodo. A continuación, Trident actualizará la política de exportación para los backends que administra para reflejar este cambio de IP.

Preparar el aprovisionamiento de volúmenes SMB

Con un poco de preparación adicional, puede aprovisionar volúmenes SMB usando `ontap-nas` conductores.

> Debes configurar los protocolos NFS y SMB/CIFS en la SVM para crear una `ontap-nas-economy` Volumen SMB para clústeres ONTAP locales. Si no se configura alguno de estos protocolos, la creación del volumen SMB fallará.

`autoExportPolicy`No es compatible con volúmenes SMB.

Antes de empezar

Antes de poder aprovisionar volúmenes SMB, debe tener lo siguiente.

- Un clúster de Kubernetes con un nodo controlador Linux y al menos un nodo de trabajo Windows que ejecuta Windows Server 2022. Trident solo admite volúmenes SMB montados en pods que se ejecutan en nodos Windows.
- Al menos un secreto de Trident que contenga sus credenciales de Active Directory. Para generar secretos `smbcreds`:

```
kubectl create secret generic smbcreds --from-literal username=user
--from-literal password='password'
```

- Un proxy CSI configurado como servicio de Windows. Para configurar un `csi-proxy`, consulte a"GitHub: Proxy CSI" o"GitHub: Proxy CSI para Windows" para nodos de Kubernetes que se ejecutan en Windows.

Pasos

1. Para ONTAP local, opcionalmente puede crear un recurso compartido SMB o Trident puede crearlo por usted.

   Se requieren recursos compartidos SMB para Amazon FSx para ONTAP.

   Puedes crear los recursos compartidos de administración SMB de dos maneras: utilizando…"Consola de administración de Microsoft" Complemento de carpetas compartidas o mediante la CLI de ONTAP . Para crear los recursos compartidos SMB mediante la CLI de ONTAP :

   a. Si es necesario, cree la estructura de rutas de directorio para el recurso compartido.

      El `vserver cifs share create` El comando verifica la ruta especificada en la opción -path durante la creación del recurso compartido. Si la ruta especificada no existe, el comando falla.

   b. Cree un recurso compartido SMB asociado con la SVM especificada:

```
vserver cifs share create -vserver vserver_name -share-name
share_name -path path [-share-properties share_properties,...]
[other_attributes] [-comment text]
```

   c. Verifique que se haya creado el recurso compartido:

```
vserver cifs share show -share-name share_name
```

   Referirse a"Crear un recurso compartido SMB" Para más detalles.

2. Al crear el backend, debe configurar lo siguiente para especificar los volúmenes SMB. Para conocer todas las opciones de configuración del backend de FSx para ONTAP , consulte"Opciones de configuración y

ejemplos de FSx para ONTAP" .

Parámetro	Descripción	Ejemplo
`smbShare`	Puede especificar una de las siguientes opciones: el nombre de un recurso compartido SMB creado mediante la Consola de administración de Microsoft o la CLI de ONTAP ; un nombre para permitir que Trident cree el recurso compartido SMB; o puede dejar el parámetro en blanco para evitar el acceso compartido común a los volúmenes. Este parámetro es opcional para ONTAP local. Este parámetro es obligatorio para los backends de Amazon FSx para ONTAP y no puede estar en blanco.	`smb-share`
`nasType`	**Debe configurarse en `smb` .** Si es nulo, el valor predeterminado es `nfs` .	`smb`
`securityStyle`	Estilo de seguridad para nuevos volúmenes. **Debe configurarse en `ntfs` o `mixed` para volúmenes SMB.**	`ntfs`o `mixed` para volúmenes SMB
`unixPermissions`	Modo para nuevos volúmenes. **Debe dejarse vacío para volúmenes SMB.**	""

Habilitar SMB seguro

A partir de la versión 25.06, NetApp Trident admite el aprovisionamiento seguro de volúmenes SMB creados mediante `ontap-nas` y `ontap-nas-economy` backends. Cuando SMB seguro está habilitado, puede proporcionar acceso controlado a los recursos compartidos SMB para usuarios y grupos de usuarios de Active Directory (AD) mediante listas de control de acceso (ACL).

Puntos para recordar

- Importador `ontap-nas-economy` No se admiten volúmenes.
- Solo se admiten clones de solo lectura para `ontap-nas-economy` volúmenes.
- Si Secure SMB está habilitado, Trident ignorará el recurso compartido SMB mencionado en el backend.
- La actualización de la anotación PVC, la anotación de la clase de almacenamiento y el campo backend no actualiza la ACL del recurso compartido SMB.
- Las ACL de recursos compartidos SMB especificadas en la anotación del PVC clonado tendrán prioridad sobre las del PVC de origen.
- Asegúrese de proporcionar usuarios de AD válidos al habilitar SMB seguro. Los usuarios no válidos no se agregarán a la ACL.
- Si se proporciona el mismo usuario de AD en el backend, la clase de almacenamiento y el PVC con diferentes permisos, la prioridad de permisos será: PVC, clase de almacenamiento y, por último, backend.
- Se admite Secure SMB para `ontap-nas` Se aplica a las importaciones de volumen gestionadas y no a las importaciones de volumen no gestionadas.

Pasos

1. Especifique adAdminUser en TridentBackendConfig como se muestra en el siguiente ejemplo:

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-ontap
  namespace: trident
spec:
  version: 1
  storageDriverName: ontap-nas
  managementLIF: 10.193.176.x
  svm: svm0
  useREST: true
  defaults:
    adAdminUser: tridentADtest
  credentials:
    name: backend-tbc-ontap-invest-secret
```

2. Agregue la anotación en la clase de almacenamiento.

   Añade el `trident.netapp.io/smbShareAdUser` Anotación a la clase de almacenamiento para habilitar SMB seguro sin fallos. El valor de usuario especificado para la anotación `trident.netapp.io/smbShareAdUser` debe ser el mismo que el nombre de usuario especificado en el `smbcreds` secreto. Puedes elegir una de las siguientes opciones para `smbShareAdUserPermission` : `full_control` , `change` , o `read` . El permiso predeterminado es `full_control` .

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontap-smb-sc
  annotations:
    trident.netapp.io/smbShareAdUserPermission: change
    trident.netapp.io/smbShareAdUser: tridentADuser
parameters:
  backendType: ontap-nas
  csi.storage.k8s.io/node-stage-secret-name: smbcreds
  csi.storage.k8s.io/node-stage-secret-namespace: trident
  trident.netapp.io/nasType: smb
provisioner: csi.trident.netapp.io
reclaimPolicy: Delete
volumeBindingMode: Immediate
```

1. Crea un tubo de PVC.

   El siguiente ejemplo crea un PVC:

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: my-pvc4
  namespace: trident
  annotations:
    trident.netapp.io/snapshotDirectory: "true"
    trident.netapp.io/smbShareAccessControl: |
      read:
        - tridentADtest
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: ontap-smb-sc
```

Opciones y ejemplos de configuración de ONTAP NAS

Aprenda a crear y utilizar controladores NAS ONTAP con su instalación de Trident . Esta sección proporciona ejemplos de configuración de backend y detalles para mapear backends a StorageClasses.

Opciones de configuración del backend

Consulte la siguiente tabla para ver las opciones de configuración del backend:

Parámetro	Descripción	Por defecto
`version`		Siempre 1
`storageDriverName`	Nombre del controlador de almacenamiento	`ontap-nas`, `ontap-nas-economy` , o `ontap-nas-flexgroup`
`backendName`	Nombre personalizado o el backend de almacenamiento	Nombre del controlador + "_" + dataLIF
`managementLIF`	Dirección IP de un clúster o LIF de administración de SVM Se puede especificar un nombre de dominio completo (FQDN). Se puede configurar para usar direcciones IPv6 si Trident se instaló usando la bandera IPv6. Las direcciones IPv6 deben definirse entre corchetes, como por ejemplo: `[28e8:d9fb:a825:b7bf:69a8:d02f:9e7b:3555]` . Para una transición fluida a MetroCluster , consulteEjemplo de MetroCluster .	"10.0.0.1", "[2001:1234:abcd::fefe]"

Parámetro	Descripción	Por defecto
`dataLIF`	Dirección IP del protocolo LIF. NetApp recomienda especificar `dataLIF` . Si no se proporcionan, Trident obtiene los dataLIF del SVM. Puede especificar un nombre de dominio completo (FQDN) para usarlo en las operaciones de montaje NFS, lo que le permite crear un DNS round-robin para equilibrar la carga entre varios dataLIF. Puede modificarse después de la configuración inicial. Referirse a . Se puede configurar para usar direcciones IPv6 si Trident se instaló usando la bandera IPv6. Las direcciones IPv6 deben definirse entre corchetes, como por ejemplo: `[28e8:d9fb:a825:b7bf:69a8:d02f:9e7b:3555]` . **Omitir para Metrocluster.** Ver elEjemplo de MetroCluster .	Dirección especificada o derivada de SVM, si no se especifica (no recomendado).
`svm`	Máquina virtual de almacenamiento a utilizar **Omitir para Metrocluster.** Ver elEjemplo de MetroCluster .	Derivado si se trata de una SVM `managementLIF` se especifica
`autoExportPolicy`	Habilitar la creación y actualización automática de políticas de exportación [Booleano]. Utilizando el `autoExportPolicy` y `autoExportCIDRs` Con algunas opciones, Trident puede gestionar las políticas de exportación automáticamente.	FALSO
`autoExportCIDRs`	Lista de CIDR para filtrar las direcciones IP de los nodos de Kubernetes cuando `autoExportPolicy` está habilitado. Utilizando el `autoExportPolicy` y `autoExportCIDRs` Con algunas opciones, Trident puede gestionar las políticas de exportación automáticamente.	["0.0.0.0/0", "::/0"]`
`labels`	Conjunto de etiquetas arbitrarias con formato JSON para aplicar a los volúmenes	""
`clientCertificate`	Valor codificado en Base64 del certificado del cliente. Se utiliza para la autenticación basada en certificados.	""
`clientPrivateKey`	Valor codificado en Base64 de la clave privada del cliente. Se utiliza para la autenticación basada en certificados.	""
`trustedCACertificate`	Valor codificado en Base64 del certificado de CA de confianza. Opcional. Se utiliza para la autenticación basada en certificados.	""
`username`	Nombre de usuario para conectarse al clúster/SVM. Se utiliza para la autenticación basada en credenciales. Para la autenticación de Active Directory, consulte "Autenticar Trident en un SVM backend mediante credenciales de Active Directory".	

Parámetro	Descripción	Por defecto
`password`	Contraseña para conectarse al cluster/SVM. Se utiliza para la autenticación basada en credenciales. Para la autenticación de Active Directory, consulte "Autenticar Trident en un SVM backend mediante credenciales de Active Directory".	
`storagePrefix`	Prefijo utilizado al aprovisionar nuevos volúmenes en la SVM. No se puede actualizar después de configurarlo. Cuando se utiliza ontap-nas-economy y un prefijo de almacenamiento de 24 caracteres o más, los qtrees no tendrán el prefijo de almacenamiento incrustado, aunque sí estará en el nombre del volumen.	"tridente"
`aggregate`	Agregado para aprovisionamiento (opcional; si se establece, debe asignarse a la SVM). Para el `ontap-nas-flexgroup` conductor, esta opción se ignora. Si no se asigna, cualquiera de los agregados disponibles se puede utilizar para aprovisionar un volumen FlexGroup . Cuando se actualiza el agregado en SVM, se actualiza automáticamente en Trident mediante sondeos a SVM sin necesidad de reiniciar el controlador Trident . Cuando se ha configurado un agregado específico en Trident para aprovisionar volúmenes, si el agregado se renombra o se mueve fuera del SVM, el backend pasará a un estado de error en Trident mientras consulta el agregado del SVM. Debe cambiar el agregado por uno que esté presente en la SVM o eliminarlo por completo para volver a poner en línea el backend.	""
`limitAggregateUsage`	Fallará el aprovisionamiento si el uso supera este porcentaje. **No se aplica a Amazon FSx para ONTAP**.	" (no se aplica por defecto)

Parámetro	Descripción	Por defecto
Lista agregada de flexgroup	Lista de agregados para el aprovisionamiento (opcional; si se establece, debe asignarse a la SVM). Todos los agregados asignados al SVM se utilizan para aprovisionar un volumen FlexGroup . Compatible con el controlador de almacenamiento **ontap-nas-flexgroup**. Cuando se actualiza la lista agregada en SVM, la lista se actualiza automáticamente en Trident mediante sondeos a SVM sin necesidad de reiniciar el controlador Trident . Cuando se ha configurado una lista agregada específica en Trident para aprovisionar volúmenes, si la lista agregada se renombra o se mueve fuera de SVM, el backend pasará a un estado de error en Trident mientras consulta el agregado de SVM. Debe cambiar la lista agregada por una que esté presente en la SVM o eliminarla por completo para volver a poner en línea el backend.	""
`limitVolumeSize`	Fallará el aprovisionamiento si el tamaño de volumen solicitado supera este valor. También restringe el tamaño máximo de los volúmenes que administra para los qtrees, y el `qtreesPerFlexvol` Esta opción permite personalizar el número máximo de qtrees por FlexVol volume.	" (no se aplica por defecto)
`debugTraceFlags`	Indicadores de depuración para usar al solucionar problemas. Ejemplo: {"api":false, "method":true} No usar `debugTraceFlags` a menos que esté solucionando problemas y necesite un registro detallado.	nulo
`nasType`	Configure la creación de volúmenes NFS o SMB. Las opciones son `nfs` , `smb` o nulo. Si se establece en nulo, se utilizarán volúmenes NFS por defecto.	`nfs`

Parámetro	Descripción	Por defecto
`nfsMountOptions`	Lista de opciones de montaje NFS separadas por comas. Las opciones de montaje para volúmenes persistentes de Kubernetes normalmente se especifican en las clases de almacenamiento, pero si no se especifican opciones de montaje en una clase de almacenamiento, Trident recurrirá a las opciones de montaje especificadas en el archivo de configuración del backend de almacenamiento. Si no se especifican opciones de montaje en la clase de almacenamiento o en el archivo de configuración, Trident no establecerá ninguna opción de montaje en un volumen persistente asociado.	""
`qtreesPerFlexvol`	Número máximo de Qtrees por FlexVol, debe estar en el rango [50, 300]	"200"
`smbShare`	Puede especificar una de las siguientes opciones: el nombre de un recurso compartido SMB creado mediante la Consola de administración de Microsoft o la CLI de ONTAP ; un nombre para permitir que Trident cree el recurso compartido SMB; o puede dejar el parámetro en blanco para evitar el acceso compartido común a los volúmenes. Este parámetro es opcional para ONTAP local. Este parámetro es obligatorio para los backends de Amazon FSx para ONTAP y no puede estar en blanco.	`smb-share`
`useREST`	Parámetro booleano para utilizar las API REST de ONTAP . `useREST`Cuando se configura para `true` Trident utiliza las API REST de ONTAP para comunicarse con el backend; cuando se configura en `false` Trident utiliza llamadas ONTAPI (ZAPI) para comunicarse con el backend. Esta función requiere ONTAP 9.11.1 y versiones posteriores. Además, el rol de inicio de sesión de ONTAP utilizado debe tener acceso a `ontapi` solicitud. Esto se satisface mediante lo predefinido. `vsadmin` y `cluster-admin` roles. A partir de la versión Trident 24.06 y ONTAP 9.15.1 o posterior, `useREST` está configurado para `true` por defecto; cambiar `useREST` a `false` para utilizar llamadas ONTAPI (ZAPI).	`true`para ONTAP 9.15.1 o posterior, de lo contrario `false`.
`limitVolumePoolSize`	Tamaño máximo de FlexVol que se puede solicitar al usar Qtrees en el backend ontap-nas-economy.	" (no se aplica por defecto)
`denyNewVolumePools`	Restringe `ontap-nas-economy` backends que crean nuevos volúmenes FlexVol para contener sus Qtrees. Solo se utilizan Flexvols preexistentes para el aprovisionamiento de nuevos PV.	

Parámetro	Descripción	Por defecto
adAdminUser	Usuario o grupo de usuarios administradores de Active Directory con acceso completo a los recursos compartidos SMB. Utilice este parámetro para otorgar derechos de administrador al recurso compartido SMB con control total.	

Opciones de configuración de backend para el aprovisionamiento de volúmenes

Puedes controlar el aprovisionamiento predeterminado utilizando estas opciones en el `defaults` sección de la configuración. Para ver un ejemplo, consulte los ejemplos de configuración a continuación.

Parámetro	Descripción	Por defecto
spaceAllocation	Asignación de espacio para Qtrees	"verdadero"
spaceReserve	Modo de reserva de espacio: "ninguno" (delgado) o "volumen" (grueso).	"ninguno"
snapshotPolicy	Política de instantáneas a utilizar	"ninguno"
qosPolicy	Grupo de políticas QoS que se asignará a los volúmenes creados. Elija una de las opciones qosPolicy o adaptiveQosPolicy por grupo de almacenamiento/backend.	""
adaptiveQosPolicy	Grupo de políticas QoS adaptativas para asignar a los volúmenes creados. Elija una de las opciones qosPolicy o adaptiveQosPolicy por grupo de almacenamiento/backend. No compatible con ontap-nas-economy.	""
snapshotReserve	Porcentaje de volumen reservado para instantáneas	"0" si `snapshotPolicy` es "ninguno", de lo contrario ""
splitOnClone	Separar un clon de su progenitor al crearlo	"FALSO"
encryption	Habilite el cifrado de volumen de NetApp (NVE) en el nuevo volumen; el valor predeterminado es `false` . Para utilizar esta opción, NVE debe estar licenciado y habilitado en el clúster. Si NAE está habilitado en el backend, cualquier volumen aprovisionado en Trident tendrá NAE habilitado. Para obtener más información, consulte:"Cómo funciona Trident con NVE y NAE" .	"FALSO"
tieringPolicy	Política de niveles para usar "ninguno"	
unixPermissions	Modo para nuevos volúmenes	"777" para volúmenes NFS; vacío (no aplicable) para volúmenes SMB
snapshotDir	Controla el acceso a `.snapshot` directorio	"verdadero" para NFSv4, "falso" para NFSv3
exportPolicy	Política de exportación a utilizar	"por defecto"

Parámetro	Descripción	Por defecto
securityStyle	Estilo de seguridad para nuevos volúmenes. NFS admite mixed y unix Estilos de seguridad. Las PYMES son compatibles con el soporte. mixed y ntfs Estilos de seguridad.	El valor predeterminado de NFS es unix . El valor predeterminado de SMB es ntfs .
nameTemplate	Plantilla para crear nombres de volumen personalizados.	""

El uso de grupos de políticas QoS con Trident requiere ONTAP 9.8 o posterior. Debe utilizar un grupo de políticas QoS no compartido y asegurarse de que el grupo de políticas se aplique a cada componente individualmente. Un grupo de políticas QoS compartidas impone un límite máximo al rendimiento total de todas las cargas de trabajo.

Ejemplos de aprovisionamiento por volumen

Aquí tenéis un ejemplo con valores predeterminados definidos:

```
---
version: 1
storageDriverName: ontap-nas
backendName: customBackendName
managementLIF: 10.0.0.1
dataLIF: 10.0.0.2
labels:
  k8scluster: dev1
  backend: dev1-nasbackend
svm: trident_svm
username: cluster-admin
password: <password>
limitAggregateUsage: 80%
limitVolumeSize: 50Gi
nfsMountOptions: nfsvers=4
debugTraceFlags:
  api: false
  method: true
defaults:
  spaceReserve: volume
  qosPolicy: premium
  exportPolicy: myk8scluster
  snapshotPolicy: default
  snapshotReserve: "10"
```

Para ontap-nas y ontap-nas-flexgroups Trident ahora utiliza un nuevo cálculo para garantizar que el FlexVol tenga el tamaño correcto con el porcentaje de snapshotReserve y el PVC. Cuando el usuario solicita un PVC, Trident crea el FlexVol original con más espacio mediante el nuevo cálculo. Este cálculo garantiza que el usuario reciba el espacio de escritura solicitado en el PVC, y no menos espacio del solicitado. Antes de

la versión v21.07, cuando el usuario solicitaba un PVC (por ejemplo, 5 GiB), con el snapshotReserve al 50 %, obtenía solo 2,5 GiB de espacio de escritura. Esto se debe a que lo que el usuario solicitó fue el volumen completo y `snapshotReserve` es un porcentaje de eso. Con Trident 21.07, lo que el usuario solicita es el espacio de escritura y Trident define el `snapshotReserve` número como porcentaje del volumen total. Esto no se aplica a `ontap-nas-economy` . Vea el siguiente ejemplo para ver cómo funciona

El cálculo es el siguiente:

```
Total volume size = (PVC requested size) / (1 - (snapshotReserve
percentage) / 100)
```

Para snapshotReserve = 50% y la solicitud de PVC = 5 GiB, el tamaño total del volumen es 5/.5 = 10 GiB y el tamaño disponible es 5 GiB, que es lo que el usuario solicitó en la solicitud de PVC El `volume show` El comando debería mostrar resultados similares a este ejemplo:

Los backends existentes de instalaciones anteriores aprovisionarán volúmenes como se explicó anteriormente al actualizar Trident. Para los volúmenes creados antes de la actualización, debe redimensionarlos para que se observe el cambio. Por ejemplo, un PVC de 2 GiB con `snapshotReserve=50` Anteriormente se obtuvo un volumen que proporciona 1 GiB de espacio de escritura. Por ejemplo, al redimensionar el volumen a 3 GiB, la aplicación obtiene 3 GiB de espacio de escritura en un volumen de 6 GiB.

Ejemplos de configuración mínima

Los siguientes ejemplos muestran configuraciones básicas que dejan la mayoría de los parámetros con sus valores predeterminados. Esta es la forma más sencilla de definir un backend.

> Si está utilizando Amazon FSx en NetApp ONTAP con Trident, se recomienda especificar nombres DNS para las LIF en lugar de direcciones IP.

Ejemplo de economía NAS de ONTAP

```
---
version: 1
storageDriverName: ontap-nas-economy
managementLIF: 10.0.0.1
dataLIF: 10.0.0.2
svm: svm_nfs
username: vsadmin
password: password
```

Ejemplo de ONTAP NAS Flexgroup

```
---
version: 1
storageDriverName: ontap-nas-flexgroup
managementLIF: 10.0.0.1
dataLIF: 10.0.0.2
svm: svm_nfs
username: vsadmin
password: password
```

Ejemplo de MetroCluster

Puede configurar el backend para evitar tener que actualizar manualmente la definición del backend después del cambio de estado y el cambio de estado durante"replicación y recuperación de SVM" .

Para una conmutación y recuperación sin interrupciones, especifique el SVM utilizando `managementLIF` y omitir el `dataLIF` y `svm` parámetros. Por ejemplo:

```
---
version: 1
storageDriverName: ontap-nas
managementLIF: 192.168.1.66
username: vsadmin
password: password
```

Ejemplo de volúmenes SMB

```
---
version: 1
backendName: ExampleBackend
storageDriverName: ontap-nas
managementLIF: 10.0.0.1
nasType: smb
securityStyle: ntfs
unixPermissions: ""
dataLIF: 10.0.0.2
svm: svm_nfs
username: vsadmin
password: password
```

Ejemplo de autenticación basada en certificados

Este es un ejemplo mínimo de configuración de backend. `clientCertificate` , `clientPrivateKey` , y `trustedCACertificate` (opcional, si se utiliza una CA de confianza) se rellenan en `backend.json` y tome los valores codificados en base64 del certificado del cliente, la clave privada y el certificado de CA de confianza, respectivamente.

```
---
version: 1
backendName: DefaultNASBackend
storageDriverName: ontap-nas
managementLIF: 10.0.0.1
dataLIF: 10.0.0.15
svm: nfs_svm
clientCertificate: ZXR0ZXJwYXB...ICMgJ3BhcGVyc2
clientPrivateKey: vciwKIyAgZG...0cnksIGRlc2NyaX
trustedCACertificate: zcyBbaG...b3Igb3duIGNsYXNz
storagePrefix: myPrefix_
```

Ejemplo de política de exportación automática

Este ejemplo muestra cómo puede configurar Trident para que utilice políticas de exportación dinámicas para crear y gestionar automáticamente la política de exportación. Esto funciona igual para el `ontap-nas-economy` y `ontap-nas-flexgroup` conductores.

```
---
version: 1
storageDriverName: ontap-nas
managementLIF: 10.0.0.1
dataLIF: 10.0.0.2
svm: svm_nfs
labels:
  k8scluster: test-cluster-east-1a
  backend: test1-nasbackend
autoExportPolicy: true
autoExportCIDRs:
- 10.0.0.0/24
username: admin
password: password
nfsMountOptions: nfsvers=4
```

Ejemplo de direcciones IPv6

Este ejemplo muestra `managementLIF` utilizando una dirección IPv6.

```
---
version: 1
storageDriverName: ontap-nas
backendName: nas_ipv6_backend
managementLIF: "[5c5d:5edf:8f:7657:bef8:109b:1b41:d491]"
labels:
  k8scluster: test-cluster-east-1a
  backend: test1-ontap-ipv6
svm: nas_ipv6_svm
username: vsadmin
password: password
```

Ejemplo de Amazon FSx para ONTAP con volúmenes SMB

El `smbShare` Este parámetro es necesario para FSx para ONTAP que utiliza volúmenes SMB.

```
---
version: 1
backendName: SMBBackend
storageDriverName: ontap-nas
managementLIF: example.mgmt.fqdn.aws.com
nasType: smb
dataLIF: 10.0.0.15
svm: nfs_svm
smbShare: smb-share
clientCertificate: ZXR0ZXJwYXB...ICMgJ3BhcGVyc2
clientPrivateKey: vciwKIyAgZG...0cnksIGRlc2NyaX
trustedCACertificate: zcyBbaG...b3Igb3duIGNsYXNz
storagePrefix: myPrefix_
```

Ejemplo de configuración de backend con plantilla de nombre

```
---
version: 1
storageDriverName: ontap-nas
backendName: ontap-nas-backend
managementLIF: <ip address>
svm: svm0
username: <admin>
password: <password>
defaults:
  nameTemplate:
"{{.volume.Name}}_{{.labels.cluster}}_{{.volume.Namespace}}_{{.vo\
    lume.RequestName}}"
labels:
  cluster: ClusterA
  PVC: "{{.volume.Namespace}}_{{.volume.RequestName}}"
```

Ejemplos de backends con pools virtuales

En los archivos de definición de backend de ejemplo que se muestran a continuación, se establecen valores predeterminados específicos para todos los grupos de almacenamiento, tales como: `spaceReserve` en ninguno, `spaceAllocation` en falso, y `encryption` en falso. Los grupos virtuales se definen en la sección de almacenamiento.

Trident establece las etiquetas de aprovisionamiento en el campo "Comentarios". Los comentarios están configurados en FlexVol para `ontap-nas` o FlexGroup para `ontap-nas-flexgroup` . Trident copia todas las etiquetas presentes en un grupo virtual al volumen de almacenamiento durante el aprovisionamiento. Para mayor comodidad, los administradores de almacenamiento pueden definir etiquetas por grupo virtual y agrupar volúmenes por etiqueta.

En estos ejemplos, algunos de los grupos de almacenamiento establecen sus propias configuraciones. `spaceReserve` , `spaceAllocation` , y `encryption` valores, y algunos pools anulan los valores predeterminados.

Ejemplo de ONTAP NAS

```
---
version: 1
storageDriverName: ontap-nas
managementLIF: 10.0.0.1
svm: svm_nfs
username: admin
password: <password>
nfsMountOptions: nfsvers=4
defaults:
  spaceReserve: none
  encryption: "false"
  qosPolicy: standard
labels:
  store: nas_store
  k8scluster: prod-cluster-1
region: us_east_1
storage:
  - labels:
      app: msoffice
      cost: "100"
    zone: us_east_1a
    defaults:
      spaceReserve: volume
      encryption: "true"
      unixPermissions: "0755"
      adaptiveQosPolicy: adaptive-premium
  - labels:
      app: slack
      cost: "75"
    zone: us_east_1b
    defaults:
      spaceReserve: none
      encryption: "true"
      unixPermissions: "0755"
  - labels:
      department: legal
      creditpoints: "5000"
    zone: us_east_1b
    defaults:
      spaceReserve: none
      encryption: "true"
      unixPermissions: "0755"
  - labels:
      app: wordpress
```

```
      cost: "50"
    zone: us_east_1c
    defaults:
      spaceReserve: none
      encryption: "true"
      unixPermissions: "0775"
  - labels:
      app: mysqldb
      cost: "25"
    zone: us_east_1d
    defaults:
      spaceReserve: volume
      encryption: "false"
      unixPermissions: "0775"
```

Ejemplo de ONTAP NAS FlexGroup

```
---
version: 1
storageDriverName: ontap-nas-flexgroup
managementLIF: 10.0.0.1
svm: svm_nfs
username: vsadmin
password: <password>
defaults:
  spaceReserve: none
  encryption: "false"
labels:
  store: flexgroup_store
  k8scluster: prod-cluster-1
region: us_east_1
storage:
  - labels:
      protection: gold
      creditpoints: "50000"
    zone: us_east_1a
    defaults:
      spaceReserve: volume
      encryption: "true"
      unixPermissions: "0755"
  - labels:
      protection: gold
      creditpoints: "30000"
    zone: us_east_1b
    defaults:
      spaceReserve: none
      encryption: "true"
      unixPermissions: "0755"
  - labels:
      protection: silver
      creditpoints: "20000"
    zone: us_east_1c
    defaults:
      spaceReserve: none
      encryption: "true"
      unixPermissions: "0775"
  - labels:
      protection: bronze
      creditpoints: "10000"
    zone: us_east_1d
    defaults:
```

```
      spaceReserve: volume
      encryption: "false"
      unixPermissions: "0775"
```

Ejemplo de economía NAS de ONTAP

```
---
version: 1
storageDriverName: ontap-nas-economy
managementLIF: 10.0.0.1
svm: svm_nfs
username: vsadmin
password: <password>
defaults:
  spaceReserve: none
  encryption: "false"
labels:
  store: nas_economy_store
region: us_east_1
storage:
  - labels:
      department: finance
      creditpoints: "6000"
    zone: us_east_1a
    defaults:
      spaceReserve: volume
      encryption: "true"
      unixPermissions: "0755"
  - labels:
      protection: bronze
      creditpoints: "5000"
    zone: us_east_1b
    defaults:
      spaceReserve: none
      encryption: "true"
      unixPermissions: "0755"
  - labels:
      department: engineering
      creditpoints: "3000"
    zone: us_east_1c
    defaults:
      spaceReserve: none
      encryption: "true"
      unixPermissions: "0775"
  - labels:
      department: humanresource
      creditpoints: "2000"
    zone: us_east_1d
    defaults:
      spaceReserve: volume
```

```
      encryption: "false"
      unixPermissions: "0775"
```

Asignar backends a StorageClasses

Las siguientes definiciones de StorageClass hacen referencia aEjemplos de backends con pools virtuales . Utilizando el `parameters.selector` En cada campo, cada StorageClass especifica qué grupos virtuales se pueden usar para alojar un volumen. El volumen tendrá los aspectos definidos en el pool virtual elegido.

- El `protection-gold` StorageClass se asignará al primer y segundo grupo virtual en el `ontap-nas-flexgroup` backend. Estas son las únicas piscinas que ofrecen protección de nivel oro.

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: protection-gold
provisioner: csi.trident.netapp.io
parameters:
  selector: "protection=gold"
  fsType: "ext4"
```

- El `protection-not-gold` StorageClass se asignará al tercer y cuarto grupo virtual en el `ontap-nas-flexgroup` backend. Estas son las únicas piscinas que ofrecen un nivel de protección distinto al oro.

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: protection-not-gold
provisioner: csi.trident.netapp.io
parameters:
  selector: "protection!=gold"
  fsType: "ext4"
```

- El `app-mysqldb` StorageClass se asignará al cuarto grupo virtual en el `ontap-nas` backend. Este es el único pool que ofrece configuración de pool de almacenamiento para aplicaciones de tipo mysqldb.

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: app-mysqldb
provisioner: csi.trident.netapp.io
parameters:
  selector: "app=mysqldb"
  fsType: "ext4"
```

- El `protection-silver-creditpoints-20k` StorageClass se asignará al tercer grupo virtual en el `ontap-nas-flexgroup` backend. Este es el único fondo que ofrece protección de nivel plata y 20000 puntos de crédito.

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: protection-silver-creditpoints-20k
provisioner: csi.trident.netapp.io
parameters:
  selector: "protection=silver; creditpoints=20000"
  fsType: "ext4"
```

- El `creditpoints-5k` StorageClass se asignará al tercer grupo virtual en el `ontap-nas` backend y el segundo grupo virtual en el `ontap-nas-economy` backend. Estas son las únicas ofertas de pool con 5000 puntos de crédito.

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: creditpoints-5k
provisioner: csi.trident.netapp.io
parameters:
  selector: "creditpoints=5000"
  fsType: "ext4"
```

Trident decidirá qué grupo virtual se selecciona y garantiza que se cumplan los requisitos de almacenamiento.

Actualizar `dataLIF` después de la configuración inicial

Puede cambiar el dataLIF después de la configuración inicial ejecutando el siguiente comando para proporcionar el nuevo archivo JSON de backend con el dataLIF actualizado.

```
tridentctl update backend <backend-name> -f <path-to-backend-json-file-
with-updated-dataLIF>
```

Si los PVC están conectados a uno o varios pods, debe desconectar todos los pods correspondientes y luego volver a conectarlos para que el nuevo dataLIF surta efecto.

Ejemplos de seguridad para pymes

Configuración del backend con el controlador ontap-nas

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-ontap-nas
  namespace: trident
spec:
  version: 1
  storageDriverName: ontap-nas
  managementLIF: 10.0.0.1
  svm: svm2
  nasType: smb
  defaults:
    adAdminUser: tridentADtest
  credentials:
    name: backend-tbc-ontap-invest-secret
```

Configuración de backend con el controlador ontap-nas-economy

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-ontap-nas
  namespace: trident
spec:
  version: 1
  storageDriverName: ontap-nas-economy
  managementLIF: 10.0.0.1
  svm: svm2
  nasType: smb
  defaults:
    adAdminUser: tridentADtest
  credentials:
    name: backend-tbc-ontap-invest-secret
```

Configuración del backend con grupo de almacenamiento

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-ontap-nas
  namespace: trident
spec:
  version: 1
  storageDriverName: ontap-nas
  managementLIF: 10.0.0.1
  svm: svm0
  useREST: false
  storage:
  - labels:
      app: msoffice
    defaults:
      adAdminUser: tridentADuser
  nasType: smb
  credentials:
    name: backend-tbc-ontap-invest-secret
```

Ejemplo de clase de almacenamiento con controlador ontap-nas

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontap-smb-sc
  annotations:
    trident.netapp.io/smbShareAdUserPermission: change
    trident.netapp.io/smbShareAdUser: tridentADtest
parameters:
  backendType: ontap-nas
  csi.storage.k8s.io/node-stage-secret-name: smbcreds
  csi.storage.k8s.io/node-stage-secret-namespace: trident
  trident.netapp.io/nasType: smb
provisioner: csi.trident.netapp.io
reclaimPolicy: Delete
volumeBindingMode: Immediate
```

Asegúrese de agregar `annotations` para habilitar SMB seguro. El protocolo SMB seguro no funciona sin las anotaciones, independientemente de las configuraciones establecidas en el backend o en el PVC.

Ejemplo de clase de almacenamiento con controlador ontap-nas-economy

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontap-smb-sc
  annotations:
    trident.netapp.io/smbShareAdUserPermission: change
    trident.netapp.io/smbShareAdUser: tridentADuser3
parameters:
  backendType: ontap-nas-economy
  csi.storage.k8s.io/node-stage-secret-name: smbcreds
  csi.storage.k8s.io/node-stage-secret-namespace: trident
  trident.netapp.io/nasType: smb
provisioner: csi.trident.netapp.io
reclaimPolicy: Delete
volumeBindingMode: Immediate
```

Ejemplo de PVC con un solo usuario de AD

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: my-pvc4
  namespace: trident
  annotations:
    trident.netapp.io/smbShareAccessControl: |
      change:
        - tridentADtest
      read:
        - tridentADuser
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: ontap-smb-sc
```

Ejemplo de PVC con múltiples usuarios de AD

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: my-test-pvc
  annotations:
    trident.netapp.io/smbShareAccessControl: |
      full_control:
        - tridentTestuser
        - tridentuser
        - tridentTestuser1
        - tridentuser1
      change:
        - tridentADuser
        - tridentADuser1
        - tridentADuser4
        - tridentTestuser2
      read:
        - tridentTestuser2
        - tridentTestuser3
        - tridentADuser2
        - tridentADuser3
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
```

Amazon FSx for NetApp ONTAP

Utilice Trident con Amazon FSx for NetApp ONTAP

"Amazon FSx for NetApp ONTAP"Es un servicio de AWS totalmente administrado que permite a los clientes lanzar y ejecutar sistemas de archivos basados en el sistema operativo de almacenamiento NetApp ONTAP . FSx para ONTAP le permite aprovechar las características, el rendimiento y las capacidades administrativas de NetApp con las que ya está familiarizado, al tiempo que se beneficia de la simplicidad, la agilidad, la seguridad y la escalabilidad del almacenamiento de datos en AWS. FSx para ONTAP admite las funciones del sistema de archivos y las API de administración de ONTAP .

Puede integrar su sistema de archivos Amazon FSx for NetApp ONTAP con Trident para garantizar que los clústeres de Kubernetes que se ejecutan en Amazon Elastic Kubernetes Service (EKS) puedan aprovisionar volúmenes persistentes de bloques y archivos respaldados por ONTAP.

En Amazon FSx, el sistema de archivos es el recurso principal, análogo a un clúster ONTAP local. Dentro de cada SVM puedes crear uno o varios volúmenes, que son contenedores de datos que almacenan los archivos

y carpetas de tu sistema de archivos. Con Amazon FSx for NetApp ONTAP se proporcionará como un sistema de archivos administrado en la nube. El nuevo tipo de sistema de archivos se llama * NetApp ONTAP*.

Al usar Trident con Amazon FSx for NetApp ONTAP, puede garantizar que los clústeres de Kubernetes que se ejecutan en Amazon Elastic Kubernetes Service (EKS) puedan aprovisionar volúmenes persistentes de bloques y archivos respaldados por ONTAP.

Requisitos

Además de"Requisitos de Trident" Para integrar FSx para ONTAP con Trident, necesitas:

- Un clúster de Amazon EKS existente o un clúster de Kubernetes autogestionado con `kubectl` instalado.
- Una máquina virtual de almacenamiento y sistema de archivos (SVM) Amazon FSx for NetApp ONTAP existente a la que se pueda acceder desde los nodos de trabajo de su clúster.
- Nodos de trabajo que están preparados para"NFS o iSCSI" .

> Asegúrese de seguir los pasos de preparación de nodos necesarios para Amazon Linux y Ubuntu. "Imágenes de máquinas de Amazon" (AMI) dependiendo de su tipo de AMI EKS.

Consideraciones

- Volúmenes SMB:
  - Los volúmenes SMB son compatibles mediante el uso de `ontap-nas` Solo el conductor.
  - Los volúmenes SMB no son compatibles con el complemento Trident EKS.
  - Trident solo admite volúmenes SMB montados en pods que se ejecutan en nodos Windows. Referirse a "Preparar el aprovisionamiento de volúmenes SMB" Para más detalles.
- Antes de Trident 24.02, los volúmenes creados en sistemas de archivos Amazon FSx que tenían habilitadas las copias de seguridad automáticas no podían ser eliminados por Trident. Para evitar este problema en Trident 24.02 o posterior, especifique el `fsxFilesystemID` AWS `apiRegion` AWS `apikey` y AWS `secretKey` en el archivo de configuración de backend para AWS FSx para ONTAP.

> Si especificas un rol de IAM para Trident, puedes omitir la especificación del `apiRegion` , `apiKey` , y `secretKey` campos a Trident explícitamente. Para obtener más información, consulte"Opciones de configuración y ejemplos de FSx para ONTAP" .

Uso simultáneo del controlador Trident SAN/iSCSI y EBS-CSI

Si planea usar controladores ontap-san (por ejemplo, iSCSI) con AWS (EKS, ROSA, EC2 o cualquier otra instancia), la configuración de múltiples rutas requerida en los nodos podría entrar en conflicto con el controlador CSI de Amazon Elastic Block Store (EBS). Para garantizar que las funciones de rutas múltiples funcionen sin interferir con los discos EBS en el mismo nodo, debe excluir EBS en su configuración de rutas múltiples. Este ejemplo muestra un `multipath.conf` Archivo que incluye la configuración necesaria de Trident excluyendo los discos EBS del multipathing:

```
defaults {
    find_multipaths no
}
blacklist {
    device {
        vendor "NVME"
        product "Amazon Elastic Block Store"
    }
}
```

Autenticación

Trident ofrece dos modos de autenticación.

- Basado en credenciales (recomendado): Almacena las credenciales de forma segura en AWS Secrets Manager. Puedes utilizar el `fsxadmin` usuario para su sistema de archivos o el `vsadmin` Usuario configurado para su SVM.

> Trident espera ser gestionado como un `vsadmin` Usuario de SVM o como usuario con un nombre diferente que tenga el mismo rol. Amazon FSx for NetApp ONTAP tiene un `fsxadmin` usuario que es un reemplazo limitado del ONTAP `admin` Usuario del clúster. Recomendamos encarecidamente utilizar `vsadmin` con Trident.

- Basado en certificados: Trident se comunicará con la SVM en su sistema de archivos FSx utilizando un certificado instalado en su SVM.

Para obtener detalles sobre cómo habilitar la autenticación, consulte la autenticación correspondiente a su tipo de controlador:

- "Autenticación NAS de ONTAP"
- "Autenticación SAN de ONTAP"

Imágenes de máquina de Amazon (AMI) probadas

El clúster EKS admite varios sistemas operativos, pero AWS ha optimizado ciertas imágenes de máquina de Amazon (AMI) para contenedores y EKS. Las siguientes AMI se han probado con NetApp Trident 25.02.

IAM	NAS	Economía NAS	iSCSI	Economía iSCSI
AL2023_x86_64_STANDARD	Sí	Sí	Sí	Sí
AL2_x86_64	Sí	Sí	Sí*	Sí*
BOTTLEROCKET_x86_64	Sí**	Sí	N/A	N/A
AL2023_ARM_64_STANDARD	Sí	Sí	Sí	Sí
AL2_ARM_64	Sí	Sí	Sí*	Sí*

BOTTLEROCKET_ARM_64	Sí**	Sí	N/A	N/A

- * No se puede eliminar el PV sin reiniciar el nodo
- ** No funciona con NFSv3 con Trident versión 25.02.

> Si la AMI que desea no aparece en esta lista, no significa que no sea compatible; simplemente significa que no se ha probado. Esta lista sirve de guía para las AMI que se sabe que funcionan.

Pruebas realizadas con:

- Versión de EKS: 1.32
- Método de instalación: Helm 25.06 y como complemento de AWS 25.06
- Para NAS se probaron tanto NFSv3 como NFSv4.1.
- Para SAN solo se probó iSCSI, no NVMe-oF.

Pruebas realizadas:

- Crear: Clase de almacenamiento, PVC, cápsula
- Eliminar: pod, pvc (normal, qtree/lun – económico, NAS con copia de seguridad de AWS)

Encuentra más información

- "Documentación de Amazon FSx for NetApp ONTAP"
- "Artículo de blog sobre Amazon FSx for NetApp ONTAP"

Crea un rol de IAM y un secreto de AWS

Puedes configurar los pods de Kubernetes para que accedan a los recursos de AWS autenticándose como un rol de AWS IAM en lugar de proporcionar credenciales explícitas de AWS.

> Para autenticarse utilizando un rol de AWS IAM, debe tener un clúster de Kubernetes implementado utilizando EKS.

Crear secreto de AWS Secrets Manager

Dado que Trident emitirá API contra un servidor virtual FSx para administrar el almacenamiento por usted, necesitará credenciales para hacerlo. La forma segura de transmitir esas credenciales es mediante un secreto de AWS Secrets Manager. Por lo tanto, si aún no tiene uno, deberá crear un secreto de AWS Secrets Manager que contenga las credenciales de la cuenta vsadmin.

Este ejemplo crea un secreto de AWS Secrets Manager para almacenar las credenciales de Trident CSI:

```
aws secretsmanager create-secret --name trident-secret --description
"Trident CSI credentials"\
  --secret-string
"{\"username\":\"vsadmin\",\"password\":\"<svmpassword>\"}"
```

Crear política de IAM

Trident también necesita permisos de AWS para funcionar correctamente. Por lo tanto, debe crear una política que otorgue a Trident los permisos que necesita.

Los siguientes ejemplos crean una política de IAM utilizando la CLI de AWS:

```
aws iam create-policy --policy-name AmazonFSxNCSIDriverPolicy --policy
-document file://policy.json
  --description "This policy grants access to Trident CSI to FSxN and
Secrets manager"
```

Ejemplo de JSON de política:

```
{
  "Statement": [
    {
      "Action": [
        "fsx:DescribeFileSystems",
        "fsx:DescribeVolumes",
        "fsx:CreateVolume",
        "fsx:RestoreVolumeFromSnapshot",
        "fsx:DescribeStorageVirtualMachines",
        "fsx:UntagResource",
        "fsx:UpdateVolume",
        "fsx:TagResource",
        "fsx:DeleteVolume"
      ],
      "Effect": "Allow",
      "Resource": "*"
    },
    {
      "Action": "secretsmanager:GetSecretValue",
      "Effect": "Allow",
      "Resource": "arn:aws:secretsmanager:<aws-region>:<aws-account-
id>:secret:<aws-secret-manager-name>*"
    }
  ],
  "Version": "2012-10-17"
}
```

Crear identidad de pod o rol de IAM para la asociación de cuenta de servicio (IRSA)

Puede configurar una cuenta de servicio de Kubernetes para que asuma un rol de AWS Identity and Access Management (IAM) con EKS Pod Identity o IAM role for Service account association (IRSA). Cualquier Pod que esté configurado para usar la cuenta de servicio podrá acceder a cualquier servicio de AWS al que el rol tenga permisos de acceso.

Identidad de pod

Las asociaciones de identidad de pods de Amazon EKS ofrecen la capacidad de administrar las credenciales de sus aplicaciones, de forma similar a como los perfiles de instancias de Amazon EC2 proporcionan credenciales a las instancias de Amazon EC2.

Instale Pod Identity en su clúster EKS:

Puede crear la identidad del Pod a través de la consola de AWS o utilizando el siguiente comando de la CLI de AWS:

```
aws eks create-addon --cluster-name <EKS_CLUSTER_NAME> --addon-name
eks-pod-identity-agent
```

Para obtener más información, consulte"Configurar el agente de identidad de pod de Amazon EKS" .

Crear trust-relationship.json:

Cree el archivo trust-relationship.json para permitir que la entidad de servicio de EKS asuma este rol para la identidad del pod. A continuación, cree un rol con esta política de confianza:

```
aws iam create-role \
  --role-name fsxn-csi-role --assume-role-policy-document file://trust-
relationship.json \
  --description "fsxn csi pod identity role"
```

archivo trust-relationship.json:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "pods.eks.amazonaws.com"
      },
      "Action": [
        "sts:AssumeRole",
        "sts:TagSession"
      ]
    }
  ]
}
```

Adjunte la política de rol al rol de IAM:

Asocie la política de rol del paso anterior al rol de IAM que se creó:

```
aws iam attach-role-policy \
  --policy-arn arn:aws:iam::aws:111122223333:policy/fsxn-csi-policy \
  --role-name fsxn-csi-role
```

Crear una asociación de identidad de pod:

Cree una asociación de identidad de pod entre el rol de IAM y la cuenta de servicio de Trident (trident-controller).

```
aws eks create-pod-identity-association \
  --cluster-name <EKS_CLUSTER_NAME> \
  --role-arn arn:aws:iam::111122223333:role/fsxn-csi-role \
  --namespace trident --service-account trident-controller
```

Rol de IAM para la asociación de cuentas de servicio (IRSA)

Utilizando la CLI de AWS:

```
aws iam create-role --role-name AmazonEKS_FSxN_CSI_DriverRole \
  --assume-role-policy-document file://trust-relationship.json
```

Archivo trust-relationship.json:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Federated": "arn:aws:iam::<account_id>:oidc-
provider/<oidc_provider>"
      },
      "Action": "sts:AssumeRoleWithWebIdentity",
      "Condition": {
        "StringEquals": {
          "<oidc_provider>:aud": "sts.amazonaws.com",
          "<oidc_provider>:sub":
"system:serviceaccount:trident:trident-controller"
        }
      }
    }
  ]
}
```

Actualiza los siguientes valores en el `trust-relationship.json` archivo:

- **<account_id>** - Tu ID de cuenta de AWS
- **<oidc_provider>** - El OIDC de su clúster EKS. Puede obtener el proveedor oidc ejecutando:

```
aws eks describe-cluster --name my-cluster --query
"cluster.identity.oidc.issuer"\
  --output text | sed -e "s/^https:\/\///"
```

Asocia el rol de IAM con la política de IAM:

Una vez creado el rol, adjúntele la política (creada en el paso anterior) mediante este comando:

```
aws iam attach-role-policy --role-name my-role --policy-arn <IAM policy
ARN>
```

Verificar que el proveedor de OICD esté asociado:

Verifique que su proveedor OIDC esté asociado con su clúster. Puedes verificarlo usando este comando:

```
aws iam list-open-id-connect-providers | grep $oidc_id | cut -d "/" -f4
```

Si la salida está vacía, utilice el siguiente comando para asociar IAM OIDC a su clúster:

```
eksctl utils associate-iam-oidc-provider --cluster $cluster_name
--approve
```

Si utiliza eksctl, siga el siguiente ejemplo para crear un rol de IAM para la cuenta de servicio en EKS:

```
eksctl create iamserviceaccount --name trident-controller --namespace
trident \
  --cluster <my-cluster> --role-name AmazonEKS_FSxN_CSI_DriverRole
--role-only \
  --attach-policy-arn <IAM-Policy ARN> --approve
```

Instalar Trident

Trident simplifica la gestión del almacenamiento de Amazon FSx for NetApp ONTAP en Kubernetes para que sus desarrolladores y administradores puedan centrarse en el despliegue de aplicaciones.

Puedes instalar Trident utilizando uno de los siguientes métodos:

- Timón
- Complemento EKS

Si desea utilizar la funcionalidad de instantáneas, instale el complemento CSI snapshot controller. Referirse a"Habilitar la funcionalidad de instantáneas para volúmenes CSI" Para más información.

Instala Trident mediante Helm.

Identidad de pod

1. Añadir el repositorio Trident Helm:

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

2. Instala Trident siguiendo el siguiente ejemplo:

```
helm install trident-operator netapp-trident/trident-operator --version 100.2502.1 --namespace trident --create-namespace
```

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

```
helm list -n trident
```

```
NAME                NAMESPACE   REVISION    UPDATED
STATUS       CHART                           APP VERSION

trident-operator    trident     1           2024-10-14
14:31:22.463122 +0300 IDT    deployed     trident-operator-
100.2502.0    25.02.0
```

Asociación de cuentas de servicio (IRSA)

1. Añadir el repositorio Trident Helm:

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

2. Establezca los valores para **proveedor de nube** e **identidad de nube**:

```
helm install trident-operator netapp-trident/trident-operator
--version 100.2502.1  \
--set cloudProvider="AWS" \
--set cloudIdentity="'eks.amazonaws.com/role-arn:
arn:aws:iam::<accountID>:role/<AmazonEKS_FSxN_CSI_DriverRole>'" \
--namespace trident \
--create-namespace
```

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

```
helm list -n trident
```

```
NAME                NAMESPACE   REVISION    UPDATED
STATUS       CHART                          APP VERSION

trident-operator    trident     1           2024-10-14
14:31:22.463122 +0300 IDT    deployed     trident-operator-
100.2506.0    25.06.0
```

Si planea utilizar iSCSI, asegúrese de que iSCSI esté habilitado en su máquina cliente. Si utiliza el sistema operativo del nodo de trabajo AL2023, puede automatizar la instalación del cliente iSCSI agregando el parámetro node prep en la instalación de helm:

```
helm install trident-operator netapp-trident/trident-operator
--version 100.2502.1 --namespace trident --create-namespace --
set nodePrep={iscsi}
```

Instala Trident mediante el complemento EKS.

El complemento Trident EKS incluye los últimos parches de seguridad, correcciones de errores y está validado por AWS para funcionar con Amazon EKS. El complemento EKS le permite garantizar de forma consistente que sus clústeres de Amazon EKS sean seguros y estables, y reduce la cantidad de trabajo que necesita realizar para instalar, configurar y actualizar complementos.

Prerrequisitos

Asegúrese de tener lo siguiente antes de configurar el complemento Trident para AWS EKS:

- Una cuenta de clúster de Amazon EKS con suscripción adicional
- Permisos de AWS para el mercado de AWS:
  `"aws-marketplace:ViewSubscriptions",`
  `"aws-marketplace:Subscribe",`
  `"aws-marketplace:Unsubscribe`
- Tipo de AMI: Amazon Linux 2 (AL2_x86_64) o Amazon Linux 2 Arm (AL2_ARM_64)
- Tipo de nodo: AMD o ARM
- Un sistema de archivos Amazon FSx for NetApp ONTAP existente

Habilita el complemento Trident para AWS

Consola de administración

1. Abra la consola de Amazon EKS en https://console.aws.amazon.com/eks/home#/clusters .
2. En el panel de navegación izquierdo, seleccione **Clústeres**.
3. Seleccione el nombre del clúster para el que desea configurar el complemento NetApp Trident CSI.
4. Seleccione **Complementos** y luego seleccione **Obtener más complementos**.
5. Siga estos pasos para seleccionar el complemento:
   a. Desplácese hacia abajo hasta la sección **Complementos de AWS Marketplace** y escriba **"Trident"** en el cuadro de búsqueda.
   b. Seleccione la casilla de verificación en la esquina superior derecha del cuadro Trident by NetApp .
   c. Seleccione **Siguiente**.
6. En la página de configuración **Configurar complementos seleccionados**, haga lo siguiente:

   Omite estos pasos si estás usando la asociación de identidad de pod.

   a. Seleccione la **Versión** que desea utilizar.
   b. Si utiliza la autenticación IRSA, asegúrese de configurar los valores disponibles en la configuración opcional:
      - Seleccione la **Versión** que desea utilizar.
      - Siga el **Esquema de configuración del complemento** y configure el parámetro **configurationValues** en la sección **Valores de configuración** con el ARN del rol que creó en el paso anterior (el valor debe tener el siguiente formato):

```
{

  "cloudIdentity": "'eks.amazonaws.com/role-arn: <role ARN>'",
  "cloudProvider": "AWS"

}
```

   +
   Si selecciona Anular como método de resolución de conflictos, una o más de las configuraciones del complemento existente pueden sobrescribirse con la configuración del complemento de Amazon EKS. Si no habilita esta opción y hay un conflicto con su configuración existente, la operación fallará. Puede utilizar el mensaje de error resultante para solucionar el conflicto. Antes de seleccionar esta opción, asegúrese de que el complemento Amazon EKS no gestione configuraciones que usted deba administrar manualmente.

7. Seleccione **Siguiente**.
8. En la página **Revisar y agregar**, seleccione **Crear**.

   Una vez finalizada la instalación del complemento, verá el complemento instalado.

CLI de AWS

1. Crea el `add-on.json` archivo:

Para la identidad del pod, utilice el siguiente formato:

```
{
  "clusterName": "<eks-cluster>",
  "addonName": "netapp_trident-operator",
  "addonVersion": "v25.6.0-eksbuild.1",
}
```

Para la autenticación IRSA, utilice el siguiente formato:

```
{
  "clusterName": "<eks-cluster>",
  "addonName": "netapp_trident-operator",
  "addonVersion": "v25.6.0-eksbuild.1",
  "serviceAccountRoleArn": "<role ARN>",
  "configurationValues": {
    "cloudIdentity": "'eks.amazonaws.com/role-arn: <role ARN>'",
    "cloudProvider": "AWS"
  }
}
```

Reemplazar `<role ARN>` con el ARN del rol que se creó en el paso anterior.

2. Instala el complemento Trident EKS.

```
aws eks create-addon --cli-input-json file://add-on.json
```

eksctl

El siguiente comando de ejemplo instala el complemento Trident EKS:

```
eksctl create addon --name netapp_trident-operator --cluster
<cluster_name> --force
```

Actualizar el complemento Trident EKS

Consola de administración

1. Abra la consola de Amazon EKS https://console.aws.amazon.com/eks/home#/clusters .
2. En el panel de navegación izquierdo, seleccione **Clústeres**.
3. Seleccione el nombre del clúster para el que desea actualizar el complemento NetApp Trident CSI.
4. Seleccione la pestaña **Complementos**.
5. Seleccione * Trident by NetApp* y luego seleccione **Editar**.
6. En la página **Configurar Trident de NetApp**, haga lo siguiente:
   a. Seleccione la **Versión** que desea utilizar.
   b. Amplíe la sección **Ajustes de configuración opcionales** y modifíquela según sea necesario.
   c. Seleccione **Guardar cambios**.

CLI de AWS

El siguiente ejemplo actualiza el complemento EKS:

```
aws eks update-addon --cluster-name <eks_cluster_name> --addon-name
netapp_trident-operator --addon-version v25.6.0-eksbuild.1 \
  --service-account-role-arn <role-ARN> --resolve-conflict preserve \
  --configuration-values "{\"cloudIdentity\":
\"'eks.amazonaws.com/role-arn: <role ARN>'\"}"
```

eksctl

- Comprueba la versión actual de tu complemento FSxN Trident CSI. Reemplazar `my-cluster` con el nombre de su clúster.

  ```
  eksctl get addon --name netapp_trident-operator --cluster my-cluster
  ```

  Ejemplo de salida:

```
NAME                          VERSION              STATUS    ISSUES
IAMROLE    UPDATE AVAILABLE    CONFIGURATION VALUES
netapp_trident-operator    v25.6.0-eksbuild.1    ACTIVE    0
{"cloudIdentity":"'eks.amazonaws.com/role-arn:
arn:aws:iam::139763910815:role/AmazonEKS_FSXN_CSI_DriverRole'"}
```

- Actualiza el complemento a la versión que se muestra en ACTUALIZACIÓN DISPONIBLE en el resultado del paso anterior.

  ```
  eksctl update addon --name netapp_trident-operator --version
  v25.6.0-eksbuild.1 --cluster my-cluster --force
  ```

Si eliminas el `--force` Si alguna de las opciones y alguna de las configuraciones del complemento de Amazon EKS entra en conflicto con su configuración existente, la actualización del complemento de Amazon EKS fallará; recibirá un mensaje de error para ayudarle a resolver el conflicto. Antes de especificar esta opción, asegúrese de que el complemento Amazon EKS no gestione configuraciones que usted necesite administrar, ya que estas configuraciones se sobrescriben con esta opción. Para obtener más información sobre otras opciones para esta configuración, consulte"Complementos" . Para obtener más información sobre la administración de campos de Amazon EKS Kubernetes, consulte"Gestión de campos de Kubernetes" .

Desinstala/elimina el complemento Trident EKS.

Tienes dos opciones para eliminar un complemento de Amazon EKS:

- **Conservar el software complementario en su clúster** – Esta opción elimina la administración de Amazon EKS de cualquier configuración. También elimina la capacidad de Amazon EKS para notificarle sobre actualizaciones y actualizar automáticamente el complemento de Amazon EKS después de que usted inicie una actualización. Sin embargo, conserva el software adicional en su clúster. Esta opción convierte el complemento en una instalación autogestionada, en lugar de un complemento de Amazon EKS. Con esta opción, no hay tiempo de inactividad para el complemento. Conservar el `--preserve` opción en el comando para conservar el complemento.
- **Elimine por completo el software complementario de su clúster** – NetApp recomienda que elimine el complemento Amazon EKS de su clúster solo si no hay recursos en su clúster que dependan de él. Quitar el `--preserve` opción de la `delete` comando para eliminar el complemento.

Si el complemento tiene una cuenta IAM asociada, dicha cuenta no se eliminará.

Consola de administración

1. Abra la consola de Amazon EKS en https://console.aws.amazon.com/eks/home#/clusters .
2. En el panel de navegación izquierdo, seleccione **Clústeres**.
3. Seleccione el nombre del clúster para el que desea eliminar el complemento NetApp Trident CSI.
4. Seleccione la pestaña **Complementos** y luego seleccione * Trident de NetApp*.
5. Seleccione **Eliminar**.
6. En el cuadro de diálogo **Eliminar confirmación de netapp_trident-operator**, haga lo siguiente:
   a. Si desea que Amazon EKS deje de administrar la configuración del complemento, seleccione **Conservar en el clúster**. Haz esto si quieres conservar el software adicional en tu clúster para poder gestionar tú mismo todas las configuraciones del complemento.
   b. Introduzca **netapp_trident-operator**.
   c. Seleccione **Eliminar**.

CLI de AWS

Reemplazar `my-cluster` con el nombre de su clúster y, a continuación, ejecute el siguiente comando.

```
aws eks delete-addon --cluster-name my-cluster --addon-name
netapp_trident-operator --preserve
```

eksctl

El siguiente comando desinstala el complemento Trident EKS:

```
eksctl delete addon --cluster K8s-arm --name netapp_trident-operator
```

Configurar el backend de almacenamiento

Integración de controladores SAN y NAS de ONTAP

Para crear un backend de almacenamiento, necesitas crear un archivo de configuración en formato JSON o YAML. El archivo debe especificar el tipo de almacenamiento que desea (NAS o SAN), el sistema de archivos, la SVM de la que se obtendrá y cómo autenticarse con ella. El siguiente ejemplo muestra cómo definir el almacenamiento basado en NAS y cómo usar un secreto de AWS para almacenar las credenciales de la SVM que desea usar:

YAML

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-ontap-nas
  namespace: trident
spec:
  version: 1
  storageDriverName: ontap-nas
  backendName: tbc-ontap-nas
  svm: svm-name
  aws:
    fsxFilesystemID: fs-xxxxxxxxxx
  credentials:
    name: "arn:aws:secretsmanager:us-west-2:xxxxxxxx:secret:secret-
name"
    type: awsarn
```

JSON

```
{
  "apiVersion": "trident.netapp.io/v1",
  "kind": "TridentBackendConfig",
  "metadata": {
    "name": "backend-tbc-ontap-nas"
    "namespace": "trident"
  },
  "spec": {
    "version": 1,
    "storageDriverName": "ontap-nas",
    "backendName": "tbc-ontap-nas",
    "svm": "svm-name",
    "aws": {
      "fsxFilesystemID": "fs-xxxxxxxxxx"
    },
    "managementLIF": null,
    "credentials": {
      "name": "arn:aws:secretsmanager:us-west-2:xxxxxxxx:secret:secret-
name",
      "type": "awsarn"
    }
  }
}
```

Ejecute los siguientes comandos para crear y validar la configuración del backend de Trident (TBC):

- Crea la configuración del backend de Trident (TBC) a partir del archivo YAML y ejecuta el siguiente comando:

```
kubectl create -f backendconfig.yaml -n trident
```

```
tridentbackendconfig.trident.netapp.io/backend-tbc-ontap-nas created
```

- Validar que la configuración del backend de Trident (TBC) se creó correctamente:

```
Kubectl get tbc -n trident
```

```
NAME                           BACKEND NAME          BACKEND UUID
PHASE   STATUS

backend-tbc-ontap-nas          tbc-ontap-nas         933e0071-66ce-4324-
b9ff-f96d916ac5e9   Bound   Success
```

Detalles del controlador FSx para ONTAP

Puede integrar Trident con Amazon FSx for NetApp ONTAP utilizando los siguientes controladores:

- `ontap-san`Cada PV aprovisionado es un LUN dentro de su propio volumen Amazon FSx for NetApp ONTAP . Recomendado para almacenamiento en bloque.
- `ontap-nas`Cada PV aprovisionado es un volumen completo de Amazon FSx for NetApp ONTAP . Recomendado para NFS y SMB.
- `ontap-san-economy`Cada PV aprovisionado es un LUN con un número configurable de LUN por volumen de Amazon FSx for NetApp ONTAP .
- `ontap-nas-economy`Cada PV aprovisionado es un qtree, con un número configurable de qtrees por volumen de Amazon FSx for NetApp ONTAP .
- `ontap-nas-flexgroup`Cada PV aprovisionado es un volumen completo de Amazon FSx for NetApp ONTAP FlexGroup .

Para obtener detalles sobre el conductor, consulte"Controladores NAS" y"Controladores SAN" .

Una vez creado el archivo de configuración, ejecute este comando para crearlo en su EKS:

```
kubectl create -f configuration_file
```

Para verificar el estado, ejecute este comando:

```
kubectl get tbc -n trident
```

```
NAME                     BACKEND NAME            BACKEND UUID
PHASE   STATUS
backend-fsx-ontap-nas   backend-fsx-ontap-nas   7a551921-997c-4c37-a1d1-
f2f4c87fa629   Bound   Success
```

Configuración avanzada del backend y ejemplos

Consulte la siguiente tabla para ver las opciones de configuración del backend:

Parámetro	Descripción	Ejemplo
`version`		Siempre 1
`storageDriverName`	Nombre del controlador de almacenamiento	`ontap-nas`, `ontap-nas-economy` , `ontap-nas-flexgroup` , `ontap-san` , `ontap-san-economy`
`backendName`	Nombre personalizado o el backend de almacenamiento	Nombre del controlador + "_" + dataLIF
`managementLIF`	Dirección IP de un clúster o LIF de administración de SVM Se puede especificar un nombre de dominio completo (FQDN). Se puede configurar para usar direcciones IPv6 si Trident se instaló usando la bandera IPv6. Las direcciones IPv6 deben definirse entre corchetes, como por ejemplo [28e8:d9fb:a825:b7bf:69a8:d02f:9e7b:3555]. Si usted proporciona el `fsxFilesystemID` bajo el `aws` campo, no es necesario que proporcione el `managementLIF` porque Trident recupera la SVM `managementLIF` Información de AWS. Por lo tanto, debe proporcionar las credenciales de un usuario en la SVM (por ejemplo: vsadmin) y el usuario debe tener `vsadmin` role.	"10.0.0.1", "[2001:1234:abcd::fefe]"

Parámetro	Descripción	Ejemplo
`dataLIF`	Dirección IP del protocolo LIF. * Controladores NAS ONTAP **: NetApp recomienda especificar dataLIF. Si no se proporcionan, Trident obtiene los dataLIF del SVM. Puede especificar un nombre de dominio completo (FQDN) para usarlo en las operaciones de montaje NFS, lo que le permite crear un DNS round-robin para equilibrar la carga en múltiples dataLIF. Puede modificarse después de la configuración inicial. Referirse a . * Controladores ONTAP SAN**: No especificar para iSCSI. Trident utiliza ONTAP Selective LUN Map para descubrir los LIF iSCI necesarios para establecer una sesión de múltiples rutas. Se genera una advertencia si dataLIF se define explícitamente. Se puede configurar para usar direcciones IPv6 si Trident se instaló usando la bandera IPv6. Las direcciones IPv6 deben definirse entre corchetes, como por ejemplo [28e8:d9fb:a825:b7bf:69a8:d02f:9e7b:3555].	
`autoExportPolicy`	Habilitar la creación y actualización automática de políticas de exportación [Booleano]. Utilizando el `autoExportPolicy` y `autoExportCIDRs` Con algunas opciones, Trident puede gestionar las políticas de exportación automáticamente.	`false`
`autoExportCIDRs`	Lista de CIDR para filtrar las direcciones IP de los nodos de Kubernetes cuando `autoExportPolicy` está habilitado. Utilizando el `autoExportPolicy` y `autoExportCIDRs` Con algunas opciones, Trident puede gestionar las políticas de exportación automáticamente.	"["0.0.0.0/0", "::/0"]"
`labels`	Conjunto de etiquetas arbitrarias con formato JSON para aplicar a los volúmenes	""

Parámetro	Descripción	Ejemplo
`clientCertificate`	Valor codificado en Base64 del certificado del cliente. Se utiliza para la autenticación basada en certificados.	""
`clientPrivateKey`	Valor codificado en Base64 de la clave privada del cliente. Se utiliza para la autenticación basada en certificados.	""
`trustedCACertificate`	Valor codificado en Base64 del certificado de CA de confianza. Opcional. Se utiliza para la autenticación basada en certificados.	""
`username`	Nombre de usuario para conectarse al clúster o SVM. Se utiliza para la autenticación basada en credenciales. Por ejemplo, vsadmin.	
`password`	Contraseña para conectarse al clúster o SVM. Se utiliza para la autenticación basada en credenciales.	
`svm`	máquina virtual de almacenamiento a utilizar	Se deriva si se especifica un SVM managementLIF.
`storagePrefix`	Prefijo utilizado al aprovisionar nuevos volúmenes en la SVM. No se puede modificar después de su creación. Para actualizar este parámetro, deberá crear un nuevo backend.	`trident`
`limitAggregateUsage`	**No especificar para Amazon FSx for NetApp ONTAP.** El proporcionado `fsxadmin` y `vsadmin` No contienen los permisos necesarios para recuperar el uso agregado y limitarlo mediante Trident.	No utilizar.
`limitVolumeSize`	Fallará el aprovisionamiento si el tamaño de volumen solicitado supera este valor. También restringe el tamaño máximo de los volúmenes que administra para qtrees y LUN, y el `qtreesPerFlexvol` Esta opción permite personalizar el número máximo de qtrees por FlexVol volume.	" (no se aplica por defecto)

Parámetro	Descripción	Ejemplo
`lunsPerFlexvol`	El número máximo de LUN por volumen de Flexvol debe estar en el rango [50, 200]. Solo SAN.	“100”
`debugTraceFlags`	Indicadores de depuración para usar al solucionar problemas. Ejemplo: {"api":false, "method":true} No usar `debugTraceFlags` a menos que esté solucionando problemas y necesite un registro detallado.	nulo
`nfsMountOptions`	Lista de opciones de montaje NFS separadas por comas. Las opciones de montaje para volúmenes persistentes de Kubernetes normalmente se especifican en las clases de almacenamiento, pero si no se especifican opciones de montaje en una clase de almacenamiento, Trident recurrirá a las opciones de montaje especificadas en el archivo de configuración del backend de almacenamiento. Si no se especifican opciones de montaje en la clase de almacenamiento o en el archivo de configuración, Trident no establecerá ninguna opción de montaje en un volumen persistente asociado.	""
`nasType`	Configure la creación de volúmenes NFS o SMB. Las opciones son `nfs` , `smb` , o nulo. **Debe configurarse en `smb` para volúmenes SMB.** Si se establece en nulo, se utilizarán volúmenes NFS por defecto.	`nfs`
`qtreesPerFlexvol`	Número máximo de Qtrees por FlexVol volume, debe estar en el rango [50, 300]	`"200"`
`smbShare`	Puede especificar una de las siguientes opciones: el nombre de un recurso compartido SMB creado mediante la Consola de administración de Microsoft o la CLI de ONTAP , o un nombre para permitir que Trident cree el recurso compartido SMB. Este parámetro es necesario para los backends de Amazon FSx para ONTAP .	`smb-share`

Parámetro	Descripción	Ejemplo
`useREST`	Parámetro booleano para utilizar las API REST de ONTAP . Cuando se configura para `true` Trident utilizará las API REST de ONTAP para comunicarse con el backend. Esta función requiere ONTAP 9.11.1 y versiones posteriores. Además, el rol de inicio de sesión de ONTAP utilizado debe tener acceso a `ontap` solicitud. Esto se satisface mediante lo predefinido. `vsadmin` y `cluster-admin` roles.	`false`
`aws`	En el archivo de configuración de AWS FSx para ONTAP puede especificar lo siguiente: `fsxFilesystemID` : Especifique el ID del sistema de archivos AWS FSx. - `apiRegion` Nombre de la región de la API de AWS. - `apikey` Clave de API de AWS. - `secretKey` Clave secreta de AWS.	"" "" ""
`credentials`	Especifique las credenciales de FSx SVM que se almacenarán en AWS Secrets Manager. - `name` : Nombre de recurso de Amazon (ARN) del secreto, que contiene las credenciales de SVM. - `type` : Configurado a `awsarn` . Referirse a"Crea un secreto de AWS Secrets Manager" Para más información.	

Opciones de configuración de backend para el aprovisionamiento de volúmenes

Puedes controlar el aprovisionamiento predeterminado utilizando estas opciones en el `defaults` sección de la configuración. Para ver un ejemplo, consulte los ejemplos de configuración a continuación.

Parámetro	Descripción	Por defecto
`spaceAllocation`	Asignación de espacio para LUN	`true`
`spaceReserve`	Modo de reserva de espacio: "ninguno" (delgado) o "volumen" (grueso).	`none`
`snapshotPolicy`	Política de instantáneas a utilizar	`none`

Parámetro	Descripción	Por defecto
`qosPolicy`	Grupo de políticas QoS que se asignará a los volúmenes creados. Elija una de las opciones qosPolicy o adaptiveQosPolicy por grupo de almacenamiento o backend. El uso de grupos de políticas QoS con Trident requiere ONTAP 9.8 o posterior. Debe utilizar un grupo de políticas QoS no compartido y asegurarse de que el grupo de políticas se aplique a cada componente individualmente. Un grupo de políticas QoS compartidas impone un límite máximo al rendimiento total de todas las cargas de trabajo.	""
`adaptiveQosPolicy`	Grupo de políticas QoS adaptativas para asignar a los volúmenes creados. Elija una de las opciones qosPolicy o adaptiveQosPolicy por grupo de almacenamiento o backend. No compatible con ontap-nas-economy.	""
`snapshotReserve`	Porcentaje de volumen reservado para instantáneas "0"	Si `snapshotPolicy` es `none` , `else` ""
`splitOnClone`	Separar un clon de su progenitor al crearlo	`false`
`encryption`	Habilite el cifrado de volumen de NetApp (NVE) en el nuevo volumen; el valor predeterminado es `false` . Para utilizar esta opción, NVE debe estar licenciado y habilitado en el clúster. Si NAE está habilitado en el backend, cualquier volumen aprovisionado en Trident tendrá NAE habilitado. Para obtener más información, consulte:"Cómo funciona Trident con NVE y NAE" .	`false`
`luksEncryption`	Habilitar el cifrado LUKS. Referirse a"Utilice la configuración de clave unificada de Linux (LUKS)." . Solo SAN.	""
`tieringPolicy`	Política de niveles a utilizar `none`	
`unixPermissions`	Modo para nuevos volúmenes. **Dejar en blanco para volúmenes SMB.**	""

Parámetro	Descripción	Por defecto
`securityStyle`	Estilo de seguridad para nuevos volúmenes. NFS admite `mixed` y `unix` Estilos de seguridad. Las PYMES son compatibles con el soporte. `mixed` y `ntfs` Estilos de seguridad.	El valor predeterminado de NFS es `unix` . El valor predeterminado de SMB es `ntfs` .

Preparar el aprovisionamiento de volúmenes SMB

Puede aprovisionar volúmenes SMB utilizando `ontap-nas` conductor. Antes de completarIntegración de controladores SAN y NAS de ONTAP Complete los siguientes pasos.

Antes de empezar

Antes de poder aprovisionar volúmenes SMB utilizando el `ontap-nas` Conductor, usted debe tener lo siguiente.

- Un clúster de Kubernetes con un nodo controlador Linux y al menos un nodo de trabajo Windows que ejecuta Windows Server 2019. Trident solo admite volúmenes SMB montados en pods que se ejecutan en nodos Windows.
- Al menos un secreto de Trident que contenga sus credenciales de Active Directory. Para generar secretos `smbcreds` :

```
kubectl create secret generic smbcreds --from-literal username=user
--from-literal password='password'
```

- Un proxy CSI configurado como servicio de Windows. Para configurar un `csi-proxy` , consulte a"GitHub: Proxy CSI" o"GitHub: Proxy CSI para Windows" para nodos de Kubernetes que se ejecutan en Windows.

Pasos

1. Crear recursos compartidos SMB. Puedes crear los recursos compartidos de administración SMB de dos maneras: utilizando…"Consola de administración de Microsoft" Complemento de carpetas compartidas o mediante la CLI de ONTAP . Para crear los recursos compartidos SMB mediante la CLI de ONTAP :
   a. Si es necesario, cree la estructura de rutas de directorio para el recurso compartido.

      El `vserver cifs share create` El comando verifica la ruta especificada en la opción -path durante la creación del recurso compartido. Si la ruta especificada no existe, el comando falla.

   b. Cree un recurso compartido SMB asociado con la SVM especificada:

```
vserver cifs share create -vserver vserver_name -share-name
share_name -path path [-share-properties share_properties,...]
[other_attributes] [-comment text]
```

   c. Verifique que se haya creado el recurso compartido:

```
vserver cifs share show -share-name share_name
```

Referirse a"Crear un recurso compartido SMB" Para más detalles.

2. Al crear el backend, debe configurar lo siguiente para especificar los volúmenes SMB. Para conocer todas las opciones de configuración del backend de FSx para ONTAP , consulte"Opciones de configuración y ejemplos de FSx para ONTAP" .

Parámetro	Descripción	Ejemplo
`smbShare`	Puede especificar una de las siguientes opciones: el nombre de un recurso compartido SMB creado mediante la Consola de administración de Microsoft o la CLI de ONTAP , o un nombre para permitir que Trident cree el recurso compartido SMB. Este parámetro es necesario para los backends de Amazon FSx para ONTAP .	`smb-share`
`nasType`	**Debe configurarse en `smb` .** Si es nulo, el valor predeterminado es `nfs` .	`smb`
`securityStyle`	Estilo de seguridad para nuevos volúmenes. **Debe configurarse en `ntfs` o `mixed` para volúmenes SMB.**	`ntfs`o `mixed` para volúmenes SMB
`unixPermissions`	Modo para nuevos volúmenes. **Debe dejarse vacío para volúmenes SMB.**	""

Configure una clase de almacenamiento y un PVC

Configure un objeto StorageClass de Kubernetes y cree la clase de almacenamiento para indicar a Trident cómo aprovisionar volúmenes. Cree un PersistentVolumeClaim (PVC) que utilice la StorageClass de Kubernetes configurada para solicitar acceso al PV. Luego puedes montar el panel fotovoltaico en un soporte.

Crear una clase de almacenamiento

Configurar un objeto StorageClass de Kubernetes

El "Objeto StorageClass de Kubernetes" El objeto identifica a Trident como el proveedor que se utiliza para esa clase e indica a Trident cómo aprovisionar un volumen. Utilice este ejemplo para configurar Storageclass para volúmenes que utilizan NFS (consulte la sección Atributos de Trident a continuación para obtener la lista completa de atributos):

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontap-gold
provisioner: csi.trident.netapp.io
parameters:
  backendType: "ontap-nas"
  provisioningType: "thin"
  snapshots: "true"
```

Utilice este ejemplo para configurar Storageclass para volúmenes que utilizan iSCSI:

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontap-gold
provisioner: csi.trident.netapp.io
parameters:
  backendType: "ontap-san"
  provisioningType: "thin"
  snapshots: "true"
```

Para aprovisionar volúmenes NFSv3 en AWS Bottlerocket, agregue los necesarios. `mountOptions` a la clase de almacenamiento:

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontap-gold
provisioner: csi.trident.netapp.io
parameters:
  backendType: "ontap-nas"
  media: "ssd"
  provisioningType: "thin"
  snapshots: "true"
mountOptions:
  - nfsvers=3
  - nolock
```

Referirse a"Objetos de Kubernetes y Trident" Para obtener detalles sobre cómo interactúan las clases de almacenamiento con `PersistentVolumeClaim` y parámetros para controlar cómo Trident gestiona los volúmenes.

Crear una clase de almacenamiento

Pasos

1. Este es un objeto de Kubernetes, así que use `kubectl` para crearlo en Kubernetes.

```
kubectl create -f storage-class-ontapnas.yaml
```

2. Ahora debería ver una clase de almacenamiento **basic-csi** tanto en Kubernetes como en Trident, y Trident debería haber detectado los pools en el backend.

```
kubectl get sc basic-csi
```

```
NAME         PROVISIONER             AGE
basic-csi    csi.trident.netapp.io   15h
```

Crea el PVC

A "*Reclamación de volumen persistente*" (PVC) es una solicitud de acceso al PersistentVolume en el clúster.

El PVC se puede configurar para solicitar el almacenamiento de un tamaño o modo de acceso determinado. Mediante la StorageClass asociada, el administrador del clúster puede controlar más que el tamaño del PersistentVolume y el modo de acceso, como el rendimiento o el nivel de servicio.

Una vez creado el PVC, puede montar el volumen en un soporte.

Ejemplos de manifiestos

Muestra de PersistentVolumeClaim

Estos ejemplos muestran opciones básicas de configuración de PVC.

PVC con acceso RWX

Este ejemplo muestra un PVC básico con acceso RWX que está asociado a una StorageClass llamada `basic-csi` .

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: pvc-storage
spec:
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 1Gi
  storageClassName: ontap-gold
```

Ejemplo de PVC usando iSCSI

Este ejemplo muestra un PVC básico para iSCSI con acceso RWO que está asociado con una StorageClass llamada `protection-gold` .

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
name: pvc-san
spec:
accessModes:
  - ReadWriteOnce
resources:
  requests:
    storage: 1Gi
storageClassName: protection-gold
```

Crear PVC

Pasos

1. Crear el PVC.

```
kubectl create -f pvc.yaml
```

2. Verifique el estado del PVC.

```
kubectl get pvc
```

```
NAME          STATUS VOLUME     CAPACITY ACCESS MODES STORAGECLASS AGE
pvc-storage Bound  pv-name 2Gi      RWO                           5m
```

Referirse a"Objetos de Kubernetes y Trident" Para obtener detalles sobre cómo interactúan las clases de almacenamiento con `PersistentVolumeClaim` y parámetros para controlar cómo Trident gestiona los volúmenes.

Atributos del Trident

Estos parámetros determinan qué pools de almacenamiento gestionados por Trident deben utilizarse para aprovisionar volúmenes de un tipo determinado.

Atributo	Tipo	Valores	Oferta	Pedido	Con el apoyo de
medios[1]	cadena	disco duro, híbrido, SSD	La piscina contiene medios de este tipo; híbrido significa ambos	Tipo de medio especificado	ontap-nas, ontap-nas-economy, ontap-nas-flexgroup, ontap-san, solidfire-san
tipo de aprovisionamiento	cadena	delgado, grueso	Pool admite este método de aprovisionamiento.	Método de aprovisionamiento especificado	Espeso: todo Ontap; fino: todo Ontap y Solidfire-san
Tipo de backend	cadena	ontap-nas, ontap-nas-economy, ontap-nas-flexgroup, ontap-san, solidfire-san, gcp-cvs, azure-netapp-files, ontap-san-economy	Pool pertenece a este tipo de backend.	Backend especificado	Todos los conductores
instantáneas	bool	verdadero, falso	El pool admite volúmenes con instantáneas.	Volumen con instantáneas habilitadas	ontap-nas, ontap-san, solidfire-san, gcp-cvs
clones	bool	verdadero, falso	Pool admite la clonación de volúmenes.	Volumen con clones habilitado	ontap-nas, ontap-san, solidfire-san, gcp-cvs

Atributo	Tipo	Valores	Oferta	Pedido	Con el apoyo de
cifrado	bool	verdadero, falso	Pool admite volúmenes cifrados	Volumen con cifrado habilitado	ontap-nas, ontap-nas-economy, ontap-nas-flexgroups, ontap-san
IOPS	int	entero positivo	Pool es capaz de garantizar IOPS en este rango.	El volumen garantizaba estas IOPS	solidfire-san

[1]: No compatible con los sistemas ONTAP Select

Implementar aplicación de muestra

Una vez creadas la clase de almacenamiento y el PVC, puede montar el PV en un pod. Esta sección enumera el comando de ejemplo y la configuración para adjuntar el PV a un pod.

Pasos

1. Monte el volumen en una cápsula.

```
kubectl create -f pv-pod.yaml
```

Estos ejemplos muestran configuraciones básicas para conectar el PVC a un pod: **Configuración básica**:

```
kind: Pod
apiVersion: v1
metadata:
  name: pv-pod
spec:
  volumes:
    - name: pv-storage
      persistentVolumeClaim:
       claimName: basic
  containers:
    - name: pv-container
      image: nginx
      ports:
        - containerPort: 80
          name: "http-server"
      volumeMounts:
        - mountPath: "/my/mount/path"
          name: pv-storage
```

Puedes supervisar el progreso usando `kubectl get pod --watch`.

2. Verifique que el volumen esté montado en `/my/mount/path`.

```
kubectl exec -it pv-pod -- df -h /my/mount/path
```

```
Filesystem                                                      Size
Used  Avail Use%  Mounted on
192.168.188.78:/trident_pvc_ae45ed05_3ace_4e7c_9080_d2a83ae03d06  1.1G
320K  1.0G   1%   /my/mount/path
```

Ahora puedes eliminar el Pod. La aplicación Pod dejará de existir, pero el volumen se mantendrá.

```
kubectl delete pod pv-pod
```

Configure el complemento Trident EKS en un clúster EKS.

NetApp Trident simplifica la gestión del almacenamiento de Amazon FSx for NetApp ONTAP en Kubernetes para que sus desarrolladores y administradores puedan centrarse en el despliegue de aplicaciones. El complemento NetApp Trident EKS incluye los últimos parches de seguridad, correcciones de errores y está validado por AWS para funcionar con Amazon EKS. El complemento EKS le permite garantizar de forma consistente que sus clústeres de Amazon EKS sean seguros y estables, y reduce la cantidad de trabajo que necesita realizar para instalar, configurar y actualizar complementos.

Prerrequisitos

Asegúrese de tener lo siguiente antes de configurar el complemento Trident para AWS EKS:

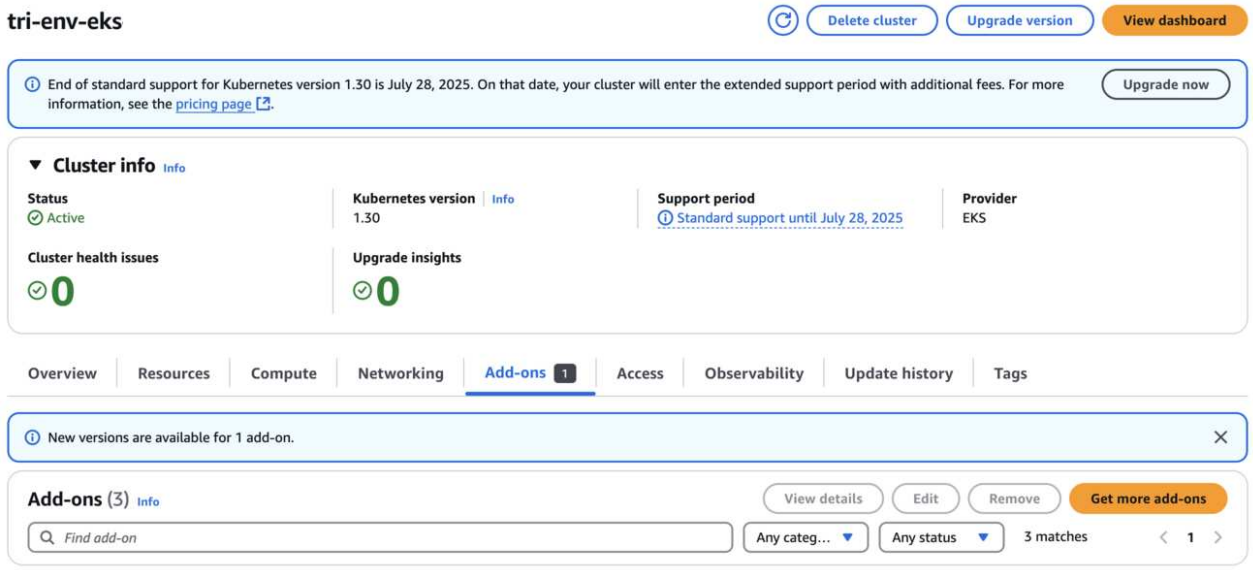
- Una cuenta de clúster de Amazon EKS con permisos para trabajar con complementos. Referirse a"Complementos de Amazon EKS" .
- Permisos de AWS para el mercado de AWS:
  `"aws-marketplace:ViewSubscriptions",`
  `"aws-marketplace:Subscribe",`
  `"aws-marketplace:Unsubscribe`
- Tipo de AMI: Amazon Linux 2 (AL2_x86_64) o Amazon Linux 2 Arm (AL2_ARM_64)
- Tipo de nodo: AMD o ARM
- Un sistema de archivos Amazon FSx for NetApp ONTAP existente

Pasos

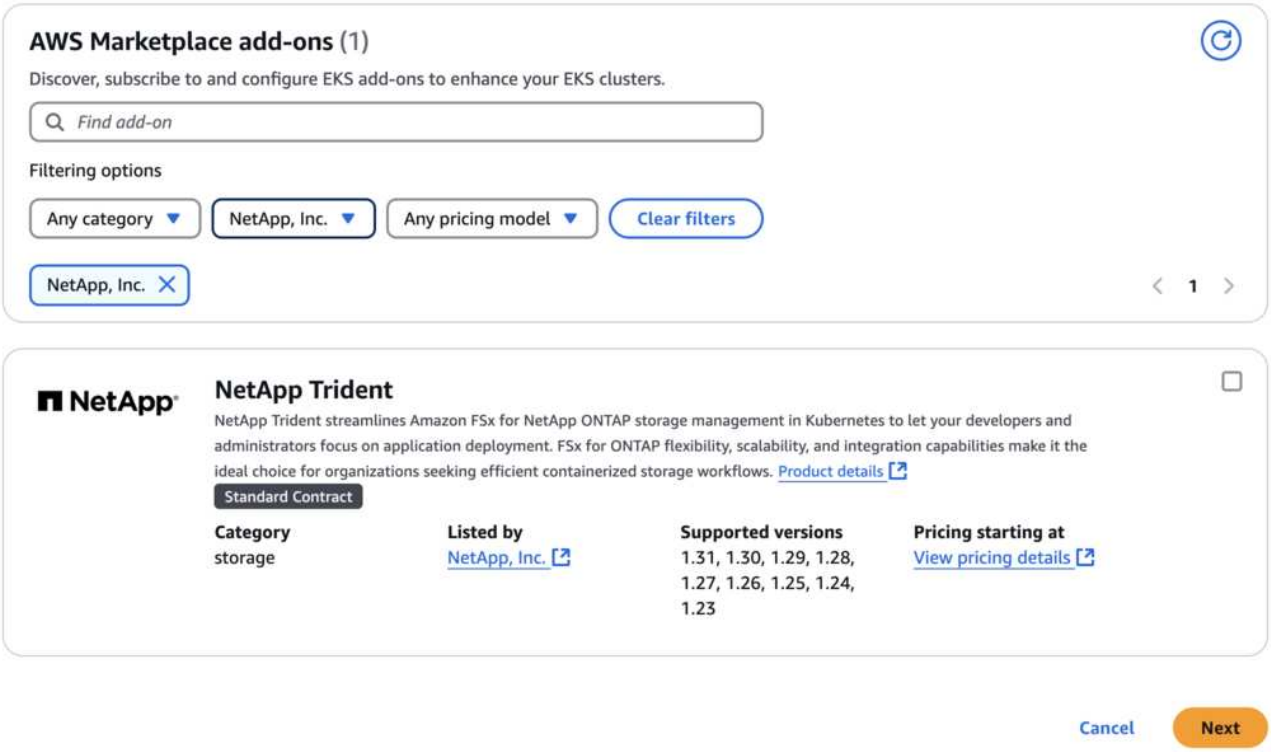
1. Asegúrese de crear un rol de IAM y un secreto de AWS para permitir que los pods de EKS accedan a los

recursos de AWS. Para obtener instrucciones, consulte"Crea un rol de IAM y un secreto de AWS" .

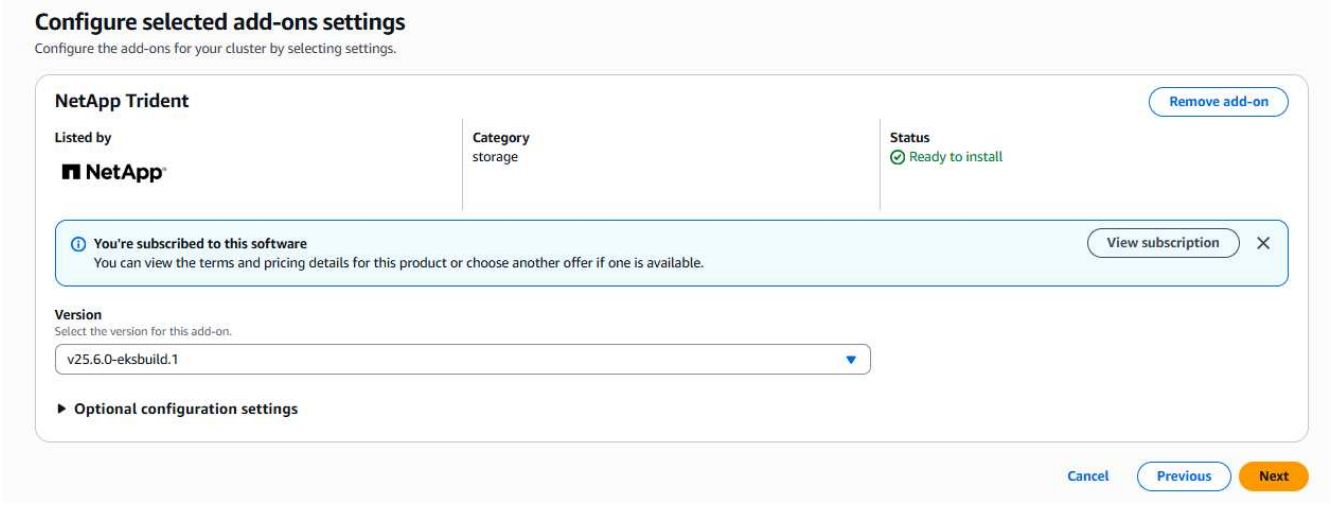
2. En su clúster de Kubernetes EKS, vaya a la pestaña **Complementos**.



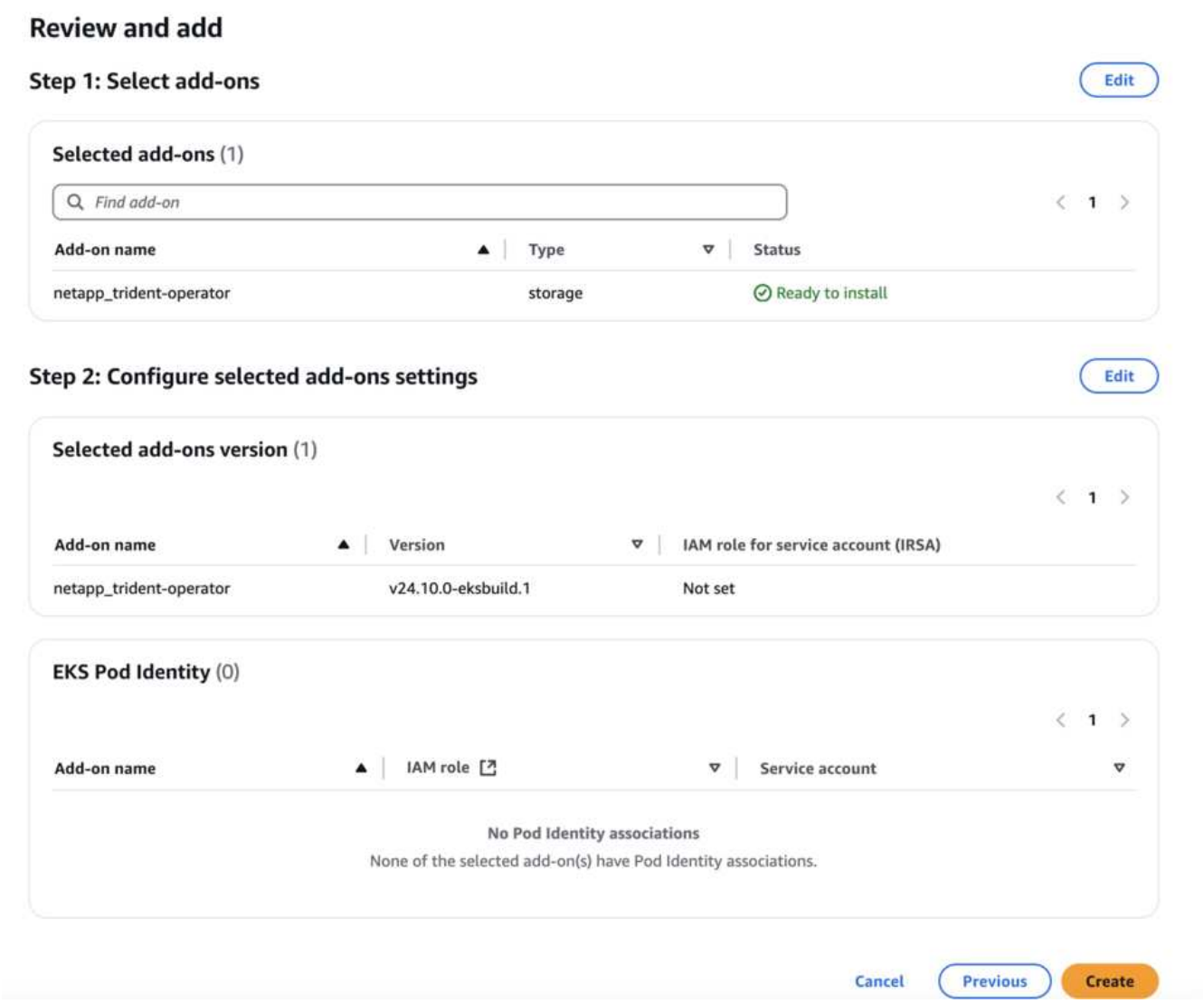
3. Vaya a **Complementos de AWS Marketplace** y elija la categoría *almacenamiento*.



4. Localice * NetApp Trident* y seleccione la casilla de verificación del complemento Trident , y haga clic en **Siguiente**.
5. Elige la versión deseada del complemento.

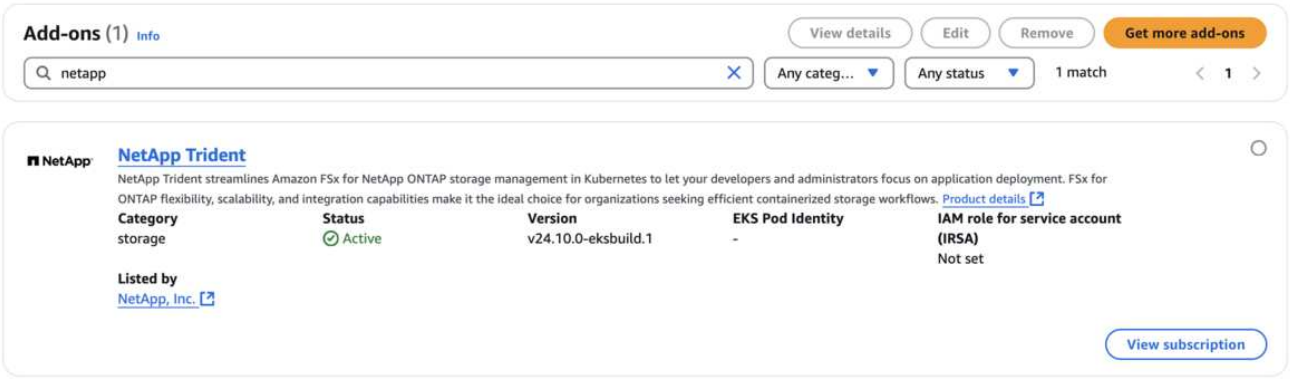


6. Configure los ajustes complementarios necesarios.



7. Si utiliza IRSA (roles de IAM para cuentas de servicio), consulte los pasos de configuración adicionales."aquí" .
8. Seleccione **Crear**.

9. Verifique que el estado del complemento sea *Activo*.



10. Ejecute el siguiente comando para verificar que Trident esté correctamente instalado en el clúster:

```
kubectl get pods -n trident
```

11. Continúe con la configuración y configure el backend de almacenamiento. Para obtener más información, consulte"Configurar el backend de almacenamiento" .

Instalar/desinstalar el complemento Trident EKS mediante la CLI

Instale el complemento NetApp Trident EKS mediante la CLI:

El siguiente comando de ejemplo instala el complemento Trident EKS: `eksctl create addon --cluster clusterName --name netapp_trident-operator --version v25.6.0-eksbuild.1` (con una versión dedicada)

Desinstale el complemento NetApp Trident EKS mediante la CLI:

El siguiente comando desinstala el complemento Trident EKS:

```
eksctl delete addon --cluster K8s-arm --name netapp_trident-operator
```

Crea backends con kubectl

Un backend define la relación entre Trident y un sistema de almacenamiento. Le indica a Trident cómo comunicarse con ese sistema de almacenamiento y cómo Trident aprovisionar volúmenes desde él. Una vez instalado Trident , el siguiente paso es crear un backend. El `TridentBackendConfig` La definición de recursos personalizados (CRD) le permite crear y administrar backends de Trident directamente a través de la interfaz de Kubernetes. Puedes hacerlo utilizando `kubectl` o la herramienta CLI equivalente para su distribución de Kubernetes.

`TridentBackendConfig`

`TridentBackendConfig` (`tbc`, `tbconfig` , `tbackendconfig`) es un CRD frontend con espacios de nombres que le permite administrar backends de Trident usando `kubectl` . Los administradores de Kubernetes y almacenamiento ahora pueden crear y administrar backends directamente a través de la CLI de

Kubernetes sin necesidad de una utilidad de línea de comandos dedicada.(`tridentctl`).

Tras la creación de un `TridentBackendConfig` objeto, sucede lo siguiente:

- Trident crea automáticamente un backend basándose en la configuración que proporciones. Esto se representa internamente como un `TridentBackend` (`tbe` , `tridentbackend`) CR.
- El `TridentBackendConfig` está singularmente ligado a un `TridentBackend` que fue creada por Trident.

Cada `TridentBackendConfig` mantiene una correspondencia uno a uno con un `TridentBackend` La primera es la interfaz que se proporciona al usuario para diseñar y configurar backends; la segunda es la forma en que Trident representa el objeto backend real.

> `TridentBackend`Trident crea automáticamente los CR. No debes modificarlos. Si deseas realizar actualizaciones en los backends, hazlo modificando el `TridentBackendConfig` objeto.

Consulte el siguiente ejemplo para ver el formato de `TridentBackendConfig` CR:

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-ontap-san
spec:
  version: 1
  backendName: ontap-san-backend
  storageDriverName: ontap-san
  managementLIF: 10.0.0.1
  dataLIF: 10.0.0.2
  svm: trident_svm
  credentials:
    name: backend-tbc-ontap-san-secret
```

También puedes consultar los ejemplos en el "instalador de trident" Directorio con configuraciones de ejemplo para la plataforma/servicio de almacenamiento deseado.

El `spec` Toma parámetros de configuración específicos del backend. En este ejemplo, el backend utiliza el `ontap-san` El controlador de almacenamiento utiliza los parámetros de configuración que se tabulan aquí. Para obtener la lista de opciones de configuración para el controlador de almacenamiento que desee, consulte el"Información de configuración de backend para su controlador de almacenamiento" .

El `spec` Esta sección también incluye `credentials` y `deletionPolicy` campos, que se introducen recientemente en el `TridentBackendConfig` CR:

- `credentials`Este parámetro es un campo obligatorio y contiene las credenciales utilizadas para autenticarse con el sistema/servicio de almacenamiento. Esto se configura con un secreto de Kubernetes creado por el usuario. Las credenciales no se pueden pasar en texto plano y se producirá un error.
- `deletionPolicy`Este campo define qué debe suceder cuando el

`TridentBackendConfig se elimina. Puede tomar uno de dos valores posibles:

- `delete`Esto da como resultado la eliminación de ambos. `TridentBackendConfig` CR y el backend asociado. Este es el valor predeterminado.
- `retain`Cuando un `TridentBackendConfig` Si se elimina el CR, la definición del backend seguirá presente y se podrá gestionar con `tridentctl` . Configurar la política de eliminación a `retain` permite a los usuarios regresar a una versión anterior (anterior a la 21.04) y conservar los backends creados. El valor de este campo se puede actualizar después de un `TridentBackendConfig` se crea

El nombre de un backend se establece mediante `spec.backendName` . Si no se especifica, el nombre del backend se establece como el nombre del `TridentBackendConfig` objeto (metadata.nombre). Se recomienda establecer explícitamente los nombres de los backends utilizando `spec.backendName` .

Backends que fueron creados con `tridentctl` no tienen un asociado `TridentBackendConfig` objeto. Puedes optar por gestionar dichos backends con `kubectl` al crear un `TridentBackendConfig` CR. Se debe tener cuidado al especificar parámetros de configuración idénticos (como por ejemplo `spec.backendName` , `spec.storagePrefix` , `spec.storageDriverName` , etcétera). Trident enlazará automáticamente el recién creado `TridentBackendConfig` con el backend preexistente.

Resumen de pasos

Para crear un nuevo backend utilizando `kubectl` , debes hacer lo siguiente:

1. Crear una "Secreto de Kubernetes" El secreto contiene las credenciales que Trident necesita para comunicarse con el clúster/servicio de almacenamiento.
2. Crear una `TridentBackendConfig` objeto. Esto contiene detalles sobre el clúster/servicio de almacenamiento y hace referencia al secreto creado en el paso anterior.

Después de crear un backend, puede observar su estado mediante el uso de `kubectl get tbc <tbc-name> -n <trident-namespace>` y recabar detalles adicionales.

Paso 1: Crear un secreto de Kubernetes

Crea un secreto que contenga las credenciales de acceso para el backend. Esto es específico de cada servicio/plataforma de almacenamiento. He aquí un ejemplo:

```
kubectl -n trident create -f backend-tbc-ontap-san-secret.yaml
```

```
apiVersion: v1
kind: Secret
metadata:
  name: backend-tbc-ontap-san-secret
type: Opaque
stringData:
  username: cluster-admin
  password: password
```

Esta tabla resume los campos que deben incluirse en el secreto para cada plataforma de almacenamiento:

Descripción de los campos secretos de la plataforma de almacenamiento	Secreto	Descripción de los campos
Azure NetApp Files	ID de cliente	El ID de cliente de un registro de aplicación
Cloud Volumes Service para GCP	ID de clave privada	Identificador de la clave privada. Parte de la clave API para la cuenta de servicio de GCP con rol de administrador de CVS
Cloud Volumes Service para GCP	clave_privada	Clave privada. Parte de la clave API para la cuenta de servicio de GCP con rol de administrador de CVS
Elemento (NetApp HCI/ SolidFire)	Punto final	MVIP para el clúster SolidFire con credenciales de inquilino
ONTAP	nombre de usuario	Nombre de usuario para conectarse al clúster/SVM. Se utiliza para la autenticación basada en credenciales.
ONTAP	contraseña	Contraseña para conectarse al cluster/SVM. Se utiliza para la autenticación basada en credenciales.
ONTAP	clave privada del cliente	Valor codificado en Base64 de la clave privada del cliente. Se utiliza para la autenticación basada en certificados.

Descripción de los campos secretos de la plataforma de almacenamiento	Secreto	Descripción de los campos
ONTAP	chapNombre de usuario	Nombre de usuario entrante. Requerido si useCHAP=true. Para `ontap-san` y `ontap-san-economy`
ONTAP	Secreto del iniciador del capítulo	Secreto del iniciador de CHAP. Requerido si useCHAP=true. Para `ontap-san` y `ontap-san-economy`
ONTAP	chapTargetUsername	Nombre de usuario objetivo. Requerido si useCHAP=true. Para `ontap-san` y `ontap-san-economy`
ONTAP	chapTargetInitiatorSecret	Secreto del iniciador del objetivo CHAP. Requerido si useCHAP=true. Para `ontap-san` y `ontap-san-economy`

El secreto creado en este paso se utilizará como referencia en el `spec.credentials` campo de la `TridentBackendConfig` objeto que se crea en el siguiente paso.

Paso 2: Crear el `TridentBackendConfig` CR

Ahora estás listo para crear tu `TridentBackendConfig` CR. En este ejemplo, un backend que utiliza el `ontap-san` El controlador se crea utilizando el `TridentBackendConfig` objeto que se muestra a continuación:

```
kubectl -n trident create -f backend-tbc-ontap-san.yaml
```

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-ontap-san
spec:
  version: 1
  backendName: ontap-san-backend
  storageDriverName: ontap-san
  managementLIF: 10.0.0.1
  dataLIF: 10.0.0.2
  svm: trident_svm
  credentials:
    name: backend-tbc-ontap-san-secret
```

Paso 3: Verificar el estado del `TridentBackendConfig` CR

Ahora que has creado el `TridentBackendConfig` CR, puede verificar el estado. Vea el siguiente ejemplo:

```
kubectl -n trident get tbc backend-tbc-ontap-san
NAME                       BACKEND NAME          BACKEND UUID
PHASE   STATUS
backend-tbc-ontap-san   ontap-san-backend     8d24fce7-6f60-4d4a-8ef6-
bab2699e6ab8   Bound   Success
```

Se creó correctamente un backend y se vinculó al `TridentBackendConfig` CR.

La fase puede tomar uno de los siguientes valores:

- `Bound`: El `TridentBackendConfig` CR está asociado a un backend, y ese backend contiene `configRef` puesto a `TridentBackendConfig` UID de CR.
- `Unbound`Representado mediante `"" . El `TridentBackendConfig` El objeto no está vinculado a un backend. Todo recién creado `TridentBackendConfig` Los CR se encuentran en esta fase por defecto. Tras el cambio de fase, no puede volver a estar sin ataduras.
- `Deleting`: El `TridentBackendConfig` CR `deletionPolicy` estaba configurado para eliminar. Cuando el `TridentBackendConfig` Cuando se elimina el CR, pasa al estado de Eliminación.
  - Si no existen reclamaciones de volumen persistentes (PVC) en el backend, eliminar el `TridentBackendConfig` Esto provocará que Trident elimine tanto el backend como el `TridentBackendConfig` CR.
  - Si hay uno o más PVC presentes en el backend, pasa a un estado de eliminación. El `TridentBackendConfig` Posteriormente, CR también entra en la fase de eliminación. El backend y `TridentBackendConfig` se eliminan solo después de que se hayan eliminado todos los PVC.
- `Lost`El backend asociado con el `TridentBackendConfig` CR fue borrado accidental o deliberadamente y el `TridentBackendConfig` CR aún conserva una referencia al backend eliminado. El `TridentBackendConfig` CR aún puede eliminarse independientemente de `deletionPolicy` valor.

- `Unknown`Trident no puede determinar el estado o la existencia del backend asociado con el `TridentBackendConfig` CR. Por ejemplo, si el servidor API no responde o si `tridentbackends.trident.netapp.io` Falta el CRD. Esto podría requerir intervención.

¡En esta etapa, el backend se ha creado con éxito! Existen varias operaciones que también se pueden realizar, como por ejemplo:"actualizaciones y eliminaciones de backend" .

(Opcional) Paso 4: Obtenga más detalles

Puedes ejecutar el siguiente comando para obtener más información sobre tu backend:

```
kubectl -n trident get tbc backend-tbc-ontap-san -o wide
```

```
NAME                      BACKEND NAME        BACKEND UUID
PHASE   STATUS    STORAGE DRIVER   DELETION POLICY
backend-tbc-ontap-san   ontap-san-backend   8d24fce7-6f60-4d4a-8ef6-
bab2699e6ab8   Bound   Success   ontap-san        delete
```

Además, también puede obtener un volcado YAML/JSON de `TridentBackendConfig` .

```
kubectl -n trident get tbc backend-tbc-ontap-san -o yaml
```

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  creationTimestamp: 2021-04-21T20:45:11Z
  finalizers:
    - trident.netapp.io
  generation: 1
  name: backend-tbc-ontap-san
  namespace: trident
  resourceVersion: "947143"
  uid: 35b9d777-109f-43d5-8077-c74a4559d09c
spec:
  backendName: ontap-san-backend
  credentials:
    name: backend-tbc-ontap-san-secret
  managementLIF: 10.0.0.1
  dataLIF: 10.0.0.2
  storageDriverName: ontap-san
  svm: trident_svm
  version: 1
status:
  backendInfo:
    backendName: ontap-san-backend
    backendUUID: 8d24fce7-6f60-4d4a-8ef6-bab2699e6ab8
  deletionPolicy: delete
  lastOperationStatus: Success
  message: Backend 'ontap-san-backend' created
  phase: Bound
```

`backendInfo`contiene el `backendName` y el `backendUUID` del backend que se creó en respuesta a `TridentBackendConfig` CR. El `lastOperationStatus` El campo representa el estado de la última operación de `TridentBackendConfig` CR, que puede ser activado por el usuario (por ejemplo, el usuario cambió algo en `spec`) o activado por Trident (por ejemplo, durante los reinicios de Trident). Puede ser un éxito o un fracaso. `phase` representa el estado de la relación entre el `TridentBackendConfig` CR y el backend. En el ejemplo anterior, `phase` tiene el valor Bound, lo que significa que el `TridentBackendConfig` CR está asociado con el backend.

Puedes ejecutar el `kubectl -n trident describe tbc <tbc-cr-name>` comando para obtener detalles de los registros de eventos.

> No se puede actualizar ni eliminar un backend que contenga un asociado `TridentBackendConfig` objeto usando `tridentctl` . Para comprender los pasos que implica el cambio entre `tridentctl` y `TridentBackendConfig` ,"ver aquí" .

Gestionar backends

Realizar la gestión del backend con kubectl

Aprenda cómo realizar operaciones de administración de backend utilizando `kubectl` .

Eliminar un backend

Al eliminar un `TridentBackendConfig` , le indicas a Trident que elimine/conserve los backends (según `deletionPolicy`). Para eliminar un backend, asegúrese de que `deletionPolicy` está configurado para eliminar. Para eliminar solo el `TridentBackendConfig` , asegúrese de que `deletionPolicy` está previsto que se mantenga. Esto garantiza que el backend siga presente y pueda gestionarse mediante `tridentctl` .

Ejecute el siguiente comando:

```
kubectl delete tbc <tbc-name> -n trident
```

Trident no elimina los secretos de Kubernetes que estaban en uso por `TridentBackendConfig` . El usuario de Kubernetes es responsable de la limpieza de secretos. Hay que tener cuidado al borrar secretos. Solo debes eliminar los secretos si los backends no los están utilizando.

Ver los backends existentes

Ejecute el siguiente comando:

```
kubectl get tbc -n trident
```

También puedes correr `tridentctl get backend -n trident` o `tridentctl get backend -o yaml -n trident` para obtener una lista de todos los backends existentes. Esta lista también incluirá backends que fueron creados con `tridentctl` .

Actualizar un backend

Existen múltiples razones para actualizar un backend:

- Las credenciales del sistema de almacenamiento han cambiado. Para actualizar las credenciales, el secreto de Kubernetes que se utiliza en el `TridentBackendConfig` El objeto debe actualizarse. Trident actualizará automáticamente el sistema backend con las últimas credenciales proporcionadas. Ejecute el siguiente comando para actualizar el secreto de Kubernetes:

  ```
  kubectl apply -f <updated-secret-file.yaml> -n trident
  ```

- Es necesario actualizar los parámetros (como el nombre de la SVM de ONTAP que se está utilizando).
  - Puedes actualizar `TridentBackendConfig` objetos directamente a través de Kubernetes usando el siguiente comando:

```
kubectl apply -f <updated-backend-file.yaml>
```

- Alternativamente, puede realizar cambios en el sistema existente. `TridentBackendConfig` CR usando el siguiente comando:

```
kubectl edit tbc <tbc-name> -n trident
```

- Si falla una actualización del backend, este permanecerá en su última configuración conocida. Puedes ver los registros para determinar la causa ejecutando `kubectl get tbc <tbc-name> -o yaml -n trident` o `kubectl describe tbc <tbc-name> -n trident` .
- Después de identificar y corregir el problema con el archivo de configuración, puede volver a ejecutar el comando de actualización.

Realizar la gestión del backend con tridentctl

Aprenda cómo realizar operaciones de administración de backend utilizando `tridentctl` .

Crear un backend

Después de crear un"archivo de configuración del backend" , ejecute el siguiente comando:

```
tridentctl create backend -f <backend-file> -n trident
```

Si falla la creación del backend, algo falló en la configuración del backend. Puedes consultar los registros para determinar la causa ejecutando el siguiente comando:

```
tridentctl logs -n trident
```

Una vez que haya identificado y corregido el problema con el archivo de configuración, simplemente puede ejecutar el programa. `create` Orden de nuevo.

Eliminar un backend

Para eliminar un backend de Trident, siga estos pasos:

1. Recuperar el nombre del backend:

```
tridentctl get backend -n trident
```

2. Eliminar el backend:

```
tridentctl delete backend <backend-name> -n trident
```

Si Trident ha aprovisionado volúmenes e instantáneas desde este backend que aún existen, eliminar el backend impide que se aprovisionen nuevos volúmenes desde él. El backend seguirá existiendo en estado "Eliminando".

Ver los backends existentes

Para ver los backends que Trident conoce, haga lo siguiente:

- Para obtener un resumen, ejecute el siguiente comando:

```
tridentctl get backend -n trident
```

- Para obtener todos los detalles, ejecute el siguiente comando:

```
tridentctl get backend -o json -n trident
```

Actualizar un backend

Después de crear un nuevo archivo de configuración de backend, ejecute el siguiente comando:

```
tridentctl update backend <backend-name> -f <backend-file> -n trident
```

Si falla la actualización del backend, algo falló en la configuración del backend o intentaste una actualización no válida. Puedes consultar los registros para determinar la causa ejecutando el siguiente comando:

```
tridentctl logs -n trident
```

Una vez que haya identificado y corregido el problema con el archivo de configuración, simplemente puede ejecutar el programa. `update` Orden de nuevo.

Identifique las clases de almacenamiento que utilizan un backend.

Este es un ejemplo del tipo de preguntas que puedes responder con el JSON que `tridentctl` Salidas para objetos de backend. Esto utiliza el `jq` utilidad que necesitas instalar.

```
tridentctl get backend -o json | jq '[.items[] | {backend: .name,
storageClasses: [.storage[].storageClasses]|unique}]'
```

Esto también se aplica a los backends que se crearon utilizando `TridentBackendConfig` .

Cambiar entre las opciones de administración de backend

Aprende sobre las diferentes formas de gestionar los backends en Trident.

Opciones para la gestión de backends

Con la introducción de `TridentBackendConfig` Ahora, los administradores disponen de dos formas únicas de gestionar los sistemas backend. Esto plantea las siguientes preguntas:

- ¿Se pueden crear backends usando `tridentctl` ser gestionado con `TridentBackendConfig` ?
- ¿Se pueden crear backends usando `TridentBackendConfig` ser gestionado mediante `tridentctl` ?

Administrar `tridentctl` backends que utilizan `TridentBackendConfig`

Esta sección abarca los pasos necesarios para gestionar los backends que se crearon utilizando `tridentctl` directamente a través de la interfaz de Kubernetes creando `TridentBackendConfig` objetos.

Esto se aplicará a los siguientes escenarios:

- Sistemas backend preexistentes que no tienen un `TridentBackendConfig` porque fueron creados con `tridentctl` .
- Nuevos backends que se crearon con `tridentctl` , mientras que otros `TridentBackendConfig` Los objetos existen.

En ambos escenarios, los backends seguirán presentes, con Trident programando volúmenes y operando sobre ellos. Los administradores tienen dos opciones:

- Continuar usando `tridentctl` para gestionar los backends que se crearon utilizándolo.
- Enlace de backends creados usando `tridentctl` a un nuevo `TridentBackendConfig` objeto. Hacerlo significaría que los backends se gestionarían utilizando `kubectl` y no `tridentctl` .

Para gestionar un backend preexistente utilizando `kubectl` , necesitarás crear un `TridentBackendConfig` que se vincula al backend existente. Aquí tenéis una descripción general de cómo funciona:

1. Crea un secreto de Kubernetes. El secreto contiene las credenciales que Trident necesita para comunicarse con el clúster/servicio de almacenamiento.
2. Crear una `TridentBackendConfig` objeto. Esto contiene detalles sobre el clúster/servicio de almacenamiento y hace referencia al secreto creado en el paso anterior. Se debe tener cuidado al especificar parámetros de configuración idénticos (como por ejemplo `spec.backendName` , `spec.storagePrefix` , `spec.storageDriverName` , etcétera). `spec.backendName` Debe configurarse con el nombre del backend existente.

Paso 0: Identificar el backend

Para crear una `TridentBackendConfig` Si se vincula a un backend existente, deberá obtener la configuración del backend. En este ejemplo, supongamos que se creó un backend utilizando la siguiente definición JSON:

```
tridentctl get backend ontap-nas-backend -n trident
+---------------------+----------------
+--------------------------------------+--------+---------+
|          NAME       | STORAGE DRIVER |                 UUID
| STATE  | VOLUMES |
+---------------------+----------------
+--------------------------------------+--------+---------+
| ontap-nas-backend   | ontap-nas      | 52f2eb10-e4c6-4160-99fc-
96b3be5ab5d7 | online |      25 |
+---------------------+----------------
+--------------------------------------+--------+---------+
```

```
cat ontap-nas-backend.json
```

```
{
  "version": 1,
  "storageDriverName": "ontap-nas",
  "managementLIF": "10.10.10.1",
  "dataLIF": "10.10.10.2",
  "backendName": "ontap-nas-backend",
  "svm": "trident_svm",
  "username": "cluster-admin",
  "password": "admin-password",
  "defaults": {
    "spaceReserve": "none",
    "encryption": "false"
  },
  "labels": {
    "store": "nas_store"
  },
  "region": "us_east_1",
  "storage": [
    {
      "labels": {
        "app": "msoffice",
        "cost": "100"
      },
      "zone": "us_east_1a",
      "defaults": {
        "spaceReserve": "volume",
        "encryption": "true",
        "unixPermissions": "0755"
      }
    },
    {
      "labels": {
        "app": "mysqldb",
        "cost": "25"
      },
      "zone": "us_east_1d",
      "defaults": {
        "spaceReserve": "volume",
        "encryption": "false",
        "unixPermissions": "0775"
      }
    }
  ]
}
```

Paso 1: Crear un secreto de Kubernetes

Crea un secreto que contenga las credenciales para el backend, como se muestra en este ejemplo:

```
cat tbc-ontap-nas-backend-secret.yaml
```

```
apiVersion: v1
kind: Secret
metadata:
  name: ontap-nas-backend-secret
type: Opaque
stringData:
  username: cluster-admin
  password: admin-password
```

```
kubectl create -f tbc-ontap-nas-backend-secret.yaml -n trident
secret/backend-tbc-ontap-san-secret created
```

Paso 2: Crear un `TridentBackendConfig` CR

El siguiente paso es crear un `TridentBackendConfig` CR que se vinculará automáticamente al preexistente `ontap-nas-backend` (como en este ejemplo). Asegúrese de que se cumplan los siguientes requisitos:

- El mismo nombre de backend se define en `spec.backendName` .
- Los parámetros de configuración son idénticos a los del backend original.
- Los pools virtuales (si existen) deben mantener el mismo orden que en el backend original.
- Las credenciales se proporcionan a través de un secreto de Kubernetes y no en texto plano.

En este caso, el `TridentBackendConfig` Se verá así:

```
cat backend-tbc-ontap-nas.yaml
```

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: tbc-ontap-nas-backend
spec:
  version: 1
  storageDriverName: ontap-nas
  managementLIF: 10.10.10.1
  dataLIF: 10.10.10.2
  backendName: ontap-nas-backend
  svm: trident_svm
  credentials:
    name: mysecret
  defaults:
    spaceReserve: none
    encryption: 'false'
  labels:
    store: nas_store
  region: us_east_1
  storage:
  - labels:
      app: msoffice
      cost: '100'
    zone: us_east_1a
    defaults:
      spaceReserve: volume
      encryption: 'true'
      unixPermissions: '0755'
  - labels:
      app: mysqldb
      cost: '25'
    zone: us_east_1d
    defaults:
      spaceReserve: volume
      encryption: 'false'
      unixPermissions: '0775'
```

```
kubectl create -f backend-tbc-ontap-nas.yaml -n trident
tridentbackendconfig.trident.netapp.io/tbc-ontap-nas-backend created
```

Paso 3: Verificar el estado del `TridentBackendConfig` CR

Después de `TridentBackendConfig` Se ha creado, su fase debe ser `Bound` . También debe reflejar el mismo nombre de backend y UUID que el backend existente.

```
kubectl get tbc tbc-ontap-nas-backend -n trident
NAME                   BACKEND NAME          BACKEND UUID
PHASE   STATUS
tbc-ontap-nas-backend  ontap-nas-backend     52f2eb10-e4c6-4160-99fc-
96b3be5ab5d7   Bound   Success

#confirm that no new backends were created (i.e., TridentBackendConfig did
not end up creating a new backend)
tridentctl get backend -n trident
+---------------------+----------------
+--------------------------------------+--------+---------+
|          NAME       | STORAGE DRIVER |                 UUID
| STATE  | VOLUMES |
+---------------------+----------------
+--------------------------------------+--------+---------+
| ontap-nas-backend   | ontap-nas      | 52f2eb10-e4c6-4160-99fc-
96b3be5ab5d7 | online |      25 |
+---------------------+----------------
+--------------------------------------+--------+---------+
```

El backend ahora se gestionará completamente utilizando `tbc-ontap-nas-backend TridentBackendConfig` objeto.

Administrar `TridentBackendConfig` backends que utilizan `tridentctl`

```
`tridentctl`se puede utilizar para listar los backends que se crearon
utilizando `TridentBackendConfig` .  Además, los administradores también
pueden optar por gestionar completamente dichos sistemas backend mediante
`tridentctl` al eliminar `TridentBackendConfig` y asegurándose
`spec.deletionPolicy` está configurado para `retain` .
```

Paso 0: Identificar el backend

Por ejemplo, supongamos que el siguiente backend se creó utilizando `TridentBackendConfig` :

```
kubectl get tbc backend-tbc-ontap-san -n trident -o wide
NAME                    BACKEND NAME        BACKEND UUID
PHASE   STATUS    STORAGE DRIVER   DELETION POLICY
backend-tbc-ontap-san   ontap-san-backend   81abcb27-ea63-49bb-b606-
0a5315ac5f82   Bound   Success   ontap-san        delete

tridentctl get backend ontap-san-backend -n trident
+-------------------+----------------
+--------------------------------------+--------+---------+
|       NAME        | STORAGE DRIVER |                 UUID
| STATE  | VOLUMES |
+-------------------+----------------
+--------------------------------------+--------+---------+
| ontap-san-backend | ontap-san      | 81abcb27-ea63-49bb-b606-
0a5315ac5f82 | online |      33 |
+-------------------+----------------
+--------------------------------------+--------+---------+
```

De los resultados se observa que `TridentBackendConfig` Se creó correctamente y está vinculado a un backend [observe el UUID del backend].

Paso 1: Confirmar `deletionPolicy` **está configurado para** `retain`

Analicemos el valor de `deletionPolicy` . Esto debe configurarse a `retain` . Esto garantiza que cuando un `TridentBackendConfig` Si se elimina el CR, la definición del backend seguirá presente y se podrá gestionar con `tridentctl` .

```
kubectl get tbc backend-tbc-ontap-san -n trident -o wide
NAME                    BACKEND NAME        BACKEND UUID
PHASE   STATUS    STORAGE DRIVER   DELETION POLICY
backend-tbc-ontap-san   ontap-san-backend   81abcb27-ea63-49bb-b606-
0a5315ac5f82   Bound   Success   ontap-san        delete

# Patch value of deletionPolicy to retain
kubectl patch tbc backend-tbc-ontap-san --type=merge -p
'{"spec":{"deletionPolicy":"retain"}}' -n trident
tridentbackendconfig.trident.netapp.io/backend-tbc-ontap-san patched

#Confirm the value of deletionPolicy
kubectl get tbc backend-tbc-ontap-san -n trident -o wide
NAME                    BACKEND NAME        BACKEND UUID
PHASE   STATUS    STORAGE DRIVER   DELETION POLICY
backend-tbc-ontap-san   ontap-san-backend   81abcb27-ea63-49bb-b606-
0a5315ac5f82   Bound   Success   ontap-san        retain
```

No continúe con el siguiente paso a menos que `deletionPolicy` está configurado para `retain` .

Paso 2: Eliminar el `TridentBackendConfig` CR

El último paso es eliminar el `TridentBackendConfig` CR. Tras confirmar el `deletionPolicy` está configurado para `retain` , puedes proceder con la eliminación:

```
kubectl delete tbc backend-tbc-ontap-san -n trident
tridentbackendconfig.trident.netapp.io "backend-tbc-ontap-san" deleted

tridentctl get backend ontap-san-backend -n trident
+-------------------+----------------
+--------------------------------------+--------+---------+
|       NAME        | STORAGE DRIVER |                 UUID
| STATE   | VOLUMES |
+-------------------+----------------
+--------------------------------------+--------+---------+
| ontap-san-backend | ontap-san      | 81abcb27-ea63-49bb-b606-
0a5315ac5f82 | online |      33 |
+-------------------+----------------
+--------------------------------------+--------+---------+
```

Tras la eliminación de `TridentBackendConfig` Trident simplemente elimina el objeto sin borrar realmente el backend.

Crear y gestionar clases de almacenamiento

Crear una clase de almacenamiento

Configure un objeto StorageClass de Kubernetes y cree la clase de almacenamiento para indicar a Trident cómo aprovisionar volúmenes.

Configurar un objeto StorageClass de Kubernetes

El "Objeto StorageClass de Kubernetes" identifica a Trident como el proveedor que se utiliza para esa clase e indica a Trident cómo aprovisionar un volumen. Por ejemplo:

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontap-gold
provisioner: csi.trident.netapp.io
mountOptions:
  - nfsvers=3
  - nolock
parameters:
  backendType: "ontap-nas"
  media: "ssd"
allowVolumeExpansion: true
volumeBindingMode: Immediate
```

Referirse a"Objetos de Kubernetes y Trident" Para obtener detalles sobre cómo interactúan las clases de almacenamiento con `PersistentVolumeClaim` y parámetros para controlar cómo Trident gestiona los volúmenes.

Crear una clase de almacenamiento

Después de crear el objeto StorageClass, puede crear la clase de almacenamiento. muestras de clase de almacenamiento proporciona algunos ejemplos básicos que puede usar o modificar.

Pasos

1. Este es un objeto de Kubernetes, así que use `kubectl` para crearlo en Kubernetes.

```
kubectl create -f sample-input/storage-class-basic-csi.yaml
```

2. Ahora debería ver una clase de almacenamiento **basic-csi** tanto en Kubernetes como en Trident, y Trident debería haber detectado los pools en el backend.

```
kubectl get sc basic-csi
```

```
NAME         PROVISIONER             AGE
basic-csi    csi.trident.netapp.io   15h
```

```
./tridentctl -n trident get storageclass basic-csi -o json
```

```
{
  "items": [
    {
      "Config": {
        "version": "1",
        "name": "basic-csi",
        "attributes": {
          "backendType": "ontap-nas"
        },
        "storagePools": null,
        "additionalStoragePools": null
      },
      "storage": {
        "ontapnas_10.0.0.1": [
          "aggr1",
          "aggr2",
          "aggr3",
          "aggr4"
        ]
      }
    }
  ]
}
```


muestras de clase de almacenamiento

Trident proporciona "Definiciones de clases de almacenamiento simples para backends específicos" .

Alternativamente, puedes editar `sample-input/storage-class-csi.yaml.templ` archivo que viene con el instalador y reemplazar *`BACKEND_TYPE`* con el nombre del controlador de almacenamiento.

```
./tridentctl -n trident get backend
+-------------+---------------+--------------------------------------
+--------+---------+
|    NAME     | STORAGE DRIVER |                 UUID                 |
STATE  | VOLUMES |
+-------------+---------------+--------------------------------------
+--------+---------+
| nas-backend | ontap-nas      | 98e19b74-aec7-4a3d-8dcf-128e5033b214 |
online |       0 |
+-------------+---------------+--------------------------------------
+--------+---------+

cp sample-input/storage-class-csi.yaml.templ sample-input/storage-class-
basic-csi.yaml

# Modify __BACKEND_TYPE__ with the storage driver field above (e.g.,
ontap-nas)
vi sample-input/storage-class-basic-csi.yaml
```

Gestionar clases de almacenamiento

Puede ver las clases de almacenamiento existentes, establecer una clase de almacenamiento predeterminada, identificar el backend de la clase de almacenamiento y eliminar clases de almacenamiento.

Vea las clases de almacenamiento existentes

- Para ver las clases de almacenamiento de Kubernetes existentes, ejecute el siguiente comando:

```
kubectl get storageclass
```

- Para ver los detalles de la clase de almacenamiento de Kubernetes, ejecute el siguiente comando:

```
kubectl get storageclass <storage-class> -o json
```

- Para ver las clases de almacenamiento sincronizadas de Trident, ejecute el siguiente comando:

```
tridentctl get storageclass
```

- Para ver los detalles de la clase de almacenamiento sincronizado de Trident, ejecute el siguiente comando:

```
tridentctl get storageclass <storage-class> -o json
```

Establecer una clase de almacenamiento predeterminada

Kubernetes 1.6 añadió la capacidad de establecer una clase de almacenamiento predeterminada. Esta es la clase de almacenamiento que se utilizará para aprovisionar un volumen persistente si un usuario no especifica uno en una reclamación de volumen persistente (PVC).

- Define una clase de almacenamiento predeterminada configurando la anotación. `storageclass.kubernetes.io/is-default-class` a verdadero en la definición de la clase de almacenamiento. Según la especificación, cualquier otro valor o la ausencia de la anotación se interpreta como falso.
- Puede configurar una clase de almacenamiento existente como clase de almacenamiento predeterminada mediante el siguiente comando:

```
kubectl patch storageclass <storage-class-name> -p '{"metadata":
{"annotations":{"storageclass.kubernetes.io/is-default-class":"true"}}}'
```

- De igual forma, puede eliminar la anotación de la clase de almacenamiento predeterminada mediante el siguiente comando:

```
kubectl patch storageclass <storage-class-name> -p '{"metadata":
{"annotations":{"storageclass.kubernetes.io/is-default-class":"false"}}}'
```

También hay ejemplos en el paquete de instalación de Trident que incluyen esta anotación.

> Solo debe haber una clase de almacenamiento predeterminada en su clúster a la vez. Técnicamente, Kubernetes no impide tener más de una, pero se comportará como si no existiera ninguna clase de almacenamiento predeterminada.

Identificar el backend para una clase de almacenamiento

Este es un ejemplo del tipo de preguntas que puedes responder con el JSON que `tridentctl` Salidas para objetos backend de Trident . Esto utiliza el `jq` utilidad, que tal vez necesites instalar primero.

```
tridentctl get storageclass -o json | jq  '[.items[] | {storageClass:
.Config.name, backends: [.storage]|unique}]'
```

Eliminar una clase de almacenamiento

Para eliminar una clase de almacenamiento de Kubernetes, ejecute el siguiente comando:

```
kubectl delete storageclass <storage-class>
```

`<storage-class>`debe reemplazarse con su clase de almacenamiento.

Los volúmenes persistentes que se hayan creado mediante esta clase de almacenamiento permanecerán intactos y Trident seguirá gestionándolos.

> Trident impone un espacio en blanco `fsType` por los volúmenes que crea. Para los backends iSCSI, se recomienda aplicar `parameters.fsType` en la clase de almacenamiento. Debes eliminar las StorageClasses existentes y volver a crearlas con `parameters.fsType` especificado.

Provisión y gestión de volúmenes

Provisión de un volumen

Cree un PersistentVolumeClaim (PVC) que utilice la StorageClass de Kubernetes configurada para solicitar acceso al PV. Luego puedes montar el panel fotovoltaico en un soporte.

Descripción general

A "*Reclamación de volumen persistente*" (PVC) es una solicitud de acceso al PersistentVolume en el clúster.

El PVC se puede configurar para solicitar el almacenamiento de un tamaño o modo de acceso determinado. Mediante la StorageClass asociada, el administrador del clúster puede controlar más que el tamaño del PersistentVolume y el modo de acceso, como el rendimiento o el nivel de servicio.

Una vez creado el PVC, puede montar el volumen en un soporte.

Crea el PVC

Pasos

1. Crear el PVC.

```
kubectl create -f pvc.yaml
```

2. Verifique el estado del PVC.

```
kubectl get pvc
```

```
NAME          STATUS   VOLUME     CAPACITY   ACCESS MODES   STORAGECLASS AGE
pvc-storage   Bound    pv-name    1Gi        RWO                    5m
```

1. Monte el volumen en una cápsula.

```
kubectl create -f pv-pod.yaml
```

Puedes supervisar el progreso usando `kubectl get pod --watch`.

2. Verifique que el volumen esté montado en `/my/mount/path`.

```
kubectl exec -it task-pv-pod -- df -h /my/mount/path
```

3. Ahora puedes eliminar el Pod. La aplicación Pod dejará de existir, pero el volumen se mantendrá.

```
kubectl delete pod pv-pod
```

Ejemplos de manifiestos

Muestra de PersistentVolumeClaim

Estos ejemplos muestran opciones básicas de configuración de PVC.

PVC con acceso RWO

Este ejemplo muestra un PVC básico con acceso RWO que está asociado a una StorageClass llamada `basic-csi` .

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: pvc-storage
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: basic-csi
```

PVC con NVMe/TCP

Este ejemplo muestra un PVC básico para NVMe/TCP con acceso RWO que está asociado con una StorageClass llamada `protection-gold` .

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
name: pvc-san-nvme
spec:
accessModes:
  - ReadWriteOnce
resources:
  requests:
    storage: 300Mi
storageClassName: protection-gold
```

Ejemplos de manifiesto de Pod

Estos ejemplos muestran configuraciones básicas para unir el PVC a un módulo.

Configuración básica

```
kind: Pod
apiVersion: v1
metadata:
  name: pv-pod
spec:
  volumes:
    - name: storage
      persistentVolumeClaim:
       claimName: pvc-storage
  containers:
    - name: pv-container
      image: nginx
      ports:
        - containerPort: 80
          name: "http-server"
      volumeMounts:
        - mountPath: "/my/mount/path"
          name: storage
```

Configuración básica NVMe/TCP

```
apiVersion: v1
kind: Pod
metadata:
  name: pod-nginx
spec:
  volumes:
    - name: basic-pvc
      persistentVolumeClaim:
        claimName: pvc-san-nvme
  containers:
    - name: task-pv-container
      image: nginx
      volumeMounts:
        - mountPath: "/my/mount/path"
          name: basic-pvc
```

Referirse a"Objetos de Kubernetes y Trident" Para obtener detalles sobre cómo interactúan las clases de almacenamiento con `PersistentVolumeClaim` y parámetros para controlar cómo Trident gestiona los volúmenes.

Ampliar volúmenes

Trident brinda a los usuarios de Kubernetes la capacidad de expandir sus volúmenes después de su creación. Encuentre información sobre las configuraciones necesarias para expandir volúmenes iSCSI, NFS, SMB, NVMe/TCP y FC.

Expandir un volumen iSCSI

Puede expandir un volumen persistente iSCSI (PV) utilizando el aprovisionador CSI.

> La expansión de volumen iSCSI es compatible con `ontap-san` , `ontap-san-economy` , `solidfire-san` controladores y requiere Kubernetes 1.16 y posterior.

Paso 1: Configure la StorageClass para que admita la expansión de volumen.

Edite la definición de StorageClass para configurar `allowVolumeExpansion` campo a `true` .

```
cat storageclass-ontapsan.yaml
```

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontap-san
provisioner: csi.trident.netapp.io
parameters:
  backendType: "ontap-san"
allowVolumeExpansion: True
```

Para una StorageClass ya existente, edítela para incluir la `allowVolumeExpansion` parámetro.

Paso 2: Cree un PVC con la StorageClass que creó.

Edite la definición de PVC y actualice la `spec.resources.requests.storage` para reflejar el nuevo tamaño deseado, que debe ser mayor que el tamaño original.

```
cat pvc-ontapsan.yaml
```

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: san-pvc
spec:
  accessModes:
  - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: ontap-san
```

Trident crea un Volumen Persistente (PV) y lo asocia con esta Reclamación de Volumen Persistente (PVC).

```
kubectl get pvc
NAME      STATUS   VOLUME                                     CAPACITY
ACCESS MODES   STORAGECLASS   AGE
san-pvc   Bound    pvc-8a814d62-bd58-4253-b0d1-82f2885db671   1Gi
RWO            ontap-san      8s

kubectl get pv
NAME                                       CAPACITY   ACCESS MODES
RECLAIM POLICY   STATUS   CLAIM              STORAGECLASS   REASON   AGE
pvc-8a814d62-bd58-4253-b0d1-82f2885db671   1Gi        RWO
Delete           Bound    default/san-pvc    ontap-san               10s
```

Paso 3: Defina un módulo que conecte el PVC.

Conecte el PV a una cápsula para poder cambiar su tamaño. Existen dos escenarios al redimensionar un PV iSCSI:

- Si el PV está conectado a un pod, Trident expande el volumen en el backend de almacenamiento, vuelve a escanear el dispositivo y redimensiona el sistema de archivos.
- Al intentar cambiar el tamaño de un PV no conectado, Trident expande el volumen en el backend de almacenamiento. Una vez que el PVC se vincula a un pod, Trident vuelve a escanear el dispositivo y redimensiona el sistema de archivos. Kubernetes actualiza entonces el tamaño del PVC una vez que la operación de expansión se ha completado correctamente.

En este ejemplo, se crea un pod que utiliza el `san-pvc` .

```
 kubectl get pod
NAME         READY   STATUS    RESTARTS   AGE
ubuntu-pod   1/1     Running   0          65s

 kubectl describe pvc san-pvc
Name:          san-pvc
Namespace:     default
StorageClass:  ontap-san
Status:        Bound
Volume:        pvc-8a814d62-bd58-4253-b0d1-82f2885db671
Labels:        <none>
Annotations:   pv.kubernetes.io/bind-completed: yes
               pv.kubernetes.io/bound-by-controller: yes
               volume.beta.kubernetes.io/storage-provisioner:
csi.trident.netapp.io
Finalizers:    [kubernetes.io/pvc-protection]
Capacity:      1Gi
Access Modes:  RWO
VolumeMode:    Filesystem
Mounted By:    ubuntu-pod
```

Paso 4: Ampliar el PV

Para cambiar el tamaño del PV creado de 1Gi a 2Gi, edite la definición del PVC y actualícela. `spec.resources.requests.storage` a 2Gi.

```
kubectl edit pvc san-pvc
```

```
# Please edit the object below. Lines beginning with a '#' will be
ignored,
# and an empty file will abort the edit. If an error occurs while saving
this file will be
# reopened with the relevant failures.
#
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  annotations:
    pv.kubernetes.io/bind-completed: "yes"
    pv.kubernetes.io/bound-by-controller: "yes"
    volume.beta.kubernetes.io/storage-provisioner: csi.trident.netapp.io
  creationTimestamp: "2019-10-10T17:32:29Z"
  finalizers:
  - kubernetes.io/pvc-protection
  name: san-pvc
  namespace: default
  resourceVersion: "16609"
  selfLink: /api/v1/namespaces/default/persistentvolumeclaims/san-pvc
  uid: 8a814d62-bd58-4253-b0d1-82f2885db671
spec:
  accessModes:
  - ReadWriteOnce
  resources:
    requests:
      storage: 2Gi
 # ...
```

Paso 5: Validar la expansión

Puedes comprobar que la expansión ha funcionado correctamente verificando el tamaño del PVC, el PV y el volumen del Trident :

```
kubectl get pvc san-pvc
NAME      STATUS   VOLUME                                     CAPACITY
ACCESS MODES   STORAGECLASS   AGE
san-pvc   Bound    pvc-8a814d62-bd58-4253-b0d1-82f2885db671   2Gi
RWO            ontap-san      11m
kubectl get pv
NAME                                       CAPACITY   ACCESS MODES
RECLAIM POLICY   STATUS   CLAIM              STORAGECLASS   REASON   AGE
pvc-8a814d62-bd58-4253-b0d1-82f2885db671   2Gi        RWO
Delete           Bound    default/san-pvc    ontap-san               12m
tridentctl get volumes -n trident
+-----------------------------------------+---------+---------------
+----------+-----------------------------------------+--------+---------+
|                   NAME                  |  SIZE   | STORAGE CLASS |
PROTOCOL |             BACKEND UUID              | STATE  | MANAGED |
+-----------------------------------------+---------+---------------
+----------+-----------------------------------------+--------+---------+
| pvc-8a814d62-bd58-4253-b0d1-82f2885db671 | 2.0 GiB | ontap-san     |
block    | a9b7bfff-0505-4e31-b6c5-59f492e02d33 | online | true    |
+-----------------------------------------+---------+---------------
+----------+-----------------------------------------+--------+---------+
```

Ampliar un volumen FC

Puede expandir un volumen persistente FC (PV) utilizando el aprovisionador CSI.

> La expansión del volumen FC está soportada por la `ontap-san` controlador y requiere Kubernetes 1.16 y posterior.

Paso 1: Configure la StorageClass para que admita la expansión de volumen.

Edite la definición de StorageClass para configurar `allowVolumeExpansion` campo a `true` .

```
cat storageclass-ontapsan.yaml
```

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontap-san
provisioner: csi.trident.netapp.io
parameters:
  backendType: "ontap-san"
allowVolumeExpansion: True
```

Para una StorageClass ya existente, edítela para incluir la `allowVolumeExpansion` parámetro.

Paso 2: Cree un PVC con la StorageClass que creó.

Edite la definición de PVC y actualice la `spec.resources.requests.storage` para reflejar el nuevo tamaño deseado, que debe ser mayor que el tamaño original.

```
cat pvc-ontapsan.yaml
```

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: san-pvc
spec:
  accessModes:
  - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: ontap-san
```

Trident crea un Volumen Persistente (PV) y lo asocia con esta Reclamación de Volumen Persistente (PVC).

```
kubectl get pvc
NAME      STATUS   VOLUME                                     CAPACITY
ACCESS MODES   STORAGECLASS   AGE
san-pvc   Bound    pvc-8a814d62-bd58-4253-b0d1-82f2885db671   1Gi
RWO            ontap-san      8s

kubectl get pv
NAME                                       CAPACITY   ACCESS MODES
RECLAIM POLICY   STATUS   CLAIM              STORAGECLASS   REASON   AGE
pvc-8a814d62-bd58-4253-b0d1-82f2885db671   1Gi        RWO
Delete           Bound    default/san-pvc    ontap-san               10s
```

Paso 3: Defina un módulo que conecte el PVC.

Conecte el PV a una cápsula para poder cambiar su tamaño. Existen dos escenarios al redimensionar un PV de FC:

- Si el PV está conectado a un pod, Trident expande el volumen en el backend de almacenamiento, vuelve a escanear el dispositivo y redimensiona el sistema de archivos.
- Al intentar cambiar el tamaño de un PV no conectado, Trident expande el volumen en el backend de almacenamiento. Una vez que el PVC se vincula a un pod, Trident vuelve a escanear el dispositivo y redimensiona el sistema de archivos. Kubernetes actualiza entonces el tamaño del PVC una vez que la

operación de expansión se ha completado correctamente.

En este ejemplo, se crea un pod que utiliza el `san-pvc` .

```
 kubectl get pod
NAME         READY   STATUS    RESTARTS   AGE
ubuntu-pod   1/1     Running   0          65s

 kubectl describe pvc san-pvc
Name:          san-pvc
Namespace:     default
StorageClass:  ontap-san
Status:        Bound
Volume:        pvc-8a814d62-bd58-4253-b0d1-82f2885db671
Labels:        <none>
Annotations:   pv.kubernetes.io/bind-completed: yes
               pv.kubernetes.io/bound-by-controller: yes
               volume.beta.kubernetes.io/storage-provisioner:
csi.trident.netapp.io
Finalizers:    [kubernetes.io/pvc-protection]
Capacity:      1Gi
Access Modes:  RWO
VolumeMode:    Filesystem
Mounted By:    ubuntu-pod
```

Paso 4: Ampliar el PV

Para cambiar el tamaño del PV creado de 1Gi a 2Gi, edite la definición del PVC y actualícela. `spec.resources.requests.storage` a 2Gi.

```
kubectl edit pvc san-pvc
```

```
# Please edit the object below. Lines beginning with a '#' will be
ignored,
# and an empty file will abort the edit. If an error occurs while saving
this file will be
# reopened with the relevant failures.
#
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  annotations:
    pv.kubernetes.io/bind-completed: "yes"
    pv.kubernetes.io/bound-by-controller: "yes"
    volume.beta.kubernetes.io/storage-provisioner: csi.trident.netapp.io
  creationTimestamp: "2019-10-10T17:32:29Z"
  finalizers:
  - kubernetes.io/pvc-protection
  name: san-pvc
  namespace: default
  resourceVersion: "16609"
  selfLink: /api/v1/namespaces/default/persistentvolumeclaims/san-pvc
  uid: 8a814d62-bd58-4253-b0d1-82f2885db671
spec:
  accessModes:
  - ReadWriteOnce
  resources:
    requests:
      storage: 2Gi
 # ...
```

Paso 5: Validar la expansión

Puedes comprobar que la expansión ha funcionado correctamente verificando el tamaño del PVC, el PV y el volumen del Trident :

```
kubectl get pvc san-pvc
NAME      STATUS   VOLUME                                     CAPACITY
ACCESS MODES   STORAGECLASS   AGE
san-pvc   Bound    pvc-8a814d62-bd58-4253-b0d1-82f2885db671   2Gi
RWO            ontap-san      11m
kubectl get pv
NAME                                       CAPACITY   ACCESS MODES
RECLAIM POLICY   STATUS   CLAIM             STORAGECLASS   REASON   AGE
pvc-8a814d62-bd58-4253-b0d1-82f2885db671   2Gi        RWO
Delete           Bound    default/san-pvc   ontap-san               12m
tridentctl get volumes -n trident
+-----------------------------------------+---------+---------------
+----------+-----------------------------------------+--------+---------+
|                   NAME                  |  SIZE   | STORAGE CLASS |
PROTOCOL |             BACKEND UUID              | STATE  | MANAGED |
+-----------------------------------------+---------+---------------
+----------+-----------------------------------------+--------+---------+
| pvc-8a814d62-bd58-4253-b0d1-82f2885db671 | 2.0 GiB | ontap-san     |
block    | a9b7bfff-0505-4e31-b6c5-59f492e02d33 | online | true    |
+-----------------------------------------+---------+---------------
+----------+-----------------------------------------+--------+---------+
```

Expandir un volumen NFS

Trident admite la expansión de volumen para PV NFS aprovisionados en `ontap-nas` , `ontap-nas-economy` , `ontap-nas-flexgroup` , `gcp-cvs` , y `azure-netapp-files` backends.

Paso 1: Configure la StorageClass para que admita la expansión de volumen.

Para cambiar el tamaño de un volumen físico NFS, el administrador primero debe configurar la clase de almacenamiento para permitir la expansión del volumen configurando la `allowVolumeExpansion` campo a `true` :

```
cat storageclass-ontapnas.yaml
```

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

Si ya ha creado una clase de almacenamiento sin esta opción, simplemente puede editar la clase de almacenamiento existente mediante el uso de `kubectl edit storageclass` para permitir la expansión de volumen.

Paso 2: Cree un PVC con la StorageClass que creó.

```
cat pvc-ontapnas.yaml
```

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: ontapnas20mb
spec:
  accessModes:
  - ReadWriteOnce
  resources:
    requests:
      storage: 20Mi
  storageClassName: ontapnas
```

Trident debería crear un PV NFS de 20 MiB para este PVC:

```
kubectl get pvc
NAME           STATUS   VOLUME
CAPACITY     ACCESS MODES   STORAGECLASS    AGE
ontapnas20mb   Bound    pvc-08f3d561-b199-11e9-8d9f-5254004dfdb7   20Mi
RWO            ontapnas        9s

kubectl get pv pvc-08f3d561-b199-11e9-8d9f-5254004dfdb7
NAME                                       CAPACITY   ACCESS MODES
RECLAIM POLICY   STATUS   CLAIM                  STORAGECLASS    REASON
AGE
pvc-08f3d561-b199-11e9-8d9f-5254004dfdb7   20Mi       RWO
Delete           Bound    default/ontapnas20mb   ontapnas
2m42s
```

Paso 3: Ampliar el PV

Para cambiar el tamaño del PV recién creado de 20 MiB a 1 GiB, edite el PVC y configure `spec.resources.requests.storage` hasta 1 GiB:

```
kubectl edit pvc ontapnas20mb
```

```
# Please edit the object below. Lines beginning with a '#' will be
ignored,
# and an empty file will abort the edit. If an error occurs while saving
this file will be
# reopened with the relevant failures.
#
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  annotations:
    pv.kubernetes.io/bind-completed: "yes"
    pv.kubernetes.io/bound-by-controller: "yes"
    volume.beta.kubernetes.io/storage-provisioner: csi.trident.netapp.io
  creationTimestamp: 2018-08-21T18:26:44Z
  finalizers:
  - kubernetes.io/pvc-protection
  name: ontapnas20mb
  namespace: default
  resourceVersion: "1958015"
  selfLink: /api/v1/namespaces/default/persistentvolumeclaims/ontapnas20mb
  uid: c1bd7fa5-a56f-11e8-b8d7-fa163e59eaab
spec:
  accessModes:
  - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
# ...
```

Paso 4: Validar la expansión

Puede comprobar que el cambio de tamaño se ha realizado correctamente verificando el tamaño del PVC, el PV y el volumen de Trident :

```
kubectl get pvc ontapnas20mb
NAME           STATUS   VOLUME
CAPACITY   ACCESS MODES   STORAGECLASS    AGE
ontapnas20mb   Bound    pvc-08f3d561-b199-11e9-8d9f-5254004dfdb7   1Gi
RWO            ontapnas        4m44s

kubectl get pv pvc-08f3d561-b199-11e9-8d9f-5254004dfdb7
NAME                                       CAPACITY   ACCESS MODES
RECLAIM POLICY   STATUS   CLAIM                  STORAGECLASS    REASON
AGE
pvc-08f3d561-b199-11e9-8d9f-5254004dfdb7   1Gi        RWO
Delete           Bound    default/ontapnas20mb   ontapnas
5m35s

tridentctl get volume pvc-08f3d561-b199-11e9-8d9f-5254004dfdb7 -n trident
+-----------------------------------------+---------+---------------
+----------+-----------------------------------------+--------+---------+
|                   NAME                  |  SIZE   | STORAGE CLASS |
PROTOCOL |             BACKEND UUID              | STATE  | MANAGED |
+-----------------------------------------+---------+---------------
+----------+-----------------------------------------+--------+---------+
| pvc-08f3d561-b199-11e9-8d9f-5254004dfdb7 | 1.0 GiB | ontapnas      |
file     | c5a6f6a4-b052-423b-80d4-8fb491a14a22 | online | true    |
+-----------------------------------------+---------+---------------
+----------+-----------------------------------------+--------+---------+
```

volúmenes de importación

Puedes importar volúmenes de almacenamiento existentes como un PV de Kubernetes usando `tridentctl import`.

Descripción general y consideraciones

Podrías importar un volumen a Trident para:

- Conteneriza una aplicación y reutiliza su conjunto de datos existente.
- Utilice un clon de un conjunto de datos para una aplicación efímera.
- Reconstruir un clúster de Kubernetes que ha fallado
- Migrar los datos de la aplicación durante la recuperación ante desastres

Consideraciones

Antes de importar un volumen, revise las siguientes consideraciones.

- Trident solo puede importar volúmenes ONTAP de tipo RW (lectura-escritura). Los volúmenes de tipo DP (protección de datos) son volúmenes de destino SnapMirror . Debes romper la relación de espejo antes de importar el volumen a Trident.

- Recomendamos importar volúmenes sin conexiones activas. Para importar un volumen que se esté utilizando activamente, clone el volumen y luego realice la importación.

  Esto es especialmente importante para los volúmenes en bloque, ya que Kubernetes desconocería la conexión anterior y podría fácilmente adjuntar un volumen activo a un pod. Esto puede provocar corrupción de datos.

- Aunque `StorageClass` Debe especificarse en un PVC; Trident no utiliza este parámetro durante la importación. Las clases de almacenamiento se utilizan durante la creación de volúmenes para seleccionar entre los grupos disponibles en función de las características de almacenamiento. Dado que el volumen ya existe, no es necesario seleccionar ningún pool durante la importación. Por lo tanto, la importación no fallará incluso si el volumen existe en un backend o pool que no coincide con la clase de almacenamiento especificada en el PVC.
- El volumen existente se determina y se establece en el PVC. Una vez que el controlador de almacenamiento importa el volumen, se crea el PV con una ClaimRef al PVC.
  - La política de reclamaciones está inicialmente establecida para `retain` en el PV. Una vez que Kubernetes enlaza correctamente el PVC y el PV, la política de recuperación se actualiza para que coincida con la política de recuperación de la clase de almacenamiento.
  - Si la política de recuperación de la Clase de Almacenamiento es `delete` El volumen de almacenamiento se eliminará cuando se elimine el PV.
- Por defecto, Trident gestiona el PVC y cambia el nombre del FlexVol volume y del LUN en el backend. Puedes pasar el `--no-manage` bandera para importar un volumen no administrado. Si utilizas `--no-manage` Trident no realiza ninguna operación adicional en el PVC o PV durante el ciclo de vida de los objetos. El volumen de almacenamiento no se elimina cuando se elimina el PV y otras operaciones como la clonación y el cambio de tamaño del volumen también se ignoran.

  Esta opción resulta útil si desea utilizar Kubernetes para cargas de trabajo en contenedores, pero por lo demás desea gestionar el ciclo de vida del volumen de almacenamiento fuera de Kubernetes.

- Se añade una anotación al PVC y al PV que cumple una doble función: indicar si el volumen fue importado y si el PVC y el PV están gestionados. Esta anotación no debe modificarse ni eliminarse.

Importar un volumen

Puedes utilizar `tridentctl import` para importar un volumen.

Pasos

1. Cree el archivo de reclamación de volumen persistente (PVC) (por ejemplo, `pvc.yaml`) que se utilizará para crear el PVC. El archivo de PVC debe incluir `name` , `namespace` , `accessModes` , y `storageClassName` . Opcionalmente, puede especificar `unixPermissions` en su definición de PVC.

   El siguiente es un ejemplo de una especificación mínima:

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: my_claim
  namespace: my_namespace
spec:
  accessModes:
    - ReadWriteOnce
  storageClassName: my_storage_class
```

No incluya parámetros adicionales como el nombre del PV o el tamaño del volumen. Esto puede provocar que falle el comando de importación.

2. Utilice el `tridentctl import` comando para especificar el nombre del backend de Trident que contiene el volumen y el nombre que identifica de forma única el volumen en el almacenamiento (por ejemplo: ONTAP FlexVol, Element Volume, ruta del Cloud Volumes Service). El `-f` Se requiere un argumento para especificar la ruta al archivo PVC.

```
tridentctl import volume <backendName> <volumeName> -f <path-to-pvc-
file>
```

Ejemplos

Revise los siguientes ejemplos de importación de volumen para los controladores compatibles.

ONTAP NAS y ONTAP NAS FlexGroup

Trident admite la importación de volúmenes mediante el `ontap-nas` y `ontap-nas-flexgroup` conductores.

- Trident no admite la importación de volumen mediante el `ontap-nas-economy` conductor.
- El `ontap-nas` y `ontap-nas-flexgroup` Los controladores no permiten nombres de volumen duplicados.

Cada volumen creado con el `ontap-nas` El controlador es un FlexVol volume en el clúster ONTAP . Importación de volúmenes FlexVol con el `ontap-nas` El controlador funciona igual. Un volumen FlexVol que ya existe en un clúster ONTAP se puede importar como un `ontap-nas` CLORURO DE POLIVINILO. De forma similar, los volúmenes de FlexGroup se pueden importar como `ontap-nas-flexgroup` PVC.

Ejemplos de ONTAP NAS

A continuación se muestra un ejemplo de importación de un volumen administrado y un volumen no administrado.

Volumen gestionado

El siguiente ejemplo importa un volumen llamado `managed_volume` en un backend llamado `ontap_nas` :

```
tridentctl import volume ontap_nas managed_volume -f <path-to-pvc-file>

+-----------------------------------------+---------+---------------
+----------+-----------------------------------------+--------+---------+
|                   NAME                  |  SIZE   | STORAGE CLASS |
PROTOCOL |             BACKEND UUID              | STATE  | MANAGED |
+-----------------------------------------+---------+---------------
+----------+-----------------------------------------+--------+---------+
| pvc-bf5ad463-afbb-11e9-8d9f-5254004dfdb7 | 1.0 GiB | standard      |
file     | c5a6f6a4-b052-423b-80d4-8fb491a14a22 | online | true    |
+-----------------------------------------+---------+---------------
+----------+-----------------------------------------+--------+---------+
```

Volumen no gestionado

Al usar el `--no-manage` argumento, Trident no cambia el nombre del volumen.

El siguiente ejemplo importa `unmanaged_volume` en el `ontap_nas` backend:

```
tridentctl import volume nas_blog unmanaged_volume -f <path-to-pvc-
file> --no-manage

+-----------------------------------------+---------+---------------
+----------+-----------------------------------------+--------+---------+
|                   NAME                  |  SIZE   | STORAGE CLASS |
PROTOCOL |             BACKEND UUID              | STATE  | MANAGED |
+-----------------------------------------+---------+---------------
+----------+-----------------------------------------+--------+---------+
| pvc-df07d542-afbc-11e9-8d9f-5254004dfdb7 | 1.0 GiB | standard      |
file     | c5a6f6a4-b052-423b-80d4-8fb491a14a22 | online | false   |
+-----------------------------------------+---------+---------------
+----------+-----------------------------------------+--------+---------+
```

ONTAP SAN

Trident admite la importación de volumen mediante el `ontap-san` (iSCSI, NVMe/TCP y FC) y `ontap-san-economy` Conductores.

Trident puede importar volúmenes ONTAP SAN FlexVol que contengan un solo LUN. Esto es coherente con la `ontap-san` controlador, que crea un FlexVol volume para cada PVC y un LUN dentro del FlexVol volume. Trident importa el FlexVol volume y lo asocia con la definición de PVC. Trident puede importar `ontap-san-`

`economy` volúmenes que contienen múltiples LUN.

Ejemplos de ONTAP SAN

A continuación se muestra un ejemplo de importación de un volumen administrado y un volumen no administrado.

Volumen gestionado

Para los volúmenes administrados, Trident cambia el nombre del FlexVol volume a `pvc-<uuid>` formato y el LUN dentro del FlexVol volume a `lun0` .

El siguiente ejemplo importa el `ontap-san-managed` FlexVol volume que está presente en el `ontap_san_default` backend:

```
tridentctl import volume ontapsan_san_default ontap-san-managed -f pvc-
basic-import.yaml -n trident -d

+----------------------------------------+--------+---------------
+----------+----------------------------------------+--------+---------+
|                   NAME                 |  SIZE  | STORAGE CLASS |
PROTOCOL |             BACKEND UUID             | STATE  | MANAGED |
+----------------------------------------+--------+---------------
+----------+----------------------------------------+--------+---------+
| pvc-d6ee4f54-4e40-4454-92fd-d00fc228d74a | 20 MiB | basic         |
block    | cd394786-ddd5-4470-adc3-10c5ce4ca757 | online | true    |
+----------------------------------------+--------+---------------
+----------+----------------------------------------+--------+---------+
```

Volumen no gestionado

El siguiente ejemplo importa `unmanaged_example_volume` en el `ontap_san` backend:

```
tridentctl import volume -n trident san_blog unmanaged_example_volume
-f pvc-import.yaml --no-manage
+----------------------------------------+---------+---------------
+----------+----------------------------------------+--------+---------+
|                   NAME                 |  SIZE   | STORAGE CLASS |
PROTOCOL |             BACKEND UUID             | STATE  | MANAGED |
+----------------------------------------+---------+---------------
+----------+----------------------------------------+--------+---------+
| pvc-1fc999c9-ce8c-459c-82e4-ed4380a4b228 | 1.0 GiB | san-blog      |
block    | e3275890-7d80-4af6-90cc-c7a0759f555a | online | false   |
+----------------------------------------+---------+---------------
+----------+----------------------------------------+--------+---------+
```

Si tiene LUNS asignados a igroups que comparten un IQN con un IQN de nodo de Kubernetes, como se

muestra en el siguiente ejemplo, recibirá el error: `LUN already mapped to initiator(s) in this group` . Deberá eliminar el iniciador o desasignar el LUN para importar el volumen.

```
Vserver   Igroup       Protocol OS Type  Initiators
--------- ------------ -------- -------- ------------------------------------
svm0      k8s-nodename.example.com-fe5d36f2-cded-4f38-9eb0-c7719fc2f9f3
                       iscsi    linux    iqn.1994-05.com.redhat:4c2e1cf35e0
svm0      unmanaged-example-igroup
                       mixed    linux    iqn.1994-05.com.redhat:4c2e1cf35e0
```

Elemento

Trident admite el software NetApp Element y la importación de volúmenes NetApp HCI mediante `solidfire-san` conductor.

> El controlador Element admite nombres de volumen duplicados. Sin embargo, Trident devuelve un error si hay nombres de volumen duplicados. Como solución alternativa, clone el volumen, asígnele un nombre único e importe el volumen clonado.

Ejemplo de elemento

El siguiente ejemplo importa un `element-managed` volumen en el backend `element_default` .

```
tridentctl import volume element_default element-managed -f pvc-basic-
import.yaml -n trident -d

+----------------------------------------+--------+---------------
+----------+----------------------------------------+--------+---------+
|                   NAME                 |  SIZE  | STORAGE CLASS |
PROTOCOL |             BACKEND UUID              | STATE  | MANAGED |
+----------------------------------------+--------+---------------
+----------+----------------------------------------+--------+---------+
| pvc-970ce1ca-2096-4ecd-8545-ac7edc24a8fe | 10 GiB | basic-element |
block    | d3ba047a-ea0b-43f9-9c42-e38e58301c49 | online | true    |
+----------------------------------------+--------+---------------
+----------+----------------------------------------+--------+---------+
```

Plataforma de Google Cloud

Trident admite la importación de volúmenes mediante el `gcp-cvs` conductor.

Para importar un volumen respaldado por el Cloud Volumes Service NetApp Cloud Volumes en Google Cloud Platform, identifique el volumen por su ruta de acceso. La ruta del volumen es la porción de la ruta de exportación del volumen después de `:/` . Por ejemplo, si la ruta de exportación es `10.0.0.1:/adroit-jolly-swift` , la ruta del volumen es `adroit-jolly-swift` .

Ejemplo de Google Cloud Platform

El siguiente ejemplo importa un `gcp-cvs` volumen en el backend `gcpcvs_YEppr` con la ruta de volumen de `adroit-jolly-swift` .

```
tridentctl import volume gcpcvs_YEppr adroit-jolly-swift -f <path-to-pvc-
file> -n trident

+------------------------------------------+--------+---------------
+----------+------------------------------------------+--------+---------+
|                   NAME                   |  SIZE  | STORAGE CLASS |
PROTOCOL |              BACKEND UUID              | STATE  | MANAGED |
+------------------------------------------+--------+---------------
+----------+------------------------------------------+--------+---------+
| pvc-a46ccab7-44aa-4433-94b1-e47fc8c0fa55 | 93 GiB | gcp-storage   | file
| e1a6e65b-299e-4568-ad05-4f0a105c888f | online | true    |
+------------------------------------------+--------+---------------
+----------+------------------------------------------+--------+---------+
```

Azure NetApp Files

Trident admite la importación de volúmenes mediante el `azure-netapp-files` conductor.

Para importar un volumen de Azure NetApp Files , identifique el volumen por su ruta de acceso. La ruta del volumen es la porción de la ruta de exportación del volumen después de `:/` . Por ejemplo, si la ruta de montaje es `10.0.0.2:/importvol1` , la ruta del volumen es `importvol1` .

Ejemplo de Azure NetApp Files

El siguiente ejemplo importa un `azure-netapp-files` volumen en el backend `azurenetappfiles_40517` con la ruta del volumen `importvol1` .

```
tridentctl import volume azurenetappfiles_40517 importvol1 -f <path-to-
pvc-file> -n trident

+----------------------------------------+---------+--------------
+----------+----------------------------------------+--------+---------+
|                   NAME                 |  SIZE   | STORAGE CLASS |
PROTOCOL |             BACKEND UUID             | STATE  | MANAGED |
+----------------------------------------+---------+--------------
+----------+----------------------------------------+--------+---------+
| pvc-0ee95d60-fd5c-448d-b505-b72901b3a4ab | 100 GiB | anf-storage   |
file     | 1c01274f-d94b-44a3-98a3-04c953c9a51e | online | true    |
+----------------------------------------+---------+--------------
+----------+----------------------------------------+--------+---------+
```

Google Cloud NetApp Volumes

Trident admite la importación de volúmenes mediante el `google-cloud-netapp-volumes` conductor.

Ejemplo de Google Cloud NetApp Volumes

El siguiente ejemplo importa un `google-cloud-netapp-volumes` volumen en el backend `backend-tbc-gcnv1` con el volumen `testvoleasiaeast1` .

```
tridentctl import volume backend-tbc-gcnv1 "testvoleasiaeast1" -f < path-
to-pvc> -n trident

+----------------------------------------+---------
+----------------------+----------+--------------------------------------
+--------+---------+
|                   NAME                 |  SIZE   | STORAGE CLASS
| PROTOCOL |             BACKEND UUID             | STATE  | MANAGED |
+----------------------------------------+---------
+----------------------+----------+--------------------------------------
+--------+---------+
| pvc-a69cda19-218c-4ca9-a941-aea05dd13dc0 |  10 GiB | gcnv-nfs-sc-
identity | file     | 8c18cdf1-0770-4bc0-bcc5-c6295fe6d837 | online | true
|
+----------------------------------------+---------
+----------------------+----------+--------------------------------------
+--------+---------+
```

El siguiente ejemplo importa un `google-cloud-netapp-volumes` volumen cuando hay dos volúmenes presentes en la misma región:

```
tridentctl import volume backend-tbc-gcnv1
"projects/123456789100/locations/asia-east1-a/volumes/testvoleasiaeast1"
-f <path-to-pvc> -n trident

+------------------------------------------+---------
+----------------------+----------+--------------------------------------
+--------+---------+
|                   NAME                   |  SIZE   | STORAGE CLASS
| PROTOCOL |             BACKEND UUID             | STATE  | MANAGED |
+------------------------------------------+---------
+----------------------+----------+--------------------------------------
+--------+---------+
| pvc-a69cda19-218c-4ca9-a941-aea05dd13dc0 |  10 GiB | gcnv-nfs-sc-
identity | file     | 8c18cdf1-0770-4bc0-bcc5-c6295fe6d837 | online | true
|
+------------------------------------------+---------
+----------------------+----------+--------------------------------------
+--------+---------+
```

Personaliza los nombres y etiquetas de los volúmenes.

Con Trident, puedes asignar nombres y etiquetas descriptivos a los volúmenes que crees. Esto te ayuda a identificar y asignar fácilmente los volúmenes a sus respectivos recursos de Kubernetes (PVC). También puede definir plantillas a nivel de backend para crear nombres de volumen personalizados y etiquetas personalizadas; cualquier volumen que cree, importe o clone se ajustará a las plantillas.

Antes de empezar

Compatibilidad con nombres y etiquetas de volumen personalizables:

1. Operaciones de creación, importación y clonación de volúmenes.
2. En el caso del controlador ontap-nas-economy, solo el nombre del volumen Qtree cumple con la plantilla de nombre.
3. En el caso del controlador ontap-san-economy, solo el nombre del LUN cumple con la plantilla de nombre.

Limitaciones

1. Los nombres de volumen personalizables solo son compatibles con los controladores ONTAP locales.
2. Los nombres de volumen personalizables no se aplican a los volúmenes existentes.

Comportamientos clave de los nombres de volumen personalizables

1. Si se produce un fallo debido a una sintaxis no válida en una plantilla de nombre, falla la creación del backend. Sin embargo, si falla la aplicación de la plantilla, el volumen se nombrará de acuerdo con la convención de nomenclatura existente.

2. El prefijo de almacenamiento no es aplicable cuando un volumen se nombra utilizando una plantilla de nombre de la configuración del backend. Se puede añadir directamente a la plantilla cualquier valor de prefijo deseado.

Ejemplos de configuración de backend con plantilla de nombre y etiquetas

Se pueden definir plantillas de nombres personalizadas a nivel de raíz y/o de grupo.

Ejemplo de nivel raíz

```
{
  "version": 1,
  "storageDriverName": "ontap-nas",
  "backendName": "ontap-nfs-backend",
  "managementLIF": "<ip address>",
  "svm": "svm0",
  "username": "<admin>",
  "password": "<password>",
  "defaults": {
    "nameTemplate":
"{{.volume.Name}}_{{.labels.cluster}}_{{.volume.Namespace}}_{{.volume.Requ
estName}}"
  },
  "labels": {
    "cluster": "ClusterA",
    "PVC": "{{.volume.Namespace}}_{{.volume.RequestName}}"
  }
}
```

Ejemplo de nivel de piscina

```
{
  "version": 1,
  "storageDriverName": "ontap-nas",
  "backendName": "ontap-nfs-backend",
  "managementLIF": "<ip address>",
  "svm": "svm0",
  "username": "<admin>",
  "password": "<password>",
  "useREST": true,
  "storage": [
    {
      "labels": {
        "labelname": "label1",
        "name": "{{ .volume.Name }}"
      },
      "defaults": {
        "nameTemplate": "pool01_{{ .volume.Name }}_{{ .labels.cluster
}}_{{ .volume.Namespace }}_{{ .volume.RequestName }}"
      }
    },
    {
      "labels": {
        "cluster": "label2",
        "name": "{{ .volume.Name }}"
      },
      "defaults": {
        "nameTemplate": "pool02_{{ .volume.Name }}_{{ .labels.cluster
}}_{{ .volume.Namespace }}_{{ .volume.RequestName }}"
      }
    }
  ]
}
```

Ejemplos de plantillas de nombres

Ejemplo 1:

```
"nameTemplate": "{{ .config.StoragePrefix }}_{{ .volume.Name }}_{{
.config.BackendName }}"
```

Ejemplo 2:

```
"nameTemplate": "pool_{{ .config.StoragePrefix }}_{{ .volume.Name }}_{{
slice .volume.RequestName 1 5 }}""
```

Puntos a considerar

1. En el caso de importaciones de volúmenes, las etiquetas se actualizan solo si el volumen existente tiene etiquetas en un formato específico. Por ejemplo: `{"provisioning":{"Cluster":"ClusterA", "PVC": "pvcname"}}`.
2. En el caso de importaciones de volúmenes gestionados, el nombre del volumen sigue la plantilla de nombre definida en el nivel raíz de la definición del backend.
3. Trident no admite el uso de un operador de segmentación con el prefijo de almacenamiento.
4. Si las plantillas no generan nombres de volumen únicos, Trident añadirá algunos caracteres aleatorios para crear nombres de volumen únicos.
5. Si el nombre personalizado de un volumen NAS economy supera los 64 caracteres, Trident nombrará los volúmenes según la convención de nomenclatura existente. Para todos los demás controladores ONTAP , si el nombre del volumen excede el límite de nombres, el proceso de creación del volumen falla.

Compartir un volumen NFS entre espacios de nombres

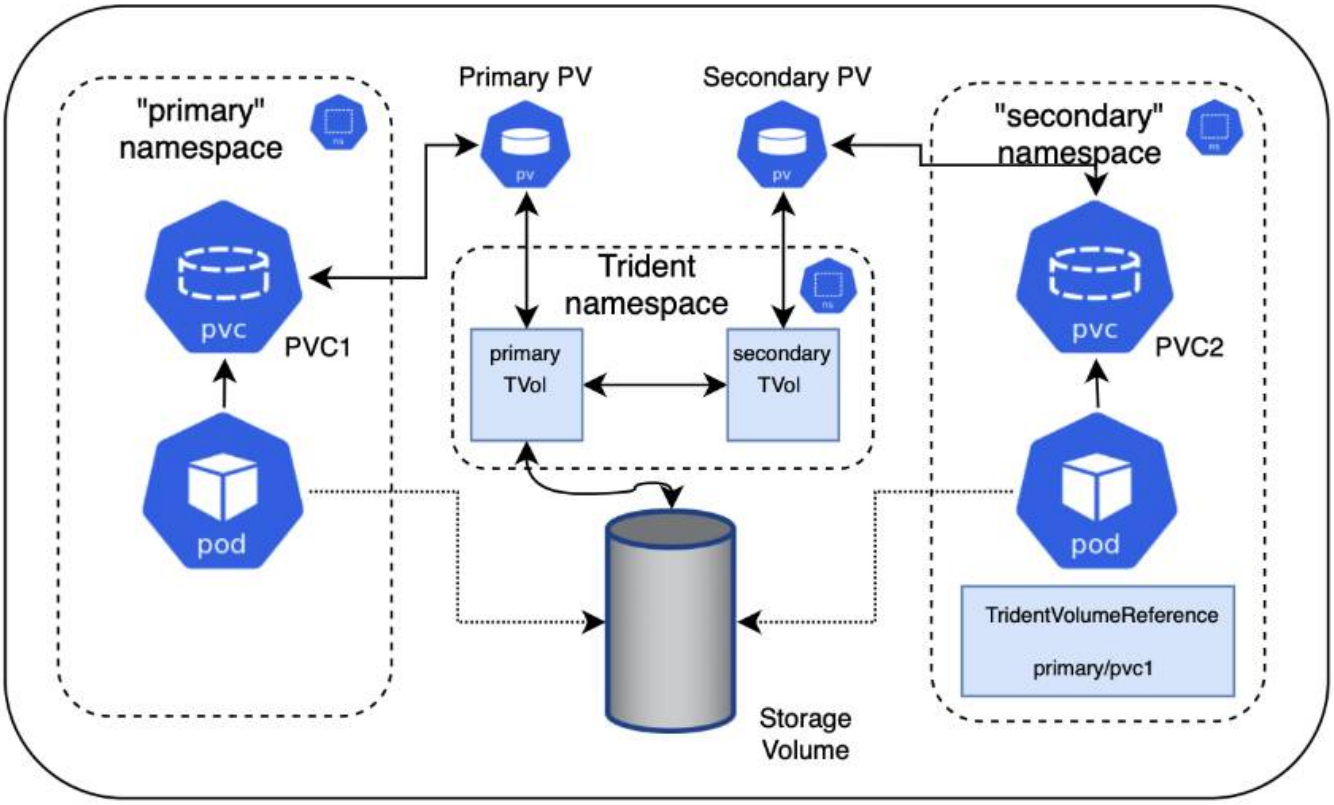
Con Trident, puedes crear un volumen en un espacio de nombres primario y compartirlo en uno o más espacios de nombres secundarios.

Funciones

El CR TridentVolumeReference le permite compartir de forma segura volúmenes NFS ReadWriteMany (RWX) a través de uno o más espacios de nombres de Kubernetes. Esta solución nativa de Kubernetes tiene las siguientes ventajas:

- Múltiples niveles de control de acceso para garantizar la seguridad
- Funciona con todos los controladores de volumen Trident NFS.
- No depende de tridentctl ni de ninguna otra función no nativa de Kubernetes.

Este diagrama ilustra el uso compartido de volúmenes NFS entre dos espacios de nombres de Kubernetes.



Inicio rápido

Puedes configurar el uso compartido de volúmenes NFS en tan solo unos pocos pasos.

1 **Configure el PVC de origen para compartir el volumen.**

El propietario del espacio de nombres de origen otorga permiso para acceder a los datos en el PVC de origen.

2 **Conceder permiso para crear un CR en el espacio de nombres de destino**

El administrador del clúster otorga permiso al propietario del espacio de nombres de destino para crear el CR TridentVolumeReference.

3 **Cree una referencia TridentVolumeReference en el espacio de nombres de destino.**

El propietario del espacio de nombres de destino crea el CR TridentVolumeReference para hacer referencia al PVC de origen.

4 **Cree el PVC subordinado en el espacio de nombres de destino.**

El propietario del espacio de nombres de destino crea el PVC subordinado para utilizar la fuente de datos del PVC de origen.

Configurar los espacios de nombres de origen y destino

Para garantizar la seguridad, el uso compartido entre espacios de nombres requiere la colaboración y la acción del propietario del espacio de nombres de origen, el administrador del clúster y el propietario del espacio de nombres de destino. El rol del usuario se designa en cada paso.

Pasos

1. **Propietario del espacio de nombres de origen:** Crear el PVC(`pvc1`) en el espacio de nombres de origen que otorga permiso para compartir con el espacio de nombres de destino(`namespace2`) usando el `shareToNamespace` anotación.

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: pvc1
  namespace: namespace1
  annotations:
    trident.netapp.io/shareToNamespace: namespace2
spec:
  accessModes:
    - ReadWriteMany
  storageClassName: trident-csi
  resources:
    requests:
      storage: 100Gi
```

   Trident crea el PV y su volumen de almacenamiento NFS de backend.

   - Puedes compartir el PVC con múltiples espacios de nombres utilizando una lista delimitada por comas. Por ejemplo, `trident.netapp.io/shareToNamespace: namespace2,namespace3,namespace4` .
   - Puedes compartir con todos los espacios de nombres usando `*` . Por ejemplo, `trident.netapp.io/shareToNamespace: *`
   - Puedes actualizar el PVC para incluir el `shareToNamespace` anotaciones en cualquier momento.

2. **Administrador del clúster:** Asegúrese de que exista un RBAC adecuado para otorgar permiso al propietario del espacio de nombres de destino para crear la CR TridentVolumeReference en el espacio de nombres de destino.
3. **Propietario del espacio de nombres de destino:** Cree un CR TridentVolumeReference en el espacio de nombres de destino que haga referencia al espacio de nombres de origen. `pvc1` .

```
apiVersion: trident.netapp.io/v1
kind: TridentVolumeReference
metadata:
  name: my-first-tvr
  namespace: namespace2
spec:
  pvcName: pvc1
  pvcNamespace: namespace1
```

4. **Propietario del espacio de nombres de destino:** Crear un PVC(`pvc2`) en el espacio de nombres de destino(`namespace2`) usando el `shareFromPVC` Anotación para designar el PVC de origen.

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  annotations:
    trident.netapp.io/shareFromPVC: namespace1/pvc1
  name: pvc2
  namespace: namespace2
spec:
  accessModes:
    - ReadWriteMany
  storageClassName: trident-csi
  resources:
    requests:
      storage: 100Gi
```

> El tamaño del tubo de PVC de destino debe ser menor o igual que el del tubo de PVC de origen.

Resultados

Trident lee el `shareFromPVC` Se realiza una anotación en el PVC de destino y se crea el PV de destino como un volumen subordinado sin recursos de almacenamiento propios que apunta al PV de origen y comparte los recursos de almacenamiento del PV de origen. Los valores de destino PVC y PV aparecen vinculados con normalidad.

Eliminar un volumen compartido

Puedes eliminar un volumen que se comparte entre varios espacios de nombres. Trident eliminará el acceso al volumen en el espacio de nombres de origen y mantendrá el acceso para otros espacios de nombres que compartan el volumen. Cuando se eliminan todos los espacios de nombres que hacen referencia al volumen, Trident borra el volumen.

Usar `tridentctl get` para consultar volúmenes subordinados

Utilizando el[`tridentctl` utilidad, puedes ejecutar la `get` comando para obtener volúmenes subordinados. Para obtener más información, consulte el siguiente enlace: ../trident-reference/tridentctl.html[`tridentctl` comandos y opciones].

```
Usage:
  tridentctl get [option]
```

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.

Limitaciones

- Trident no puede impedir que los espacios de nombres de destino escriban en el volumen compartido. Debe utilizar el bloqueo de archivos u otros procesos para evitar la sobrescritura de datos de volúmenes compartidos.
- No se puede revocar el acceso al PVC de origen eliminando el `shareToNamespace` o `shareFromNamespace` anotaciones o eliminar las `TridentVolumeReference` CR. Para revocar el acceso, debe eliminar el PVC subordinado.
- Las instantáneas, los clones y la duplicación no son posibles en volúmenes subordinados.

Para más información

Para obtener más información sobre el acceso a volúmenes entre espacios de nombres:

- Visita"Compartir volúmenes entre espacios de nombres: Descubra el acceso a volúmenes entre espacios de nombres." .
- Vea la demostración en"NetAppTV" .

Clonar volúmenes entre espacios de nombres

Con Trident, puedes crear nuevos volúmenes utilizando volúmenes existentes o instantáneas de volúmenes de un espacio de nombres diferente dentro del mismo clúster de Kubernetes.

Prerrequisitos

Antes de clonar volúmenes, asegúrese de que los backends de origen y destino sean del mismo tipo y tengan la misma clase de almacenamiento.

> La clonación entre espacios de nombres solo se admite para `ontap-san` y `ontap-nas` Controladores de almacenamiento. No se admiten clones de solo lectura.

Inicio rápido

Puedes configurar la clonación de volúmenes en tan solo unos pasos.

Configure el PVC de origen para clonar el volumen.

El propietario del espacio de nombres de origen otorga permiso para acceder a los datos en el PVC de origen.

Conceder permiso para crear un CR en el espacio de nombres de destino

El administrador del clúster otorga permiso al propietario del espacio de nombres de destino para crear el CR TridentVolumeReference.

Cree una referencia TridentVolumeReference en el espacio de nombres de destino.

El propietario del espacio de nombres de destino crea el CR TridentVolumeReference para hacer referencia al PVC de origen.

Cree el PVC clonado en el espacio de nombres de destino.

El propietario del espacio de nombres de destino crea un PVC para clonar el PVC del espacio de nombres de origen.

Configurar los espacios de nombres de origen y destino

Para garantizar la seguridad, la clonación de volúmenes entre espacios de nombres requiere la colaboración y la acción del propietario del espacio de nombres de origen, el administrador del clúster y el propietario del espacio de nombres de destino. El rol del usuario se designa en cada paso.

Pasos

1. **Propietario del espacio de nombres de origen:** Crear el PVC(`pvc1`) en el espacio de nombres de origen(`namespace1`) que otorga permiso para compartir con el espacio de nombres de destino(`namespace2`) usando el `cloneToNamespace` anotación.

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: pvc1
  namespace: namespace1
  annotations:
    trident.netapp.io/cloneToNamespace: namespace2
spec:
  accessModes:
    - ReadWriteMany
  storageClassName: trident-csi
  resources:
    requests:
      storage: 100Gi
```

Trident crea el PV y su volumen de almacenamiento backend.

- Puedes compartir el PVC con múltiples espacios de nombres utilizando una lista delimitada por comas. Por ejemplo, `trident.netapp.io/cloneToNamespace: namespace2,namespace3,namespace4` .
- Puedes compartir con todos los espacios de nombres usando `*` . Por ejemplo, `trident.netapp.io/cloneToNamespace: *`
- Puedes actualizar el PVC para incluir el `cloneToNamespace` anotaciones en cualquier momento.

2. **Administrador del clúster:** Asegúrese de que el control de acceso basado en roles (RBAC) esté configurado correctamente para otorgar permiso al propietario del espacio de nombres de destino para crear el recurso compartido TridentVolumeReference en el espacio de nombres de destino.(`namespace2`).
3. **Propietario del espacio de nombres de destino:** Cree un CR TridentVolumeReference en el espacio de nombres de destino que haga referencia al espacio de nombres de origen. `pvc1` .

```
apiVersion: trident.netapp.io/v1
kind: TridentVolumeReference
metadata:
  name: my-first-tvr
  namespace: namespace2
spec:
  pvcName: pvc1
  pvcNamespace: namespace1
```

4. **Propietario del espacio de nombres de destino:** Crear un PVC(`pvc2`) en el espacio de nombres de destino(`namespace2`) usando el `cloneFromPVC` o `cloneFromSnapshot` , y `cloneFromNamespace` Anotaciones para designar el PVC de origen.

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  annotations:
    trident.netapp.io/cloneFromPVC: pvc1
    trident.netapp.io/cloneFromNamespace: namespace1
  name: pvc2
  namespace: namespace2
spec:
  accessModes:
    - ReadWriteMany
  storageClassName: trident-csi
  resources:
    requests:
      storage: 100Gi
```

Limitaciones

- Para los PVC aprovisionados mediante controladores ontap-nas-economy, no se admiten clones de solo lectura.

Replicar volúmenes usando SnapMirror

Trident admite relaciones de espejo entre un volumen de origen en un clúster y el volumen de destino en el clúster emparejado para replicar datos para la recuperación ante desastres. Puede utilizar una definición de recurso personalizado (CRD) con espacio de nombres, denominada relación de espejo de Trident (TMR), para realizar las siguientes operaciones:

- Crear relaciones de simetría entre volúmenes (PVC)
- Eliminar las relaciones de simetría entre volúmenes
- Rompe las relaciones espejo
- Promover el volumen secundario durante situaciones de desastre (conmutaciones por error).
- Realizar una transición sin pérdidas de aplicaciones de un clúster a otro (durante conmutaciones por error o migraciones planificadas).

Requisitos previos de replicación

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

Clústeres ONTAP

- * Trident*: Debe existir la versión 22.10 o posterior de Trident tanto en los clústeres de Kubernetes de origen como de destino que utilizan ONTAP como backend.
- **Licencias**: Las licencias asíncronas de ONTAP SnapMirror que utilizan el paquete de protección de datos deben estar habilitadas tanto en el clúster ONTAP de origen como en el de destino. Referirse a "Información general sobre 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 de NetApp (NLF), que es un único archivo que habilita múltiples funciones. Referirse a"Licencias incluidas con ONTAP One" Para más información.

> Solo se admite la protección asíncrona de SnapMirror .

Mirando

- **Clúster y SVM**: Los backends de almacenamiento ONTAP deben estar interconectados. Referirse a "Descripción general del emparejamiento de clústeres y SVM" Para más información.

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

- * Trident y SVM*: Las SVM remotas emparejadas deben estar disponibles para Trident en el clúster de destino.

Controladores compatibles

NetApp Trident admite la replicación de volúmenes con la tecnología NetApp SnapMirror mediante clases de almacenamiento compatibles con los siguientes controladores: **`ontap-nas` : NFS** `ontap-san` : iSCSI **`ontap-san` : FC** `ontap-san` : NVMe/TCP (requiere como mínimo la versión 9.15.1 de ONTAP)

> La replicación de volúmenes mediante SnapMirror no es compatible con los sistemas ASA r2. Para obtener información sobre los sistemas ASA r2, consulte"Conozca los sistemas de almacenamiento ASA r2" .

Crea un PVC espejado

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

Pasos

1. Realice los siguientes pasos en el clúster principal de Kubernetes:
   a. Crea 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 un PVC con la StorageClass creada previamente.

Ejemplo

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

c. Cree un CR MirrorRelationship con información local.

Ejemplo

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

Trident recupera la información interna del volumen y el estado actual de protección de datos (DP) del volumen, y luego completa el campo de estado de MirrorRelationship.

d. Obtenga el CR TridentMirrorRelationship para obtener el nombre interno y el SVM del 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 secundario de Kubernetes:
   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 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"
      ```

Trident creará una relación SnapMirror con el nombre de política de relación configurado (o el predeterminado para ONTAP) y la inicializará.

c. Cree un PVC con la StorageClass creada previamente para que actúe como destino secundario (destino 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
```

Trident comprobará la existencia del CRD TridentMirrorRelationship y no podrá crear el volumen si la relación no existe. Si existe la relación, Trident se asegurará de que el nuevo FlexVol volume se coloque en una SVM que esté emparejada con la SVM remota definida en MirrorRelationship.

Estados de replicación de volumen

Una relación de espejo Trident (TMR) es un CRD que representa un extremo de una relación de replicación entre PVC. El TMR de destino tiene un estado, que le indica a Trident 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 de lectura/escritura y se puede montar, sin ninguna relación de espejo actualmente en vigor.
- **Restablecido**: el PVC local es el volumen de destino de una relación de espejo y también estuvo previamente en esa relación de espejo.
  - Debe utilizarse el estado restablecido si el volumen de destino alguna vez estuvo relacionado con el volumen de origen, ya que sobrescribe el contenido del volumen de destino.
  - El estado restablecido fallará si el volumen no estaba previamente relacionado con la fuente.

Promover la PVC secundaria durante una conmutación por error no planificada

Realice el siguiente paso en el clúster de Kubernetes secundario:

- Actualizar el campo *spec.state* de TridentMirrorRelationship a `promoted` .

Promover la PVC secundaria durante una conmutación por error planificada

Durante una conmutación por error planificada (migración), realice los siguientes pasos para promover el PVC secundario:

Pasos

1. En el clúster principal de Kubernetes, cree una instantánea del PVC y espere hasta que se cree la instantánea.
2. En el clúster principal de Kubernetes, cree el CR SnapshotInfo para obtener detalles internos.

   Ejemplo

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

3. En el clúster secundario de Kubernetes, actualice el campo *spec.state* del CR *TridentMirrorRelationship* a *promoted* y *spec.promotedSnapshotHandle* para que sea el internalName de la instantánea.
4. En el clúster secundario de Kubernetes, confirme que el estado (campo status.state) de TridentMirrorRelationship sea promovido.

Restablecer una relación de espejo después de una conmutación por error

Antes de restablecer una relación de espejo, elige el lado que quieres convertir en el nuevo principal.

Pasos

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

Operaciones adicionales

Trident admite las siguientes operaciones en los volúmenes primario y secundario:

Replicar el PVC primario a un nuevo PVC secundario

Asegúrese de que ya dispone de un tubo de PVC primario y un tubo de PVC secundario.

Pasos

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

Cambiar el tamaño de un PVC reflejado, primario o secundario

El tamaño del PVC se puede cambiar como de costumbre; ONTAP expandirá automáticamente cualquier destino flevxols si la cantidad de datos excede el tamaño actual.

Eliminar la replicación de un PVC

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

- Elimine la relación MirrorRelationship en el PVC secundario. Esto rompe la relación de replicación.
- O bien, actualice el campo spec.state a *promoted*.

Eliminar un PVC (que previamente se había duplicado)

Trident comprueba si existen PVC replicados y libera la relación de replicación antes de intentar eliminar el volumen.

Eliminar un TMR

Eliminar un TMR en un lado de una relación reflejada provoca que el TMR restante pase al estado *promocionado* antes de que Trident complete la eliminación. Si el TMR seleccionado para su eliminación ya se encuentra en estado *promocionado*, no existe ninguna relación de réplica y el TMR será eliminado y Trident promoverá el PVC local a *Lectura/Escritura*. Esta eliminación libera los metadatos de SnapMirror para el volumen local en ONTAP. Si este volumen se utiliza en una relación de espejo en el futuro, deberá utilizar un nuevo TMR con un estado de replicación de volumen *establecido* al crear la nueva relación de espejo.

Actualizar las relaciones de espejo cuando ONTAP esté en línea

Las relaciones de espejo se pueden actualizar en cualquier momento después de que se hayan establecido. Puedes utilizar el `state: promoted` o `state: reestablished` campos para actualizar las relaciones. Al promover un volumen de destino a un volumen ReadWrite normal, puede usar *promotedSnapshotHandle* para especificar una instantánea específica a la que restaurar el volumen actual.

Actualizar las relaciones de réplica cuando ONTAP esté fuera de línea

Puede utilizar un CRD para realizar una actualización de SnapMirror sin que Trident tenga conectividad directa con el clúster ONTAP . Consulte el siguiente ejemplo de formato 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*.

Utilizar la topología CSI

Trident puede crear y adjuntar volúmenes de forma selectiva a los nodos presentes en un clúster de Kubernetes mediante el uso de "Función de topología CSI" .

Descripción general

Al utilizar la función de topología CSI, el acceso a los volúmenes se puede limitar a un subconjunto de nodos, según las regiones y las zonas de disponibilidad. Actualmente, los proveedores de servicios en la nube permiten a los administradores de Kubernetes crear nodos basados en zonas. Los nodos pueden ubicarse en diferentes zonas de disponibilidad dentro de una región o en varias regiones. Para facilitar el aprovisionamiento de volúmenes para cargas de trabajo en una arquitectura multizona, Trident utiliza la topología CSI.

Obtenga más información sobre la función de topología CSI. "aquí" .

Kubernetes proporciona dos modos únicos de enlace de volúmenes:

- Con `VolumeBindingMode` empezar a `Immediate` Trident crea el volumen sin tener en cuenta la topología. La vinculación de volúmenes y el aprovisionamiento dinámico se gestionan cuando se crea el PVC. Este es el valor predeterminado `VolumeBindingMode` y es adecuado para clústeres que no imponen restricciones de topología. Los volúmenes persistentes se crean sin depender de los requisitos de programación del pod solicitante.
- Con `VolumeBindingMode` empezar a `WaitForFirstConsumer` La creación y el enlace de un volumen persistente para un PVC se retrasan hasta que se programa y se crea un pod que utiliza el PVC. De esta manera, se crean volúmenes para cumplir con las restricciones de programación impuestas por los requisitos de topología.

El `WaitForFirstConsumer` El modo de enlace no requiere etiquetas de topología. Esto se puede utilizar independientemente de la función de topología CSI.

Lo que necesitarás

Para utilizar la topología CSI, necesita lo siguiente:

- Un clúster de Kubernetes que ejecuta un"Versión de Kubernetes compatible"

```
kubectl version
Client Version: version.Info{Major:"1", Minor:"19",
GitVersion:"v1.19.3",
GitCommit:"1e11e4a2108024935ecfcb2912226cedeafd99df",
GitTreeState:"clean", BuildDate:"2020-10-14T12:50:19Z",
GoVersion:"go1.15.2", Compiler:"gc", Platform:"linux/amd64"}
Server Version: version.Info{Major:"1", Minor:"19",
GitVersion:"v1.19.3",
GitCommit:"1e11e4a2108024935ecfcb2912226cedeafd99df",
GitTreeState:"clean", BuildDate:"2020-10-14T12:41:49Z",
GoVersion:"go1.15.2", Compiler:"gc", Platform:"linux/amd64"}
```

- Los nodos del clúster deben tener etiquetas que indiquen la topología.(`topology.kubernetes.io/region` y `topology.kubernetes.io/zone`). Estas etiquetas **deben estar presentes en los nodos del clúster** antes de instalar Trident para que Trident tenga en cuenta la topología.

```
kubectl get nodes -o=jsonpath='{range .items[*]}[{.metadata.name},
{.metadata.labels}]{"\n"}{end}' | grep --color "topology.kubernetes.io"
[node1,
{"beta.kubernetes.io/arch":"amd64","beta.kubernetes.io/os":"linux","kube
rnetes.io/arch":"amd64","kubernetes.io/hostname":"node1","kubernetes.io/
os":"linux","node-
role.kubernetes.io/master":"","topology.kubernetes.io/region":"us-
east1","topology.kubernetes.io/zone":"us-east1-a"}]
[node2,
{"beta.kubernetes.io/arch":"amd64","beta.kubernetes.io/os":"linux","kube
rnetes.io/arch":"amd64","kubernetes.io/hostname":"node2","kubernetes.io/
os":"linux","node-
role.kubernetes.io/worker":"","topology.kubernetes.io/region":"us-
east1","topology.kubernetes.io/zone":"us-east1-b"}]
[node3,
{"beta.kubernetes.io/arch":"amd64","beta.kubernetes.io/os":"linux","kube
rnetes.io/arch":"amd64","kubernetes.io/hostname":"node3","kubernetes.io/
os":"linux","node-
role.kubernetes.io/worker":"","topology.kubernetes.io/region":"us-
east1","topology.kubernetes.io/zone":"us-east1-c"}]
```

Paso 1: Crear un backend que tenga en cuenta la topología

Los sistemas de almacenamiento Trident pueden diseñarse para aprovisionar volúmenes de forma selectiva en función de las zonas de disponibilidad. Cada backend puede incluir una opción. `supportedTopologies` bloque que representa una lista de zonas y regiones admitidas. Para las StorageClasses que utilizan dicho backend, solo se creará un volumen si lo solicita una aplicación programada en una zona o región compatible.

Aquí tenéis un ejemplo de definición de backend:

YAML

```yaml
---
version: 1
storageDriverName: ontap-san
backendName: san-backend-us-east1
managementLIF: 192.168.27.5
svm: iscsi_svm
username: admin
password: password
supportedTopologies:
  - topology.kubernetes.io/region: us-east1
    topology.kubernetes.io/zone: us-east1-a
  - topology.kubernetes.io/region: us-east1
    topology.kubernetes.io/zone: us-east1-b
```

JSON

```json
{
  "version": 1,
  "storageDriverName": "ontap-san",
  "backendName": "san-backend-us-east1",
  "managementLIF": "192.168.27.5",
  "svm": "iscsi_svm",
  "username": "admin",
  "password": "password",
  "supportedTopologies": [
    {
      "topology.kubernetes.io/region": "us-east1",
      "topology.kubernetes.io/zone": "us-east1-a"
    },
    {
      "topology.kubernetes.io/region": "us-east1",
      "topology.kubernetes.io/zone": "us-east1-b"
    }
  ]
}
```

`supportedTopologies`Se utiliza para proporcionar una lista de regiones y zonas por backend. Estas regiones y zonas representan la lista de valores permitidos que se pueden proporcionar en una StorageClass. Para las StorageClasses que contienen un subconjunto de las regiones y zonas proporcionadas en un backend, Trident crea un volumen en el backend.

Puedes definir `supportedTopologies` por cada grupo de almacenamiento también. Vea el siguiente

ejemplo:

```
---
version: 1
storageDriverName: ontap-nas
backendName: nas-backend-us-central1
managementLIF: 172.16.238.5
svm: nfs_svm
username: admin
password: password
supportedTopologies:
  - topology.kubernetes.io/region: us-central1
    topology.kubernetes.io/zone: us-central1-a
  - topology.kubernetes.io/region: us-central1
    topology.kubernetes.io/zone: us-central1-b
storage:
  - labels:
      workload: production
    supportedTopologies:
      - topology.kubernetes.io/region: us-central1
        topology.kubernetes.io/zone: us-central1-a
  - labels:
      workload: dev
    supportedTopologies:
      - topology.kubernetes.io/region: us-central1
        topology.kubernetes.io/zone: us-central1-b
```

En este ejemplo, el `region` y `zone` Las etiquetas indican la ubicación del depósito de almacenamiento. `topology.kubernetes.io/region` y `topology.kubernetes.io/zone` dictar desde dónde se pueden consumir los grupos de almacenamiento.

Paso 2: Defina las StorageClasses que tengan en cuenta la topología.

En función de las etiquetas de topología que se proporcionan a los nodos del clúster, se pueden definir StorageClasses para que contengan información de topología. Esto determinará los grupos de almacenamiento que sirven como candidatos para las solicitudes de PVC realizadas, y el subconjunto de nodos que pueden utilizar los volúmenes aprovisionados por Trident.

Vea el siguiente ejemplo:

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata: null
name: netapp-san-us-east1
provisioner: csi.trident.netapp.io
volumeBindingMode: WaitForFirstConsumer
allowedTopologies:
  - matchLabelExpressions: null
  - key: topology.kubernetes.io/zone
    values:
      - us-east1-a
      - us-east1-b
  - key: topology.kubernetes.io/region
    values:
      - us-east1
parameters:
  fsType: ext4
```

En la definición de StorageClass proporcionada anteriormente, `volumeBindingMode` está configurado para `WaitForFirstConsumer` . Los PVC que se soliciten con esta StorageClass no se procesarán hasta que se haga referencia a ellos en un pod. Y, `allowedTopologies` proporciona las zonas y la región que se utilizarán. El `netapp-san-us-east1` StorageClass crea PVC en el `san-backend-us-east1` Backend definido anteriormente.

Paso 3: Crear y usar un PVC

Una vez creada la StorageClass y asignada a un backend, ya puede crear PVC.

Vea el ejemplo `spec` abajo:

```
---
kind: PersistentVolumeClaim
apiVersion: v1
metadata: null
name: pvc-san
spec: null
accessModes:
  - ReadWriteOnce
resources:
  requests:
    storage: 300Mi
storageClassName: netapp-san-us-east1
```

La creación de un PVC utilizando este manifiesto daría como resultado lo siguiente:

```
kubectl create -f pvc.yaml
persistentvolumeclaim/pvc-san created
kubectl get pvc
NAME      STATUS    VOLUME   CAPACITY   ACCESS MODES   STORAGECLASS
AGE
pvc-san   Pending                                      netapp-san-us-east1
2s
kubectl describe pvc
Name:          pvc-san
Namespace:     default
StorageClass:  netapp-san-us-east1
Status:        Pending
Volume:
Labels:        <none>
Annotations:   <none>
Finalizers:    [kubernetes.io/pvc-protection]
Capacity:
Access Modes:
VolumeMode:    Filesystem
Mounted By:    <none>
Events:
  Type    Reason                Age   From                         Message
  ----    ------                ----  ----                         -------
  Normal  WaitForFirstConsumer  6s    persistentvolume-controller  waiting
for first consumer to be created before binding
```

Para que Trident cree un volumen y lo una al PVC, utilice el PVC en una cápsula. Vea el siguiente ejemplo:

```
apiVersion: v1
kind: Pod
metadata:
  name: app-pod-1
spec:
  affinity:
    nodeAffinity:
      requiredDuringSchedulingIgnoredDuringExecution:
        nodeSelectorTerms:
        - matchExpressions:
          - key: topology.kubernetes.io/region
            operator: In
            values:
            - us-east1
      preferredDuringSchedulingIgnoredDuringExecution:
      - weight: 1
        preference:
          matchExpressions:
          - key: topology.kubernetes.io/zone
            operator: In
            values:
            - us-east1-a
            - us-east1-b
  securityContext:
    runAsUser: 1000
    runAsGroup: 3000
    fsGroup: 2000
  volumes:
  - name: vol1
    persistentVolumeClaim:
      claimName: pvc-san
  containers:
  - name: sec-ctx-demo
    image: busybox
    command: [ "sh", "-c", "sleep 1h" ]
    volumeMounts:
    - name: vol1
      mountPath: /data/demo
    securityContext:
      allowPrivilegeEscalation: false
```

Esta especificación de pod indica a Kubernetes que programe el pod en los nodos que están presentes en el `us-east1` región, y elige cualquier nodo que esté presente en la `us-east1-a` o `us-east1-b` zonas.

Vea el siguiente resultado:

```
kubectl get pods -o wide
NAME        READY   STATUS    RESTARTS   AGE   IP               NODE
NOMINATED NODE   READINESS GATES
app-pod-1   1/1     Running   0          19s   192.168.25.131   node2
<none>           <none>
kubectl get pvc -o wide
NAME      STATUS   VOLUME                                     CAPACITY
ACCESS MODES   STORAGECLASS          AGE   VOLUMEMODE
pvc-san   Bound    pvc-ecb1e1a0-840c-463b-8b65-b3d033e2e62b   300Mi
RWO            netapp-san-us-east1   48s   Filesystem
```

Actualizar los backends para incluir `supportedTopologies`

Los sistemas backend preexistentes se pueden actualizar para incluir una lista de `supportedTopologies` usando `tridentctl backend update` . Esto no afectará a los volúmenes que ya se hayan aprovisionado y solo se utilizará para los PVC posteriores.

Encuentra más información

- "Gestionar recursos para contenedores"
- "selector de nodo"
- "Afinidad y antiafinidad"
- "Manchas y tolerancias"

Trabajar con instantáneas

Las instantáneas de volúmenes persistentes (PV) de Kubernetes permiten realizar copias puntuales de los volúmenes. Puede crear una instantánea de un volumen creado con Trident, importar una instantánea creada fuera de Trident, crear un nuevo volumen a partir de una instantánea existente y recuperar datos de volumen a partir de instantáneas.

Descripción general

La instantánea de volumen es compatible con `ontap-nas` , `ontap-nas-flexgroup` , `ontap-san` , `ontap-san-economy` , `solidfire-san` , `gcp-cvs` , `azure-netapp-files` , y `google-cloud-netapp-volumes` conductores.

Antes de empezar

Para trabajar con instantáneas, debe disponer de un controlador de instantáneas externo y definiciones de recursos personalizados (CRD). Esta es responsabilidad del orquestador de Kubernetes (por ejemplo: Kubeadm, GKE, OpenShift).

Si su distribución de Kubernetes no incluye el controlador de instantáneas ni los CRD, consulteImplementar un controlador de instantáneas de volumen .

No cree un controlador de instantáneas si va a crear instantáneas de volumen bajo demanda en un entorno GKE. GKE utiliza un controlador de instantáneas integrado y oculto.

Cree una instantánea de volumen

Pasos

1. Crear una `VolumeSnapshotClass` Para obtener más información, consulte"Clase de instantánea de volumen" .
   - El `driver` señala al controlador Trident CSI.
   - `deletionPolicy`puede ser `Delete o `Retain` . Cuando se configura para `Retain` La instantánea física subyacente en el clúster de almacenamiento se conserva incluso cuando `VolumeSnapshot` El objeto ha sido eliminado.

     Ejemplo

     ```
     cat snap-sc.yaml
     ```

     ```
     apiVersion: snapshot.storage.k8s.io/v1
     kind: VolumeSnapshotClass
     metadata:
       name: csi-snapclass
     driver: csi.trident.netapp.io
     deletionPolicy: Delete
     ```

2. Cree una instantánea de un PVC existente.

   Ejemplos

   - Este ejemplo crea una instantánea de un PVC existente.

     ```
     cat snap.yaml
     ```

     ```
     apiVersion: snapshot.storage.k8s.io/v1
     kind: VolumeSnapshot
     metadata:
       name: pvc1-snap
     spec:
       volumeSnapshotClassName: csi-snapclass
       source:
         persistentVolumeClaimName: pvc1
     ```

   - Este ejemplo crea un objeto de instantánea de volumen para un PVC llamado `pvc1` y el nombre de la instantánea se establece en `pvc1-snap` . Un VolumeSnapshot es análogo a un PVC y está asociado con un `VolumeSnapshotContent` objeto que representa la instantánea real.

```
kubectl create -f snap.yaml
volumesnapshot.snapshot.storage.k8s.io/pvc1-snap created

kubectl get volumesnapshots
NAME                    AGE
pvc1-snap               50s
```

- Puedes identificar el `VolumeSnapshotContent` objeto para el `pvc1-snap` VolumeSnapshot describiéndolo. El `Snapshot Content Name` identifica el objeto VolumeSnapshotContent que sirve a esta instantánea. El `Ready To Use` El parámetro indica que la instantánea se puede utilizar para crear un nuevo PVC.

```
kubectl describe volumesnapshots pvc1-snap
Name:         pvc1-snap
Namespace:    default
...
Spec:
  Snapshot Class Name:    pvc1-snap
  Snapshot Content Name:  snapcontent-e8d8a0ca-9826-11e9-9807-
525400f3f660
  Source:
    API Group:
    Kind:       PersistentVolumeClaim
    Name:       pvc1
Status:
  Creation Time:  2019-06-26T15:27:29Z
  Ready To Use:   true
  Restore Size:   3Gi
...
```

Cree un PVC a partir de una instantánea de volumen.

Puedes utilizar `dataSource` para crear un PVC utilizando un VolumeSnapshot llamado `<pvc-name>` como fuente de los datos. Una vez creado el PVC, se puede acoplar a una cápsula y utilizar como cualquier otro PVC.

> El PVC se creará en el mismo backend que el volumen fuente. Referirse a"KB: No se puede crear un PVC a partir de una instantánea de PVC de Trident en un backend alternativo." .

El siguiente ejemplo crea el PVC utilizando `pvc1-snap` como fuente de datos.

```
cat pvc-from-snap.yaml
```

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: pvc-from-snap
spec:
  accessModes:
    - ReadWriteOnce
  storageClassName: golden
  resources:
    requests:
      storage: 3Gi
  dataSource:
    name: pvc1-snap
    kind: VolumeSnapshot
    apiGroup: snapshot.storage.k8s.io
```

Importar una instantánea de volumen

Trident apoya a"Proceso de instantáneas preaprovisionadas de Kubernetes" para permitir que el administrador del clúster cree un `VolumeSnapshotContent` Objetos e instantáneas de importación creadas fuera de Trident.

Antes de empezar

Trident debe haber creado o importado el volumen principal de la instantánea.

Pasos

1. **Administrador del clúster:** Crear un `VolumeSnapshotContent` Objeto que hace referencia a la instantánea del backend. Esto inicia el flujo de trabajo de instantáneas en Trident.
   - Especifique el nombre de la instantánea del backend en `annotations` como `trident.netapp.io/internalSnapshotName: <"backend-snapshot-name">`.
   - Especificar `<name-of-parent-volume-in-trident>/<volume-snapshot-content-name>` en `snapshotHandle` Esta es la única información proporcionada a Trident por el capturador de instantáneas externo en el `ListSnapshots` llamar.

     > El `<volumeSnapshotContentName>` No siempre puede coincidir el nombre de la instantánea del backend debido a las restricciones de nomenclatura de CR.

     Ejemplo

     El siguiente ejemplo crea un `VolumeSnapshotContent` Objeto que hace referencia a la instantánea del backend `snap-01`.

```
apiVersion: snapshot.storage.k8s.io/v1
kind: VolumeSnapshotContent
metadata:
  name: import-snap-content
  annotations:
    trident.netapp.io/internalSnapshotName: "snap-01"  # This is the
name of the snapshot on the backend
spec:
  deletionPolicy: Retain
  driver: csi.trident.netapp.io
  source:
    snapshotHandle: pvc-f71223b5-23b9-4235-bbfe-e269ac7b84b0/import-
snap-content # <import PV name or source PV name>/<volume-snapshot-
content-name>
  volumeSnapshotRef:
    name: import-snap
    namespace: default
```

2. **Administrador del clúster:** Crear el `VolumeSnapshot` CR que hace referencia al `VolumeSnapshotContent` objeto. Esta solicitud permite el uso de `VolumeSnapshot` en un espacio de nombres determinado.

   Ejemplo

   El siguiente ejemplo crea un `VolumeSnapshot` CR llamado `import-snap` que hace referencia a `VolumeSnapshotContent` llamado `import-snap-content` .

   ```
   apiVersion: snapshot.storage.k8s.io/v1
   kind: VolumeSnapshot
   metadata:
     name: import-snap
   spec:
     # volumeSnapshotClassName: csi-snapclass (not required for pre-
   provisioned or imported snapshots)
     source:
       volumeSnapshotContentName: import-snap-content
   ```

3. **Procesamiento interno (no se requiere ninguna acción):** El generador de instantáneas externo reconoce la instantánea recién creada `VolumeSnapshotContent` y ejecuta el `ListSnapshots` llamar. Trident crea el `TridentSnapshot` .
   - El capturador externo establece el `VolumeSnapshotContent` a `readyToUse` y el `VolumeSnapshot` a `true` .
   - El Trident regresa `readyToUse=true` .
4. **Cualquier usuario:** Crea un `PersistentVolumeClaim` para hacer referencia al nuevo `VolumeSnapshot` , donde el `spec.dataSource` (o `spec.dataSourceRef`) el nombre es el

`VolumeSnapshot` nombre.

Ejemplo

El siguiente ejemplo crea un PVC que hace referencia al `VolumeSnapshot` llamado `import-snap` .

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: pvc-from-snap
spec:
  accessModes:
    - ReadWriteOnce
  storageClassName: simple-sc
  resources:
    requests:
      storage: 1Gi
  dataSource:
    name: import-snap
    kind: VolumeSnapshot
    apiGroup: snapshot.storage.k8s.io
```

Recuperar datos de volumen mediante instantáneas

El directorio de instantáneas está oculto de forma predeterminada para facilitar la máxima compatibilidad de los volúmenes aprovisionados mediante `ontap-nas` y `ontap-nas-economy` conductores. Habilitar el `.snapshot` directorio para recuperar datos directamente desde instantáneas.

Utilice la CLI de ONTAP para restaurar instantáneas de volumen para restaurar un volumen a un estado registrado en una instantánea anterior.

```
cluster1::*> volume snapshot restore -vserver vs0 -volume vol3 -snapshot
vol3_snap_archive
```

> Al restaurar una copia de instantánea, se sobrescribe la configuración de volumen existente. Los cambios realizados en los datos del volumen después de la creación de la copia de instantánea se pierden.

Restauración de volumen in situ a partir de una instantánea

Trident proporciona una restauración de volumen rápida e in situ a partir de una instantánea utilizando la `TridentActionSnapshotRestore` (TASR) CR. Esta solicitud de cambio (CR) funciona como una acción imperativa de Kubernetes y no persiste una vez finalizada la operación.

Trident admite la restauración de instantáneas en el `ontap-san` , `ontap-san-economy` , `ontap-nas` , `ontap-nas-flexgroup` , `azure-netapp-files` , `gcp-cvs` , `google-cloud-netapp-volumes` , y `solidfire-san` conductores.

Antes de empezar

Debe tener un PVC vinculado y una instantánea de volumen disponible.

- Verifique que el estado del PVC esté vinculado.

```
kubectl get pvc
```

- Verifique que la instantánea del volumen esté lista para usar.

```
kubectl get vs
```

Pasos

1. Cree el TASR CR. Este ejemplo crea una solicitud de cambio para PVC. `pvc1` y instantánea de volumen `pvc1-snapshot` .

   El CR TASR debe estar en un espacio de nombres donde existan el PVC y el VS.

```
cat tasr-pvc1-snapshot.yaml
```

```
apiVersion: trident.netapp.io/v1
kind: TridentActionSnapshotRestore
metadata:
  name: trident-snap
  namespace: trident
spec:
  pvcName: pvc1
  volumeSnapshotName: pvc1-snapshot
```

2. Aplique el CR para restaurar desde la instantánea. Este ejemplo restaura desde una instantánea. `pvc1` .

```
kubectl create -f tasr-pvc1-snapshot.yaml
```

```
tridentactionsnapshotrestore.trident.netapp.io/trident-snap created
```

Resultados

Trident restaura los datos a partir de la instantánea. Puedes verificar el estado de restauración de la instantánea:

```
kubectl get tasr -o yaml
```

```
apiVersion: trident.netapp.io/v1
items:
- apiVersion: trident.netapp.io/v1
  kind: TridentActionSnapshotRestore
  metadata:
    creationTimestamp: "2023-04-14T00:20:33Z"
    generation: 3
    name: trident-snap
    namespace: trident
    resourceVersion: "3453847"
    uid: <uid>
  spec:
    pvcName: pvc1
    volumeSnapshotName: pvc1-snapshot
  status:
    startTime: "2023-04-14T00:20:34Z"
    completionTime: "2023-04-14T00:20:37Z"
    state: Succeeded
kind: List
metadata:
  resourceVersion: ""
```

- En la mayoría de los casos, Trident no reintentará automáticamente la operación en caso de fallo. Deberá repetir la operación.
- Los usuarios de Kubernetes sin acceso de administrador podrían necesitar que el administrador les otorgue permiso para crear un CR TASR en el espacio de nombres de su aplicación.

Eliminar un PV con instantáneas asociadas

Al eliminar un volumen persistente con instantáneas asociadas, el volumen Trident correspondiente se actualiza a un estado de "Eliminación". Elimine las instantáneas de volumen para borrar el volumen de Trident .

Implementar un controlador de instantáneas de volumen

Si tu distribución de Kubernetes no incluye el controlador de instantáneas y los CRD, puedes implementarlos de la siguiente manera.

Pasos

1. Crear CRD de instantáneas de volumen.

```
cat snapshot-setup.sh
```

```
#!/bin/bash
# Create volume snapshot CRDs
kubectl apply -f https://raw.githubusercontent.com/kubernetes-
csi/external-snapshotter/release-
6.1/client/config/crd/snapshot.storage.k8s.io_volumesnapshotclasses.yaml
kubectl apply -f https://raw.githubusercontent.com/kubernetes-
csi/external-snapshotter/release-
6.1/client/config/crd/snapshot.storage.k8s.io_volumesnapshotcontents.yam
l
kubectl apply -f https://raw.githubusercontent.com/kubernetes-
csi/external-snapshotter/release-
6.1/client/config/crd/snapshot.storage.k8s.io_volumesnapshots.yaml
```

2. Crea el controlador de instantáneas.

```
kubectl apply -f https://raw.githubusercontent.com/kubernetes-
csi/external-snapshotter/release-6.1/deploy/kubernetes/snapshot-
controller/rbac-snapshot-controller.yaml
```

```
kubectl apply -f https://raw.githubusercontent.com/kubernetes-
csi/external-snapshotter/release-6.1/deploy/kubernetes/snapshot-
controller/setup-snapshot-controller.yaml
```

Si es necesario, abra `deploy/kubernetes/snapshot-controller/rbac-snapshot-controller.yaml` y actualizar `namespace` a tu espacio de nombres.

Enlaces relacionados

- "Instantáneas de volumen"
- "Clase de instantánea de volumen"

Trabajar con instantáneas de grupos de volúmenes

Instantáneas de grupos de volúmenes de Kubernetes de volúmenes persistentes (PV) NetApp Trident ofrece la capacidad de crear instantáneas de múltiples volúmenes (un grupo de instantáneas de volumen). Esta instantánea del grupo de volúmenes representa copias de múltiples volúmenes tomadas en el mismo momento.

VolumeGroupSnapshot es una función beta en Kubernetes con API beta. Kubernetes 1.32 es la versión mínima requerida para VolumeGroupSnapshot.

Crear instantáneas de grupos de volúmenes

La instantánea de grupo de volúmenes es compatible con `ontap-san` Controlador, solo para el protocolo iSCSI, aún no compatible con Fibre Channel (FCP) ni NVMe/TCP. Antes de comenzar

- Asegúrese de que su versión de Kubernetes sea K8s 1.32 o superior.
- Para trabajar con instantáneas, debe disponer de un controlador de instantáneas externo y definiciones de recursos personalizados (CRD). Esta es responsabilidad del orquestador de Kubernetes (por ejemplo: Kubeadm, GKE, OpenShift).

  Si su distribución de Kubernetes no incluye el controlador de instantáneas externo ni los CRD, consulteImplementar un controlador de instantáneas de volumen .

  No cree un controlador de instantáneas si va a crear instantáneas de grupos de volúmenes bajo demanda en un entorno GKE. GKE utiliza un controlador de instantáneas integrado y oculto.

- En el archivo YAML del controlador de instantáneas, configure el `CSIVolumeGroupSnapshot` Establezca el valor de la función en 'true' para garantizar que la instantánea del grupo de volúmenes esté habilitada.
- Cree las clases de instantáneas de grupo de volúmenes necesarias antes de crear una instantánea de grupo de volúmenes.
- Asegúrese de que todos los PVC/volúmenes estén en el mismo SVM para poder crear VolumeGroupSnapshot.

Pasos

- Cree una clase VolumeGroupSnapshotClass antes de crear un objeto VolumeGroupSnapshot. Para obtener más información, consulte"Clase de instantánea de grupo de volumen" .

```
apiVersion: groupsnapshot.storage.k8s.io/v1beta1
kind: VolumeGroupSnapshotClass
metadata:
  name: csi-group-snap-class
  annotations:
    kubernetes.io/description: "Trident group snapshot class"
driver: csi.trident.netapp.io
deletionPolicy: Delete
```

- Cree PVC con las etiquetas necesarias utilizando las clases de almacenamiento existentes, o agregue estas etiquetas a los PVC existentes.

  El siguiente ejemplo crea el PVC utilizando `pvc1-group-snap` como fuente de datos y etiqueta `consistentGroupSnapshot: groupA` . Define la clave y el valor de la etiqueta según tus requisitos.

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: pvc1-group-snap
  labels:
    consistentGroupSnapshot: groupA
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 100Mi
  storageClassName: sc1-1
```

- Crea un VolumeGroupSnapshot con la misma etiqueta(`consistentGroupSnapshot: groupA`) especificado en el PVC.

  Este ejemplo crea una instantánea de grupo de volúmenes:

```
apiVersion: groupsnapshot.storage.k8s.io/v1beta1
kind: VolumeGroupSnapshot
metadata:
  name: "vgs1"
  namespace: trident
spec:
  volumeGroupSnapshotClassName: csi-group-snap-class
  source:
    selector:
      matchLabels:
        consistentGroupSnapshot: groupA
```

Recuperar datos de volumen mediante una instantánea de grupo

Puede restaurar volúmenes persistentes individuales utilizando las instantáneas individuales que se han creado como parte de la instantánea del grupo de volúmenes. No se puede recuperar la instantánea del grupo de volúmenes como una unidad.

Utilice la CLI de ONTAP para restaurar instantáneas de volumen para restaurar un volumen a un estado registrado en una instantánea anterior.

```
cluster1::*> volume snapshot restore -vserver vs0 -volume vol3 -snapshot
vol3_snap_archive
```

Al restaurar una copia de instantánea, se sobrescribe la configuración de volumen existente. Los cambios realizados en los datos del volumen después de la creación de la copia de instantánea se pierden.

Restauración de volumen in situ a partir de una instantánea

Trident proporciona una restauración de volumen rápida e in situ a partir de una instantánea utilizando la `TridentActionSnapshotRestore` (TASR) CR. Esta solicitud de cambio (CR) funciona como una acción imperativa de Kubernetes y no persiste una vez finalizada la operación.

Para obtener más información, consulte "Restauración de volumen in situ a partir de una instantánea".

Eliminar un PV con instantáneas de grupo asociadas

Al eliminar una instantánea de volumen de grupo:

- Puede eliminar VolumeGroupSnapshots en su totalidad, no instantáneas individuales dentro del grupo.
- Si se eliminan volúmenes persistentes mientras existe una instantánea para ese volumen persistente, Trident moverá ese volumen a un estado de "eliminación" porque la instantánea debe eliminarse antes de que el volumen pueda eliminarse de forma segura.
- Si se ha creado un clon utilizando una instantánea agrupada y luego se va a eliminar el grupo, se iniciará una operación de división en clon y el grupo no se podrá eliminar hasta que se complete la división.

Implementar un controlador de instantáneas de volumen

Si tu distribución de Kubernetes no incluye el controlador de instantáneas y los CRD, puedes implementarlos de la siguiente manera.

Pasos

1. Crear CRD de instantáneas de volumen.

```
cat snapshot-setup.sh
```

```
#!/bin/bash
# Create volume snapshot CRDs
kubectl apply -f https://raw.githubusercontent.com/kubernetes-csi/external-snapshotter/release-8.2/client/config/crd/groupsnapshot.storage.k8s.io_volumegroupsnapshotclasses.yaml
kubectl apply -f https://raw.githubusercontent.com/kubernetes-csi/external-snapshotter/release-8.2/client/config/crd/groupsnapshot.storage.k8s.io_volumegroupsnapshotcontents.yaml
kubectl apply -f https://raw.githubusercontent.com/kubernetes-csi/external-snapshotter/release-8.2/client/config/crd/groupsnapshot.storage.k8s.io_volumegroupsnapshots.yaml
```

2. Crea el controlador de instantáneas.

```
kubectl apply -f https://raw.githubusercontent.com/kubernetes-csi/external-snapshotter/release-8.2/deploy/kubernetes/snapshot-controller/rbac-snapshot-controller.yaml
```

```
kubectl apply -f https://raw.githubusercontent.com/kubernetes-csi/external-snapshotter/release-8.2/deploy/kubernetes/snapshot-controller/setup-snapshot-controller.yaml
```

Si es necesario, abra `deploy/kubernetes/snapshot-controller/rbac-snapshot-controller.yaml` y actualizar `namespace` a tu espacio de nombres.

Enlaces relacionados

- "Clase de instantánea de grupo de volumen"
- "Instantáneas de volumen"

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>
```

Trident para Docker

Requisitos previos para la implementación

Debes instalar y configurar los requisitos previos del protocolo necesarios en tu host antes de poder implementar Trident.

Verificar los requisitos

- Verifique que su implementación cumpla con todos los requisitos."requisitos" .
- Verifique que tiene instalada una versión compatible de Docker. Si tu versión de Docker está desactualizada, "instalarlo o actualizarlo" .

```
docker --version
```

- Verifique que los requisitos previos del protocolo estén instalados y configurados en su host.

Herramientas NFS

Instala las herramientas NFS utilizando los comandos correspondientes a tu sistema operativo.

RHEL 8+

```
sudo yum install -y nfs-utils
```

Ubuntu

```
sudo apt-get install -y nfs-common
```

Reinicie sus nodos de trabajo después de instalar las herramientas NFS para evitar fallos al conectar volúmenes a los contenedores.

herramientas iSCSI

Instale las herramientas iSCSI utilizando los comandos correspondientes a su sistema operativo.

RHEL 8+

1. Instale los siguientes paquetes del sistema:

```
sudo yum install -y lsscsi iscsi-initiator-utils sg3_utils device-
mapper-multipath
```

2. Compruebe que la versión de iscsi-initiator-utils sea la 6.2.0.874-2.el7 o posterior:

```
rpm -q iscsi-initiator-utils
```

3. Configurar el escaneo en modo manual:

```
sudo sed -i 's/^\(node.session.scan\).*/\1 = manual/'
/etc/iscsi/iscsid.conf
```

4. Habilitar rutas múltiples:

```
sudo mpathconf --enable --with_multipathd y --find_multipaths n
```

Asegurar `etc/multipath.conf` contiene `find_multipaths no` bajo `defaults` .

5. Asegúrese de que `iscsid` y `multipathd` están en funcionamiento:

```
sudo systemctl enable --now iscsid multipathd
```

6. Habilitar e iniciar `iscsi` :

```
sudo systemctl enable --now iscsi
```

Ubuntu

1. Instale los siguientes paquetes del sistema:

```
sudo apt-get install -y open-iscsi lsscsi sg3-utils multipath-tools
scsitools
```

2. Compruebe que la versión de open-iscsi sea 2.0.874-5ubuntu2.10 o posterior (para bionic) o 2.0.874-7.1ubuntu6.1 o posterior (para focal):

```
dpkg -l open-iscsi
```

3. Configurar el escaneo en modo manual:

```
sudo sed -i 's/^\(node.session.scan\).*/\1 = manual/'
/etc/iscsi/iscsid.conf
```

4. Habilitar rutas múltiples:

```
sudo tee /etc/multipath.conf <<-EOF
defaults {
    user_friendly_names yes
    find_multipaths no
}
EOF
sudo systemctl enable --now multipath-tools.service
sudo service multipath-tools restart
```

> Asegurar `etc/multipath.conf` contiene `find_multipaths no` bajo `defaults` .

5. Asegúrese de que `open-iscsi` y `multipath-tools` están habilitados y en funcionamiento:

```
sudo systemctl status multipath-tools
sudo systemctl enable --now open-iscsi.service
sudo systemctl status open-iscsi
```

Herramientas NVMe

Instala las herramientas NVMe utilizando los comandos correspondientes a tu sistema operativo.

- NVMe requiere RHEL 9 o posterior.
- Si la versión del kernel de su nodo de Kubernetes es demasiado antigua o si el paquete NVMe no está disponible para su versión del kernel, es posible que deba actualizar la versión del kernel de su nodo a una que incluya el paquete NVMe.

RHEL 9

```
sudo yum install nvme-cli
sudo yum install linux-modules-extra-$(uname -r)
sudo modprobe nvme-tcp
```

Ubuntu

```
sudo apt install nvme-cli
sudo apt -y install linux-modules-extra-$(uname -r)
sudo modprobe nvme-tcp
```

Herramientas FC

Instala las herramientas FC utilizando los comandos correspondientes a tu sistema operativo.

- Cuando utilice nodos de trabajo que ejecuten RHEL/Red Hat Enterprise Linux CoreOS (RHCOS) con PV FC, especifique el `discard` mountOption en la StorageClass para realizar la recuperación de espacio en línea. Referirse a "Documentación de Red Hat" .

RHEL 8+

1. Instale los siguientes paquetes del sistema:

```
sudo yum install -y lsscsi device-mapper-multipath
```

2. Habilitar rutas múltiples:

```
sudo mpathconf --enable --with_multipathd y --find_multipaths n
```

Asegurar `etc/multipath.conf` contiene `find_multipaths no` bajo `defaults` .

3. Asegúrese de que `multipathd` está en funcionamiento:

```
sudo systemctl enable --now multipathd
```

Ubuntu

1. Instale los siguientes paquetes del sistema:

```
sudo apt-get install -y lsscsi sg3-utils multipath-tools scsitools
```

2. Habilitar rutas múltiples:

```
sudo tee /etc/multipath.conf <<-EOF
defaults {
    user_friendly_names yes
    find_multipaths no
}
EOF
sudo systemctl enable --now multipath-tools.service
sudo service multipath-tools restart
```

Asegurar `etc/multipath.conf` contiene `find_multipaths no` bajo `defaults` .

3. Asegúrese de que `multipath-tools` está habilitado y en funcionamiento:

```
sudo systemctl status multipath-tools
```

Despliegue Trident

Trident para Docker proporciona integración directa con el ecosistema Docker para las plataformas de almacenamiento NetApp . Admite el aprovisionamiento y la gestión de recursos de almacenamiento desde la plataforma de almacenamiento hasta los hosts de Docker, con un marco para agregar plataformas adicionales en el futuro.

Se pueden ejecutar varias instancias de Trident simultáneamente en el mismo host. Esto permite conexiones simultáneas a múltiples sistemas y tipos de almacenamiento, con la posibilidad de personalizar el almacenamiento utilizado para los volúmenes de Docker.

Lo que necesitarás

Ver el"Requisitos previos para la implementación" . Una vez que te asegures de que se cumplen los requisitos previos, estarás listo para implementar Trident.

Método de complemento gestionado por Docker (versión 1.13/17.03 y posteriores)

Antes de empezar

Si ha utilizado Trident anterior a Docker 1.13/17.03 en el método daemon tradicional, asegúrese de detener el proceso Trident y reiniciar su daemon Docker antes de utilizar el método de complemento administrado.

1. Detener todas las instancias en ejecución:

```
pkill /usr/local/bin/netappdvp
pkill /usr/local/bin/trident
```

2. Reinicia Docker.

```
systemctl restart docker
```

3. Asegúrese de tener instalado Docker Engine 17.03 (nuevo 1.13) o posterior.

```
docker --version
```

Si su versión está desactualizada, "instalar o actualizar su instalación" .

Pasos

1. Cree un archivo de configuración y especifique las opciones de la siguiente manera:
   - `config`El nombre de archivo predeterminado es `config.json` Sin embargo, puedes usar el nombre que elijas especificándolo. `config` opción con el nombre del archivo. El archivo de configuración debe estar ubicado en el `/etc/netappdvp` directorio en el sistema anfitrión.
   - `log-level`Especifique el nivel de registro(`debug`, `info`, `warn`, `error`, `fatal`). El valor predeterminado es `info` .

- `debug`Especifique si el registro de depuración está habilitado. El valor predeterminado es falso. Anula el nivel de registro si es verdadero.

  i. Cree una ubicación para el archivo de configuración:

  ```
  sudo mkdir -p /etc/netappdvp
  ```

  ii. Crear el archivo de configuración:

  ```
  cat << EOF > /etc/netappdvp/config.json
  ```

  ```
  {
    "version": 1,
    "storageDriverName": "ontap-nas",
    "managementLIF": "10.0.0.1",
    "dataLIF": "10.0.0.2",
    "svm": "svm_nfs",
    "username": "vsadmin",
    "password": "password",
    "aggregate": "aggr1"
  }
  EOF
  ```

2. Inicie Trident utilizando el sistema de plugins gestionados. Reemplazar `<version>` con la versión del plugin (xxx.xx.x) que está utilizando.

   ```
   docker plugin install --grant-all-permissions --alias netapp
   netapp/trident-plugin:<version> config=myConfigFile.json
   ```

3. Comience a utilizar Trident para consumir almacenamiento del sistema configurado.

   a. Crea un volumen llamado "firstVolume":

   ```
   docker volume create -d netapp --name firstVolume
   ```

   b. Crea un volumen predeterminado cuando se inicie el contenedor:

   ```
   docker run --rm -it --volume-driver netapp --volume
   secondVolume:/my_vol alpine ash
   ```

   c. Eliminar el volumen "firstVolume":

```
docker volume rm firstVolume
```

Método tradicional (versión 1.12 o anterior)

Antes de empezar

1. Asegúrese de tener Docker versión 1.10 o posterior.

   ```
   docker --version
   ```

   Si su versión está desactualizada, actualice su instalación.

   ```
   curl -fsSL https://get.docker.com/ | sh
   ```

   O, "Siga las instrucciones para su distribución." .

2. Asegúrese de que NFS y/o iSCSI estén configurados para su sistema.

Pasos

1. Instalar y configurar el complemento NetApp Docker Volume:
   a. Descarga y descomprime la aplicación:

   ```
   wget
   https://github.com/NetApp/trident/releases/download/v25.06.0/trident-
   installer-25.06.0.tar.gz
   tar zxf trident-installer-25.06.0.tar.gz
   ```

   b. Muévete a una ubicación en la ruta del contenedor:

   ```
   sudo mv trident-installer/extras/bin/trident /usr/local/bin/
   sudo chown root:root /usr/local/bin/trident
   sudo chmod 755 /usr/local/bin/trident
   ```

   c. Cree una ubicación para el archivo de configuración:

   ```
   sudo mkdir -p /etc/netappdvp
   ```

   d. Crear el archivo de configuración:

   ```
   cat << EOF > /etc/netappdvp/ontap-nas.json
   ```

```
{
  "version": 1,
  "storageDriverName": "ontap-nas",
  "managementLIF": "10.0.0.1",
  "dataLIF": "10.0.0.2",
  "svm": "svm_nfs",
  "username": "vsadmin",
  "password": "password",
  "aggregate": "aggr1"
}
EOF
```

2. Después de colocar el binario y crear el archivo de configuración, inicie el demonio Trident utilizando el archivo de configuración deseado.

```
sudo trident --config=/etc/netappdvp/ontap-nas.json
```

> A menos que se especifique lo contrario, el nombre predeterminado para el controlador de volumen es "netapp".

Una vez iniciado el demonio, puede crear y administrar volúmenes utilizando la interfaz de línea de comandos de Docker.

3. Crear un volumen:

```
docker volume create -d netapp --name trident_1
```

4. Aprovisionar un volumen de Docker al iniciar un contenedor:

```
docker run --rm -it --volume-driver netapp --volume trident_2:/my_vol
alpine ash
```

5. Eliminar un volumen de Docker:

```
docker volume rm trident_1
```

```
docker volume rm trident_2
```

Inicie Trident al arrancar el sistema.

Un archivo de unidad de ejemplo para sistemas basados en systemd se puede encontrar en `contrib/trident.service.example` en el repositorio Git. Para usar el archivo con RHEL, haga lo siguiente:

1. Copie el archivo a la ubicación correcta.

   Debe utilizar nombres únicos para los archivos de unidad si tiene más de una instancia en ejecución.

   ```
   cp contrib/trident.service.example
   /usr/lib/systemd/system/trident.service
   ```

2. Edite el archivo, cambie la descripción (línea 2) para que coincida con el nombre del controlador y la ruta del archivo de configuración (línea 9) para que refleje su entorno.
3. Recarga systemd para que aplique los cambios:

   ```
   systemctl daemon-reload
   ```

4. Habilite el servicio.

   Este nombre varía dependiendo del nombre que le hayas dado al archivo en el `/usr/lib/systemd/system` directorio.

   ```
   systemctl enable trident
   ```

5. Iniciar el servicio.

   ```
   systemctl start trident
   ```

6. Consulta el estado.

   ```
   systemctl status trident
   ```

Cada vez que modifique el archivo de la unidad, ejecute el `systemctl daemon-reload` comando para que esté al tanto de los cambios.

Actualizar o desinstalar Trident

Puede actualizar Trident para Docker de forma segura sin que ello afecte a los volúmenes que estén en uso. Durante el proceso de actualización habrá un breve período en el que `docker volume` Los comandos dirigidos al complemento no tendrán

éxito y las aplicaciones no podrán montar volúmenes hasta que el complemento vuelva a estar en funcionamiento. En la mayoría de los casos, esto es cuestión de segundos.

Mejora

Siga los pasos a continuación para actualizar Trident para Docker.

Pasos

1. Enumere los volúmenes existentes:

```
docker volume ls
DRIVER              VOLUME NAME
netapp:latest       my_volume
```

2. Desactivar el plugin:

```
docker plugin disable -f netapp:latest
docker plugin ls
ID                  NAME                DESCRIPTION
ENABLED
7067f39a5df5        netapp:latest       nDVP - NetApp Docker Volume
Plugin   false
```

3. Actualizar el plugin:

```
docker plugin upgrade --skip-remote-check --grant-all-permissions
netapp:latest netapp/trident-plugin:21.07
```

> La versión 18.01 de Trident reemplaza a nDVP. Deberías actualizar directamente desde el `netapp/ndvp-plugin` imagen a la `netapp/trident-plugin` imagen.

4. Habilitar el plugin:

```
docker plugin enable netapp:latest
```

5. Verifique que el plugin esté habilitado:

```
docker plugin ls
ID                  NAME                DESCRIPTION
ENABLED
7067f39a5df5        netapp:latest       Trident - NetApp Docker Volume
Plugin   true
```

6. Verifique que los volúmenes sean visibles:

```
docker volume ls
DRIVER              VOLUME NAME
netapp:latest       my_volume
```

> Si está actualizando desde una versión antigua de Trident (anterior a la 20.10) a Trident 20.10 o posterior, podría encontrarse con un error. Para obtener más información, consulte"Problemas conocidos" . Si se produce este error, primero debe desactivar el plugin, luego eliminarlo y, a continuación, instalar la versión de Trident necesaria pasando un parámetro de configuración adicional: `docker plugin install netapp/trident-plugin:20.10 --alias netapp --grant-all-permissions config=config.json`

Desinstalar

Siga los pasos que se indican a continuación para desinstalar Trident para Docker.

Pasos

1. Elimine cualquier volumen que haya creado el plugin.
2. Desactivar el plugin:

```
docker plugin disable netapp:latest
docker plugin ls
ID                  NAME                DESCRIPTION
ENABLED
7067f39a5df5        netapp:latest       nDVP - NetApp Docker Volume
Plugin   false
```

3. Eliminar el plugin:

```
docker plugin rm netapp:latest
```

Trabajar con volúmenes

Puede crear, clonar y eliminar volúmenes fácilmente utilizando el estándar. `docker volume` comandos con el nombre del controlador Trident especificado cuando sea necesario.

Crear un volumen

- Crea un volumen con una unidad utilizando el nombre predeterminado:

```
docker volume create -d netapp --name firstVolume
```

- Crear un volumen con una instancia específica de Trident :

```
docker volume create -d ntap_bronze --name bronzeVolume
```

Si no especifica ninguno"opciones" Se utilizan los valores predeterminados del controlador.

- Anule el tamaño de volumen predeterminado. Consulte el siguiente ejemplo para crear un volumen de 20 GiB con un controlador:

```
docker volume create -d netapp --name my_vol --opt size=20G
```

Los tamaños de volumen se expresan como cadenas que contienen un valor entero con unidades opcionales (ejemplo: 10G, 20GB, 3TiB). Si no se especifican unidades, el valor predeterminado es G. Las unidades de tamaño se pueden expresar como potencias de 2 (B, KiB, MiB, GiB, TiB) o potencias de 10 (B, KB, MB, GB, TB). Las unidades abreviadas utilizan potencias de 2 (G = GiB, T = TiB, …).

Eliminar un volumen

- Elimine el volumen como cualquier otro volumen de Docker:

```
docker volume rm firstVolume
```

Al usar el `solidfire-san` El controlador, en el ejemplo anterior, elimina y purga el volumen.

Siga los pasos a continuación para actualizar Trident para Docker.

Clonar un volumen

Al usar el `ontap-nas` , `ontap-san` , `solidfire-san` , y `gcp-cvs storage drivers` Trident puede clonar volúmenes. Al usar el `ontap-nas-flexgroup` o `ontap-nas-economy` controladores, la clonación no es compatible. La creación de un nuevo volumen a partir de un volumen existente dará como resultado la creación de una nueva instantánea.

- Inspeccione el volumen para enumerar las instantáneas:

```
docker volume inspect <volume_name>
```

- Crear un nuevo volumen a partir de un volumen existente. Esto dará como resultado la creación de una

nueva instantánea:

```
docker volume create -d <driver_name> --name <new_name> -o from
=<source_docker_volume>
```

- Crea un nuevo volumen a partir de una instantánea existente en un volumen. Esto no creará una nueva instantánea:

```
docker volume create -d <driver_name> --name <new_name> -o from
=<source_docker_volume> -o fromSnapshot=<source_snap_name>
```

Ejemplo

```
docker volume inspect firstVolume

[
  {
    "Driver": "ontap-nas",
    "Labels": null,
    "Mountpoint": "/var/lib/docker-volumes/ontap-
nas/netappdvp_firstVolume",
    "Name": "firstVolume",
    "Options": {},
    "Scope": "global",
    "Status": {
      "Snapshots": [
        {
          "Created": "2017-02-10T19:05:00Z",
          "Name": "hourly.2017-02-10_1505"
        }
      ]
    }
  }
]

docker volume create -d ontap-nas --name clonedVolume -o from=firstVolume
clonedVolume

docker volume rm clonedVolume
docker volume create -d ontap-nas --name volFromSnap -o from=firstVolume
-o fromSnapshot=hourly.2017-02-10_1505
volFromSnap

docker volume rm volFromSnap
```

Acceso a volúmenes creados externamente

Los contenedores que utilizan Trident solo pueden acceder a dispositivos de bloque creados externamente (o sus clones) si no tienen particiones y si su sistema de archivos es compatible con Trident (por ejemplo: un `ext4` -formateado `/dev/sdc1` no será accesible a través de Trident).

Opciones de volumen específicas del conductor

Cada controlador de almacenamiento tiene un conjunto diferente de opciones, que puede especificar al momento de crear el volumen para personalizar el resultado. Consulte a continuación las opciones que se aplican a su sistema de almacenamiento configurado.

Utilizar estas opciones durante la operación de creación de volumen es sencillo. Proporcione la opción y el

valor utilizando el `-o` operador durante la operación CLI. Estos valores anulan cualquier valor equivalente del archivo de configuración JSON.

Opciones de volumen de ONTAP

Las opciones de creación de volúmenes para NFS, iSCSI y FC incluyen lo siguiente:

Opción	Descripción
`size`	El tamaño del volumen, por defecto, es de 1 GiB.
`spaceReserve`	El volumen se define como fino o grueso; por defecto, se utiliza fino. Los valores válidos son `none` (con provisiones escasas) y `volume` (provisionado abundantemente).
`snapshotPolicy`	Esto configurará la política de instantáneas al valor deseado. El valor predeterminado es `none` , lo que significa que no se crearán instantáneas automáticamente para el volumen. A menos que su administrador de almacenamiento la modifique, en todos los sistemas ONTAP existe una política denominada "predeterminada" que crea y conserva seis instantáneas horarias, dos diarias y dos semanales. Los datos conservados en una instantánea se pueden recuperar navegando hasta el archivo `.snapshot` directorio en cualquier directorio del volumen.
`snapshotReserve`	Esto establecerá la reserva de instantáneas en el porcentaje deseado. El valor predeterminado es ningún valor, lo que significa que ONTAP seleccionará snapshotReserve (normalmente 5%) si ha seleccionado snapshotPolicy, o 0% si snapshotPolicy es ninguno. Puede establecer el valor predeterminado de snapshotReserve en el archivo de configuración para todos los backends de ONTAP , y puede usarlo como una opción de creación de volúmenes para todos los backends de ONTAP excepto ontap-nas-economy.
`splitOnClone`	Al clonar un volumen, esto hará que ONTAP separe inmediatamente el clon de su volumen padre. El valor predeterminado es `false` . En algunos casos, la clonación de volúmenes se resuelve mejor separando el clon de su padre inmediatamente después de su creación, ya que es poco probable que exista alguna oportunidad para lograr eficiencias de almacenamiento. Por ejemplo, clonar una base de datos vacía puede ofrecer un gran ahorro de tiempo pero poco ahorro de almacenamiento, por lo que es mejor dividir el clon inmediatamente.

Opción	Descripción
`encryption`	Habilite el cifrado de volumen de NetApp (NVE) en el nuevo volumen; el valor predeterminado es `false` . Para utilizar esta opción, NVE debe estar licenciado y habilitado en el clúster. Si NAE está habilitado en el backend, cualquier volumen aprovisionado en Trident tendrá NAE habilitado. Para obtener más información, consulte:"Cómo funciona Trident con NVE y NAE" .
`tieringPolicy`	Establece la política de niveles que se utilizará para el volumen. Esto decide si los datos se transfieren a la capa de nube cuando se vuelven inactivos (fríos).

Las siguientes opciones adicionales son **únicamente** para NFS:

Opción	Descripción
`unixPermissions`	Esto controla el conjunto de permisos para el propio volumen. Por defecto, los permisos se establecerán en `---rwxr-xr-x , o en notación numérica 0755, y `root` será el propietario. Funcionará tanto el formato de texto como el numérico.
`snapshotDir`	Configurando esto a `true` hará el `.snapshot` directorio visible para los clientes que acceden al volumen. El valor predeterminado es `false` , lo que significa que la visibilidad de `.snapshot` El directorio está deshabilitado por defecto. Algunas imágenes, por ejemplo la imagen oficial de MySQL, no funcionan como se espera cuando `.snapshot` El directorio es visible.
`exportPolicy`	Establece la política de exportación que se utilizará para el volumen. El valor predeterminado es `default` .
`securityStyle`	Establece el estilo de seguridad que se utilizará para acceder al volumen. El valor predeterminado es `unix` . Los valores válidos son `unix` y `mixed` .

Las siguientes opciones adicionales son **únicamente** para iSCSI:

Opción	Descripción
`fileSystemType`	Establece el sistema de archivos utilizado para formatear volúmenes iSCSI. El valor predeterminado es `ext4` . Los valores válidos son `ext3` , `ext4` , y `xfs` .
`spaceAllocation`	Configurando esto a `false` Desactivará la función de asignación de espacio del LUN. El valor predeterminado es `true` , lo que significa que ONTAP notifica al host cuando el volumen se ha quedado sin espacio y la LUN del volumen no puede aceptar escrituras. Esta opción también permite a ONTAP recuperar espacio automáticamente cuando el host elimina datos.

Ejemplos

Vea los ejemplos a continuación:

- Cree un volumen de 10 GiB:

```
docker volume create -d netapp --name demo -o size=10G -o
encryption=true
```

- Cree un volumen de 100 GiB con instantáneas:

```
docker volume create -d netapp --name demo -o size=100G -o
snapshotPolicy=default -o snapshotReserve=10
```

- Cree un volumen que tenga habilitado el bit setUID:

```
docker volume create -d netapp --name demo -o unixPermissions=4755
```

El tamaño mínimo del volumen es 20 MiB.

Si no se especifica la reserva de instantáneas y la política de instantáneas es `none` Trident utiliza una reserva de instantáneas del 0%.

- Cree un volumen sin política de instantáneas ni reserva de instantáneas:

```
docker volume create -d netapp --name my_vol --opt snapshotPolicy=none
```

- Cree un volumen sin política de instantáneas y con una reserva de instantáneas personalizada del 10%:

```
docker volume create -d netapp --name my_vol --opt snapshotPolicy=none
--opt snapshotReserve=10
```

- Cree un volumen con una política de instantáneas y una reserva de instantáneas personalizada del 10%:

```
docker volume create -d netapp --name my_vol --opt
snapshotPolicy=myPolicy --opt snapshotReserve=10
```

- Cree un volumen con una política de instantáneas y acepte la reserva de instantáneas predeterminada de ONTAP (normalmente el 5 %):

```
docker volume create -d netapp --name my_vol --opt
snapshotPolicy=myPolicy
```

Opciones de volumen del software Element

Las opciones del software Element exponen el tamaño y las políticas de calidad de servicio (QoS) asociadas con el volumen. Cuando se crea el volumen, la política de QoS asociada a él se especifica mediante `-o type=service_level` nomenclatura.

El primer paso para definir un nivel de servicio QoS con el controlador Element es crear al menos un tipo y especificar las IOPS mínimas, máximas y de ráfaga asociadas a un nombre en el archivo de configuración.

Otras opciones de creación de volúmenes del software Element incluyen las siguientes:

Opción	Descripción
`size`	El tamaño del volumen, por defecto es 1 GiB o entrada de configuración… "defaults": {"size": "5G"}.
`blocksize`	Utilice 512 o 4096, el valor predeterminado es 512 o la entrada de configuración DefaultBlockSize.

Ejemplo

Consulte el siguiente archivo de configuración de ejemplo con definiciones de QoS:

```
{
  "Types": [
    {
      "Type": "Bronze",
      "Qos": {
        "minIOPS": 1000,
        "maxIOPS": 2000,
        "burstIOPS": 4000
      }
    },
    {
      "Type": "Silver",
      "Qos": {
        "minIOPS": 4000,
        "maxIOPS": 6000,
        "burstIOPS": 8000
      }
    },
    {
      "Type": "Gold",
      "Qos": {
        "minIOPS": 6000,
        "maxIOPS": 8000,
        "burstIOPS": 10000
      }
    }
  ]
}
```

En la configuración anterior, tenemos tres definiciones de políticas: Bronce, Plata y Oro. Estos nombres son arbitrarios.

- Cree un volumen Gold de 10 GiB:

```
docker volume create -d solidfire --name sfGold -o type=Gold -o size=10G
```

- Cree un volumen Bronze de 100 GiB:

```
docker volume create -d solidfire --name sfBronze -o type=Bronze -o
size=100G
```

Recopilar registros

Puedes recopilar registros para obtener ayuda en la resolución de problemas. El método que utilice para recopilar los registros varía según cómo esté ejecutando el complemento de Docker.

Recopilar registros para la resolución de problemas

Pasos

1. Si está ejecutando Trident utilizando el método de plugin gestionado recomendado (es decir, utilizando `docker plugin` comandos), consúltelos de la siguiente manera:

```
docker plugin ls
```

```
ID                  NAME                DESCRIPTION
ENABLED
4fb97d2b956b        netapp:latest       nDVP - NetApp Docker Volume
Plugin   false
journalctl -u docker | grep 4fb97d2b956b
```

   El nivel de registro estándar debería permitirle diagnosticar la mayoría de los problemas. Si considera que eso no es suficiente, puede habilitar el registro de depuración.

2. Para habilitar el registro de depuración, instale el complemento con el registro de depuración habilitado:

```
docker plugin install netapp/trident-plugin:<version> --alias <alias>
debug=true
```

   O bien, habilite el registro de depuración cuando el complemento ya esté instalado:

```
docker plugin disable <plugin>
```

```
docker plugin set <plugin> debug=true
```

```
docker plugin enable <plugin>
```

3. Si ejecutas el propio binario en el host, los registros estarán disponibles en el archivo de registro del host. `/var/log/netappdvp` directorio. Para habilitar el registro de depuración, especifique `-debug` cuando ejecutas el plugin.

Consejos generales para la solución de problemas

- El problema más común con el que se encuentran los nuevos usuarios es una configuración incorrecta que impide que el plugin se inicialice. Cuando esto ocurra, probablemente verá un mensaje como este al intentar instalar o habilitar el complemento:

  ```
  Error response from daemon: dial unix /run/docker/plugins/<id>/netapp.sock:
  connect: no such file or directory
  ```

  Esto significa que el plugin no se pudo iniciar. Por suerte, el plugin se ha desarrollado con una completa capacidad de registro que debería ayudarle a diagnosticar la mayoría de los problemas con los que pueda encontrarse.

- Si hay problemas al montar un panel fotovoltaico en un contenedor, asegúrese de que `rpcbind` Está instalado y en funcionamiento. Utilice el gestor de paquetes requerido para el sistema operativo anfitrión y compruebe si `rpcbind` está en funcionamiento. Puedes comprobar el estado del servicio rpcbind ejecutando un `systemctl status rpcbind` o su equivalente.

Gestionar múltiples instancias de Trident

Se necesitan varias instancias de Trident cuando se desea tener varias configuraciones de almacenamiento disponibles simultáneamente. La clave para múltiples instancias es darles nombres diferentes usando el `--alias` opción con el plugin en contenedor, o `--volume-driver` opción al instanciar Trident en el host.

Pasos para el plugin gestionado por Docker (versión 1.13/17.03 o posterior)

1. Inicie la primera instancia especificando un alias y un archivo de configuración.

   ```
   docker plugin install --grant-all-permissions --alias silver
   netapp/trident-plugin:21.07 config=silver.json
   ```

2. Inicie la segunda instancia, especificando un alias y un archivo de configuración diferentes.

   ```
   docker plugin install --grant-all-permissions --alias gold
   netapp/trident-plugin:21.07 config=gold.json
   ```

3. Cree volúmenes especificando el alias como nombre del controlador.

   Por ejemplo, en el caso del volumen de oro:

   ```
   docker volume create -d gold --name ntapGold
   ```

   Por ejemplo, para el volumen de plata:

```
docker volume create -d silver --name ntapSilver
```

Pasos para la versión tradicional (versión 1.12 o anterior)

1. Inicie el complemento con una configuración NFS utilizando un ID de controlador personalizado:

```
sudo trident --volume-driver=netapp-nas --config=/path/to/config
-nfs.json
```

2. Inicie el complemento con una configuración iSCSI utilizando un ID de controlador personalizado:

```
sudo trident --volume-driver=netapp-san --config=/path/to/config
-iscsi.json
```

3. Aprovisionar volúmenes de Docker para cada instancia del controlador:

   Por ejemplo, para NFS:

```
docker volume create -d netapp-nas --name my_nfs_vol
```

   Por ejemplo, para iSCSI:

```
docker volume create -d netapp-san --name my_iscsi_vol
```

Opciones de configuración de almacenamiento

Consulte las opciones de configuración disponibles para sus configuraciones de Trident .

Opciones de configuración global

Estas opciones de configuración se aplican a todas las configuraciones de Trident , independientemente de la plataforma de almacenamiento que se utilice.

Opción	Descripción	Ejemplo
`version`	Número de versión del archivo de configuración	`1`

Opción	Descripción	Ejemplo
`storageDriverName`	Nombre del controlador de almacenamiento	`ontap-nas`, `ontap-san` , `ontap-nas-economy` , `ontap-nas-flexgroup` , `solidfire-san`
`storagePrefix`	Prefijo opcional para los nombres de los volúmenes. Por defecto: `netappdvp_` .	`staging_`
`limitVolumeSize`	Restricción opcional en los tamaños de volumen. Valor predeterminado: "" (no se aplica)	`10g`

No utilizar `storagePrefix` (incluido el valor predeterminado) para los backends de Element. De forma predeterminada, el `solidfire-san` El controlador ignorará esta configuración y no utilizará un prefijo. NetApp recomienda usar un tenantID específico para la asignación de volúmenes de Docker o usar los datos de atributos que se completan con la versión de Docker, la información del controlador y el nombre sin procesar de Docker en los casos en que se haya utilizado alguna manipulación de nombres.

Se ofrecen opciones predeterminadas para evitar tener que especificarlas en cada volumen que cree. El `size` Esta opción está disponible para todos los tipos de controladores. Consulte la sección de configuración de ONTAP para ver un ejemplo de cómo configurar el tamaño de volumen predeterminado.

Opción	Descripción	Ejemplo
`size`	Tamaño predeterminado opcional para nuevos volúmenes. Por defecto: `1G`	`10G`

Configuración de ONTAP

Además de los valores de configuración global anteriores, al usar ONTAP, están disponibles las siguientes opciones de nivel superior.

Opción	Descripción	Ejemplo
`managementLIF`	Dirección IP de la LIF de gestión de ONTAP . Puede especificar un nombre de dominio completo (FQDN).	`10.0.0.1`

Opción	Descripción	Ejemplo
`dataLIF`	Dirección IP del protocolo LIF. • Controladores NAS ONTAP *: NetApp recomienda especificar `dataLIF` . Si no se proporcionan, Trident obtiene los dataLIF del SVM. Puede especificar un nombre de dominio completo (FQDN) para usarlo en las operaciones de montaje NFS, lo que le permite crear un DNS round-robin para equilibrar la carga entre varios dataLIF. • Controladores SAN ONTAP *: No especifique para iSCSI o FC. Trident utiliza"Mapa selectivo de LUN de ONTAP" para descubrir los LIF iSCSI o FC necesarios para establecer una sesión de múltiples rutas. Se genera una advertencia si `dataLIF` está definido explícitamente.	`10.0.0.2`
`svm`	Máquina virtual de almacenamiento a utilizar (obligatoria si la LIF de gestión es una LIF de clúster)	`svm_nfs`
`username`	Nombre de usuario para conectarse al dispositivo de almacenamiento	`vsadmin`
`password`	Contraseña para conectarse al dispositivo de almacenamiento	`secret`
`aggregate`	Agregado para aprovisionamiento (opcional; si se establece, debe asignarse a la SVM). Para el `ontap-nas-flexgroup` conductor, esta opción se ignora. Todos los agregados asignados al SVM se utilizan para aprovisionar un volumen FlexGroup .	`aggr1`
`limitAggregateUsage`	Opcional: el aprovisionamiento puede fallar si el uso supera este porcentaje.	`75%`

Opción	Descripción	Ejemplo
`nfsMountOptions`	Control preciso de las opciones de montaje NFS; el valor predeterminado es "-o nfsvers=3". **Disponible solo para `ontap-nas` y `ontap-nas-economy` conductores**. "Consulte aquí la información de configuración del host NFS." .	`-o nfsvers=4`
`igroupName`	Trident crea y gestiona por nodo `igroups` como `netappdvp` . Este valor no se puede cambiar ni omitir. **Disponible solo para `ontap-san` conductor**.	`netappdvp`
`limitVolumeSize`	Tamaño máximo de volumen solicitable.	`300g`
`qtreesPerFlexvol`	Número máximo de qtrees por FlexVol, debe estar en el rango [50, 300], el valor predeterminado es 200. **Para el `ontap-nas-economy` controlador, esta opción permite personalizar el número máximo de qtrees por FlexVol**.	`300`
`sanType`	**Compatible con `ontap-san` Solo el conductor.** Utilice para seleccionar `iscsi` para iSCSI, `nvme` para NVMe/TCP o `fcp` para SCSI sobre Fibre Channel (FC).	`iscsi`si está en blanco
`limitVolumePoolSize`	**Compatible con `ontap-san-economy` y `ontap-san-economy` Solo conductores.** Limita los tamaños de FlexVol en los controladores ONTAP ontap-nas-economy y ontap-SAN-economy.	`300g`

Se ofrecen opciones predeterminadas para evitar tener que especificarlas en cada volumen que cree:

Opción	Descripción	Ejemplo
`spaceReserve`	Modo de reserva de espacio; `none` (con provisiones escasas) o `volume` (grueso)	`none`

Opción	Descripción	Ejemplo
`snapshotPolicy`	Política de instantáneas a utilizar; la predeterminada es `none`	`none`
`snapshotReserve`	Porcentaje de reserva de instantánea, el valor predeterminado es "" para aceptar el valor predeterminado de ONTAP	`10`
`splitOnClone`	Al crear un clon, se separa de su padre; por defecto, se utiliza el método `divide`. `false`	`false`
`encryption`	Habilita el cifrado de volumen de NetApp (NVE) en el nuevo volumen; el valor predeterminado es `false` . Para utilizar esta opción, NVE debe estar licenciado y habilitado en el clúster. Si NAE está habilitado en el backend, cualquier volumen aprovisionado en Trident tendrá NAE habilitado. Para obtener más información, consulte:"Cómo funciona Trident con NVE y NAE" .	verdadero
`unixPermissions`	Opción NAS para volúmenes NFS aprovisionados, valor predeterminado `777`	`777`
`snapshotDir`	Opción NAS para acceder a `.snapshot` directorio.	"verdadero" para NFSv4, "falso" para NFSv3
`exportPolicy`	Opción NAS para la política de exportación NFS a utilizar, por defecto a `default`	`default`
`securityStyle`	Opción NAS para acceder al volumen NFS aprovisionado. NFS admite `mixed` y `unix` Estilos de seguridad. El valor predeterminado es `unix` .	`unix`
`fileSystemType`	Opción SAN para seleccionar el tipo de sistema de archivos, por defecto es `ext4`	`xfs`
`tieringPolicy`	La política de niveles a utilizar, por defecto es `none` .	`none`

Opciones de escala

El `ontap-nas` y `ontap-san` Los controladores crean un ONTAP FlexVol para cada volumen de Docker. ONTAP admite hasta 1000 FlexVols por nodo de clúster, con un máximo de 12 000 volúmenes FlexVol por clúster. Si sus requisitos de volumen de Docker se ajustan a esa limitación, `ontap-nas` El controlador es la

solución NAS preferida debido a las características adicionales que ofrece FlexVols, como instantáneas granulares de volumen Docker y clonación.

Si necesita más volúmenes de Docker de los que pueden admitir los límites de FlexVol , elija la opción `ontap-nas-economy` o el `ontap-san-economy` conductor.

El `ontap-nas-economy` El controlador crea volúmenes Docker como Qtrees de ONTAP dentro de un grupo de volúmenes FlexVol gestionados automáticamente. Los Qtrees ofrecen una escalabilidad mucho mayor, hasta 100.000 por nodo de clúster y 2.400.000 por clúster, a costa de algunas características. El `ontap-nas-economy` El controlador no admite instantáneas granulares de volumen de Docker ni clonación.

> El `ontap-nas-economy` Actualmente, el controlador no es compatible con Docker Swarm, ya que Docker Swarm no orquesta la creación de volúmenes en múltiples nodos.

El `ontap-san-economy` El controlador crea volúmenes Docker como LUN ONTAP dentro de un grupo compartido de volúmenes FlexVol gestionados automáticamente. De esta forma, cada FlexVol no se limita a un solo LUN y ofrece una mejor escalabilidad para las cargas de trabajo SAN. Dependiendo de la matriz de almacenamiento, ONTAP admite hasta 16384 LUN por clúster. Debido a que los volúmenes subyacentes son LUN, este controlador admite instantáneas granulares de volumen de Docker y clonación.

Elige el `ontap-nas-flexgroup` Controlador para aumentar el paralelismo en un único volumen que puede crecer hasta el rango de petabytes con miles de millones de archivos. Algunos casos de uso ideales para FlexGroups incluyen IA/ML/DL, big data y análisis, compilaciones de software, streaming, repositorios de archivos, etc. Trident utiliza todos los agregados asignados a una SVM al aprovisionar un volumen FlexGroup . La compatibilidad con FlexGroup en Trident también tiene las siguientes consideraciones:

- Requiere ONTAP versión 9.2 o superior.
- Al momento de escribir este artículo, FlexGroups solo admite NFS v3.
- Se recomienda habilitar los identificadores NFSv3 de 64 bits para la SVM.
- El tamaño mínimo recomendado para el miembro/volumen de FlexGroup es 100 GiB.
- La clonación no es compatible con volúmenes FlexGroup .

Para obtener información sobre FlexGroups y las cargas de trabajo adecuadas para FlexGroups, consulte la "Guía de implementación y mejores prácticas de NetApp FlexGroup volume" .

Para obtener funciones avanzadas y gran escala en el mismo entorno, puede ejecutar varias instancias del complemento Docker Volume, una de las cuales utilizará `ontap-nas` y otro usando `ontap-nas-economy` .

Rol ONTAP personalizado para Trident

Puede crear un rol de clúster ONTAP con privilegios mínimos para que no tenga que usar el rol de administrador de ONTAP para realizar operaciones en Trident. Cuando incluyes el nombre de usuario en una configuración de backend de Trident , Trident utiliza el rol de clúster ONTAP que creaste para realizar las operaciones.

Referirse a"Generador de roles personalizados de Trident" Para obtener más información sobre la creación de roles personalizados de Trident .

Uso de la CLI de ONTAP

1. Crea un nuevo rol utilizando el siguiente comando:

```
security login role create <role_name\> -cmddirname "command" -access all
-vserver <svm_name\>
```

2. Crea un nombre de usuario para el usuario de Trident :

```
security login create -username <user_name\> -application ontapi
-authmethod password -role <name_of_role_in_step_1\> -vserver <svm_name\>
-comment "user_description"
security login create -username <user_name\> -application http -authmethod
password -role <name_of_role_in_step_1\> -vserver <svm_name\> -comment
"user_description"
```

3. Asigna el rol al usuario:

```
security login modify username <user_name\> -vserver <svm_name\> -role
<role_name\> -application ontapi -application console -authmethod
<password\>
```

Usando el Administrador del sistema

Realice los siguientes pasos en ONTAP System Manager:

1. **Crea un rol personalizado**:
   a. Para crear un rol personalizado a nivel de clúster, seleccione **Clúster > Configuración**.

      (O) Para crear un rol personalizado a nivel de SVM, seleccione **Almacenamiento > Máquinas virtuales de almacenamiento > `required SVM` > Configuración > Usuarios y roles**.

   b. Seleccione el icono de flecha (→) junto a **Usuarios y roles**.
   c. Seleccione ****Agregar** en **Roles**.
   d. Define las reglas para el rol y haz clic en **Guardar**.
2. **Asigna el rol al usuario de Trident *: + Realiza los siguientes pasos en la página *Usuarios y roles**:
   a. Seleccione el icono Agregar ***+** debajo de **Usuarios**.
   b. Seleccione el nombre de usuario requerido y seleccione un rol en el menú desplegable para **Rol**.
   c. Haga clic en **Guardar**.

Para obtener más información, consulte las siguientes páginas:

- "Roles personalizados para la administración de ONTAP"o"Definir roles personalizados"
- "Trabajar con roles y usuarios"

Ejemplos de archivos de configuración de ONTAP

Ejemplo de NFS para el controlador <code>ontap-nas</code>

```
{
  "version": 1,
  "storageDriverName": "ontap-nas",
  "managementLIF": "10.0.0.1",
  "dataLIF": "10.0.0.2",
  "svm": "svm_nfs",
  "username": "vsadmin",
  "password": "password",
  "aggregate": "aggr1",
  "defaults": {
    "size": "10G",
    "spaceReserve": "none",
    "exportPolicy": "default"
  }
}
```

Ejemplo de NFS para el controlador <code>ontap-nas-flexgroup</code>

```
{
  "version": 1,
  "storageDriverName": "ontap-nas-flexgroup",
  "managementLIF": "10.0.0.1",
  "dataLIF": "10.0.0.2",
  "svm": "svm_nfs",
  "username": "vsadmin",
  "password": "password",
  "defaults": {
    "size": "100G",
    "spaceReserve": "none",
    "exportPolicy": "default"
  }
}
```

Ejemplo de NFS para el controlador <code>ontap-nas-economy</code>

```
{
  "version": 1,
  "storageDriverName": "ontap-nas-economy",
  "managementLIF": "10.0.0.1",
  "dataLIF": "10.0.0.2",
  "svm": "svm_nfs",
  "username": "vsadmin",
  "password": "password",
  "aggregate": "aggr1"
}
```

Ejemplo de iSCSI para el controlador <code>ontap-san</code>

```
{
  "version": 1,
  "storageDriverName": "ontap-san",
  "managementLIF": "10.0.0.1",
  "dataLIF": "10.0.0.3",
  "svm": "svm_iscsi",
  "username": "vsadmin",
  "password": "password",
  "aggregate": "aggr1",
  "igroupName": "netappdvp"
}
```

Ejemplo de NFS para el controlador <code>ontap-san-economy</code>

```
{
  "version": 1,
  "storageDriverName": "ontap-san-economy",
  "managementLIF": "10.0.0.1",
  "dataLIF": "10.0.0.3",
  "svm": "svm_iscsi_eco",
  "username": "vsadmin",
  "password": "password",
  "aggregate": "aggr1",
  "igroupName": "netappdvp"
}
```

Ejemplo NVMe/TCP para el controlador <code>ontap-san</code>

```
{
  "version": 1,
  "backendName": "NVMeBackend",
  "storageDriverName": "ontap-san",
  "managementLIF": "10.0.0.1",
  "svm": "svm_nvme",
  "username": "vsadmin",
  "password": "password",
  "sanType": "nvme",
  "useREST": true
}
```

Ejemplo de SCSI sobre FC para el controlador <code>ontap-san</code>

```
{
  "version": 1,
  "backendName": "ontap-san-backend",
  "storageDriverName": "ontap-san",
  "managementLIF": "10.0.0.1",
  "sanType": "fcp",
  "svm": "trident_svm",
  "username": "vsadmin",
  "password": "password",
  "useREST": true
}
```

Configuración del software Element

Además de los valores de configuración global, al usar el software Element (NetApp HCI/ SolidFire), están disponibles estas opciones.

Opción	Descripción	Ejemplo
`Endpoint`	<a href="https://<login>:<password>@<mvip>/json-rpc/<element-version&gt" class="bare">https://<login>:<password>@<mvip>/json-rpc/<element-version&gt;	https://admin:admin@192.168.160.3/json-rpc/8.0

Opción	Descripción	Ejemplo
`SVIP`	Dirección IP y puerto iSCSI	10.0.0.7:3260
`TenantName`	Tenent de SolidFireF a utilizar (se creará si no se encuentra)	`docker`
`InitiatorIFace`	Especifique la interfaz al restringir el tráfico iSCSI a una interfaz no predeterminada.	`default`
`Types`	Especificaciones de QoS	Vea el ejemplo a continuación.
`LegacyNamePrefix`	Prefijo para instalaciones Trident actualizadas. Si utilizó una versión de Trident anterior a la 1.3.2 y realiza una actualización con volúmenes existentes, deberá establecer este valor para acceder a sus volúmenes antiguos que se asignaron mediante el método nombre de volumen.	`netappdvp-`

El `solidfire-san` El controlador no es compatible con Docker Swarm.

Archivo de configuración de software de ejemplo de Element

```
{
  "version": 1,
  "storageDriverName": "solidfire-san",
  "Endpoint": "https://admin:admin@192.168.160.3/json-rpc/8.0",
  "SVIP": "10.0.0.7:3260",
  "TenantName": "docker",
  "InitiatorIFace": "default",
  "Types": [
    {
      "Type": "Bronze",
      "Qos": {
        "minIOPS": 1000,
        "maxIOPS": 2000,
        "burstIOPS": 4000
      }
    },
    {
      "Type": "Silver",
      "Qos": {
        "minIOPS": 4000,
        "maxIOPS": 6000,
        "burstIOPS": 8000
      }
    },
    {
      "Type": "Gold",
      "Qos": {
        "minIOPS": 6000,
        "maxIOPS": 8000,
        "burstIOPS": 10000
      }
    }
  ]
}
```

Problemas y limitaciones conocidos

Encuentre información sobre problemas conocidos y limitaciones al usar Trident con Docker.

La actualización del complemento Trident Docker Volume a la versión 20.10 o posterior desde versiones anteriores produce un fallo en la actualización con el error "no existe el archivo o directorio".

Solución alternativa

1. Desactive el plugin.

```
docker plugin disable -f netapp:latest
```

2. Elimina el plugin.

```
docker plugin rm -f netapp:latest
```

3. Reinstale el plugin proporcionando el archivo adicional. `config` parámetro.

```
docker plugin install netapp/trident-plugin:20.10 --alias netapp --grant
-all-permissions config=config.json
```

Los nombres de los volúmenes deben tener una longitud mínima de 2 caracteres.

Esta es una limitación del cliente Docker. El cliente interpretará un nombre de un solo carácter como una ruta de Windows. "Ver error 25773" .

Docker Swarm presenta ciertos comportamientos que impiden que Trident lo admita con todas las combinaciones de almacenamiento y controladores.

- Actualmente, Docker Swarm utiliza el nombre del volumen en lugar del ID del volumen como identificador único de volumen.
- Las solicitudes de volumen se envían simultáneamente a cada nodo de un clúster Swarm.
- Los complementos de volumen (incluido Trident) deben ejecutarse de forma independiente en cada nodo de un clúster Swarm. Debido al funcionamiento de ONTAP y a cómo `ontap-nas` y `ontap-san` Los conductores funcionan; son los únicos que, por casualidad, pueden operar dentro de estas limitaciones.

El resto de los controladores están sujetos a problemas como las condiciones de carrera que pueden dar lugar a la creación de una gran cantidad de volúmenes para una sola solicitud sin un "ganador" claro; por ejemplo, Element tiene una función que permite que los volúmenes tengan el mismo nombre pero diferentes identificadores.

NetApp ha enviado sus comentarios al equipo de Docker, pero no tiene ninguna indicación de que vaya a tomar medidas en el futuro.

Si se está aprovisionando un FlexGroup , ONTAP no aprovisiona un segundo FlexGroup si este FlexGroup tiene uno o más agregados en común con el FlexGroup que se está aprovisionando.

Mejores prácticas y recomendaciones

Despliegue

Utilice las recomendaciones que se enumeran aquí al implementar Trident.

Implementar en un espacio de nombres dedicado

"Espacios de nombres"Proporcionan separación administrativa entre diferentes aplicaciones y constituyen una barrera para el intercambio de recursos. Por ejemplo, un PVC de un espacio de nombres no se puede consumir desde otro. Trident proporciona recursos PV a todos los espacios de nombres del clúster de Kubernetes y, en consecuencia, utiliza una cuenta de servicio con privilegios elevados.

Además, el acceso al pod Trident podría permitir a un usuario acceder a las credenciales del sistema de almacenamiento y otra información confidencial. Es importante garantizar que los usuarios de la aplicación y las aplicaciones de gestión no tengan la capacidad de acceder a las definiciones de objetos de Trident ni a los propios pods.

Utilice cuotas y límites de rango para controlar el consumo de almacenamiento.

Kubernetes cuenta con dos características que, combinadas, proporcionan un mecanismo potente para limitar el consumo de recursos por parte de las aplicaciones. El "mecanismo de cuota de almacenamiento" permite al administrador implementar límites de consumo de capacidad y recuento de objetos globales y específicos de la clase de almacenamiento por espacio de nombres. Además, utilizando un "límite de rango" Garantiza que las solicitudes de PVC se encuentren dentro de un valor mínimo y máximo antes de que la solicitud se envíe al proveedor.

Estos valores se definen para cada espacio de nombres, lo que significa que cada espacio de nombres debe tener valores definidos que se ajusten a sus requisitos de recursos. Consulte aquí para obtener información sobre "cómo aprovechar las cuotas" .

Configuración de almacenamiento

Cada plataforma de almacenamiento del portafolio de NetApp tiene capacidades únicas que benefician a las aplicaciones, ya sean contenerizadas o no.

Descripción general de la plataforma

Trident funciona con ONTAP y Element. No existe una plataforma que se adapte mejor a todas las aplicaciones y escenarios que otra; sin embargo, al elegir una plataforma se deben tener en cuenta las necesidades de la aplicación y del equipo que administra el dispositivo.

Debes seguir las mejores prácticas básicas para el sistema operativo anfitrión con el protocolo que estés utilizando. Opcionalmente, puede considerar incorporar las mejores prácticas de la aplicación, cuando estén disponibles, con la configuración de backend, clase de almacenamiento y PVC para optimizar el almacenamiento para aplicaciones específicas.

Mejores prácticas de ONTAP y Cloud Volumes ONTAP

Aprenda las mejores prácticas para configurar ONTAP y Cloud Volumes ONTAP para Trident.

Las siguientes recomendaciones son pautas para configurar ONTAP para cargas de trabajo en contenedores, que consumen volúmenes aprovisionados dinámicamente por Trident. Cada una debe ser considerada y evaluada para determinar su idoneidad en su entorno.

Utilice SVM dedicados a Trident.

Las máquinas virtuales de almacenamiento (SVM) proporcionan aislamiento y separación administrativa entre inquilinos en un sistema ONTAP . Dedicar una SVM a las aplicaciones permite la delegación de privilegios y la aplicación de las mejores prácticas para limitar el consumo de recursos.

Existen varias opciones disponibles para la gestión de la SVM:

- Proporcione la interfaz de administración del clúster en la configuración del backend, junto con las credenciales apropiadas, y especifique el nombre de la SVM.
- Cree una interfaz de administración dedicada para la SVM utilizando ONTAP System Manager o la CLI.
- Comparta el rol de gestión con una interfaz de datos NFS.

En cada caso, la interfaz debe estar en el DNS y el nombre DNS debe usarse al configurar Trident. Esto ayuda a facilitar algunos escenarios de recuperación ante desastres, por ejemplo, SVM-DR sin el uso de retención de identidad de red.

No hay preferencia entre tener una LIF de administración dedicada o compartida para la SVM; sin embargo, debe asegurarse de que sus políticas de seguridad de red se alineen con el enfoque que elija. En cualquier caso, la LIF de gestión debería ser accesible a través de DNS para facilitar la máxima flexibilidad. "SVM-DR" debe utilizarse conjuntamente con Trident.

Limitar el recuento de volumen máximo

Los sistemas de almacenamiento ONTAP tienen un número máximo de volúmenes, que varía según la versión del software y la plataforma de hardware. Referirse a "NetApp Hardware Universe" para su plataforma específica y versión de ONTAP para determinar los límites exactos. Cuando se agota el número de volúmenes, las operaciones de aprovisionamiento fallan no solo para Trident, sino para todas las solicitudes de almacenamiento.

Trident's `ontap-nas` y `ontap-san` Los controladores aprovisionan un FlexVolume para cada Volumen Persistente (PV) de Kubernetes que se crea. El `ontap-nas-economy` El controlador crea aproximadamente un FlexVolume por cada 200 PV (configurable entre 50 y 300). El `ontap-san-economy` El controlador crea aproximadamente un FlexVolume por cada 100 PV (configurable entre 50 y 200). Para evitar que Trident consuma todos los volúmenes disponibles en el sistema de almacenamiento, debe establecer un límite en la SVM. Puedes hacer esto desde la línea de comando:

```
vserver modify -vserver <svm_name> -max-volumes <num_of_volumes>
```

El valor para `max-volumes` varía en función de varios criterios específicos de su entorno:

- Número de volúmenes existentes en el clúster ONTAP
- El número de volúmenes que prevé aprovisionar fuera de Trident para otras aplicaciones
- Número de volúmenes persistentes que se espera que consuman las aplicaciones de Kubernetes

El `max-volumes` El valor es el total de volúmenes aprovisionados en todos los nodos del clúster ONTAP , y no en un nodo ONTAP individual. Como resultado, es posible que encuentre algunas condiciones en las que

un nodo de clúster de ONTAP podría tener muchos más o menos volúmenes aprovisionados Trident que otro nodo.

Por ejemplo, un clúster ONTAP de dos nodos tiene la capacidad de alojar un máximo de 2000 volúmenes FlexVol . Establecer el recuento máximo de volumen en 1250 parece muy razonable. Sin embargo, si tan solo "agregados" Si los agregados asignados desde un nodo se asignan al SVM, o si los agregados asignados desde un nodo no se pueden aprovisionar (por ejemplo, debido a la capacidad), entonces el otro nodo se convierte en el destino de todos los volúmenes aprovisionados de Trident . Esto significa que el límite de volumen podría alcanzarse para ese nodo antes de que… `max-volumes` Se alcanza un valor determinado, lo que repercute tanto en Trident como en otras operaciones de volumen que utilizan ese nodo. **Puede evitar esta situación asegurándose de que los agregados de cada nodo del clúster se asignen a la SVM utilizada por Trident en igual número.**

Clonar un volumen

NetApp Trident admite la clonación de volúmenes al usar `ontap-nas` , `ontap-san` , `solidfire-san` , y `gcp-cvs` Controladores de almacenamiento. Al usar el `ontap-nas-flexgroup` o `ontap-nas-economy` controladores, la clonación no es compatible. La creación de un nuevo volumen a partir de un volumen existente dará como resultado la creación de una nueva instantánea.

> Evite clonar un PVC que esté asociado con una StorageClass diferente. Realice operaciones de clonación dentro de la misma StorageClass para garantizar la compatibilidad y evitar comportamientos inesperados.

Limitar el tamaño máximo de los volúmenes creados por Trident

Para configurar el tamaño máximo de los volúmenes que puede crear Trident, utilice el `limitVolumeSize` parámetro en su `backend.json` definición.

Además de controlar el tamaño del volumen en la matriz de almacenamiento, también debe aprovechar las capacidades de Kubernetes.

Limitar el tamaño máximo de los FlexVols creados por Trident

Para configurar el tamaño máximo de los FlexVols utilizados como pools para los controladores ontap-san-economy y ontap-nas-economy, utilice el siguiente comando: `limitVolumePoolSize` parámetro en su `backend.json` definición.

Configure Trident para usar CHAP bidireccional.

Puedes especificar el iniciador CHAP y los nombres de usuario y contraseñas de destino en la definición de tu backend y hacer que Trident habilite CHAP en la SVM. Utilizando el `useCHAP` En la configuración de su backend, Trident autentica las conexiones iSCSI para backends ONTAP con CHAP.

Cree y utilice una política de QoS para SVM.

Al aprovechar una política QoS de ONTAP , aplicada a la SVM, se limita el número de IOPS que pueden consumir los volúmenes aprovisionados de Trident . Esto ayuda a "prevenir el acoso" o un contenedor fuera de control que afecte a cargas de trabajo externas a la SVM de Trident .

Puedes crear una política QoS para la SVM en pocos pasos. Consulte la documentación de su versión de ONTAP para obtener la información más precisa. El ejemplo a continuación crea una política de QoS que limita el total de IOPS disponibles para la SVM a 5000.

```
# create the policy group for the SVM
qos policy-group create -policy-group <policy_name> -vserver <svm_name>
-max-throughput 5000iops

# assign the policy group to the SVM, note this will not work
# if volumes or files in the SVM have existing QoS policies
vserver modify -vserver <svm_name> -qos-policy-group <policy_name>
```

Además, si su versión de ONTAP lo admite, puede considerar el uso de un mínimo de QoS para garantizar una cantidad de rendimiento para las cargas de trabajo en contenedores. La QoS adaptativa no es compatible con una política a nivel de SVM.

El número de IOPS dedicadas a las cargas de trabajo en contenedores depende de muchos aspectos. Entre otras cosas, estas incluyen:

- Otras cargas de trabajo que utilizan la matriz de almacenamiento. Si existen otras cargas de trabajo, no relacionadas con la implementación de Kubernetes, que utilizan los recursos de almacenamiento, se debe tener cuidado para garantizar que dichas cargas de trabajo no se vean afectadas negativamente de forma accidental.
- Cargas de trabajo previstas que se ejecutan en contenedores. Si las cargas de trabajo con altos requisitos de IOPS se ejecutan en contenedores, una política de QoS baja resulta en una mala experiencia.

Es importante recordar que una política de QoS asignada a nivel de SVM da como resultado que todos los volúmenes aprovisionados al SVM compartan el mismo grupo de IOPS. Si una, o un pequeño número, de las aplicaciones en contenedores tienen un alto requerimiento de IOPS, podrían convertirse en un problema para las demás cargas de trabajo en contenedores. Si este es el caso, quizás le convenga considerar el uso de automatización externa para asignar políticas de QoS por volumen.

> Debe asignar el grupo de políticas QoS al SVM **únicamente** si su versión de ONTAP es anterior a la 9.8.

Cree grupos de políticas de QoS para Trident.

La calidad de servicio (QoS) garantiza que el rendimiento de las cargas de trabajo críticas no se vea degradado por cargas de trabajo competidoras. Los grupos de políticas QoS de ONTAP proporcionan opciones de QoS para volúmenes y permiten a los usuarios definir el límite máximo de rendimiento para una o más cargas de trabajo. Para obtener más información sobre QoS, consulte "Garantizar el rendimiento con QoS" . Puede especificar grupos de políticas QoS en el backend o en un grupo de almacenamiento, y estos se aplicarán a cada volumen creado en ese grupo o backend.

ONTAP cuenta con dos tipos de grupos de políticas QoS: tradicionales y adaptativas. Los grupos de políticas tradicionales proporcionan un rendimiento máximo (o mínimo, en versiones posteriores) fijo en IOPS. La QoS adaptativa ajusta automáticamente el rendimiento al tamaño de la carga de trabajo, manteniendo la relación de IOPS a TB/GB a medida que cambia el tamaño de la carga de trabajo. Esto proporciona una ventaja significativa cuando se gestionan cientos o miles de cargas de trabajo en una implementación de gran tamaño.

Tenga en cuenta lo siguiente al crear grupos de políticas de QoS:

- Debes configurar el `qosPolicy` clave en el `defaults` bloque de la configuración del backend. Vea el siguiente ejemplo de configuración de backend:

```
---
version: 1
storageDriverName: ontap-nas
managementLIF: 0.0.0.0
dataLIF: 0.0.0.0
svm: svm0
username: user
password: pass
defaults:
  qosPolicy: standard-pg
storage:
  - labels:
      performance: extreme
    defaults:
      adaptiveQosPolicy: extremely-adaptive-pg
  - labels:
      performance: premium
    defaults:
      qosPolicy: premium-pg
```

- Debe aplicar los grupos de políticas por volumen, de modo que cada volumen obtenga el rendimiento total especificado por el grupo de políticas. No se admiten grupos de políticas compartidas.

Para obtener más información sobre los grupos de políticas de QoS, consulte "Referencia de comandos de ONTAP" .

Limitar el acceso a los recursos de almacenamiento a los miembros del clúster de Kubernetes

Limitar el acceso a los volúmenes NFS, LUN iSCSI y LUN FC creados por Trident es un componente crítico de la postura de seguridad para su implementación de Kubernetes. Al hacerlo, se evita que los hosts que no forman parte del clúster de Kubernetes accedan a los volúmenes y potencialmente modifiquen los datos de forma inesperada.

Es importante comprender que los espacios de nombres son el límite lógico de los recursos en Kubernetes. Se asume que los recursos en el mismo espacio de nombres se pueden compartir; sin embargo, es importante destacar que no existe capacidad para compartir entre espacios de nombres. Esto significa que, aunque los PV son objetos globales, cuando están vinculados a un PVC solo son accesibles para los pods que se encuentran en el mismo espacio de nombres. **Es fundamental garantizar que se utilicen espacios de nombres para proporcionar separación cuando sea apropiado.**

La principal preocupación para la mayoría de las organizaciones con respecto a la seguridad de los datos en un contexto de Kubernetes es que un proceso en un contenedor pueda acceder al almacenamiento montado en el host, pero que no está destinado al contenedor. "Espacios de nombres" están diseñados para prevenir este tipo de vulneración. Sin embargo, existe una excepción: los contenedores privilegiados.

Un contenedor privilegiado es aquel que se ejecuta con muchos más permisos a nivel de host de lo normal. Estas opciones no están denegadas de forma predeterminada, así que asegúrese de deshabilitar la funcionalidad mediante el uso de "políticas de seguridad de pods" .

Para los volúmenes en los que se desea acceso tanto desde Kubernetes como desde hosts externos, el

almacenamiento debe gestionarse de forma tradicional, con el PV introducido por el administrador y no gestionado por Trident. Esto garantiza que el volumen de almacenamiento se destruya solo cuando tanto Kubernetes como los hosts externos se hayan desconectado y ya no estén utilizando el volumen. Además, se puede aplicar una política de exportación personalizada, que permite el acceso desde los nodos del clúster de Kubernetes y servidores específicos fuera del clúster de Kubernetes.

Para implementaciones que tienen nodos de infraestructura dedicados (por ejemplo, OpenShift) u otros nodos que no pueden programar aplicaciones de usuario, se deben utilizar políticas de exportación separadas para limitar aún más el acceso a los recursos de almacenamiento. Esto incluye la creación de una política de exportación para los servicios que se implementan en esos nodos de infraestructura (por ejemplo, los servicios de métricas y registro de OpenShift) y las aplicaciones estándar que se implementan en nodos que no son de infraestructura.

Utilice una política de exportación específica

Debe asegurarse de que exista una política de exportación para cada backend que solo permita el acceso a los nodos presentes en el clúster de Kubernetes. Trident puede crear y gestionar automáticamente políticas de exportación. De esta manera, Trident limita el acceso a los volúmenes que aprovisiona a los nodos del clúster de Kubernetes y simplifica la adición o eliminación de nodos.

Como alternativa, también puede crear una política de exportación manualmente y completarla con una o más reglas de exportación que procesen cada solicitud de acceso a un nodo:

- Utilice el `vserver export-policy create` Comando CLI de ONTAP para crear la política de exportación.
- Agregue reglas a la política de exportación utilizando el `vserver export-policy rule create` Comando CLI de ONTAP .

Al ejecutar estos comandos, podrá restringir qué nodos de Kubernetes tienen acceso a los datos.

Desactivar `showmount` para la aplicación SVM

El `showmount` Esta función permite que un cliente NFS consulte al SVM para obtener una lista de las exportaciones NFS disponibles. Un pod desplegado en el clúster de Kubernetes puede emitir el `showmount -e` Ejecuta el comando contra el servidor y recibe una lista de los puntos de montaje disponibles, incluidos aquellos a los que no tiene acceso. Si bien esto, por sí solo, no constituye una vulneración de la seguridad, sí proporciona información innecesaria que podría ayudar a un usuario no autorizado a conectarse a una exportación NFS.

Debes desactivar `showmount` utilizando el comando CLI de ONTAP a nivel de SVM:

```
vserver nfs modify -vserver <svm_name> -showmount disabled
```

Mejores prácticas de SolidFire

Aprenda las mejores prácticas para configurar el almacenamiento SolidFire para Trident.

Crear cuenta de Solidfire

Cada cuenta de SolidFire representa un propietario de volumen único y recibe su propio conjunto de credenciales del Protocolo de Autenticación por Desafío-Respuesta (CHAP). Puede acceder a los volúmenes asignados a una cuenta utilizando el nombre de la cuenta y las credenciales CHAP correspondientes o a

través de un grupo de acceso a volúmenes. Una cuenta puede tener asignados hasta dos mil volúmenes, pero un volumen solo puede pertenecer a una cuenta.

Crea una política de QoS

Utilice las políticas de Calidad de Servicio (QoS) de SolidFire si desea crear y guardar una configuración de calidad de servicio estandarizada que se pueda aplicar a muchos volúmenes.

Puede configurar los parámetros de QoS para cada volumen. El rendimiento de cada volumen se puede garantizar configurando tres parámetros configurables que definen la QoS: IOPS mínimas, IOPS máximas e IOPS de ráfaga.

Aquí están los posibles valores mínimos, máximos y de ráfaga de IOPS para el tamaño de bloque de 4Kb.

Parámetro IOPS	Definición	Valor mínimo	Valor predeterminado	Valor máximo (4 KB)
IOPS mínimas	El nivel de rendimiento garantizado para un volumen.	50	50	15000
IOPS máximas	El rendimiento no excederá este límite.	50	15000	200.000
Burst IOPS	IOPS máximo permitido en un escenario de ráfaga corta.	50	15000	200.000

Aunque el IOPS máximo y el IOPS de ráfaga se pueden configurar hasta en 200.000, el rendimiento máximo real de un volumen está limitado por el uso del clúster y el rendimiento por nodo.

El tamaño del bloque y el ancho de banda influyen directamente en el número de IOPS. A medida que aumenta el tamaño de los bloques, el sistema incrementa el ancho de banda hasta el nivel necesario para procesar los bloques de mayor tamaño. A medida que aumenta el ancho de banda, disminuye el número de IOPS que el sistema puede alcanzar. Referirse a "Calidad de servicio de SolidFire" Para obtener más información sobre QoS y rendimiento.

Autenticación de SolidFire

Element admite dos métodos de autenticación: CHAP y Grupos de Acceso por Volumen (VAG). CHAP utiliza el protocolo CHAP para autenticar el host ante el servidor backend. Los grupos de acceso a volúmenes controlan el acceso a los volúmenes que aprovisionan. NetApp recomienda utilizar CHAP para la autenticación, ya que es más sencillo y no tiene límites de escalabilidad.

Trident con el aprovisionador CSI mejorado admite el uso de la autenticación CHAP. Los VAG solo deben utilizarse en el modo de funcionamiento tradicional no CSI.

La autenticación CHAP (verificación de que el iniciador es el usuario de volumen previsto) solo es compatible con el control de acceso basado en cuentas. Si utiliza CHAP para la autenticación, dispone de dos opciones:

CHAP unidireccional y CHAP bidireccional. El protocolo CHAP unidireccional autentica el acceso al volumen utilizando el nombre de cuenta de SolidFire y el secreto del iniciador. La opción CHAP bidireccional proporciona la forma más segura de autenticar el volumen porque el volumen autentica al host a través del nombre de cuenta y el secreto del iniciador, y luego el host autentica al volumen a través del nombre de cuenta y el secreto del destino.

Sin embargo, si no se puede habilitar CHAP y se requieren VAG, cree el grupo de acceso y agregue los iniciadores de host y los volúmenes al grupo de acceso. Cada IQN que agregue a un grupo de acceso puede acceder a cada volumen del grupo con o sin autenticación CHAP. Si el iniciador iSCSI está configurado para usar la autenticación CHAP, se utiliza el control de acceso basado en cuentas. Si el iniciador iSCSI no está configurado para usar la autenticación CHAP, entonces se utiliza el control de acceso de Grupo de Acceso a Volumen.

¿Dónde puedo encontrar más información?

A continuación se enumeran algunos de los documentos sobre mejores prácticas. Buscar en el "Biblioteca NetApp" para las versiones más recientes.

- ONTAP*
- "Guía de buenas prácticas e implementación de NFS"
- "Administración SAN"(para iSCSI)
- "Configuración de iSCSI Express para RHEL"

Software Element

- "Configuración de SolidFire para Linux"
- NetApp HCI*
- "Requisitos previos para la implementación de NetApp HCI"
- "Acceda al motor de implementación de NetApp"

Información sobre las mejores prácticas de la aplicación

- "Mejores prácticas para MySQL en ONTAP"
- "Mejores prácticas para MySQL en SolidFire"
- "NetApp SolidFire y Cassandra"
- "Mejores prácticas de Oracle en SolidFire"
- "Mejores prácticas de PostgreSQL en SolidFire"

No todas las aplicaciones tienen directrices específicas; es importante trabajar con su equipo de NetApp y utilizar las "Biblioteca NetApp" para encontrar la documentación más actualizada.

Integrar Trident

Para integrar Trident, se requiere la integración de los siguientes elementos de diseño y arquitectura: selección e implementación de controladores, diseño de clases de almacenamiento, diseño de grupos virtuales, impactos de Persistent Volume Claim (PVC) en el aprovisionamiento de almacenamiento, operaciones de volumen e implementación de servicios de OpenShift utilizando Trident.

Selección y despliegue de controladores

Seleccione e implemente un controlador de backend para su sistema de almacenamiento.

Controladores de backend de ONTAP

Los controladores backend de ONTAP se diferencian por el protocolo utilizado y la forma en que se aprovisionan los volúmenes en el sistema de almacenamiento. Por lo tanto, considere cuidadosamente qué controlador implementar.

En un nivel superior, si su aplicación tiene componentes que necesitan almacenamiento compartido (varios pods que acceden al mismo PVC), los controladores basados en NAS serían la opción predeterminada, mientras que los controladores iSCSI basados en bloques satisfacen las necesidades de almacenamiento no compartido. Elija el protocolo en función de los requisitos de la aplicación y el nivel de familiaridad de los equipos de almacenamiento e infraestructura. En términos generales, hay poca diferencia entre ellos para la mayoría de las aplicaciones, por lo que a menudo la decisión se basa en si se necesita o no almacenamiento compartido (donde más de un pod necesitará acceso simultáneo).

Los controladores de backend de ONTAP disponibles son:

- `ontap-nas`Cada PV aprovisionado es un ONTAP FlexVolume completo.
- `ontap-nas-economy`Cada PV aprovisionado es un qtree, con un número configurable de qtrees por FlexVolume (el valor predeterminado es 200).
- `ontap-nas-flexgroup`Cada PV se aprovisiona como un ONTAP FlexGroup completo y se utilizan todos los agregados asignados a una SVM.
- `ontap-san`Cada PV aprovisionado es un LUN dentro de su propio FlexVolume.
- `ontap-san-economy`Cada PV aprovisionado es un LUN, con un número configurable de LUN por FlexVolume (el valor predeterminado es 100).

Elegir entre los tres controladores NAS tiene algunas repercusiones en las funciones que se ponen a disposición de la aplicación.

Tenga en cuenta que, en las tablas siguientes, no todas las capacidades están expuestas a través de Trident. Algunas de estas funcionalidades deben ser aplicadas por el administrador de almacenamiento después del aprovisionamiento si se desea dicha funcionalidad. Las notas a pie de página en superíndice distinguen la funcionalidad de cada característica y controlador.

Controladores NAS ONTAP	Instantáneas	Clones	Políticas de exportación dinámicas	Multi-attach	Calidad de servicio	Cambiar tamaño	Replicación
`ontap-nas`	Sí	Sí	Sí, nota al pie:5[]	Sí	Sí, nota al pie:1[]	Sí	Sí, nota al pie:1[]
`ontap-nas-economy`	Sin nota al pie:3[]	Sin nota al pie:3[]	Sí, nota al pie:5[]	Sí	Sin nota al pie:3[]	Sí	Sin nota al pie:3[]
`ontap-nas-flexgroup`	Sí, nota al pie:1[]	NO	Sí, nota al pie:5[]	Sí	Sí, nota al pie:1[]	Sí	Sí, nota al pie:1[]

Trident ofrece 2 controladores SAN para ONTAP, cuyas capacidades se muestran a continuación.

Controladores SAN de ONTAP	Instantáneas	Clones	Multi-attach	CHAP bidireccional	Calidad de servicio	Cambiar tamaño	Replicación
`ontap-san`	Sí	Sí	Sí, nota al pie:4[]	Sí	Sí, nota al pie:1[]	Sí	Sí, nota al pie:1[]
`ontap-san-economy`	Sí	Sí	Sí, nota al pie:4[]	Sí	Sin nota al pie:3[]	Sí	Sin nota al pie:3[]

> Nota al pie de las tablas anteriores: Sí (nota al pie 1): No gestionado por Trident. Sí (nota al pie 2): Gestionado por Trident, pero sin granularidad PV. No (nota al pie 3): No gestionado por Trident ni con granularidad PV. Sí (nota al pie 4): Compatible con volúmenes de bloques sin formato. Sí (nota al pie 5): Compatible con Trident.

Las características que no son granulares a nivel de PV se aplican a todo el FlexVolume y todos los PV (es decir, qtrees o LUN en FlexVols compartidos) compartirán una programación común.

Como podemos ver en las tablas anteriores, gran parte de la funcionalidad entre los `ontap-nas` y `ontap-nas-economy` es lo mismo. Sin embargo, debido a que `ontap-nas-economy` El controlador limita la capacidad de controlar la programación a nivel de cada PV, lo que puede afectar especialmente a la planificación de la recuperación ante desastres y las copias de seguridad. Para los equipos de desarrollo que deseen aprovechar la funcionalidad de clonación de PVC en el almacenamiento ONTAP , esto solo es posible al usar el `ontap-nas` , `ontap-san` o `ontap-san-economy` conductores.

> El `solidfire-san` El controlador también es capaz de clonar PVC.

Controladores de backend de Cloud Volumes ONTAP

Cloud Volumes ONTAP proporciona control de datos junto con funciones de almacenamiento de clase empresarial para diversos casos de uso, incluyendo recursos compartidos de archivos y almacenamiento a nivel de bloque que admiten protocolos NAS y SAN (NFS, SMB/CIFS e iSCSI). Los controladores compatibles para Cloud Volume ONTAP son `ontap-nas` , `ontap-nas-economy` , `ontap-san` y `ontap-san-economy` . Estas son aplicables a Cloud Volume ONTAP para Azure y Cloud Volume ONTAP para GCP.

Controladores de backend de Amazon FSx para ONTAP

Amazon FSx for NetApp ONTAP le permite aprovechar las características, el rendimiento y las capacidades administrativas de NetApp con las que ya está familiarizado, al tiempo que se beneficia de la simplicidad, la agilidad, la seguridad y la escalabilidad del almacenamiento de datos en AWS. FSx para ONTAP admite muchas funciones del sistema de archivos ONTAP y API de administración. Los controladores compatibles para Cloud Volume ONTAP son `ontap-nas` , `ontap-nas-economy` , `ontap-nas-flexgroup` , `ontap-san` y `ontap-san-economy` .

Controladores de backend NetApp HCI/ SolidFire

El `solidfire-san` El controlador utilizado con las plataformas NetApp HCI/ SolidFire ayuda al administrador a configurar un backend de Element para Trident en función de los límites de QoS. Si desea diseñar su backend para establecer límites de QoS específicos en los volúmenes aprovisionados por Trident, utilice `type` parámetro en el archivo backend. El administrador también puede restringir el tamaño del volumen que se puede crear en el almacenamiento mediante la `limitVolumeSize` parámetro. Actualmente, las funciones de almacenamiento de Element, como el cambio de tamaño y la replicación de volúmenes, no son compatibles

con la versión anterior. `solidfire-san` conductor. Estas operaciones deben realizarse manualmente a través de la interfaz web de Element Software.

Controlador SolidFire	Instantáneas	Clones	Multi-attach	CHAP	Calidad de servicio	Cambiar tamaño	Replicación
`solidfire-san`	Sí	Sí	Sí, nota al pie:2[]	Sí	Sí	Sí	Sí, nota al pie:1[]

> Nota al pie: Sí (nota al pie 1): No gestionado por Trident. Sí (nota al pie 2): Compatible con volúmenes de bloques sin formato.

Controladores de backend de Azure NetApp Files

Trident utiliza el `azure-netapp-files` conductor para gestionar el"Azure NetApp Files" servicio.

Encontrará más información sobre este controlador y cómo configurarlo en"Configuración de backend de Trident para Azure NetApp Files" .

Controlador de Azure NetApp Files	Instantáneas	Clones	Multi-attach	Calidad de servicio	Expandir	Replicación
`azure-netapp-files`	Sí	Sí	Sí	Sí	Sí	Sí, nota al pie:1[]

> Nota al pie: Sí. Nota al pie 1: No gestionado por Trident.

Cloud Volumes Service en el controlador backend de Google Cloud

Trident utiliza el `gcp-cvs` Controlador para conectarse con el Cloud Volumes Service en Google Cloud.

El `gcp-cvs` El controlador utiliza grupos virtuales para abstraer el backend y permitir que Trident determine la ubicación del volumen. El administrador define los grupos virtuales en el `backend.json` archivos. Las clases de almacenamiento utilizan selectores para identificar los grupos virtuales por etiqueta.

- Si se definen grupos virtuales en el backend, Trident intentará crear un volumen en los grupos de almacenamiento de Google Cloud a los que están limitados esos grupos virtuales.
- Si no se definen grupos virtuales en el backend, Trident seleccionará un grupo de almacenamiento de Google Cloud de entre los grupos de almacenamiento disponibles en la región.

Para configurar el backend de Google Cloud en Trident, debe especificar `projectNumber` , `apiRegion` , y `apiKey` en el archivo de backend. Puedes encontrar el número de proyecto en la consola de Google Cloud. La clave API se toma del archivo de clave privada de la cuenta de servicio que creó al configurar el acceso a la API para Cloud Volumes Service en Google Cloud.

Para obtener detalles sobre los tipos y niveles de servicio de Cloud Volumes Service en Google Cloud, consulte:"Obtenga más información sobre la compatibilidad de Trident con CVS para GCP" .

Controlador del Cloud Volumes Service para Google Cloud	Instantáneas	Clones	Multi-attach	Calidad de servicio	Expandir	Replicación
`gcp-cvs`	Sí	Sí	Sí	Sí	Sí	Disponible únicamente en el tipo de servicio CVS-Performance.

Notas de replicación

- La replicación no la gestiona Trident.
- El clon se creará en el mismo grupo de almacenamiento que el volumen de origen.

Diseño de clase de almacenamiento

Es necesario configurar y aplicar clases de almacenamiento individuales para crear un objeto de clase de almacenamiento de Kubernetes. En esta sección se explica cómo diseñar una clase de almacenamiento para su aplicación.

Utilización específica del backend

El filtrado se puede utilizar dentro de un objeto de clase de almacenamiento específico para determinar qué grupo o conjunto de grupos de almacenamiento se utilizará con esa clase de almacenamiento específica. Se pueden configurar tres conjuntos de filtros en la clase de almacenamiento: `storagePools` , `additionalStoragePools` , y/o `excludeStoragePools` .

El `storagePools` Este parámetro ayuda a restringir el almacenamiento al conjunto de pools que coincidan con los atributos especificados. El `additionalStoragePools` El parámetro se utiliza para ampliar el conjunto de pools que Trident utiliza para el aprovisionamiento, junto con el conjunto de pools seleccionados por los atributos y `storagePools` parámetros. Puede utilizar cualquiera de los parámetros por separado o ambos juntos para asegurarse de que se selecciona el conjunto adecuado de grupos de almacenamiento.

El `excludeStoragePools` El parámetro se utiliza para excluir específicamente el conjunto de pools enumerados que coinciden con los atributos.

Emular políticas de QoS

Si desea diseñar clases de almacenamiento para emular políticas de calidad de servicio, cree una clase de almacenamiento con la siguiente estructura: `media` atributo como `hdd` o `ssd` . Basado en el `media` Trident seleccionará el backend apropiado que corresponda al atributo mencionado en la clase de almacenamiento. `hdd` o `ssd` agrega para que coincida con el atributo de medios y luego dirige el aprovisionamiento de los volúmenes al agregado específico. Por lo tanto, podemos crear una clase de almacenamiento PREMIUM que tendría `media` atributo establecido como `ssd` lo cual podría clasificarse como la política QoS PREMIUM. Podemos crear otra clase de almacenamiento STANDARD que tendría el atributo de medios establecido como `hdd`, que podría clasificarse como la política QoS STANDARD. También podríamos usar el atributo "IOPS" en la clase de almacenamiento para redirigir el aprovisionamiento a un dispositivo Element que se puede definir como una política de QoS.

Utilizar el backend en función de características específicas

Las clases de almacenamiento se pueden diseñar para dirigir el aprovisionamiento de volúmenes en un backend específico donde se habilitan funciones como el aprovisionamiento ligero y grueso, instantáneas, clones y cifrado. Para especificar qué almacenamiento utilizar, cree clases de almacenamiento que especifiquen el backend apropiado con la función requerida habilitada.

piscinas virtuales

Los pools virtuales están disponibles para todos los backends de Trident . Puedes definir grupos virtuales para cualquier backend, utilizando cualquier controlador que proporcione Trident .

Los grupos virtuales permiten a un administrador crear un nivel de abstracción sobre los backends a los que se puede hacer referencia a través de clases de almacenamiento, para una mayor flexibilidad y una ubicación eficiente de los volúmenes en los backends. Se pueden definir diferentes backends con la misma clase de servicio. Además, se pueden crear múltiples grupos de almacenamiento en el mismo backend, pero con características diferentes. Cuando una clase de almacenamiento se configura con un selector con etiquetas específicas, Trident elige un backend que coincida con todas las etiquetas del selector para ubicar el volumen. Si las etiquetas del selector de clase de almacenamiento coinciden con varios grupos de almacenamiento, Trident elegirá uno de ellos para aprovisionar el volumen.

Diseño de piscina virtual

Al crear un backend, generalmente se puede especificar un conjunto de parámetros. Antes, el administrador no podía crear otro backend con las mismas credenciales de almacenamiento y un conjunto de parámetros diferente. Con la introducción de los grupos virtuales, este problema se ha solucionado. Un grupo virtual es una abstracción de nivel entre el backend y la clase de almacenamiento de Kubernetes, que permite al administrador definir parámetros y etiquetas, a los que se puede hacer referencia mediante clases de almacenamiento de Kubernetes como selector, de forma independiente del backend. Se pueden definir grupos virtuales para todos los backends de NetApp compatibles con Trident. Esta lista incluye SolidFire/ NetApp HCI, ONTAP, Cloud Volumes Service en GCP y Azure NetApp Files.

> Al definir grupos virtuales, se recomienda no intentar reorganizar el orden de los grupos virtuales existentes en una definición de backend. También es recomendable no editar/modificar los atributos de un grupo virtual existente y, en su lugar, definir un nuevo grupo virtual.

Emulación de diferentes niveles de servicio/QoS

Es posible diseñar pools virtuales para emular clases de servicio. Utilizando la implementación de grupo virtual para Cloud Volume Service para Azure NetApp Files, examinemos cómo podemos configurar diferentes clases de servicio. Configure el backend de Azure NetApp Files con varias etiquetas que representen diferentes niveles de rendimiento. Colocar `servicelevel` ajuste cada aspecto al nivel de rendimiento adecuado y añada los demás aspectos necesarios bajo cada etiqueta. Ahora crea diferentes clases de almacenamiento de Kubernetes que se correspondan con diferentes grupos virtuales. Utilizando el `parameters.selector` En cada campo, cada StorageClass especifica qué grupos virtuales se pueden usar para alojar un volumen.

Asignación de un conjunto específico de aspectos

Se pueden diseñar múltiples pools virtuales con un conjunto específico de aspectos a partir de un único backend de almacenamiento. Para ello, configure el backend con múltiples etiquetas y establezca los aspectos necesarios bajo cada etiqueta. Ahora crea diferentes clases de almacenamiento de Kubernetes usando `parameters.selector` campo que se asignaría a diferentes grupos virtuales. Los volúmenes que se aprovisionan en el backend tendrán los aspectos definidos en el grupo virtual elegido.

Características del PVC que afectan al aprovisionamiento de almacenamiento

Algunos parámetros que van más allá de la clase de almacenamiento solicitada pueden afectar el proceso de decisión de aprovisionamiento de Trident al crear un PVC.

Modo de acceso

Al solicitar almacenamiento a través de un PVC, uno de los campos obligatorios es el modo de acceso. El modo deseado puede afectar al backend seleccionado para alojar la solicitud de almacenamiento.

Trident intentará hacer coincidir el protocolo de almacenamiento utilizado con el método de acceso especificado según la siguiente matriz. Esto es independiente de la plataforma de almacenamiento subyacente.

	Leer y escribir una vez	Solo lecturaMuchos	LeerEscribirMuchos
iSCSI	Sí	Sí	Sí (Bloque sin procesar)
Sistema Nacional de Archivos	Sí	Sí	Sí

Una solicitud de PVC ReadWriteMany enviada a una implementación de Trident sin un backend NFS configurado dará como resultado que no se aprovisione ningún volumen. Por este motivo, el solicitante deberá utilizar el modo de acceso adecuado para su aplicación.

Operaciones de volumen

Modificar volúmenes persistentes

Los volúmenes persistentes son, con dos excepciones, objetos inmutables en Kubernetes. Una vez creada, la política de reclamaciones y su tamaño pueden modificarse. Sin embargo, esto no impide que algunos aspectos del volumen se modifiquen fuera de Kubernetes. Esto puede ser conveniente para personalizar el volumen para aplicaciones específicas, para garantizar que la capacidad no se consuma accidentalmente o simplemente para mover el volumen a un controlador de almacenamiento diferente por cualquier motivo.

> Actualmente, los provisionadores integrados de Kubernetes no admiten operaciones de cambio de tamaño de volumen para PV NFS, iSCSI o FC. Trident admite la expansión de volúmenes NFS, iSCSI y FC.

Los detalles de conexión del PV no se pueden modificar después de su creación.

Crear instantáneas de volumen bajo demanda

Trident admite la creación de instantáneas de volumen bajo demanda y la creación de PVC a partir de instantáneas utilizando el marco CSI. Las instantáneas proporcionan un método conveniente para mantener copias puntuales de los datos y tienen un ciclo de vida independiente del PV de origen en Kubernetes. Estas instantáneas se pueden utilizar para clonar PVC.

Crear volúmenes a partir de instantáneas

Trident también admite la creación de PersistentVolumes a partir de instantáneas de volumen. Para lograr esto, simplemente cree un PersistentVolumeClaim y mencione el `datasource` como la instantánea requerida a partir de la cual se debe crear el volumen. Trident gestionará este PVC creando un volumen con los datos presentes en la instantánea. Con esta función, es posible duplicar datos entre regiones, crear entornos de

prueba, reemplazar un volumen de producción dañado o corrupto en su totalidad, o recuperar archivos y directorios específicos y transferirlos a otro volumen conectado.

Mover volúmenes en el clúster

Los administradores de almacenamiento tienen la capacidad de mover volúmenes entre agregados y controladores en el clúster ONTAP sin interrumpir el funcionamiento del consumidor de almacenamiento. Esta operación no afecta a Trident ni al clúster de Kubernetes, siempre y cuando el agregado de destino sea uno al que tenga acceso la SVM que utiliza Trident . Es importante destacar que, si el agregado se ha añadido recientemente al SVM, será necesario actualizar el backend volviéndolo a añadir a Trident. Esto hará que Trident reinvente la SVM para que se reconozca el nuevo agregado.

Sin embargo, Trident no admite automáticamente el traslado de volúmenes entre backends. Esto incluye transferencias entre SVM en el mismo clúster, entre clústeres o a una plataforma de almacenamiento diferente (incluso si ese sistema de almacenamiento está conectado a Trident).

Si se copia un volumen a otra ubicación, se puede utilizar la función de importación de volúmenes para importar los volúmenes actuales a Trident.

Ampliar volúmenes

Trident admite el cambio de tamaño de los volúmenes persistentes (PV) NFS, iSCSI y FC. Esto permite a los usuarios cambiar el tamaño de sus volúmenes directamente a través de la capa de Kubernetes. La expansión de volumen es posible para todas las principales plataformas de almacenamiento de NetApp , incluidos los backends de ONTAP, SolidFire/ NetApp HCI y Cloud Volumes Service . Para permitir una posible expansión posterior, establezca `allowVolumeExpansion` a `true` en su StorageClass asociada con el volumen. Siempre que sea necesario cambiar el tamaño del volumen persistente, edite el `spec.resources.requests.storage` anotación en la reclamación de volumen persistente al tamaño de volumen requerido. Trident se encargará automáticamente de redimensionar el volumen en el clúster de almacenamiento.

Importar un volumen existente a Kubernetes

La importación de volúmenes ofrece la posibilidad de importar un volumen de almacenamiento existente a un entorno Kubernetes. Actualmente, esto cuenta con el apoyo de `ontap-nas` , `ontap-nas-flexgroup` , `solidfire-san` , `azure-netapp-files` , y `gcp-cvs` conductores. Esta función resulta útil al migrar una aplicación existente a Kubernetes o durante escenarios de recuperación ante desastres.

Al utilizar ONTAP y `solidfire-san` Conductores, utilicen el comando `tridentctl import volume <backend-name> <volume-name> -f /path/pvc.yaml` Importar un volumen existente a Kubernetes para que sea administrado por Trident. El archivo PVC YAML o JSON utilizado en el comando de importación de volumen apunta a una clase de almacenamiento que identifica a Trident como el proveedor. Cuando utilice un backend NetApp HCI/ SolidFire , asegúrese de que los nombres de los volúmenes sean únicos. Si los nombres de los volúmenes están duplicados, clone el volumen con un nombre único para que la función de importación de volúmenes pueda distinguirlos.

Si el `azure-netapp-files` o `gcp-cvs` Se utiliza el controlador, utilice el comando `tridentctl import volume <backend-name> <volume path> -f /path/pvc.yaml` para importar el volumen a Kubernetes para que sea administrado por Trident. Esto garantiza una referencia de volumen única.

Cuando se ejecute el comando anterior, Trident encontrará el volumen en el servidor y leerá su tamaño. Añadirá automáticamente (y sobrescribirá si es necesario) el tamaño de volumen configurado del PVC. Luego, Trident crea el nuevo PV y Kubernetes vincula el PVC al PV.

Si un contenedor se desplegara de forma que requiriera el PVC importado específico, permanecería en estado

pendiente hasta que el par PVC/PV se vinculara mediante el proceso de importación de volumen. Una vez unidos los pares PVC/PV, el contenedor debería subir, siempre que no haya otros problemas.

Servicio de registro

La implementación y la gestión del almacenamiento para el registro se han documentado en"netapp.io" en el"blog" .

Servicio de registro

Al igual que otros servicios de OpenShift, el servicio de registro se implementa utilizando Ansible con parámetros de configuración proporcionados por el archivo de inventario, también conocido como hosts, que se proporciona al playbook. Se abordarán dos métodos de instalación: la implementación del registro durante la instalación inicial de OpenShift y la implementación del registro después de que se haya instalado OpenShift.

A partir de la versión 3.9 de Red Hat OpenShift, la documentación oficial desaconseja el uso de NFS para el servicio de registro debido a la preocupación por la corrupción de datos. Esto se basa en las pruebas realizadas por Red Hat a sus productos. El servidor NFS de ONTAP no presenta estos problemas y puede respaldar fácilmente una implementación de registro. En última instancia, la elección del protocolo para el servicio de registro depende de usted; tenga en cuenta que ambos funcionarán perfectamente al usar plataformas NetApp y no hay razón para evitar NFS si esa es su preferencia.

Si elige usar NFS con el servicio de registro, deberá configurar la variable de Ansible. `openshift_enable_unsupported_configurations` a `true` para evitar que falle el instalador.

Empezar

El servicio de registro puede, opcionalmente, implementarse tanto para las aplicaciones como para las operaciones centrales del propio clúster de OpenShift. Si elige implementar el registro de operaciones, especifique la variable `openshift_logging_use_ops` como `true` Se crearán dos instancias del servicio. Las variables que controlan la instancia de registro para operaciones contienen "ops" en ellas, mientras que la instancia para aplicaciones no.

Configurar las variables de Ansible según el método de despliegue es importante para garantizar que los servicios subyacentes utilicen el almacenamiento correcto. Analicemos las opciones para cada uno de los métodos de implementación.

Las tablas que aparecen a continuación contienen únicamente las variables relevantes para la configuración del almacenamiento en lo que respecta al servicio de registro. Puedes encontrar otras opciones en"Documentación de registro de Red Hat OpenShift" que deberá revisarse, configurarse y utilizarse de acuerdo con su implementación.

Las variables de la tabla siguiente harán que el playbook de Ansible cree un PV y un PVC para el servicio de registro utilizando los detalles proporcionados. Este método es significativamente menos flexible que usar el playbook de instalación de componentes después de la instalación de OpenShift; sin embargo, si dispone de volúmenes existentes, es una opción.

Variable	Detalles
`openshift_logging_storage_kind`	Empezar a `nfs` para que el instalador cree un PV NFS para el servicio de registro.

Variable	Detalles
`openshift_logging_storage_host`	El nombre de host o la dirección IP del host NFS. Esto debe configurarse con el dataLIF de su máquina virtual.
`openshift_logging_storage_nfs_directory`	La ruta de montaje para la exportación NFS. Por ejemplo, si el volumen está unido como `/openshift_logging` , usarías esa ruta para esta variable.
`openshift_logging_storage_volume_name`	El nombre, por ejemplo `pv_ose_logs` , del PV para crear.
`openshift_logging_storage_volume_size`	El tamaño de la exportación NFS, por ejemplo `100Gi` .

Si su clúster de OpenShift ya está en funcionamiento y, por lo tanto, Trident se ha implementado y configurado, el instalador puede utilizar el aprovisionamiento dinámico para crear los volúmenes. Será necesario configurar las siguientes variables.

Variable	Detalles
`openshift_logging_es_pvc_dynamic`	Establézcalo en verdadero para usar volúmenes aprovisionados dinámicamente.
`openshift_logging_es_pvc_storage_class_name`	El nombre de la clase de almacenamiento que se utilizará en el PVC.
`openshift_logging_es_pvc_size`	El tamaño del volumen solicitado en el PVC.
`openshift_logging_es_pvc_prefix`	Un prefijo para los PVC utilizados por el servicio de registro.
`openshift_logging_es_ops_pvc_dynamic`	Empezar a `true` utilizar volúmenes aprovisionados dinámicamente para la instancia de registro de operaciones.
`openshift_logging_es_ops_pvc_storage_class_name`	El nombre de la clase de almacenamiento para la instancia de registro de operaciones.
`openshift_logging_es_ops_pvc_size`	El tamaño de la solicitud de volumen para la instancia de operaciones.
`openshift_logging_es_ops_pvc_prefix`	Un prefijo para los PVC de la instancia ops.

Implementar la pila de registro

Si va a implementar el registro como parte del proceso de instalación inicial de OpenShift, solo necesita seguir el proceso de implementación estándar. Ansible configurará e implementará los servicios y objetos de OpenShift necesarios para que el servicio esté disponible tan pronto como Ansible finalice.

Sin embargo, si realiza la implementación después de la instalación inicial, Ansible deberá utilizar el playbook del componente. Este proceso puede variar ligeramente con diferentes versiones de OpenShift, así que asegúrese de leer y seguir las instrucciones."Documentación de Red Hat OpenShift Container Platform 3.11" para tu versión.

Servicio de métricas

El servicio de métricas proporciona información valiosa al administrador sobre el estado, la utilización de recursos y la disponibilidad del clúster de OpenShift. También es necesario para la funcionalidad de escalado automático de pods y muchas organizaciones utilizan datos del servicio de métricas para sus aplicaciones de facturación interna y/o de visualización de datos.

Al igual que con el servicio de registro y con OpenShift en su conjunto, Ansible se utiliza para implementar el servicio de métricas. Asimismo, al igual que el servicio de registro, el servicio de métricas se puede implementar durante la configuración inicial del clúster o después de que esté operativo utilizando el método de instalación de componentes. Las siguientes tablas contienen las variables importantes a la hora de configurar el almacenamiento persistente para el servicio de métricas.

Las tablas que aparecen a continuación solo contienen las variables relevantes para la configuración del almacenamiento en lo que respecta al servicio de métricas. En la documentación se incluyen muchas otras opciones que deben revisarse, configurarse y utilizarse de acuerdo con su implementación.

Variable	Detalles
`openshift_metrics_storage_kind`	Empezar a `nfs` para que el instalador cree un PV NFS para el servicio de registro.
`openshift_metrics_storage_host`	El nombre de host o la dirección IP del host NFS. Esto debe configurarse con el dataLIF de su SVM.
`openshift_metrics_storage_nfs_directory`	La ruta de montaje para la exportación NFS. Por ejemplo, si el volumen está unido como `/openshift_metrics` , usarías esa ruta para esta variable.
`openshift_metrics_storage_volume_name`	El nombre, por ejemplo `pv_ose_metrics` , del PV para crear.
`openshift_metrics_storage_volume_size`	El tamaño de la exportación NFS, por ejemplo `100Gi` .

Si su clúster de OpenShift ya está en funcionamiento y, por lo tanto, Trident se ha implementado y configurado, el instalador puede utilizar el aprovisionamiento dinámico para crear los volúmenes. Será necesario configurar las siguientes variables.

Variable	Detalles
`openshift_metrics_cassandra_pvc_prefix`	Un prefijo para usar en los PVC métricos.
`openshift_metrics_cassandra_pvc_size`	El tamaño de los volúmenes a solicitar.
`openshift_metrics_cassandra_storage_type`	El tipo de almacenamiento que se utilizará para las métricas debe configurarse como dinámico para que Ansible cree PVC con la clase de almacenamiento adecuada.
`openshift_metrics_cassanda_pvc_storage_class_name`	El nombre de la clase de almacenamiento que se va a utilizar.

Implementar el servicio de métricas

Con las variables de Ansible apropiadas definidas en su archivo hosts/inventory, implemente el servicio utilizando Ansible. Si realiza la implementación durante la instalación de OpenShift, el PV se creará y utilizará automáticamente. Si realiza la implementación utilizando los playbooks de componentes, después de la instalación de OpenShift, Ansible crea los PVC necesarios y, después de que Trident haya aprovisionado el almacenamiento para ellos, implementa el servicio.

Las variables anteriores, así como el proceso de implementación, pueden cambiar con cada versión de OpenShift. Asegúrese de revisar y seguir"Guía de implementación de OpenShift de Red Hat" para su versión, de modo que esté configurada para su entorno.

Protección de datos y recuperación ante desastres

Obtenga información sobre las opciones de protección y recuperación para Trident y los volúmenes creados con Trident. Debe contar con una estrategia de protección y recuperación de datos para cada aplicación con requisitos de persistencia.

replicación y recuperación de Trident

Puedes crear una copia de seguridad para restaurar Trident en caso de desastre.

replicación de Trident

Trident utiliza CRD de Kubernetes para almacenar y gestionar su propio estado y el etcd del clúster de Kubernetes para almacenar sus metadatos.

Pasos

1. Realice una copia de seguridad del clúster de Kubernetes etcd utilizando"Kubernetes: Copia de seguridad de un clúster etcd" .
2. Coloque los artefactos de copia de seguridad en un FlexVol volume.

> NetApp recomienda proteger la SVM donde reside el FlexVol con una relación SnapMirror a otra SVM.

Recuperación de Trident

Utilizando CRD de Kubernetes y la instantánea etcd del clúster de Kubernetes, puede recuperar Trident.

Pasos

1. Desde la SVM de destino, monte el volumen que contiene los archivos de datos y certificados etcd de Kubernetes en el host que se configurará como nodo maestro.
2. Copie todos los certificados necesarios relacionados con el clúster de Kubernetes en `/etc/kubernetes/pki` y los archivos miembros de etcd bajo `/var/lib/etcd` .
3. Restaure el clúster de Kubernetes desde la copia de seguridad de etcd usando"Kubernetes: Restauración de un clúster etcd" .
4. Correr `kubectl get crd` para verificar que todos los recursos personalizados de Trident se hayan cargado y recuperar los objetos de Trident para verificar que todos los datos estén disponibles.

replicación y recuperación de SVM

Trident no puede configurar relaciones de replicación; sin embargo, el administrador de almacenamiento puede usar "ONTAP SnapMirror" para replicar una SVM.

En caso de desastre, puede activar el SVM de destino SnapMirror para comenzar a servir datos. Podrás volver a la configuración principal cuando se restablezcan los sistemas.

Acerca de esta tarea

Tenga en cuenta lo siguiente al utilizar la función de replicación SVM de SnapMirror :

- Debes crear un backend distinto para cada SVM con SVM-DR habilitado.
- Configure las clases de almacenamiento para seleccionar los backends replicados solo cuando sea necesario para evitar que se aprovisionen volúmenes que no necesitan replicación en los backends que admiten SVM-DR.
- Los administradores de aplicaciones deben comprender el coste y la complejidad adicionales asociados a la replicación y considerar cuidadosamente su plan de recuperación antes de iniciar este proceso.

replicación de SVM

Puedes utilizar"ONTAP: Replicación SVM de SnapMirror" para crear la relación de replicación SVM.

SnapMirror te permite configurar opciones para controlar qué replicar. Necesitarás saber qué opciones seleccionaste al realizar la operación.Recuperación de SVM usando Trident .

- "-identity-preserve true"Replica toda la configuración SVM.
- "-descartar-configuraciones de red"Excluye las LIF y la configuración de red relacionada.
- "-identity-preserve falso"Solo replica los volúmenes y la configuración de seguridad.

Recuperación de SVM usando Trident

Trident no detecta automáticamente los fallos de SVM. En caso de desastre, el administrador puede iniciar manualmente la conmutación por error de Trident al nuevo SVM.

Pasos

1. Cancele las transferencias SnapMirror programadas y en curso, rompa la relación de replicación, detenga la SVM de origen y, a continuación, active la SVM de destino SnapMirror .
2. Si usted especificó `-identity-preserve false` o `-discard-config network` Al configurar la replicación de SVM, actualice `managementLIF` y `dataLIF` en el archivo de definición del backend de Trident .
3. Confirmar `storagePrefix` está presente en el archivo de definición del backend de Trident . Este parámetro no se puede cambiar. Omitiendo `storagePrefix` provocará un fallo en la actualización del backend.
4. Actualice todos los backends necesarios para reflejar el nuevo nombre de SVM de destino utilizando:

   ```
   ./tridentctl update backend <backend-name> -f <backend-json-file> -n
   <namespace>
   ```

5. Si usted especificó `-identity-preserve false` o `discard-config network` , debes reiniciar todos

los pods de la aplicación.

> Si usted especificó `-identity-preserve true` Todos los volúmenes aprovisionados por Trident comienzan a proporcionar datos cuando se activa la SVM de destino.

replicación y recuperación de volumen

Trident no puede configurar las relaciones de replicación de SnapMirror ; sin embargo, el administrador de almacenamiento puede usar"replicación y recuperación de ONTAP SnapMirror" para replicar volúmenes creados por Trident.

Luego puede importar los volúmenes recuperados a Trident mediante"importación de volumen de tridentctl" .

> La importación no es compatible con `ontap-nas-economy` , `ontap-san-economy` , o `ontap-flexgroup-economy` conductores.

Protección de datos de instantáneas

Puedes proteger y restaurar datos usando:

- Un controlador de instantáneas externo y CRD para crear instantáneas de volúmenes de Kubernetes de volúmenes persistentes (PV).

  "Instantáneas de volumen"

- Instantáneas de ONTAP para restaurar todo el contenido de un volumen o para recuperar archivos o LUN individuales.

  "Instantáneas de ONTAP"

Seguridad

Seguridad

Siga las recomendaciones que se enumeran aquí para garantizar la seguridad de su instalación de Trident .

Ejecutar Trident en su propio espacio de nombres.

Es importante impedir que las aplicaciones, los administradores de aplicaciones, los usuarios y las aplicaciones de gestión accedan a las definiciones de objetos de Trident o a los pods para garantizar un almacenamiento fiable y bloquear posibles actividades maliciosas.

Para separar las demás aplicaciones y usuarios de Trident, instale siempre Trident en su propio espacio de nombres de Kubernetes.(`trident`). Al colocar Trident en su propio espacio de nombres, se garantiza que solo el personal administrativo de Kubernetes tenga acceso al pod de Trident y a los artefactos (como los secretos de backend y CHAP, si corresponde) almacenados en los objetos CRD con espacio de nombres. Debe asegurarse de permitir únicamente a los administradores el acceso al espacio de nombres de Trident y, por lo tanto, el acceso a `tridentctl` solicitud.

Utilice la autenticación CHAP con los backends SAN de ONTAP.

Trident admite la autenticación basada en CHAP para cargas de trabajo SAN de ONTAP (utilizando el `ontap-san` y `ontap-san-economy` conductores). NetApp recomienda utilizar CHAP bidireccional con Trident para la autenticación entre un host y el backend de almacenamiento.

Para los backends ONTAP que utilizan los controladores de almacenamiento SAN, Trident puede configurar CHAP bidireccional y administrar los nombres de usuario y secretos de CHAP a través de `tridentctl` . Referirse a"Prepárese para configurar el backend con los controladores SAN de ONTAP." Para comprender cómo Trident configura CHAP en los sistemas backend de ONTAP .

Utilice la autenticación CHAP con los sistemas backend NetApp HCI y SolidFire.

NetApp recomienda implementar CHAP bidireccional para garantizar la autenticación entre un host y los backends de NetApp HCI y SolidFire . Trident utiliza un objeto secreto que incluye dos contraseñas CHAP por inquilino. Cuando se instala Trident , gestiona los secretos CHAP y los almacena en un `tridentvolume` Objeto CR para el PV correspondiente. Cuando se crea un PV, Trident utiliza los secretos CHAP para iniciar una sesión iSCSI y comunicarse con el sistema NetApp HCI y SolidFire a través de CHAP.

Los volúmenes creados por Trident no están asociados a ningún grupo de acceso a volúmenes.

Utilice Trident con NVE y NAE

NetApp ONTAP proporciona cifrado de datos en reposo para proteger los datos confidenciales en caso de que un disco sea robado, devuelto o reutilizado. Para más detalles, consulte"Descripción general de la configuración de NetApp Volume Encryption" .

- Si NAE está habilitado en el backend, cualquier volumen aprovisionado en Trident estará habilitado para NAE.
  - Puedes configurar la bandera de cifrado NVE a `""` para crear volúmenes compatibles con NAE.
- Si NAE no está habilitado en el backend, cualquier volumen aprovisionado en Trident tendrá NVE habilitado a menos que la bandera de cifrado NVE esté configurada en `false` (el valor predeterminado) en la configuración del backend.

Los volúmenes creados en Trident en un backend habilitado para NAE deben estar cifrados con NVE o NAE.

- Puedes configurar la bandera de cifrado NVE a `true` en la configuración del backend de Trident para anular el cifrado NAE y usar una clave de cifrado específica por volumen.
- Configurar la bandera de cifrado NVE a `false` En un backend habilitado para NAE se crea un volumen habilitado para NAE. No se puede deshabilitar el cifrado NAE configurando la bandera de cifrado NVE en `false` .

- Puede crear manualmente un volumen NVE en Trident configurando explícitamente la bandera de cifrado NVE a `true` .

Para obtener más información sobre las opciones de configuración del backend, consulte:

- "Opciones de configuración SAN de ONTAP"
- "Opciones de configuración de ONTAP NAS"

Configuración de clave unificada de Linux (LUKS)

Puede habilitar Linux Unified Key Setup (LUKS) para cifrar los volúmenes ONTAP SAN y ONTAP SAN ECONOMY en Trident. Trident admite la rotación de contraseñas y la expansión de volumen para volúmenes cifrados con LUKS.

En Trident, los volúmenes cifrados con LUKS utilizan el cifrado y el modo aes-xts-plain64, tal como recomienda"NIST" .

> El cifrado LUKS no es compatible con los sistemas ASA r2. Para obtener información sobre los sistemas ASA r2, consulte"Conozca los sistemas de almacenamiento ASA r2" .

Antes de empezar

- Los nodos de trabajo deben tener instalado cryptsetup 2.1 o superior (pero inferior a 3.0). Para obtener más información, visite"Gitlab: configuración de criptografía" .
- Por motivos de rendimiento, NetApp recomienda que los nodos de trabajo sean compatibles con las nuevas instrucciones del estándar de cifrado avanzado (AES-NI). Para verificar la compatibilidad con AES-NI, ejecute el siguiente comando:

  ```
  grep "aes" /proc/cpuinfo
  ```

  Si no se devuelve nada, su procesador no admite AES-NI. Para obtener más información sobre AES-NI, visite:"Intel: Instrucciones del estándar de cifrado avanzado (AES-NI)" .

Habilitar el cifrado LUKS

Puede habilitar el cifrado por volumen en el host mediante Linux Unified Key Setup (LUKS) para volúmenes ONTAP SAN y ONTAP SAN ECONOMY.

Pasos

1. Define los atributos de cifrado LUKS en la configuración del backend. Para obtener más información sobre las opciones de configuración de backend para ONTAP SAN, consulte"Opciones de configuración SAN de ONTAP" .

```
{
  "storage": [
    {
      "labels": {
        "luks": "true"
      },
      "zone": "us_east_1a",
      "defaults": {
        "luksEncryption": "true"
      }
    },
    {
      "labels": {
        "luks": "false"
      },
      "zone": "us_east_1a",
      "defaults": {
        "luksEncryption": "false"
      }
    }
  ]
}
```

2. Usar `parameters.selector` para definir los grupos de almacenamiento utilizando el cifrado LUKS. Por ejemplo:

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: luks
provisioner: csi.trident.netapp.io
parameters:
  selector: "luks=true"
  csi.storage.k8s.io/node-stage-secret-name: luks-${pvc.name}
  csi.storage.k8s.io/node-stage-secret-namespace: ${pvc.namespace}
```

3. Crea un secreto que contenga la contraseña LUKS. Por ejemplo:

```
kubectl -n trident create -f luks-pvc1.yaml
apiVersion: v1
kind: Secret
metadata:
  name: luks-pvc1
stringData:
  luks-passphrase-name: A
  luks-passphrase: secretA
```

Limitaciones

Los volúmenes cifrados con LUKS no pueden aprovechar la deduplicación y compresión de ONTAP .

Configuración del backend para importar volúmenes LUKS

Para importar un volumen LUKS, debe configurar `luksEncryption` a(`true` en el backend. El `luksEncryption` Esta opción le indica a Trident si el volumen es compatible con LUKS.(`true`) o no cumple con LUKS(`false`) como se muestra en el siguiente ejemplo.

```
version: 1
storageDriverName: ontap-san
managementLIF: 10.0.0.1
dataLIF: 10.0.0.2
svm: trident_svm
username: admin
password: password
defaults:
  luksEncryption: 'true'
  spaceAllocation: 'false'
  snapshotPolicy: default
  snapshotReserve: '10'
```

Configuración de PVC para importar volúmenes LUKS

Para importar volúmenes LUKS de forma dinámica, configure la anotación `trident.netapp.io/luksEncryption` a `true` e incluir una clase de almacenamiento habilitada para LUKS en el PVC como se muestra en este ejemplo.

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: luks-pvc
  namespace: trident
  annotations:
    trident.netapp.io/luksEncryption: "true"
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: luks-sc
```

Rotar una frase de contraseña LUKS

Puedes cambiar la contraseña LUKS y confirmar el cambio.

> No olvide una frase de contraseña hasta que haya verificado que ya no está siendo referenciada por ningún volumen, instantánea o secreto. Si se pierde la contraseña a la que se hace referencia, es posible que no pueda montar el volumen y los datos permanecerán cifrados e inaccesibles.

Acerca de esta tarea

La rotación de la contraseña LUKS se produce cuando se crea un pod que monta el volumen después de que se haya especificado una nueva contraseña LUKS. Cuando se crea un nuevo pod, Trident compara la contraseña LUKS del volumen con la contraseña activa del secreto.

- Si la contraseña del volumen no coincide con la contraseña activa del secreto, se produce una rotación.
- Si la contraseña del volumen coincide con la contraseña activa del secreto, `previous-luks-passphrase` El parámetro se ignora.

Pasos

1. Añade el `node-publish-secret-name` y `node-publish-secret-namespace` Parámetros de StorageClass. Por ejemplo:

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: csi-san
provisioner: csi.trident.netapp.io
parameters:
  trident.netapp.io/backendType: "ontap-san"
  csi.storage.k8s.io/node-stage-secret-name: luks
  csi.storage.k8s.io/node-stage-secret-namespace: ${pvc.namespace}
  csi.storage.k8s.io/node-publish-secret-name: luks
  csi.storage.k8s.io/node-publish-secret-namespace: ${pvc.namespace}
```

2. Identificar las contraseñas existentes en el volumen o instantánea.

Volumen

```
tridentctl -d get volume luks-pvc1
GET http://127.0.0.1:8000/trident/v1/volume/<volumeID>

...luksPassphraseNames:["A"]
```

Snapshot

```
tridentctl -d get snapshot luks-pvc1
GET http://127.0.0.1:8000/trident/v1/volume/<volumeID>/<snapshotID>

...luksPassphraseNames:["A"]
```

3. Actualice la clave secreta LUKS del volumen para especificar las contraseñas nueva y anterior. Asegurar `previous-luke-passphrase-name` y `previous-luks-passphrase` Coincide con la contraseña anterior.

```
apiVersion: v1
kind: Secret
metadata:
  name: luks-pvc1
stringData:
  luks-passphrase-name: B
  luks-passphrase: secretB
  previous-luks-passphrase-name: A
  previous-luks-passphrase: secretA
```

4. Crea un nuevo pod para montar el volumen. Esto es necesario para iniciar la rotación.
5. Verifique que la contraseña se haya rotado.

Volumen

```
tridentctl -d get volume luks-pvc1
GET http://127.0.0.1:8000/trident/v1/volume/<volumeID>

...luksPassphraseNames:["B"]
```

Snapshot

```
tridentctl -d get snapshot luks-pvc1
GET http://127.0.0.1:8000/trident/v1/volume/<volumeID>/<snapshotID>

...luksPassphraseNames:["B"]
```

Resultados

La contraseña se rotó cuando solo se devolvió la nueva contraseña en el volumen y la instantánea.

> Si se devuelven dos frases de contraseña, por ejemplo `luksPassphraseNames: ["B", "A"]` , la rotación está incompleta. Puedes activar una nueva cápsula para intentar completar la rotación.

Habilitar la expansión de volumen

Puede habilitar la expansión de volumen en un volumen cifrado con LUKS.

Pasos

1. Habilitar el `CSINodeExpandSecret` Puerta de características (beta 1.25+). Referirse a "Kubernetes 1.25: Uso de secretos para la expansión de volúmenes CSI basada en nodos" Para más detalles.
2. Añade el `node-expand-secret-name` y `node-expand-secret-namespace` Parámetros de StorageClass. Por ejemplo:

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: luks
provisioner: csi.trident.netapp.io
parameters:
  selector: "luks=true"
  csi.storage.k8s.io/node-stage-secret-name: luks-${pvc.name}
  csi.storage.k8s.io/node-stage-secret-namespace: ${pvc.namespace}
  csi.storage.k8s.io/node-expand-secret-name: luks-${pvc.name}
  csi.storage.k8s.io/node-expand-secret-namespace: ${pvc.namespace}
allowVolumeExpansion: true
```

Resultados

Cuando se inicia la expansión del almacenamiento en línea, el kubelet pasa las credenciales apropiadas al controlador.

Cifrado en vuelo Kerberos

Mediante el cifrado en tránsito de Kerberos, puede mejorar la seguridad del acceso a los datos habilitando el cifrado para el tráfico entre su clúster administrado y el backend de almacenamiento.

Trident admite el cifrado Kerberos para ONTAP como backend de almacenamiento:

- * ONTAP local* - Trident admite el cifrado Kerberos a través de conexiones NFSv3 y NFSv4 desde Red Hat OpenShift y clústeres Kubernetes ascendentes a volúmenes ONTAP locales.

Puede crear, eliminar, cambiar el tamaño, crear instantáneas, clonar, clonar en modo de solo lectura e importar volúmenes que utilicen cifrado NFS.

Configurar el cifrado Kerberos en vuelo con volúmenes ONTAP locales

Puede habilitar el cifrado Kerberos en el tráfico de almacenamiento entre su clúster administrado y un backend de almacenamiento ONTAP local.

> El cifrado Kerberos para el tráfico NFS con backends de almacenamiento ONTAP locales solo es compatible mediante el uso de `ontap-nas` Controlador de almacenamiento.

Antes de empezar

- Asegúrese de tener acceso a `tridentctl` utilidad.
- Asegúrese de tener acceso de administrador al backend de almacenamiento de ONTAP .
- Asegúrese de conocer el nombre del volumen o volúmenes que compartirá desde el backend de almacenamiento de ONTAP .
- Asegúrese de haber preparado la máquina virtual de almacenamiento ONTAP para que admita el cifrado Kerberos para volúmenes NFS. Referirse a "Habilitar Kerberos en un dataLIF" para obtener instrucciones.
- Asegúrese de que todos los volúmenes NFSv4 que utilice con cifrado Kerberos estén configurados correctamente. Consulte la sección Configuración del dominio NFSv4 de NetApp (página 13) del "Guía de mejoras y prácticas recomendadas de NetApp NFSv4" .

Agregar o modificar las políticas de exportación de ONTAP

Debe agregar reglas a las políticas de exportación de ONTAP existentes o crear nuevas políticas de exportación que admitan el cifrado Kerberos para el volumen raíz de la máquina virtual de almacenamiento de ONTAP , así como para cualquier volumen de ONTAP compartido con el clúster de Kubernetes ascendente. Las reglas de política de exportación que agregue, o las nuevas políticas de exportación que cree, deben admitir los siguientes protocolos de acceso y permisos de acceso:

Protocolos de acceso

Configure la política de exportación con los protocolos de acceso NFS, NFSv3 y NFSv4.

Detalles de acceso

Puede configurar una de las tres versiones diferentes de cifrado Kerberos, según sus necesidades de volumen:

- **Kerberos 5** - (autenticación y cifrado)
- **Kerberos 5i** - (autenticación y cifrado con protección de identidad)
- **Kerberos 5p** - (autenticación y cifrado con protección de identidad y privacidad)

Configure la regla de política de exportación de ONTAP con los permisos de acceso adecuados. Por ejemplo, si los clústeres van a montar los volúmenes NFS con una combinación de cifrado Kerberos 5i y Kerberos 5p, utilice la siguiente configuración de acceso:

Tipo	Acceso de solo lectura	Acceso de lectura/escritura	acceso de superusuario
UNIX	Habilitado	Habilitado	Habilitado
Kerberos 5i	Habilitado	Habilitado	Habilitado
Kerberos 5p	Habilitado	Habilitado	Habilitado

Consulte la siguiente documentación para obtener información sobre cómo crear políticas de exportación de ONTAP y reglas de políticas de exportación:

- "Crear una política de exportación"
- "Agregar una regla a una política de exportación"

Crea un backend de almacenamiento

Puede crear una configuración de backend de almacenamiento Trident que incluya capacidad de cifrado Kerberos.

Acerca de esta tarea

Cuando crea un archivo de configuración de backend de almacenamiento que configura el cifrado Kerberos, puede especificar una de las tres versiones diferentes de cifrado Kerberos mediante el `spec.nfsMountOptions` parámetro:

- `spec.nfsMountOptions: sec=krb5`(autenticación y cifrado)
- `spec.nfsMountOptions: sec=krb5i`(autenticación y cifrado con protección de identidad)
- `spec.nfsMountOptions: sec=krb5p`(autenticación y cifrado con protección de identidad y privacidad)

Especifique solo un nivel de Kerberos. Si se especifica más de un nivel de cifrado Kerberos en la lista de parámetros, solo se utilizará la primera opción.

Pasos

1. En el clúster administrado, cree un archivo de configuración de backend de almacenamiento utilizando el siguiente ejemplo. Reemplace los valores entre corchetes <> con información de su entorno:

```
apiVersion: v1
kind: Secret
metadata:
  name: backend-ontap-nas-secret
type: Opaque
stringData:
  clientID: <CLIENT_ID>
  clientSecret: <CLIENT_SECRET>
---
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-ontap-nas
spec:
  version: 1
  storageDriverName: "ontap-nas"
  managementLIF: <STORAGE_VM_MGMT_LIF_IP_ADDRESS>
  dataLIF: <PROTOCOL_LIF_FQDN_OR_IP_ADDRESS>
  svm: <STORAGE_VM_NAME>
  username: <STORAGE_VM_USERNAME_CREDENTIAL>
  password: <STORAGE_VM_PASSWORD_CREDENTIAL>
  nasType: nfs
  nfsMountOptions: ["sec=krb5i"] #can be krb5, krb5i, or krb5p
  qtreesPerFlexvol:
  credentials:
    name: backend-ontap-nas-secret
```

2. Utilice el archivo de configuración que creó en el paso anterior para crear el backend:

```
tridentctl create backend -f <backend-configuration-file>
```

Si falla la creación del backend, algo falla en la configuración del backend. Puedes consultar los registros para determinar la causa ejecutando el siguiente comando:

```
tridentctl logs
```

Después de identificar y corregir el problema con el archivo de configuración, puede volver a ejecutar el comando de creación.

Crear una clase de almacenamiento

Puede crear una clase de almacenamiento para aprovisionar volúmenes con cifrado Kerberos.

Acerca de esta tarea

Al crear un objeto de clase de almacenamiento, puede especificar una de las tres versiones diferentes de cifrado Kerberos mediante el `mountOptions` parámetro:

- `mountOptions: sec=krb5`(autenticación y cifrado)
- `mountOptions: sec=krb5i`(autenticación y cifrado con protección de identidad)
- `mountOptions: sec=krb5p`(autenticación y cifrado con protección de identidad y privacidad)

Especifique solo un nivel de Kerberos. Si se especifica más de un nivel de cifrado Kerberos en la lista de parámetros, solo se utilizará la primera opción. Si el nivel de cifrado que especificó en la configuración del backend de almacenamiento es diferente del nivel que especificó en el objeto de clase de almacenamiento, el objeto de clase de almacenamiento tiene prioridad.

Pasos

1. Crea un objeto StorageClass de Kubernetes, utilizando el siguiente ejemplo:

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontap-nas-sc
provisioner: csi.trident.netapp.io
mountOptions:
  - sec=krb5i #can be krb5, krb5i, or krb5p
parameters:
  backendType: ontap-nas
  storagePools: ontapnas_pool
  trident.netapp.io/nasType: nfs
allowVolumeExpansion: true
```

2. Crea la clase de almacenamiento:

```
kubectl create -f sample-input/storage-class-ontap-nas-sc.yaml
```

3. Asegúrese de que se haya creado la clase de almacenamiento:

```
kubectl get sc ontap-nas-sc
```

   Debería ver un resultado similar al siguiente:

```
NAME          PROVISIONER             AGE
ontap-nas-sc    csi.trident.netapp.io   15h
```

volúmenes de provisiones

Después de crear un backend de almacenamiento y una clase de almacenamiento, ahora puede aprovisionar un volumen. Para obtener instrucciones, consulte "Provisión de un volumen" .

Configurar el cifrado Kerberos en vuelo con volúmenes de Azure NetApp Files

Puede habilitar el cifrado Kerberos en el tráfico de almacenamiento entre su clúster administrado y un único backend de almacenamiento de Azure NetApp Files o un grupo virtual de backends de almacenamiento de Azure NetApp Files .

Antes de empezar

- Asegúrese de haber habilitado Trident en el clúster Red Hat OpenShift administrado.
- Asegúrese de tener acceso a `tridentctl` utilidad.
- Asegúrese de haber preparado el backend de almacenamiento de Azure NetApp Files para el cifrado Kerberos teniendo en cuenta los requisitos y siguiendo las instrucciones en "Documentación de Azure NetApp Files" .
- Asegúrese de que todos los volúmenes NFSv4 que utilice con cifrado Kerberos estén configurados correctamente. Consulte la sección Configuración del dominio NFSv4 de NetApp (página 13) del "Guía de mejoras y prácticas recomendadas de NetApp NFSv4" .

Crea un backend de almacenamiento

Puede crear una configuración de backend de almacenamiento de Azure NetApp Files que incluya la capacidad de cifrado Kerberos.

Acerca de esta tarea

Cuando crea un archivo de configuración de backend de almacenamiento que configura el cifrado Kerberos, puede definirlo para que se aplique en uno de dos niveles posibles:

- El **nivel de backend de almacenamiento** que utiliza el `spec.kerberos` campo
- El **nivel de piscina virtual** que utiliza el `spec.storage.kerberos` campo

Cuando se define la configuración a nivel de grupo virtual, el grupo se selecciona utilizando la etiqueta en la clase de almacenamiento.

En cualquiera de los dos niveles, puede especificar una de las tres versiones diferentes de cifrado Kerberos:

- `kerberos: sec=krb5`(autenticación y cifrado)
- `kerberos: sec=krb5i`(autenticación y cifrado con protección de identidad)
- `kerberos: sec=krb5p`(autenticación y cifrado con protección de identidad y privacidad)

Pasos

1. En el clúster administrado, cree un archivo de configuración de backend de almacenamiento utilizando uno de los siguientes ejemplos, según dónde necesite definir el backend de almacenamiento (nivel de backend de almacenamiento o nivel de grupo virtual). Reemplace los valores entre corchetes <> con información de su entorno:

Ejemplo de nivel de backend de almacenamiento

```
apiVersion: v1
kind: Secret
metadata:
  name: backend-tbc-secret
type: Opaque
stringData:
  clientID: <CLIENT_ID>
  clientSecret: <CLIENT_SECRET>

---
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc
spec:
  version: 1
  storageDriverName: azure-netapp-files
  subscriptionID: <SUBSCRIPTION_ID>
  tenantID: <TENANT_ID>
  location: <AZURE_REGION_LOCATION>
  serviceLevel: Standard
  networkFeatures: Standard
  capacityPools: <CAPACITY_POOL>
  resourceGroups: <RESOURCE_GROUP>
  netappAccounts: <NETAPP_ACCOUNT>
  virtualNetwork: <VIRTUAL_NETWORK>
  subnet: <SUBNET>
  nasType: nfs
  kerberos: sec=krb5i #can be krb5, krb5i, or krb5p
  credentials:
    name: backend-tbc-secret
```

Ejemplo de nivel de piscina virtual

```
---
apiVersion: v1
kind: Secret
metadata:
  name: backend-tbc-secret
type: Opaque
stringData:
  clientID: <CLIENT_ID>
  clientSecret: <CLIENT_SECRET>

---
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc
spec:
  version: 1
  storageDriverName: azure-netapp-files
  subscriptionID: <SUBSCRIPTION_ID>
  tenantID: <TENANT_ID>
  location: <AZURE_REGION_LOCATION>
  serviceLevel: Standard
  networkFeatures: Standard
  capacityPools: <CAPACITY_POOL>
  resourceGroups: <RESOURCE_GROUP>
  netappAccounts: <NETAPP_ACCOUNT>
  virtualNetwork: <VIRTUAL_NETWORK>
  subnet: <SUBNET>
  nasType: nfs
  storage:
    - labels:
        type: encryption
      kerberos: sec=krb5i #can be krb5, krb5i, or krb5p
  credentials:
    name: backend-tbc-secret
```

2. Utilice el archivo de configuración que creó en el paso anterior para crear el backend:

```
tridentctl create backend -f <backend-configuration-file>
```

Si falla la creación del backend, algo falla en la configuración del backend. Puedes consultar los registros para determinar la causa ejecutando el siguiente comando:

```
tridentctl logs
```

Después de identificar y corregir el problema con el archivo de configuración, puede volver a ejecutar el comando de creación.

Crear una clase de almacenamiento

Puede crear una clase de almacenamiento para aprovisionar volúmenes con cifrado Kerberos.

Pasos

1. Crea un objeto StorageClass de Kubernetes, utilizando el siguiente ejemplo:

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: sc-nfs
provisioner: csi.trident.netapp.io
parameters:
  backendType: azure-netapp-files
  trident.netapp.io/nasType: nfs
  selector: type=encryption
```

2. Crea la clase de almacenamiento:

```
kubectl create -f sample-input/storage-class-sc-nfs.yaml
```

3. Asegúrese de que se haya creado la clase de almacenamiento:

```
kubectl get sc -sc-nfs
```

Debería ver un resultado similar al siguiente:

```
NAME          PROVISIONER             AGE
sc-nfs        csi.trident.netapp.io   15h
```

volúmenes de provisiones

Después de crear un backend de almacenamiento y una clase de almacenamiento, ahora puede aprovisionar un volumen. Para obtener instrucciones, consulte "Provisión de un volumen" .

Proteja las aplicaciones con Trident Protect

Obtenga más información sobre Trident Protect

NetApp Trident Protect proporciona capacidades avanzadas de gestión de datos de aplicaciones que mejoran la funcionalidad y la disponibilidad de las aplicaciones Kubernetes con estado respaldadas por los sistemas de almacenamiento NetApp ONTAP y el aprovisionador de almacenamiento NetApp Trident CSI. Trident Protect simplifica la gestión, la protección y el movimiento de cargas de trabajo en contenedores en nubes públicas y entornos locales. También ofrece capacidades de automatización a través de su API y CLI.

Puede proteger aplicaciones con Trident Protect creando recursos personalizados (CR) o utilizando la CLI de Trident Protect.

¿Que sigue?

Puede obtener información sobre los requisitos de Trident Protect antes de instalarlo:

- "Requisitos de Trident Protect"

Instalar Trident Protect

Requisitos de Trident Protect

Comience por verificar que su entorno operativo, clústeres de aplicaciones, aplicaciones y licencias estén listos. Asegúrese de que su entorno cumpla con estos requisitos para implementar y operar Trident Protect.

Compatibilidad del clúster Kubernetes de Trident Protect

Trident Protect es compatible con una amplia gama de ofertas de Kubernetes totalmente administradas y autoadministradas, que incluyen:

- Servicio Amazon Elastic Kubernetes (EKS)
- Motor Google Kubernetes (GKE)
- Servicio de Kubernetes de Microsoft Azure (AKS)
- Red Hat OpenShift
- SUSE Rancher
- Cartera de VMware Tanzu
- Kubernetes ascendente

- Las copias de seguridad de Trident Protect solo se admiten en nodos de cómputo de Linux. Los nodos de cómputo de Windows no son compatibles con operaciones de copia de seguridad.
- Asegúrese de que el clúster en el que instala Trident Protect esté configurado con un controlador de instantáneas en ejecución y los CRD relacionados. Para instalar un controlador de instantáneas, consulte "estas instrucciones" .

Compatibilidad del backend de almacenamiento Trident Protect

Trident Protect admite los siguientes backends de almacenamiento:

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

Asegúrese de que su sistema de almacenamiento cumpla con los siguientes requisitos:

- Asegúrese de que el almacenamiento NetApp conectado al clúster utilice Trident 24.02 o posterior (se recomienda Trident 24.10).
- Asegúrese de tener un backend de almacenamiento NetApp ONTAP .
- Asegúrese de haber configurado un bucket de almacenamiento de objetos para almacenar las copias de seguridad.
- Cree los espacios de nombres de aplicación que planee utilizar para las aplicaciones o las operaciones de administración de datos de las aplicaciones. Trident Protect no crea estos espacios de nombres por usted; si especifica un espacio de nombres inexistente en un recurso personalizado, la operación fallará.

Requisitos para volúmenes nas-economy

Trident Protect admite operaciones de copia de seguridad y restauración en volúmenes nas-economy. Actualmente no se admiten instantáneas, clones ni replicación SnapMirror a volúmenes nas-economy. Debe habilitar un directorio de instantáneas para cada volumen nas-economy que planea usar con Trident Protect.

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

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

Puede habilitar el directorio de instantáneas ejecutando el siguiente comando para cada volumen nas-economy, reemplazando `<volume-UUID>` con el UUID del volumen que desea cambiar:

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

Puede habilitar los directorios de instantáneas de forma predeterminada para los nuevos volúmenes configurando la opción de configuración del backend de Trident. `snapshotDir` a `true` . Los volúmenes existentes no se ven afectados.

Protección de datos con máquinas virtuales KubeVirt

Trident Protect 24.10 y 24.10.1 y versiones más recientes tienen un comportamiento diferente cuando protege aplicaciones que se ejecutan en máquinas virtuales de KubeVirt. En ambas versiones, puede habilitar o deshabilitar la congelación y descongelación del sistema de archivos durante las operaciones de protección de datos.

Durante las operaciones de restauración, cualquier `VirtualMachineSnapshots` Los archivos creados para una máquina virtual (VM) no se restauran.

Trident Protect 24.10

Trident Protect 24.10 no garantiza automáticamente un estado consistente para los sistemas de archivos de VM KubeVirt durante las operaciones de protección de datos. Si desea proteger los datos de su máquina virtual KubeVirt con Trident Protect 24.10, debe habilitar manualmente la funcionalidad de congelamiento/descongelamiento de los sistemas de archivos antes de la operación de protección de datos. Esto garantiza que los sistemas de archivos se encuentren en un estado consistente.

Puede configurar Trident Protect 24.10 para administrar la congelación y descongelación del sistema de archivos de la máquina virtual durante las operaciones de protección de datos."configuración de la virtualización" y luego usando el siguiente comando:

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

Trident Protect 24.10.1 y posteriores

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

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

Requisitos para la replicación de SnapMirror

La replicación de NetApp SnapMirror está disponible para su uso con Trident Protect para las siguientes soluciones ONTAP :

- Clústeres NetApp FAS, AFF y ASA locales
- Selección de NetApp ONTAP Select
- Volúmenes en la nube de NetApp Cloud Volumes ONTAP
- Amazon FSx for NetApp ONTAP

Requisitos del clúster ONTAP para la replicación SnapMirror

Asegúrese de que su clúster ONTAP cumpla con los siguientes requisitos si planea utilizar la replicación SnapMirror :

- * NetApp Trident*: NetApp Trident debe existir tanto en el clúster de Kubernetes de origen como en el de destino que utilizan ONTAP como backend. Trident Protect admite la replicación con la tecnología NetApp SnapMirror utilizando clases de almacenamiento respaldadas por los siguientes controladores:
  - `ontap-nas`NFS
  - `ontap-san`iSCSI
  - `ontap-san`: FC
  - `ontap-san`: NVMe/TCP (requiere como mínimo la versión 9.15.1 de ONTAP)
- **Licencias**: Las licencias asíncronas de ONTAP SnapMirror que utilizan el paquete de protección de datos deben estar habilitadas tanto en el clúster ONTAP de origen como en el de destino. Referirse a "Información general sobre 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 de NetApp (NLF), que es un único archivo que habilita múltiples funciones. Referirse a"Licencias incluidas con ONTAP One" Para más información.

  > Solo se admite la protección asíncrona de SnapMirror .

Consideraciones de interconexión para la replicación de SnapMirror

Asegúrese de que su entorno cumpla los siguientes requisitos si planea utilizar el emparejamiento de backend de almacenamiento:

- **Clúster y SVM**: Los backends de almacenamiento ONTAP deben estar interconectados. Referirse a "Descripción general del emparejamiento de clústeres y SVM" Para más información.

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

- * NetApp Trident y SVM*: Las SVM remotas emparejadas deben estar disponibles para NetApp Trident en el clúster de destino.
- **Backends administrados**: debe agregar y administrar backends de almacenamiento ONTAP en Trident Protect para crear una relación de replicación.

Configuración de Trident / ONTAP para la replicación de SnapMirror

Trident Protect requiere que configure al menos un backend de almacenamiento que admita la replicación para los clústeres de origen y destino. Si los clústeres de origen y destino son los mismos, la aplicación de destino debería utilizar 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

Asegúrese de que sus clústeres de Kubernetes cumplan los siguientes requisitos:

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

- **Conectividad de red**: Configure las reglas del firewall, los permisos de los buckets y las listas blancas de IP para habilitar la comunicación entre ambos clústeres y AppVault a través de las WAN.

> Muchos entornos empresariales implementan políticas de firewall estrictas en las conexiones WAN. Verifique estos requisitos de red con su equipo de infraestructura antes de configurar la replicación.

Instalar y configurar Trident Protect

Si su entorno cumple con los requisitos de Trident Protect, puede seguir estos pasos para instalar Trident Protect en su clúster. Puede obtener Trident Protect de NetApp o instalarlo desde su propio registro privado. Instalar desde un registro privado resulta útil si su clúster no puede acceder a Internet.

Instalar Trident Protect

Instalar Trident Protect de NetApp

Pasos

1. Añadir el repositorio Trident Helm:

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

2. Utilice Helm para instalar Trident Protect. Reemplazar `<name-of-cluster>` con un nombre de clúster, que se asignará al clúster y se utilizará para identificar las copias de seguridad y las instantáneas del clúster:

```
helm install trident-protect netapp-trident-protect/trident-protect
--set clusterName=<name-of-cluster> --version 100.2506.0 --create
-namespace --namespace trident-protect
```

Instalar Trident Protect desde un registro privado

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

Pasos

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

```
netapp/controller:25.06.0
netapp/restic:25.06.0
netapp/kopia:25.06.0
netapp/trident-autosupport:25.06.0
netapp/exechook:25.06.0
netapp/resourcebackup:25.06.0
netapp/resourcerestore:25.06.0
netapp/resourcedelete:25.06.0
bitnami/kubectl:1.30.2
kubebuilder/kube-rbac-proxy:v0.16.0
```

   Por ejemplo:

```
docker pull netapp/controller:25.06.0
```

```
docker tag netapp/controller:25.06.0 <private-registry-
url>/controller:25.06.0
```

```
docker push <private-registry-url>/controller:25.06.0
```

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

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

3. Iniciar sesión en el registro:

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

4. Crea un secreto de extracción para usar 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. Añadir 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` . Asegúrese de que contenga las siguientes configuraciones de Trident Protect:

```
---
image:
  registry: <private-registry-url>
imagePullSecrets:
  - name: regcred
controller:
  image:
    registry: <private-registry-url>
rbacProxy:
  image:
    registry: <private-registry-url>
crCleanup:
  imagePullSecrets:
    - name: regcred
webhooksCleanup:
  imagePullSecrets:
    - name: regcred
```

7. Utilice Helm para instalar Trident Protect. Reemplazar `<name_of_cluster>` con un nombre de clúster, que se asignará al clúster y se utilizará para identificar las copias de seguridad y las instantáneas del clúster:

```
helm install trident-protect netapp-trident-protect/trident-protect
--set clusterName=<name_of_cluster> --version 100.2506.0 --create
-namespace --namespace trident-protect -f protectValues.yaml
```

Instalar el complemento CLI de Trident Protect

Puede utilizar el complemento de línea de comandos Trident Protect, que es una extensión de Trident `tridentctl` utilidad para crear e interactuar con recursos personalizados (CR) de Trident Protect.

Instalar el complemento CLI de Trident Protect

Antes de utilizar la utilidad de línea de comandos, debe instalarla en la máquina que utiliza para acceder a su clúster. Siga estos pasos, dependiendo de si su máquina utiliza una CPU x64 o ARM .

Descargar complemento para procesadores Linux AMD64

Pasos

1. Descargue el complemento CLI de Trident Protect:

```
curl -L -o tridentctl-protect https://github.com/NetApp/tridentctl-protect/releases/download/25.06.0/tridentctl-protect-linux-amd64
```

Descargar plugin para CPU Linux ARM64

Pasos

1. Descargue el complemento CLI de Trident Protect:

```
curl -L -o tridentctl-protect https://github.com/NetApp/tridentctl-protect/releases/download/25.06.0/tridentctl-protect-linux-arm64
```

Descargar plugin para procesadores Mac AMD64

Pasos

1. Descargue el complemento CLI de Trident Protect:

```
curl -L -o tridentctl-protect https://github.com/NetApp/tridentctl-protect/releases/download/25.06.0/tridentctl-protect-macos-amd64
```

Descargar plugin para procesadores Mac ARM64

Pasos

1. Descargue el complemento CLI de Trident Protect:

```
curl -L -o tridentctl-protect https://github.com/NetApp/tridentctl-protect/releases/download/25.06.0/tridentctl-protect-macos-arm64
```

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

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

2. Copie el archivo binario del plugin a una ubicación que esté definida en su variable PATH. Por ejemplo, `/usr/bin` o `/usr/local/bin` (Es posible que necesites privilegios elevados):

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

3. Opcionalmente, puede copiar el archivo binario del plugin a una ubicación en su directorio de inicio. En este caso, se recomienda asegurarse de que la ubicación forme parte de la variable PATH:

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

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

Consulta la ayuda del plugin Trident CLI.

Puedes utilizar las funciones de ayuda integradas del plugin para obtener ayuda detallada sobre sus capacidades:

Pasos

1. Utilice la función de ayuda para ver las instrucciones de uso:

```
tridentctl-protect help
```

Habilitar la autocompletación de comandos

Después de haber instalado el complemento CLI de Trident Protect, puede habilitar el autocompletado para ciertos comandos.

Habilitar la autocompletación para el shell de Bash

Pasos

1. Descarga el script de finalización:

```
curl -L -O https://github.com/NetApp/tridentctl-
protect/releases/download/25.06.0/tridentctl-completion.bash
```

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

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

3. Mueva el script descargado a la `~/.bash/completions` directorio:

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

4. Agrega la siguiente línea a la `~/.bashrc` archivo en tu directorio de inicio:

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

Habilitar la autocompletación para la shell Z

Pasos

1. Descarga el script de finalización:

```
curl -L -O https://github.com/NetApp/tridentctl-
protect/releases/download/25.06.0/tridentctl-completion.zsh
```

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

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

3. Mueva el script descargado a la `~/.zsh/completions` directorio:

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

4. Agrega la siguiente línea a la `~/.zprofile` archivo en tu directorio de inicio:

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

Resultado

En su próximo inicio de sesión en la shell, puede usar la autocompletación de comandos con el complemento tridentctl-protect.

Personalizar la instalación de Trident Protect

Puede personalizar la configuración predeterminada de Trident Protect para satisfacer los requisitos específicos de su entorno.

Especificar los límites de recursos del contenedor Trident Protect

Puede utilizar un archivo de configuración para especificar límites de recursos para los contenedores de Trident Protect después de instalar Trident Protect. Establecer límites de recursos le permite controlar qué cantidad de recursos del clúster consumen las operaciones de Trident Protect.

Pasos

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

   El siguiente archivo de configuración de ejemplo muestra las opciones 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. Aplique los valores de `resourceLimits.yaml` archivo:

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

Personalizar las restricciones del contexto de seguridad

Puede utilizar un archivo de configuración para modificar la restricción de contexto de seguridad (SCC) de OpenShift para los contenedores de Trident Protect después de instalar Trident Protect. Estas restricciones definen las limitaciones de seguridad para los pods en un clúster de Red Hat OpenShift.

Pasos

1. Crea un archivo llamado `sccconfig.yaml` .
2. Agregue la opción SCC al archivo y modifique los parámetros según las necesidades de su 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	Por defecto
crear	Determina si se puede crear un recurso SCC. Un recurso SCC se creará solo si `scc.create` está configurado para `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 para `false` , no se creará ningún recurso SCC.	verdadero
nombre	Especifica el nombre del SCC.	trabajo de protección de tridente
prioridad	Define la prioridad del SCC. Los SCC con valores de prioridad más altos se evalúan antes que aquellos con valores más bajos.	1

3. Aplique los valores de `sccconfig.yaml` archivo:

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

Esto reemplazará los valores predeterminados con los especificados en el `sccconfig.yaml` archivo.

Configurar ajustes adicionales del gráfico de timón Trident Protect

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

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

Parámetro	Tipo	Descripción
autoSupport.inseguro	booleano	Omite la verificación TLS para las conexiones proxy de AutoSupport cuando está configurado en `true` . Úselo sólo para conexiones proxy inseguras. (por defecto: `false`)
autoSupport.enabled	booleano	Habilita o deshabilita las cargas diarias de paquetes de AutoSupport de Trident Protect. Cuando se configura para `false` Las cargas diarias programadas están desactivadas, pero aún puede generar manualmente paquetes de soporte. (por defecto: `true`)
restaurar anotaciones de SkipNamespace	cadena	Lista separada por comas de anotaciones de espacios de nombres para excluir de las operaciones de copia de seguridad y restauración. Le permite filtrar espacios de nombres según anotaciones.
restaurar etiquetas de espacios de nombres de saltos	cadena	Lista separada por comas de etiquetas de espacios de nombres para excluir de las operaciones de copia de seguridad y restauración. Le permite filtrar espacios de nombres según etiquetas.

Puede configurar estas opciones mediante un archivo de configuración YAML o indicadores de línea de comandos:

Utilice el archivo YAML

Pasos

1. Crea un archivo de configuración y nómbralo. `values.yaml` .
2. En el archivo que ha creado, agregue las opciones de configuración que desea personalizar.

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

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

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

Usar la bandera CLI

Pasos

1. Utilice el siguiente comando con el `--set` bandera 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 restoreSkipNamespaceAnnotations="annotation1,annotation2" \
  --set restoreSkipNamespaceLabels="label1,label2" \
  --reuse-values
```

Restringir los pods de Trident Protect a nodos específicos

Puede utilizar la restricción de selección de nodos nodeSelector de Kubernetes para controlar cuáles de sus nodos son elegibles para ejecutar pods de Trident Protect, según las etiquetas de los nodos. De forma predeterminada, Trident Protect está restringido a los nodos que ejecutan Linux. Puedes personalizar aún más estas restricciones según tus necesidades.

Pasos

1. Crea un archivo llamado `nodeSelectorConfig.yaml` .
2. Agregue la opción nodeSelector al archivo y modifique el archivo para agregar o cambiar las etiquetas de los nodos para restringirlos según las necesidades de su entorno. Por ejemplo, el siguiente archivo

contiene la restricción predeterminada del sistema operativo, pero también se dirige a una región y un nombre de aplicación específicos:

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

3. Aplique los valores de `nodeSelectorConfig.yaml` archivo:

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

Esto reemplaza las restricciones predeterminadas con las que usted especificó en el `nodeSelectorConfig.yaml` archivo.

Administrar Trident Protect

Administrar la autorización y el control de acceso de Trident Protect

Trident Protect utiliza el modelo Kubernetes de control de acceso basado en roles (RBAC). De forma predeterminada, Trident Protect proporciona un único espacio de nombres de sistema y su cuenta de servicio predeterminada asociada. Si tiene una organización con muchos usuarios o necesidades de seguridad específicas, puede utilizar las funciones RBAC de Trident Protect para obtener un control más granular sobre el acceso a los recursos y espacios de nombres.

El administrador del clúster siempre tiene acceso a los recursos en la configuración predeterminada. `trident-protect` espacio de nombres, y también puede acceder a recursos en todos los demás espacios de nombres. Para controlar el acceso a los recursos y las aplicaciones, debe crear espacios de nombres adicionales y agregar los recursos y las aplicaciones a esos espacios de nombres.

Tenga en cuenta que ningún usuario puede crear solicitudes de cambio (CR) de administración de datos de aplicaciones en la configuración predeterminada. `trident-protect` espacio de nombres. Debe crear los CR de administración de datos de la aplicación en un espacio de nombres de la aplicación (como práctica recomendada, cree los CR de administración de datos de la aplicación en el mismo espacio de nombres que su aplicación asociada).

Solo los administradores deben tener acceso a los objetos de recursos personalizados privilegiados de Trident Protect, que incluyen:

- **AppVault**: Requiere datos de credenciales del bucket
- **AutoSupportBundle**: Recopila métricas, registros y otros datos confidenciales de Trident Protect
- **AutoSupportBundleSchedule**: Gestiona los horarios de recopilación de registros.

Como práctica recomendada, utilice RBAC para restringir el acceso a objetos privilegiados a los administradores.

Para obtener más información sobre cómo RBAC regula el acceso a los recursos y espacios de nombres, consulte la documentación. "Documentación de RBAC de Kubernetes" .

Para obtener información sobre las cuentas de servicio, consulte la "Documentación de la cuenta de servicio de Kubernetes" .

Ejemplo: Gestionar el acceso para dos grupos de usuarios

Por ejemplo, una organización tiene un administrador de clúster, un grupo de usuarios de ingeniería y un grupo de usuarios de marketing. El administrador del clúster completaría las siguientes tareas para crear un entorno donde el grupo de ingeniería y el grupo de marketing tengan acceso únicamente a los recursos asignados a sus respectivos espacios de nombres.

Paso 1: Cree un espacio de nombres para contener los recursos de cada grupo.

La creación de un espacio de nombres le permite separar lógicamente los recursos y controlar mejor quién tiene acceso a ellos.

Pasos

1. Crea un espacio de nombres para el grupo de ingeniería:

```
kubectl create ns engineering-ns
```

2. Crea un espacio de nombres para el grupo de marketing:

```
kubectl create ns marketing-ns
```

Paso 2: Cree nuevas cuentas de servicio para interactuar con los recursos en cada espacio de nombres.

Cada nuevo espacio de nombres que cree viene con una cuenta de servicio predeterminada, pero debería crear una cuenta de servicio para cada grupo de usuarios para que pueda dividir aún más los privilegios entre grupos en el futuro si fuera necesario.

Pasos

1. Cree una cuenta de servicio para el grupo de ingeniería:

```
apiVersion: v1
kind: ServiceAccount
metadata:
  name: eng-user
  namespace: engineering-ns
```

2. Crea una cuenta de servicio para el grupo de marketing:

```
apiVersion: v1
kind: ServiceAccount
metadata:
  name: mkt-user
  namespace: marketing-ns
```

Paso 3: Cree un secreto para cada nueva cuenta de servicio

Se utiliza una clave secreta de cuenta de servicio para autenticarse con la cuenta de servicio, y se puede eliminar y volver a crear fácilmente si se ve comprometida.

Pasos

1. Cree un secreto para la cuenta del servicio de ingeniería:

```
apiVersion: v1
kind: Secret
metadata:
  annotations:
    kubernetes.io/service-account.name: eng-user
  name: eng-user-secret
  namespace: engineering-ns
type: kubernetes.io/service-account-token
```

2. Crea una clave secreta para la cuenta del servicio de marketing:

```
apiVersion: v1
kind: Secret
metadata:
  annotations:
    kubernetes.io/service-account.name: mkt-user
  name: mkt-user-secret
  namespace: marketing-ns
type: kubernetes.io/service-account-token
```

Paso 4: Cree un objeto RoleBinding para vincular el objeto ClusterRole a cada nueva cuenta de servicio.

Se crea un objeto ClusterRole predeterminado cuando instala Trident Protect. Puede vincular este ClusterRole a la cuenta de servicio creando y aplicando un objeto RoleBinding.

Pasos

1. Vincule el ClusterRole a la cuenta de servicio de ingeniería:

```
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: engineering-ns-tenant-rolebinding
  namespace: engineering-ns
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: trident-protect-tenant-cluster-role
subjects:
- kind: ServiceAccount
  name: eng-user
  namespace: engineering-ns
```

2. Vincula el ClusterRole a la cuenta del servicio de marketing:

```
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: marketing-ns-tenant-rolebinding
  namespace: marketing-ns
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: trident-protect-tenant-cluster-role
subjects:
- kind: ServiceAccount
  name: mkt-user
  namespace: marketing-ns
```

Paso 5: Probar permisos

Comprueba que los permisos sean correctos.

Pasos

1. Confirme que los usuarios de ingeniería pueden acceder a los recursos de ingeniería:

```
kubectl auth can-i --as=system:serviceaccount:engineering-ns:eng-user
get applications.protect.trident.netapp.io -n engineering-ns
```

2. Confirma que los usuarios de ingeniería no pueden acceder a los recursos de marketing:

```
kubectl auth can-i --as=system:serviceaccount:engineering-ns:eng-user
get applications.protect.trident.netapp.io -n marketing-ns
```

Paso 6: Conceder acceso a los objetos de AppVault

Para realizar tareas de administración de datos como copias de seguridad e instantáneas, el administrador del clúster debe otorgar acceso a los objetos de AppVault a usuarios individuales.

Pasos

1. Cree y aplique un archivo YAML de combinación de AppVault y secreto que otorgue a un usuario acceso a un AppVault. Por ejemplo, la siguiente solicitud CR otorga acceso a un AppVault al usuario `eng-user` :

```
apiVersion: v1
data:
  accessKeyID: <ID_value>
  secretAccessKey: <key_value>
kind: Secret
metadata:
  name: appvault-for-eng-user-only-secret
  namespace: trident-protect
type: Opaque
---
apiVersion: protect.trident.netapp.io/v1
kind: AppVault
metadata:
  name: appvault-for-eng-user-only
  namespace: trident-protect # Trident Protect system namespace
spec:
  providerConfig:
    azure:
      accountName: ""
      bucketName: ""
      endpoint: ""
    gcp:
      bucketName: ""
      projectID: ""
    s3:
      bucketName: testbucket
      endpoint: 192.168.0.1:30000
      secure: "false"
      skipCertValidation: "true"
  providerCredentials:
    accessKeyID:
      valueFromSecret:
        key: accessKeyID
        name: appvault-for-eng-user-only-secret
    secretAccessKey:
      valueFromSecret:
        key: secretAccessKey
        name: appvault-for-eng-user-only-secret
  providerType: GenericS3
```

2. Cree y aplique una solicitud de cambio de rol para permitir que los administradores del clúster otorguen acceso a recursos específicos en un espacio de nombres. Por ejemplo:

```
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  name: eng-user-appvault-reader
  namespace: trident-protect
rules:
- apiGroups:
  - protect.trident.netapp.io
  resourceNames:
  - appvault-for-enguser-only
  resources:
  - appvaults
  verbs:
  - get
```

3. Cree y aplique un CR RoleBinding para vincular los permisos al usuario eng-user. Por ejemplo:

```
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: eng-user-read-appvault-binding
  namespace: trident-protect
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: Role
  name: eng-user-appvault-reader
subjects:
- kind: ServiceAccount
  name: eng-user
  namespace: engineering-ns
```

4. Verifique que los permisos sean correctos.

   a. Intento de recuperar la información de objetos de AppVault para todos los espacios de nombres:

   ```
   kubectl get appvaults -n trident-protect
   --as=system:serviceaccount:engineering-ns:eng-user
   ```

   Debería ver un resultado similar al siguiente:

```
Error from server (Forbidden): appvaults.protect.trident.netapp.io is
forbidden: User "system:serviceaccount:engineering-ns:eng-user"
cannot list resource "appvaults" in API group
"protect.trident.netapp.io" in the namespace "trident-protect"
```

b. Prueba para comprobar si el usuario puede obtener la información de AppVault a la que ahora tiene permiso de acceso:

```
kubectl auth can-i --as=system:serviceaccount:engineering-ns:eng-user
get appvaults.protect.trident.netapp.io/appvault-for-eng-user-only -n
trident-protect
```

Debería ver un resultado similar al siguiente:

```
yes
```

Resultado

Los usuarios a los que les haya concedido permisos de AppVault deberían poder utilizar los objetos autorizados de AppVault para las operaciones de gestión de datos de la aplicación, y no deberían poder acceder a ningún recurso fuera de los espacios de nombres asignados ni crear nuevos recursos a los que no tengan acceso.

Supervisar los recursos de Trident Protect

Puede utilizar las herramientas de código abierto kube-state-metrics, Prometheus y Alertmanager para supervisar el estado de los recursos protegidos por Trident Protect.

El servicio kube-state-metrics genera métricas a partir de la comunicación con la API de Kubernetes. Usarlo con Trident Protect expone información útil sobre el estado de los recursos en su entorno.

Prometheus es un conjunto de herramientas que puede ingerir los datos generados por kube-state-metrics y presentarlos como información fácilmente legible sobre estos objetos. Juntos, kube-state-metrics y Prometheus le brindan una manera de monitorear la salud y el estado de los recursos que administra con Trident Protect.

Alertmanager es un servicio que recibe las alertas enviadas por herramientas como Prometheus y las dirige a los destinos que usted configure.

> Las configuraciones y la orientación incluidas en estos pasos son solo ejemplos; debe personalizarlas para que se ajusten a su entorno. Consulte la siguiente documentación oficial para obtener instrucciones y asistencia específicas:
>
> - "Documentación de kube-state-metrics"
> - "Documentación de Prometeo"
> - "Documentación de Alertmanager"

Paso 1: Instalar las herramientas de monitorización

Para habilitar la supervisión de recursos en Trident Protect, debe instalar y configurar kube-state-metrics, Promethus y Alertmanager.

Instalar kube-state-metrics

Puedes instalar kube-state-metrics usando Helm.

Pasos

1. Agrega el gráfico Helm kube-state-metrics. Por ejemplo:

```
helm repo add prometheus-community https://prometheus-
community.github.io/helm-charts
helm repo update
```

2. Aplique el CRD de Prometheus ServiceMonitor al clúster:

```
kubectl apply -f https://raw.githubusercontent.com/prometheus-
operator/prometheus-operator/main/example/prometheus-operator-
crd/monitoring.coreos.com_servicemonitors.yaml
```

3. Crea un archivo de configuración para el gráfico de Helm (por ejemplo, `metrics-config.yaml`). Puede personalizar la siguiente configuración de ejemplo para que se ajuste a su entorno:

metrics-config.yaml: Configuración del gráfico Helm de kube-state-metrics

```
---
extraArgs:
  # Collect only custom metrics
  - --custom-resource-state-only=true

customResourceState:
  enabled: true
  config:
    kind: CustomResourceStateMetrics
    spec:
      resources:
      - groupVersionKind:
          group: protect.trident.netapp.io
          kind: "Backup"
          version: "v1"
        labelsFromPath:
          backup_uid: [metadata, uid]
          backup_name: [metadata, name]
          creation_time: [metadata, creationTimestamp]
        metrics:
        - name: backup_info
          help: "Exposes details about the Backup state"
          each:
            type: Info
            info:
              labelsFromPath:
                appVaultReference: ["spec", "appVaultRef"]
                appReference: ["spec", "applicationRef"]
rbac:
  extraRules:
  - apiGroups: ["protect.trident.netapp.io"]
    resources: ["backups"]
    verbs: ["list", "watch"]

# Collect metrics from all namespaces
namespaces: ""

# Ensure that the metrics are collected by Prometheus
prometheus:
  monitor:
    enabled: true
```

4. Instale kube-state-metrics implementando el gráfico Helm. Por ejemplo:

```
helm install custom-resource -f metrics-config.yaml prometheus-
community/kube-state-metrics --version 5.21.0
```

5. Configure kube-state-metrics para generar métricas para los recursos personalizados utilizados por Trident Protect siguiendo las instrucciones en "Documentación del recurso personalizado kube-state-metrics" .

Instalar Prometheus

Puedes instalar Prometheus siguiendo las instrucciones del manual. "Documentación de Prometeo" .

Instalar Alertmanager

Puede instalar Alertmanager siguiendo las instrucciones del siguiente enlace: "Documentación de Alertmanager" .

Paso 2: Configurar las herramientas de monitorización para que funcionen conjuntamente.

Después de instalar las herramientas de monitorización, deberá configurarlas para que funcionen conjuntamente.

Pasos

1. Integra kube-state-metrics con Prometheus. Edita el archivo de configuración de Prometheus(`prometheus.yaml`) y agregue la información del servicio kube-state-metrics. Por ejemplo:

   prometheus.yaml: Integración del servicio kube-state-metrics con Prometheus

```
---
apiVersion: v1
kind: ConfigMap
metadata:
  name: prometheus-config
  namespace: trident-protect
data:
  prometheus.yaml: |
    global:
      scrape_interval: 15s
    scrape_configs:
      - job_name: 'kube-state-metrics'
        static_configs:
          - targets: ['kube-state-metrics.trident-protect.svc:8080']
```

2. Configura Prometheus para que dirija las alertas a Alertmanager. Edita el archivo de configuración de Prometheus(`prometheus.yaml`) y agregue la siguiente sección:

prometheus.yaml: Enviar alertas a Alertmanager

```
alerting:
  alertmanagers:
    - static_configs:
        - targets:
            - alertmanager.trident-protect.svc:9093
```

Resultado

Prometheus ahora puede recopilar métricas de kube-state-metrics y enviar alertas a Alertmanager. Ahora ya puedes configurar qué condiciones activan una alerta y dónde deben enviarse las alertas.

Paso 3: Configurar alertas y destinos de alertas

Una vez configuradas las herramientas para que funcionen conjuntamente, es necesario configurar qué tipo de información activa las alertas y dónde deben enviarse dichas alertas.

Ejemplo de alerta: fallo de copia de seguridad

El siguiente ejemplo define una alerta crítica que se activa cuando el estado del recurso personalizado de copia de seguridad se establece en `Error` durante 5 segundos o más. Puedes personalizar este ejemplo para que se ajuste a tu entorno e incluir este fragmento YAML en tu `prometheus.yaml` archivo de configuración:

rules.yaml: Define una alerta de Prometheus para copias de seguridad fallidas.

```
rules.yaml: |
  groups:
    - name: fail-backup
        rules:
          - alert: BackupFailed
            expr: kube_customresource_backup_info{status="Error"}
            for: 5s
            labels:
              severity: critical
            annotations:
              summary: "Backup failed"
              description: "A backup has failed."
```

Configura Alertmanager para enviar alertas a otros canales.

Puede configurar Alertmanager para que envíe notificaciones a otros canales, como correo electrónico, PagerDuty, Microsoft Teams u otros servicios de notificación, especificando la configuración correspondiente en el archivo `alertmanager.yaml` archivo.

El siguiente ejemplo configura Alertmanager para enviar notificaciones a un canal de Slack. Para personalizar este ejemplo a su entorno, reemplace el valor de `api_url` clave con la URL del webhook de Slack utilizada en su entorno:

alertmanager.yaml: Enviar alertas a un canal de Slack

```
data:
  alertmanager.yaml: |
    global:
      resolve_timeout: 5m
    route:
      receiver: 'slack-notifications'
    receivers:
      - name: 'slack-notifications'
        slack_configs:
          - api_url: '<your-slack-webhook-url>'
            channel: '#failed-backups-channel'
            send_resolved: false
```

Generar un paquete de soporte de Trident Protect

Trident Protect permite a los administradores generar paquetes que incluyen información útil para el soporte de NetApp , incluidos registros, métricas e información de topología sobre los clústeres y las aplicaciones bajo administración. Si está conectado a Internet, puede cargar paquetes de soporte en el sitio de soporte de NetApp (NSS) mediante un archivo de recursos personalizados (CR).

Cree un paquete de soporte utilizando un CR.

Pasos

1. Cree el archivo de recursos personalizados (CR) y asígnele un nombre (por ejemplo, `trident-protect-support-bundle.yaml`).
2. Configure los siguientes atributos:
   - **metadata.name**: (*Obligatorio*) El nombre de este recurso personalizado; elija un nombre único y adecuado para su entorno.
   - **spec.triggerType**: (*Obligatorio*) Determina si el paquete de soporte se genera inmediatamente o se programa. La generación programada de paquetes se produce a las 12 AM UTC. Valores posibles:
     - Programado
     - Manual
   - **spec.uploadEnabled**: (*Opcional*) Controla si el paquete de soporte debe cargarse en el sitio de soporte de NetApp después de generarse. Si no se especifica, el valor predeterminado es `false` . Valores posibles:
     - verdadero
     - falso (predeterminado)
   - **spec.dataWindowStart**: (*Opcional*) Una cadena de fecha en formato RFC 3339 que especifica la fecha y hora en que debe comenzar la ventana de datos incluidos en el paquete de soporte. Si no se especifica, se tomará como valor predeterminado las últimas 24 horas. La fecha más antigua que puedes especificar es hace 7 días.

     Ejemplo de YAML:

```
---
apiVersion: protect.trident.netapp.io/v1
kind: AutoSupportBundle
metadata:
  name: trident-protect-support-bundle
spec:
  triggerType: Manual
  uploadEnabled: true
  dataWindowStart: 2024-05-05T12:30:00Z
```

3. Después de rellenar el `trident-protect-support-bundle.yaml` Archivo con los valores correctos, aplicar el CR:

```
kubectl apply -f trident-protect-support-bundle.yaml -n trident-protect
```

Crea un paquete de soporte usando la CLI.

Pasos

1. Cree el paquete de soporte, reemplazando los valores entre corchetes con información de su entorno. El `trigger-type` Determina si el paquete se crea inmediatamente o si el tiempo de creación viene dictado por la programación, y puede ser `Manual` o `Scheduled` . La configuración predeterminada es `Manual` .

   Por ejemplo:

   ```
   tridentctl-protect create autosupportbundle <my-bundle-name>
   --trigger-type <trigger-type> -n trident-protect
   ```

Supervisar y recuperar el paquete de soporte

Después de crear un paquete de soporte utilizando cualquiera de los métodos, puede supervisar el progreso de su generación y recuperarlo en su sistema local.

Pasos

1. Espera a que `status.generationState` para alcanzar `Completed` estado. Puede monitorear el progreso de la generación con el siguiente comando:

   ```
   kubectl get autosupportbundle trident-protect-support-bundle -n trident-protect
   ```

2. Recupere el paquete de soporte para su sistema local. Obtenga el comando de copia del paquete AutoSupport completado:

   ```
   kubectl describe autosupportbundle trident-protect-support-bundle -n trident-protect
   ```

   Encuentra el `kubectl cp` Copie el comando de la salida y ejecútelo, reemplazando el argumento de destino con el directorio local que prefiera.

Actualizar Trident Protect

Puede actualizar Trident Protect a la última versión para aprovechar nuevas funciones o correcciones de errores.

> Al actualizar desde la versión 24.10, es posible que fallen las instantáneas que se ejecutan durante la actualización. Este fallo no impide que se creen instantáneas futuras, ya sean manuales o programadas. Si una instantánea falla durante la actualización, puede crear manualmente una nueva instantánea para garantizar la protección de su aplicación.
>
> Para evitar posibles fallos, puede deshabilitar todas las programaciones de instantáneas antes de la actualización y volver a habilitarlas después. Sin embargo, esto provoca que se pierdan las instantáneas programadas durante el período de actualización.

Para actualizar Trident Protect, realice los siguientes pasos.

Pasos

1. Actualizar el repositorio de Trident Helm:

```
helm repo update
```

2. Actualice los CRD Trident Protect:

> Este paso es necesario si estás actualizando desde una versión anterior a la 25.06, ya que los CRD ahora están incluidos en la tabla de Trident Protect Helm.

   a. Ejecute este comando para transferir la administración de CRD desde `trident-protect-crds` a `trident-protect` :

```
kubectl get crd | grep protect.trident.netapp.io | awk '{print $1}' |
xargs -I {} kubectl patch crd {} --type merge -p '{"metadata":
{"annotations":{"meta.helm.sh/release-name": "trident-protect"}}}'
```

   b. Ejecuta este comando para eliminar el secreto de Helm para el `trident-protect-crds` cuadro:

> No desinstale el `trident-protect-crds` No utilice Helm para crear gráficos, ya que esto podría eliminar sus CRD y cualquier dato relacionado.

```
kubectl delete secret -n trident-protect -l name=trident-protect-
crds,owner=helm
```

3. Mejora Trident Protect:

```
helm upgrade trident-protect netapp-trident-protect/trident-protect
--version 100.2506.0 --namespace trident-protect
```

Gestionar y proteger las aplicaciones

Utilice objetos Trident Protect AppVault para administrar depósitos

El recurso personalizado (CR) de depósito para Trident Protect se conoce como AppVault. Los objetos AppVault son la representación declarativa del flujo de trabajo de Kubernetes de un depósito de almacenamiento. Un CR de AppVault contiene las configuraciones necesarias para que un bucket se utilice en operaciones de protección, como copias de seguridad, instantáneas, operaciones de restauración y replicación de SnapMirror . Solo los administradores pueden crear AppVaults.

Debe crear una CR de AppVault manualmente o desde la línea de comandos al realizar operaciones de protección de datos en una aplicación. La CR de AppVault es específica de su entorno, y puede usar los ejemplos de esta página como guía al crearlas.

Asegúrese de que AppVault CR esté en el clúster donde está instalado Trident Protect. Si el CR de AppVault no existe o no puede acceder a él, la línea de comandos mostrará un error.

Configurar la autenticación y las contraseñas de AppVault

Antes de crear un CR de AppVault, asegúrese de que AppVault y el transportador de datos que elija puedan autenticarse con el proveedor y cualquier recurso relacionado.

Contraseñas del repositorio de transferencia de datos

Cuando crea objetos de AppVault utilizando CR o el complemento CLI de Trident Protect, puede especificar un secreto de Kubernetes con contraseñas personalizadas para el cifrado de Restic y Kopia. Si no especifica un secreto, Trident Protect utiliza una contraseña predeterminada.

- Al crear manualmente solicitudes de cambio de AppVault, utilice el campo **spec.dataMoverPasswordSecretRef** para especificar el secreto.
- Al crear objetos de AppVault mediante la CLI de Trident Protect, utilice el `--data-mover-password-secret-ref` argumento para especificar el secreto.

Crea una contraseña secreta para el repositorio de transferencia de datos

Utilice los siguientes ejemplos para crear la contraseña secreta. Cuando crea objetos de AppVault, puede indicarle a Trident Protect que use este secreto para autenticarse con el repositorio de transferencia de datos.

- Dependiendo del gestor de transferencia de datos que esté utilizando, solo necesita incluir la contraseña correspondiente a ese gestor. Por ejemplo, si está utilizando Restic y no planea utilizar Kopia en el futuro, puede incluir solo la contraseña de Restic cuando cree el secreto.
- Guarde la contraseña en un lugar seguro. La necesitará para restaurar datos en el mismo clúster o en uno diferente. Si el grupo o el `trident-protect` Una vez eliminado el espacio de nombres, no podrá restaurar sus copias de seguridad ni instantáneas sin la contraseña.

Utilice un CR

```
---
apiVersion: v1
data:
  KOPIA_PASSWORD: <base64-encoded-password>
  RESTIC_PASSWORD: <base64-encoded-password>
kind: Secret
metadata:
  name: my-optional-data-mover-secret
  namespace: trident-protect
type: Opaque
```

Utilice la CLI

```
kubectl create secret generic my-optional-data-mover-secret \
--from-literal=KOPIA_PASSWORD=<plain-text-password> \
--from-literal=RESTIC_PASSWORD=<plain-text-password> \
-n trident-protect
```

Permisos IAM de almacenamiento compatibles con S3

Cuando accede a almacenamiento compatible con S3, como Amazon S3, Generic S3, "StorageGrid S3" , o "ONTAP S3" Al utilizar Trident Protect, debe asegurarse de que las credenciales de usuario que proporcione tengan los permisos necesarios para acceder al depósito. El siguiente es un ejemplo de una política que otorga los permisos mínimos requeridos para acceder con Trident Protect. Puede aplicar esta política al usuario que administra las políticas de bucket compatibles con S3.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:PutObject",
        "s3:GetObject",
        "s3:ListBucket",
        "s3:DeleteObject"
      ],
      "Resource": "*"
    }
  ]
}
```

Para obtener más información sobre las políticas de Amazon S3, consulte los ejemplos en el documento. "Documentación de Amazon S3" .

Identidad de pod EKS para la autenticación de Amazon S3 (AWS)

Trident Protect admite EKS Pod Identity para operaciones de transferencia de datos de Kopia. Esta función permite el acceso seguro a los buckets S3 sin almacenar las credenciales de AWS en los secretos de Kubernetes.

Requisitos para la identidad del pod EKS con Trident Protect

Antes de utilizar EKS Pod Identity con Trident Protect, asegúrese de lo siguiente:

- Su clúster EKS tiene Identidad de Pod habilitada.
- Ha creado una función de IAM con los permisos de bucket S3 necesarios. Para obtener más información, consulte "Permisos IAM de almacenamiento compatibles con S3".
- La función IAM está asociada con las siguientes cuentas de servicio de Trident Protect:
  - `<trident-protect>-controller-manager`
  - `<trident-protect>-resource-backup`
  - `<trident-protect>-resource-restore`
  - `<trident-protect>-resource-delete`

Para obtener instrucciones detalladas sobre cómo habilitar la identidad de pod y asociar roles de IAM con cuentas de servicio, consulte la documentación. "Documentación de identidad de pod de AWS EKS" .

Configuración de AppVault Al usar la identidad de pod de EKS, configure su CR de AppVault con la `useIAM: true` bandera en lugar de credenciales explícitas:

```
apiVersion: protect.trident.netapp.io/v1
kind: AppVault
metadata:
  name: eks-protect-vault
  namespace: trident-protect
spec:
  providerType: AWS
  providerConfig:
    s3:
      bucketName: trident-protect-aws
      endpoint: s3.example.com
      useIAM: true
```

Ejemplos de generación de claves de AppVault para proveedores de servicios en la nube

Al definir un CR de AppVault, debe incluir credenciales para acceder a los recursos alojados por el proveedor, a menos que utilice la autenticación IAM. La forma de generar las claves para las credenciales variará según el proveedor. Los siguientes son ejemplos de generación de claves de línea de comandos para varios proveedores. Puede utilizar los siguientes ejemplos para crear claves para las credenciales de cada proveedor de nube.

Google Cloud

```
kubectl create secret generic <secret-name> \
--from-file=credentials=<mycreds-file.json> \
-n trident-protect
```

Amazon S3 (AWS)

```
kubectl create secret generic <secret-name> \
--from-literal=accessKeyID=<objectstorage-accesskey> \
--from-literal=secretAccessKey=<amazon-s3-trident-protect-src-bucket
-secret> \
-n trident-protect
```

Microsoft Azure

```
kubectl create secret generic <secret-name> \
--from-literal=accountKey=<secret-name> \
-n trident-protect
```

S3 genérico

```
kubectl create secret generic <secret-name> \
--from-literal=accessKeyID=<objectstorage-accesskey> \
--from-literal=secretAccessKey=<generic-s3-trident-protect-src-bucket
-secret> \
-n trident-protect
```

ONTAP S3

```
kubectl create secret generic <secret-name> \
--from-literal=accessKeyID=<objectstorage-accesskey> \
--from-literal=secretAccessKey=<ontap-s3-trident-protect-src-bucket
-secret> \
-n trident-protect
```

StorageGrid S3

```
kubectl create secret generic <secret-name> \
--from-literal=accessKeyID=<objectstorage-accesskey> \
--from-literal=secretAccessKey=<storagegrid-s3-trident-protect-src
-bucket-secret> \
-n trident-protect
```

Ejemplos de creación de AppVault

A continuación se muestran ejemplos de definiciones de AppVault para cada proveedor.

Ejemplos de AppVault CR

Puede utilizar los siguientes ejemplos de CR para crear objetos AppVault para cada proveedor de nube.

- Opcionalmente, puede especificar un secreto de Kubernetes que contenga contraseñas personalizadas para el cifrado de los repositorios Restic y Kopia. Referirse aContraseñas del repositorio de transferencia de datos Para más información.
- Para los objetos de Amazon S3 (AWS) AppVault, puede especificar opcionalmente un sessionToken, lo cual es útil si está utilizando el inicio de sesión único (SSO) para la autenticación. Este token se crea cuando se generan las claves para el proveedor enEjemplos de generación de claves de AppVault para proveedores de servicios en la nube .
- Para los objetos S3 AppVault, puede especificar opcionalmente una URL de proxy de salida para el tráfico S3 saliente mediante `spec.providerConfig.S3.proxyURL` llave.

Google Cloud

```
apiVersion: protect.trident.netapp.io/v1
kind: AppVault
metadata:
  name: gcp-trident-protect-src-bucket
  namespace: trident-protect
spec:
  dataMoverPasswordSecretRef: my-optional-data-mover-secret
  providerType: GCP
  providerConfig:
    gcp:
      bucketName: trident-protect-src-bucket
      projectID: project-id
  providerCredentials:
    credentials:
      valueFromSecret:
        key: credentials
        name: gcp-trident-protect-src-bucket-secret
```

Amazon S3 (AWS)

```yaml
---
apiVersion: protect.trident.netapp.io/v1
kind: AppVault
metadata:
  name: amazon-s3-trident-protect-src-bucket
  namespace: trident-protect
spec:
  dataMoverPasswordSecretRef: my-optional-data-mover-secret
  providerType: AWS
  providerConfig:
    s3:
      bucketName: trident-protect-src-bucket
      endpoint: s3.example.com
      proxyURL: http://10.1.1.1:3128
  providerCredentials:
    accessKeyID:
      valueFromSecret:
        key: accessKeyID
        name: s3-secret
    secretAccessKey:
      valueFromSecret:
        key: secretAccessKey
        name: s3-secret
    sessionToken:
      valueFromSecret:
        key: sessionToken
        name: s3-secret
```

Para entornos EKS que utilizan Pod Identity con Kopia Data Mover, puede eliminar `providerCredentials` sección y agregar `useIAM: true` bajo el `s3` configuración en su lugar.

Microsoft Azure

```
apiVersion: protect.trident.netapp.io/v1
kind: AppVault
metadata:
  name: azure-trident-protect-src-bucket
  namespace: trident-protect
spec:
  dataMoverPasswordSecretRef: my-optional-data-mover-secret
  providerType: Azure
  providerConfig:
    azure:
      accountName: account-name
      bucketName: trident-protect-src-bucket
  providerCredentials:
    accountKey:
      valueFromSecret:
        key: accountKey
        name: azure-trident-protect-src-bucket-secret
```

S3 genérico

```
apiVersion: protect.trident.netapp.io/v1
kind: AppVault
metadata:
  name: generic-s3-trident-protect-src-bucket
  namespace: trident-protect
spec:
  dataMoverPasswordSecretRef: my-optional-data-mover-secret
  providerType: GenericS3
  providerConfig:
    s3:
      bucketName: trident-protect-src-bucket
      endpoint: s3.example.com
      proxyURL: http://10.1.1.1:3128
  providerCredentials:
    accessKeyID:
      valueFromSecret:
        key: accessKeyID
        name: s3-secret
    secretAccessKey:
      valueFromSecret:
        key: secretAccessKey
        name: s3-secret
```

ONTAP S3

```
apiVersion: protect.trident.netapp.io/v1
kind: AppVault
metadata:
  name: ontap-s3-trident-protect-src-bucket
  namespace: trident-protect
spec:
  dataMoverPasswordSecretRef: my-optional-data-mover-secret
  providerType: OntapS3
  providerConfig:
    s3:
      bucketName: trident-protect-src-bucket
      endpoint: s3.example.com
      proxyURL: http://10.1.1.1:3128
  providerCredentials:
    accessKeyID:
      valueFromSecret:
        key: accessKeyID
        name: s3-secret
    secretAccessKey:
      valueFromSecret:
        key: secretAccessKey
        name: s3-secret
```

StorageGrid S3

```
apiVersion: protect.trident.netapp.io/v1
kind: AppVault
metadata:
  name: storagegrid-s3-trident-protect-src-bucket
  namespace: trident-protect
spec:
  dataMoverPasswordSecretRef: my-optional-data-mover-secret
  providerType: StorageGridS3
  providerConfig:
    s3:
      bucketName: trident-protect-src-bucket
      endpoint: s3.example.com
      proxyURL: http://10.1.1.1:3128
  providerCredentials:
    accessKeyID:
      valueFromSecret:
        key: accessKeyID
        name: s3-secret
    secretAccessKey:
      valueFromSecret:
        key: secretAccessKey
        name: s3-secret
```

Ejemplos de creación de AppVault mediante la CLI de Trident Protect

Puede utilizar los siguientes ejemplos de comandos de la CLI para crear solicitudes de cambio de AppVault para cada proveedor.

- Opcionalmente, puede especificar un secreto de Kubernetes que contenga contraseñas personalizadas para el cifrado de los repositorios Restic y Kopia. Referirse aContraseñas del repositorio de transferencia de datos Para más información.
- Para los objetos S3 AppVault, puede especificar opcionalmente una URL de proxy de salida para el tráfico S3 saliente mediante `--proxy-url <ip_address:port>` argumento.

Google Cloud

```
tridentctl-protect create vault GCP <vault-name> \
--bucket <mybucket> \
--project <my-gcp-project> \
--secret <secret-name>/credentials \
--data-mover-password-secret-ref <my-optional-data-mover-secret> \
-n trident-protect
```

Amazon S3 (AWS)

```
tridentctl-protect create vault AWS <vault-name> \
--bucket <bucket-name> \
--secret  <secret-name>  \
--endpoint <s3-endpoint> \
--data-mover-password-secret-ref <my-optional-data-mover-secret> \
-n trident-protect
```

Microsoft Azure

```
tridentctl-protect create vault Azure <vault-name> \
--account <account-name> \
--bucket <bucket-name> \
--secret <secret-name> \
--data-mover-password-secret-ref <my-optional-data-mover-secret> \
-n trident-protect
```

S3 genérico

```
tridentctl-protect create vault GenericS3 <vault-name> \
--bucket <bucket-name> \
--secret  <secret-name>  \
--endpoint <s3-endpoint> \
--data-mover-password-secret-ref <my-optional-data-mover-secret> \
-n trident-protect
```

ONTAP S3

```
tridentctl-protect create vault OntapS3 <vault-name> \
--bucket <bucket-name> \
--secret  <secret-name>  \
--endpoint <s3-endpoint> \
--data-mover-password-secret-ref <my-optional-data-mover-secret> \
-n trident-protect
```

StorageGrid S3

```
tridentctl-protect create vault StorageGridS3 <vault-name> \
--bucket <bucket-name> \
--secret  <secret-name>  \
--endpoint <s3-endpoint> \
--data-mover-password-secret-ref <my-optional-data-mover-secret> \
-n trident-protect
```

Ver información de AppVault

Puede utilizar el complemento CLI de Trident Protect para ver información sobre los objetos de AppVault que ha creado en el clúster.

Pasos

1. Vea el contenido de un objeto AppVault:

```
tridentctl-protect get appvaultcontent gcp-vault \
--show-resources all \
-n trident-protect
```

Ejemplo de salida:

```
+-------------+-------+----------+-----------------------------
+--------------------------+
|   CLUSTER   |  APP  |   TYPE   |            NAME             |
TIMESTAMP         |
+-------------+-------+----------+-----------------------------
+--------------------------+
|             | mysql | snapshot | mysnap                      | 2024-
08-09 21:02:11 (UTC) |
| production1 | mysql | snapshot | hourly-e7db6-20240815180300 | 2024-
08-15 18:03:06 (UTC) |
| production1 | mysql | snapshot | hourly-e7db6-20240815190300 | 2024-
08-15 19:03:06 (UTC) |
| production1 | mysql | snapshot | hourly-e7db6-20240815200300 | 2024-
08-15 20:03:06 (UTC) |
| production1 | mysql | backup   | hourly-e7db6-20240815180300 | 2024-
08-15 18:04:25 (UTC) |
| production1 | mysql | backup   | hourly-e7db6-20240815190300 | 2024-
08-15 19:03:30 (UTC) |
| production1 | mysql | backup   | hourly-e7db6-20240815200300 | 2024-
08-15 20:04:21 (UTC) |
| production1 | mysql | backup   | mybackup5                   | 2024-
08-09 22:25:13 (UTC) |
|             | mysql | backup   | mybackup                    | 2024-
08-09 21:02:52 (UTC) |
+-------------+-------+----------+-----------------------------
+--------------------------+
```

2. Opcionalmente, para ver la ruta de AppVaultPath de cada recurso, utilice el indicador `--show-paths` .

   El nombre del clúster en la primera columna de la tabla solo está disponible si se especificó un nombre de clúster en la instalación de Trident Protect Helm. Por ejemplo: `--set clusterName=production1` .

Eliminar un AppVault

Puedes eliminar un objeto de AppVault en cualquier momento.

> No retire el `finalizers` Introduzca la clave en el registro de cambios de AppVault antes de eliminar el objeto de AppVault. Si lo hace, puede provocar que queden datos residuales en el bucket de AppVault y recursos huérfanos en el clúster.

Antes de empezar

Asegúrese de haber eliminado todos los registros CR de instantáneas y copias de seguridad que utiliza el AppVault que desea eliminar.

Eliminar un AppVault mediante la CLI de Kubernetes

1. Elimine el objeto AppVault y reemplácelo. `appvault-name` con el nombre del objeto de AppVault que se va a eliminar:

```
kubectl delete appvault <appvault-name> \
-n trident-protect
```

Eliminar un AppVault mediante la CLI de Trident Protect

1. Elimine el objeto AppVault y reemplácelo. `appvault-name` con el nombre del objeto de AppVault que se va a eliminar:

```
tridentctl-protect delete appvault <appvault-name> \
-n trident-protect
```

Definir una aplicación para la gestión con Trident Protect

Puede definir una aplicación que desee administrar con Trident Protect creando un CR de aplicación y un CR de AppVault asociado.

Crear un CR de AppVault

Debe crear un CR de AppVault que se utilizará al realizar operaciones de protección de datos en la aplicación, y el CR de AppVault debe residir en el clúster donde está instalado Trident Protect. La solicitud de cambio (CR) de AppVault es específica de su entorno; para ver ejemplos de solicitudes de cambio de cambio de AppVault, consulte"Recursos personalizados de AppVault."

Definir una aplicación

Debe definir cada aplicación que desee administrar con Trident Protect. Puede definir una aplicación para su administración ya sea creando manualmente un CR de aplicación o utilizando la CLI de Trident Protect.

Agregar una aplicación usando un CR

Pasos

1. Cree el archivo CR de la aplicación de destino:
   a. Cree el archivo de recursos personalizados (CR) y asígnele un nombre (por ejemplo, `maria-app.yaml`).
   b. Configure los siguientes atributos:
      - **metadata.name**: (*Obligatorio*) El nombre del recurso personalizado de la aplicación. Anote el nombre que elija, ya que otros archivos CR necesarios para las operaciones de protección hacen referencia a este valor.
      - **spec.includedNamespaces**: (*Obligatorio*) Utilice el selector de espacio de nombres y etiqueta para especificar los espacios de nombres y recursos que utiliza la aplicación. El espacio de nombres de la aplicación debe formar parte de esta lista. El selector de etiquetas es opcional y se puede utilizar para filtrar recursos dentro de cada espacio de nombres especificado.
      - **spec.includedClusterScopedResources**: (*Opcional*) Utilice este atributo para especificar los recursos con ámbito de clúster que se incluirán en la definición de la aplicación. Este atributo le permite seleccionar estos recursos en función de su grupo, versión, tipo y etiquetas.
        - **groupVersionKind**: (*Obligatorio*) Especifica el grupo de API, la versión y el tipo del recurso con ámbito de clúster.
        - **labelSelector**: (*Opcional*) Filtra los recursos con ámbito de clúster en función de sus etiquetas.
      - **metadata.annotations.protect.trident.netapp.io/skip-vm-freeze**: (*Opcional*) Esta anotación solo es aplicable a aplicaciones definidas desde máquinas virtuales, como en entornos KubeVirt, donde las congelaciones del sistema de archivos ocurren antes de las instantáneas. Especifique si esta aplicación puede escribir en el sistema de archivos durante una instantánea. Si se establece en verdadero, la aplicación ignora la configuración global y puede escribir en el sistema de archivos durante una instantánea. Si se establece en falso, la aplicación ignora la configuración global y el sistema de archivos se congela durante una instantánea. Si se especifica pero la aplicación no tiene máquinas virtuales en la definición de la aplicación, la anotación se ignora. Si no se especifica, la aplicación sigue el procedimiento establecido."Configuración de congelación global de Trident Protect" .

Si necesita aplicar esta anotación después de que ya se haya creado una aplicación, puede utilizar el siguiente comando:

```
kubectl annotate application -n <application CR namespace> <application CR name> protect.trident.netapp.io/skip-vm-freeze="true"
```

+
Ejemplo de YAML:

+

```
apiVersion: protect.trident.netapp.io/v1
kind: Application
metadata:
  annotations:
    protect.trident.netapp.io/skip-vm-freeze: "false"
  name: my-app-name
  namespace: my-app-namespace
spec:
  includedNamespaces:
    - namespace: namespace-1
      labelSelector:
        matchLabels:
          app: example-app
    - namespace: namespace-2
      labelSelector:
        matchLabels:
          app: another-example-app
  includedClusterScopedResources:
    - groupVersionKind:
        group: rbac.authorization.k8s.io
        kind: ClusterRole
        version: v1
      labelSelector:
        matchLabels:
          mylabel: test
```

1. (Opcional) Agregue filtros que incluyan o excluyan recursos marcados con etiquetas específicas:

   - **resourceFilter.resourceSelectionCriteria**: (Obligatorio para filtrar) Usar `Include` o `Exclude` para incluir o excluir un recurso definido en resourceMatchers. Agregue los siguientes parámetros resourceMatchers para definir los recursos que se incluirán o excluirán:

     - **resourceFilter.resourceMatchers**: Una matriz de objetos resourceMatcher. Si defines varios elementos en esta matriz, coinciden como una operación OR, y los campos dentro de cada elemento (grupo, tipo, versión) coinciden como una operación AND.
       - **resourceMatchers[].group**: (*Opcional*) Grupo del recurso a filtrar.
       - **resourceMatchers[].kind**: (*Opcional*) Tipo de recurso a filtrar.
       - **resourceMatchers[].version**: (*Opcional*) Versión del recurso a filtrar.
       - **resourceMatchers[].names**: (*Opcional*) Nombres en el campo metadata.name de Kubernetes del recurso que se va a filtrar.

- **resourceMatchers[].namespaces**: (*Opcional*) Espacios de nombres en el campo metadata.name de Kubernetes del recurso que se va a filtrar.
- **resourceMatchers[].labelSelectors**: (*Opcional*) Cadena de selector de etiquetas en el campo metadata.name de Kubernetes del recurso, tal como se define en el "Documentación de Kubernetes" . Por ejemplo: `"trident.netapp.io/os=linux"` .

> ⓘ Cuando ambos `resourceFilter` y `labelSelector` se utilizan, `resourceFilter` Primero corre, y luego `labelSelector` se aplica a los recursos resultantes.

Por ejemplo:

```
spec:
  resourceFilter:
    resourceSelectionCriteria: "Include"
    resourceMatchers:
      - group: my-resource-group-1
        kind: my-resource-kind-1
        version: my-resource-version-1
        names: ["my-resource-names"]
        namespaces: ["my-resource-namespaces"]
        labelSelectors: ["trident.netapp.io/os=linux"]
      - group: my-resource-group-2
        kind: my-resource-kind-2
        version: my-resource-version-2
        names: ["my-resource-names"]
        namespaces: ["my-resource-namespaces"]
        labelSelectors: ["trident.netapp.io/os=linux"]
```

2. Después de crear la solicitud de cambio (CR) de la aplicación para que coincida con su entorno, aplique la CR. Por ejemplo:

```
kubectl apply -f maria-app.yaml
```

Pasos

1. Cree y aplique la definición de la aplicación utilizando uno de los siguientes ejemplos, reemplazando los valores entre corchetes con información de su entorno. Puede incluir espacios de nombres y recursos en la definición de la aplicación utilizando listas separadas por comas con los argumentos que se muestran en los ejemplos.

   Opcionalmente, puede usar una anotación al crear una aplicación para especificar si la aplicación puede escribir en el sistema de archivos durante una instantánea. Esto solo es aplicable a aplicaciones definidas desde máquinas virtuales, como en entornos KubeVirt, donde se producen congelaciones del sistema de archivos antes de las instantáneas. Si configura la anotación a `true` La aplicación ignora la configuración global y puede escribir en el sistema de archivos durante una instantánea. Si lo configuras para `false` La aplicación ignora la configuración global y el sistema de

archivos se congela durante la toma de una instantánea. Si utiliza la anotación pero la aplicación no tiene máquinas virtuales en la definición de la aplicación, la anotación se ignora. Si no utilizas la anotación, la aplicación sigue el comportamiento esperado."Configuración de congelación global de Trident Protect" .

Para especificar la anotación cuando utilice la CLI para crear una aplicación, puede usar la `--annotation` bandera.

- Cree la aplicación y utilice la configuración global para el comportamiento de congelación del sistema de archivos:

```
tridentctl-protect create application <my_new_app_cr_name>
--namespaces <namespaces_to_include> --csr
<cluster_scoped_resources_to_include> --namespace <my-app-
namespace>
```

- Cree la aplicación y configure los ajustes de la aplicación local para el comportamiento de congelación del sistema de archivos:

```
tridentctl-protect create application <my_new_app_cr_name>
--namespaces <namespaces_to_include> --csr
<cluster_scoped_resources_to_include> --namespace <my-app-
namespace> --annotation protect.trident.netapp.io/skip-vm-freeze
=<"true"|"false">
```

Puedes utilizar `--resource-filter-include` y `--resource-filter-exclude` indicadores para incluir o excluir recursos en función de `resourceSelectionCriteria` como por ejemplo: grupo, tipo, versión, etiquetas, nombres y espacios de nombres, como se muestra en el siguiente ejemplo:

```
tridentctl-protect create application <my_new_app_cr_name>
--namespaces <namespaces_to_include> --csr
<cluster_scoped_resources_to_include> --namespace <my-app-namespace>
--resource-filter-include
'[{"Group":"apps","Kind":"Deployment","Version":"v1","Names":["my-
deployment"],"Namespaces":["my-
namespace"],"LabelSelectors":["app=my-app"]}]'
```

Proteja las aplicaciones con Trident Protect

Puede proteger todas las aplicaciones administradas por Trident Protect tomando instantáneas y copias de seguridad mediante una política de protección automatizada o de forma ad hoc.

Puede configurar Trident Protect para congelar y descongelar sistemas de archivos durante operaciones de protección de datos. "Obtenga más información sobre cómo configurar la congelación del sistema de archivos con Trident Protect".

Crea una instantánea bajo demanda

Puedes crear una instantánea bajo demanda en cualquier momento.

Los recursos con ámbito de clúster se incluyen en una copia de seguridad, instantánea o clon si se hace referencia a ellos explícitamente en la definición de la aplicación o si tienen referencias a cualquiera de los espacios de nombres de la aplicación.

Cree una instantánea utilizando un CR

Pasos

1. Crea el archivo de recursos personalizados (CR) y nómbralo. `trident-protect-snapshot-cr.yaml` .
2. En el archivo que creó, configure los siguientes atributos:
   - **metadata.name**: (*Obligatorio*) El nombre de este recurso personalizado; elija un nombre único y adecuado para su entorno.
   - **spec.applicationRef**: El nombre de Kubernetes de la aplicación a la que se va a crear una instantánea.
   - **spec.appVaultRef**: (*Obligatorio*) El nombre del AppVault donde se deben almacenar los contenidos de la instantánea (metadatos).
   - **spec.reclaimPolicy**: (*Opcional*) Define qué sucede con el AppArchive de una instantánea cuando se elimina el CR de la instantánea. Esto significa que incluso cuando se configura para `Retain` La instantánea será eliminada. Opciones válidas:
     - `Retain`(por defecto)
     - `Delete`

     ```
     ---
     apiVersion: protect.trident.netapp.io/v1
     kind: Snapshot
     metadata:
       namespace: my-app-namespace
       name: my-cr-name
     spec:
       applicationRef: my-application
       appVaultRef: appvault-name
       reclaimPolicy: Delete
     ```
3. Después de rellenar el `trident-protect-snapshot-cr.yaml` Archivo con los valores correctos, aplicar el CR:

   ```
   kubectl apply -f trident-protect-snapshot-cr.yaml
   ```

Crea una instantánea usando la CLI.

Pasos

1. Crea la instantánea, reemplazando los valores entre corchetes con información de tu entorno. Por ejemplo:

   ```
   tridentctl-protect create snapshot <my_snapshot_name> --appvault
   <my_appvault_name> --app <name_of_app_to_snapshot> -n
   <application_namespace>
   ```

Crea una copia de seguridad bajo demanda

Puedes realizar una copia de seguridad de una aplicación en cualquier momento.

> Los recursos con ámbito de clúster se incluyen en una copia de seguridad, instantánea o clon si se hace referencia a ellos explícitamente en la definición de la aplicación o si tienen referencias a cualquiera de los espacios de nombres de la aplicación.

Antes de empezar

Asegúrese de que la expiración del token de sesión de AWS sea suficiente para cualquier operación de copia de seguridad de S3 de larga duración. Si el token caduca durante la operación de copia de seguridad, la operación puede fallar.

- Consulte el "Documentación de AWS API" Para obtener más información sobre cómo comprobar la caducidad del token de sesión actual.
- Consulte el "Documentación de AWS IAM" Para obtener más información sobre las credenciales con los recursos de AWS.

Cree una copia de seguridad utilizando un CR.

Pasos

1. Crea el archivo de recursos personalizados (CR) y nómbralo. `trident-protect-backup-cr.yaml` .
2. En el archivo que creó, configure los siguientes atributos:
   - **metadata.name**: (*Obligatorio*) El nombre de este recurso personalizado; elija un nombre único y adecuado para su entorno.
   - **spec.applicationRef**: (*Obligatorio*) El nombre de Kubernetes de la aplicación de la que se hará la copia de seguridad.
   - **spec.appVaultRef**: (*Obligatorio*) El nombre del AppVault donde se debe almacenar el contenido de la copia de seguridad.
   - **spec.dataMover**: (*Opcional*) Una cadena que indica qué herramienta de copia de seguridad se utilizará para la operación de copia de seguridad. Valores posibles (distingue entre mayúsculas y minúsculas):
     - `Restic`
     - `Kopia`(por defecto)
   - **spec.reclaimPolicy**: (*Opcional*) Define qué sucede con una copia de seguridad cuando se libera de su reclamación. Valores posibles:
     - `Delete`
     - `Retain`(por defecto)
   - **spec.snapshotRef**: (*Opcional*): Nombre de la instantánea que se utilizará como origen de la copia de seguridad. Si no se proporciona, se creará y copiará una instantánea temporal.
   - **metadata.annotations.protect.trident.netapp.io/full-backup** : (*Opcional*) Esta anotación se utiliza para especificar si una copia de seguridad no debe ser incremental. De forma predeterminada, todas las copias de seguridad son incrementales. Sin embargo, si esta anotación está configurada para `true` , la copia de seguridad deja de ser incremental. Si no se especifica, la copia de seguridad sigue la configuración de copia de seguridad incremental predeterminada. La mejor práctica es realizar una copia de seguridad completa periódicamente y luego realizar copias de seguridad incrementales entre las copias de seguridad completas para minimizar el riesgo asociado con las restauraciones.

     Ejemplo de YAML:

```
---
apiVersion: protect.trident.netapp.io/v1
kind: Backup
metadata:
  namespace: my-app-namespace
  name: my-cr-name
  annotations:
    protect.trident.netapp.io/full-backup: "true"
spec:
  applicationRef: my-application
  appVaultRef: appvault-name
  dataMover: Kopia
```

3. Después de rellenar el `trident-protect-backup-cr.yaml` Archivo con los valores correctos, aplicar el CR:

```
kubectl apply -f trident-protect-backup-cr.yaml
```

Crea una copia de seguridad usando la CLI.

Pasos

1. Crea la copia de seguridad, reemplazando los valores entre corchetes con información de tu entorno. Por ejemplo:

```
tridentctl-protect create backup <my_backup_name> --appvault <my-
vault-name> --app <name_of_app_to_back_up> --data-mover
<Kopia_or_Restic> -n <application_namespace>
```

   Opcionalmente, puede utilizar el `--full-backup` Bandera para especificar si una copia de seguridad debe ser no incremental. De forma predeterminada, todas las copias de seguridad son incrementales. Cuando se utiliza esta bandera, la copia de seguridad deja de ser incremental. La mejor práctica es realizar una copia de seguridad completa periódicamente y luego realizar copias de seguridad incrementales entre las copias de seguridad completas para minimizar el riesgo asociado con las restauraciones.

Crea un calendario de protección de datos.

Una política de protección protege una aplicación mediante la creación de instantáneas, copias de seguridad o ambas según un cronograma definido. Puede elegir crear instantáneas y copias de seguridad cada hora, día, semana o mes, y puede especificar la cantidad de copias que desea conservar. Puede programar una copia de seguridad completa no incremental mediante la anotación full-backup-rule. De forma predeterminada, todas las copias de seguridad son incrementales. Realizar una copia de seguridad completa periódicamente, junto con copias de seguridad incrementales entre ellas, ayuda a reducir el riesgo asociado con las restauraciones.

- Solo puedes crear programaciones para instantáneas configurando `backupRetention` a cero y `snapshotRetention` a un valor mayor que cero. Configuración `snapshotRetention` A cero significa que cualquier copia de seguridad programada seguirá creando instantáneas, pero estas son temporales y se eliminan inmediatamente después de que se complete la copia de seguridad.
- Los recursos con ámbito de clúster se incluyen en una copia de seguridad, instantánea o clon si se hace referencia a ellos explícitamente en la definición de la aplicación o si tienen referencias a cualquiera de los espacios de nombres de la aplicación.

Crea un cronograma usando un CR

Pasos

1. Crea el archivo de recursos personalizados (CR) y nómbralo. `trident-protect-schedule-cr.yaml` .
2. En el archivo que creó, configure los siguientes atributos:
   - **metadata.name**: (*Obligatorio*) El nombre de este recurso personalizado; elija un nombre único y adecuado para su entorno.
   - **spec.dataMover**: (*Opcional*) Una cadena que indica qué herramienta de copia de seguridad se utilizará para la operación de copia de seguridad. Valores posibles (distingue entre mayúsculas y minúsculas):
     - `Restic`
     - `Kopia`(por defecto)
   - **spec.applicationRef**: El nombre de Kubernetes de la aplicación de la que se realizará la copia de seguridad.
   - **spec.appVaultRef**: (*Obligatorio*) El nombre del AppVault donde se debe almacenar el contenido de la copia de seguridad.
   - **spec.backupRetention**: La cantidad de copias de seguridad que se conservarán. Cero indica que no se deben crear copias de seguridad (solo instantáneas).
   - **spec.snapshotRetention**: El número de instantáneas que se conservarán. El valor cero indica que no se deben crear instantáneas.
   - **spec.granularity**: La frecuencia con la que debe ejecutarse la programación. Valores posibles, junto con los campos asociados obligatorios:
     - `Hourly`(requiere que especifique `spec.minute`)
     - `Daily`(requiere que especifique `spec.minute` y `spec.hour`)
     - `Weekly`(requiere que especifique `spec.minute, spec.hour` , y `spec.dayOfWeek`)
     - `Monthly`(requiere que especifique `spec.minute, spec.hour` , y `spec.dayOfMonth`)
     - `Custom`
   - **spec.dayOfMonth**: (*Opcional*) El día del mes (1 - 31) en que debe ejecutarse la programación. Este campo es obligatorio si la granularidad está configurada en `Monthly` . El valor debe proporcionarse como una cadena.
   - **spec.dayOfWeek**: (*Opcional*) El día de la semana (0 - 7) en que debe ejecutarse la programación. Los valores de 0 o 7 indican domingo. Este campo es obligatorio si la granularidad está configurada en `Weekly` . El valor debe proporcionarse como una cadena.
   - **spec.hour**: (*Opcional*) La hora del día (0 - 23) en que debe ejecutarse el programa. Este campo es obligatorio si la granularidad está configurada en `Daily` , `Weekly` , o `Monthly` . El valor debe proporcionarse como una cadena.
   - **spec.minute**: (*Opcional*) El minuto de la hora (0 - 59) en que debe ejecutarse el programa. Este campo es obligatorio si la granularidad está configurada en `Hourly` , `Daily` , `Weekly` , o `Monthly` . El valor debe proporcionarse como una cadena.
   - **metadata.annotations.protect.trident.netapp.io/full-backup-rule**: (*Opcional*) Esta anotación se utiliza para especificar la regla para programar la copia de seguridad completa. Puedes

configurarlo para `always` para copias de seguridad completas constantes o personalícelo según sus necesidades. Por ejemplo, si elige una granularidad diaria, puede especificar los días de la semana en los que se debe realizar la copia de seguridad completa.

Ejemplo de YAML para programación de copias de seguridad y instantáneas:

```
---
apiVersion: protect.trident.netapp.io/v1
kind: Schedule
metadata:
  namespace: my-app-namespace
  name: my-cr-name
  annotations:
    protect.trident.netapp.io/full-backup-rule: "Monday,Thursday"
spec:
  dataMover: Kopia
  applicationRef: my-application
  appVaultRef: appvault-name
  backupRetention: "15"
  snapshotRetention: "15"
  granularity: Daily
  hour: "0"
  minute: "0"
```

Ejemplo de YAML para una programación de solo instantáneas:

```
---
apiVersion: protect.trident.netapp.io/v1
kind: Schedule
metadata:
  namespace: my-app-namespace
  name: my-snapshot-schedule
spec:
  applicationRef: my-application
  appVaultRef: appvault-name
  backupRetention: "0"
  snapshotRetention: "15"
  granularity: Daily
  hour: "2"
  minute: "0"
```

3. Después de rellenar el `trident-protect-schedule-cr.yaml` Archivo con los valores correctos, aplicar el CR:

```
kubectl apply -f trident-protect-schedule-cr.yaml
```

Crea una programación usando la CLI.

Pasos

1. Cree el plan de protección, reemplazando los valores entre paréntesis con información de su entorno. Por ejemplo:

   Puedes utilizar `tridentctl-protect create schedule --help` Para ver información de ayuda detallada sobre este comando.

   ```
   tridentctl-protect create schedule <my_schedule_name> --appvault
   <my_appvault_name> --app <name_of_app_to_snapshot> --backup
   -retention <how_many_backups_to_retain> --data-mover
   <Kopia_or_Restic> --day-of-month <day_of_month_to_run_schedule>
   --day-of-week <day_of_month_to_run_schedule> --granularity
   <frequency_to_run> --hour <hour_of_day_to_run> --minute
   <minute_of_hour_to_run> --recurrence-rule <recurrence> --snapshot
   -retention <how_many_snapshots_to_retain> -n <application_namespace>
   --full-backup-rule <string>
   ```

   Puedes configurarlo `--full-backup-rule` bandera a `always` para copias de seguridad completas constantes o personalícelo según sus necesidades. Por ejemplo, si elige una granularidad diaria, puede especificar los días de la semana en los que debe realizarse la copia de seguridad completa. Por ejemplo, utilice `--full-backup-rule "Monday,Thursday"` Programar copias de seguridad completas los lunes y jueves.

   Para programaciones solo con instantáneas, configure `--backup-retention 0` y especifique un valor mayor que 0 para `--snapshot-retention` .

Eliminar una instantánea

Elimine las instantáneas programadas o bajo demanda que ya no necesite.

Pasos

1. Eliminar el CR de instantánea asociado con la instantánea:

   ```
   kubectl delete snapshot <snapshot_name> -n my-app-namespace
   ```

Eliminar una copia de seguridad

Elimine las copias de seguridad programadas o bajo demanda que ya no necesite.

Asegúrese de que la política de reclamaciones esté configurada para `Delete` Eliminar todos los datos de copia de seguridad del almacenamiento de objetos. La configuración predeterminada de la política es `Retain` para evitar la pérdida accidental de datos. Si la política no se modifica a `Delete` Los datos de copia de seguridad permanecerán en el almacenamiento de objetos y requerirán eliminación manual.

Pasos

1. Eliminar el CR de copia de seguridad asociado con la copia de seguridad:

```
kubectl delete backup <backup_name> -n my-app-namespace
```

Comprobar el estado de una operación de copia de seguridad

Puede utilizar la línea de comandos para comprobar el estado de una operación de copia de seguridad que esté en curso, haya finalizado o haya fallado.

Pasos

1. Utilice el siguiente comando para recuperar el estado de la operación de copia de seguridad, reemplazando los valores entre corchetes con información de su entorno:

```
kubectl get backup -n <namespace_name> <my_backup_cr_name> -o jsonpath
='{.status}'
```

Habilitar la copia de seguridad y la restauración para las operaciones de archivos de Azure NetApp (ANF).

Si ha instalado Trident Protect, puede habilitar la funcionalidad de copia de seguridad y restauración que ahorra espacio para los backends de almacenamiento que usan la clase de almacenamiento azure-netapp-files y se crearon antes de Trident 24.06. Esta funcionalidad funciona con volúmenes NFSv4 y no consume espacio adicional del grupo de capacidad.

Antes de empezar

Asegúrese de lo siguiente:

- Ha instalado Trident Protect.
- Ha definido una aplicación en Trident Protect. Esta aplicación tendrá una funcionalidad de protección limitada hasta que complete este procedimiento.
- Tienes `azure-netapp-files` seleccionada como la clase de almacenamiento predeterminada para su backend de almacenamiento.

Expandir para ver los pasos de configuración

1. Realice lo siguiente en Trident si el volumen ANF se creó antes de actualizar a Trident 24.10:

   a. Habilite el directorio de instantáneas para cada PV que esté basado en azure-netapp-files y asociado con la aplicación:

   ```
   tridentctl update volume <pv name> --snapshot-dir=true -n trident
   ```

   b. Confirme que el directorio de instantáneas se ha habilitado para cada PV asociado:

   ```
   tridentctl get volume <pv name> -n trident -o yaml | grep
   snapshotDir
   ```

   Respuesta:

   ```
   snapshotDirectory: "true"
   ```

   +
   Cuando el directorio de instantáneas no está habilitado, se elige la funcionalidad de copia de seguridad regular, que consume temporalmente espacio en el grupo de capacidad durante el proceso de copia de seguridad. En este caso, asegúrese de que haya suficiente espacio disponible en el grupo de capacidad para crear un volumen temporal del tamaño del volumen del que se está realizando la copia de seguridad.

Resultado

La aplicación está lista para realizar copias de seguridad y restaurar mediante Trident Protect. Cada PVC también está disponible para ser utilizado por otras aplicaciones para copias de seguridad y restauraciones.

Restaurar aplicaciones

Restaurar aplicaciones usando Trident Protect

Puede utilizar Trident Protect para restaurar su aplicación desde una instantánea o una copia de seguridad. Restaurar desde una instantánea existente será más rápido al restaurar la aplicación en el mismo clúster.

- Cuando se restaura una aplicación, todos los ganchos de ejecución configurados para la aplicación se restauran con la aplicación. Si existe un gancho de ejecución posterior a la restauración, se ejecuta automáticamente como parte de la operación de restauración.
- Se admite la restauración desde una copia de seguridad a un espacio de nombres diferente o al espacio de nombres original para volúmenes qtree. Sin embargo, la restauración desde una instantánea a un espacio de nombres diferente o al espacio de nombres original no es compatible con los volúmenes qtree.
- Puede utilizar la configuración avanzada para personalizar las operaciones de restauración. Para obtener más información, consulte "Utilice la configuración de restauración avanzada de Trident Protect".

Restaurar desde una copia de seguridad a un espacio de nombres diferente

Cuando restaura una copia de seguridad en un espacio de nombres diferente mediante un CR de BackupRestore, Trident Protect restaura la aplicación en un nuevo espacio de nombres y crea un CR de aplicación para la aplicación restaurada. Para proteger la aplicación restaurada, cree copias de seguridad o instantáneas bajo demanda, o establezca un programa de protección.

Restaurar una copia de seguridad en un espacio de nombres diferente con recursos existentes no alterará ningún recurso que comparta nombres con los de la copia de seguridad. Para restaurar todos los recursos de la copia de seguridad, elimine y vuelva a crear el espacio de nombres de destino, o restaure la copia de seguridad en un nuevo espacio de nombres.

Antes de empezar

Asegúrese de que la expiración del token de sesión de AWS sea suficiente para cualquier operación de restauración de S3 de larga duración. Si el token caduca durante la operación de restauración, la operación puede fallar.

- Consulte el "Documentación de AWS API" Para obtener más información sobre cómo comprobar la caducidad del token de sesión actual.
- Consulte el "Documentación de AWS IAM" Para obtener más información sobre las credenciales con los recursos de AWS.

Al restaurar copias de seguridad utilizando Kopia como transportador de datos, puede especificar opcionalmente anotaciones en la CR o usar la CLI para controlar el comportamiento del almacenamiento temporal utilizado por Kopia. Consulte el "Documentación de Kopia" Para obtener más información sobre las opciones que puede configurar. Utilice el `tridentctl-protect create --help` Comando para obtener más información sobre cómo especificar anotaciones con la CLI de Trident Protect.

Utilice un CR

Pasos

1. Crea el archivo de recursos personalizados (CR) y nómbralo. `trident-protect-backup-restore-cr.yaml` .
2. En el archivo que creó, configure los siguientes atributos:
   - **metadata.name**: (*Obligatorio*) El nombre de este recurso personalizado; elija un nombre único y adecuado para su entorno.
   - **spec.appArchivePath**: La ruta dentro de AppVault donde se almacenan los contenidos de la copia de seguridad. Puedes utilizar el siguiente comando para encontrar esta ruta:

     ```
     kubectl get backups <BACKUP_NAME> -n my-app-namespace -o
     jsonpath='{.status.appArchivePath}'
     ```

   - **spec.appVaultRef**: (*Obligatorio*) El nombre del AppVault donde se almacenan los contenidos de la copia de seguridad.
   - **spec.namespaceMapping**: La asignación del espacio de nombres de origen de la operación de restauración al espacio de nombres de destino. Reemplazar `my-source-namespace` y `my-destination-namespace` con información de tu entorno.

     ```
     ---
     apiVersion: protect.trident.netapp.io/v1
     kind: BackupRestore
     metadata:
       name: my-cr-name
       namespace: my-destination-namespace
     spec:
       appArchivePath: my-backup-path
       appVaultRef: appvault-name
       namespaceMapping: [{"source": "my-source-namespace",
     "destination": "my-destination-namespace"}]
     ```

3. (Opcional) Si necesita seleccionar solo ciertos recursos de la aplicación para restaurar, agregue un filtro que incluya o excluya los recursos marcados con etiquetas particulares:

   > Trident Protect selecciona algunos recursos automáticamente debido a su relación con los recursos que usted selecciona. Por ejemplo, si selecciona un recurso de reclamo de volumen persistente y tiene un pod asociado, Trident Protect también restaurará el pod asociado.

   - **resourceFilter.resourceSelectionCriteria**: (Obligatorio para filtrar) Usar `Include` o `Exclude` para incluir o excluir un recurso definido en resourceMatchers. Agregue los siguientes parámetros resourceMatchers para definir los recursos que se incluirán o excluirán:
     - **resourceFilter.resourceMatchers**: Una matriz de objetos resourceMatcher. Si defines varios elementos en esta matriz, coinciden como una operación OR, y los campos dentro de cada

elemento (grupo, tipo, versión) coinciden como una operación AND.

- **resourceMatchers[].group**: (*Opcional*) Grupo del recurso a filtrar.
- **resourceMatchers[].kind**: (*Opcional*) Tipo de recurso a filtrar.
- **resourceMatchers[].version**: (*Opcional*) Versión del recurso a filtrar.
- **resourceMatchers[].names**: (*Opcional*) Nombres en el campo metadata.name de Kubernetes del recurso que se va a filtrar.
- **resourceMatchers[].namespaces**: (*Opcional*) Espacios de nombres en el campo metadata.name de Kubernetes del recurso que se va a filtrar.
- **resourceMatchers[].labelSelectors**: (*Opcional*) Cadena de selector de etiquetas en el campo metadata.name de Kubernetes del recurso, tal como se define en el "Documentación de Kubernetes" . Por ejemplo: `"trident.netapp.io/os=linux"` .

  Por ejemplo:

```
spec:
  resourceFilter:
    resourceSelectionCriteria: "Include"
    resourceMatchers:
      - group: my-resource-group-1
        kind: my-resource-kind-1
        version: my-resource-version-1
        names: ["my-resource-names"]
        namespaces: ["my-resource-namespaces"]
        labelSelectors: ["trident.netapp.io/os=linux"]
      - group: my-resource-group-2
        kind: my-resource-kind-2
        version: my-resource-version-2
        names: ["my-resource-names"]
        namespaces: ["my-resource-namespaces"]
        labelSelectors: ["trident.netapp.io/os=linux"]
```

4. Después de rellenar el `trident-protect-backup-restore-cr.yaml` Archivo con los valores correctos, aplicar el CR:

```
kubectl apply -f trident-protect-backup-restore-cr.yaml
```

Utilice la CLI

Pasos

1. Restaure la copia de seguridad en un espacio de nombres diferente, reemplazando los valores entre corchetes con información de su entorno. El `namespace-mapping` El argumento utiliza espacios de nombres separados por dos puntos para asignar los espacios de nombres de origen a los espacios de nombres de destino correctos en el formato `source1:dest1,source2:dest2` . Por ejemplo:

```
tridentctl-protect create backuprestore <my_restore_name> \
--backup <backup_namespace>/<backup_to_restore> \
--namespace-mapping <source_to_destination_namespace_mapping> \
-n <application_namespace>
```

Restaurar desde una copia de seguridad al espacio de nombres original

Puedes restaurar una copia de seguridad al espacio de nombres original en cualquier momento.

Antes de empezar

Asegúrese de que la expiración del token de sesión de AWS sea suficiente para cualquier operación de restauración de S3 de larga duración. Si el token caduca durante la operación de restauración, la operación puede fallar.

- Consulte el "Documentación de AWS API" Para obtener más información sobre cómo comprobar la caducidad del token de sesión actual.
- Consulte el "Documentación de AWS IAM" Para obtener más información sobre las credenciales con los recursos de AWS.

> Al restaurar copias de seguridad utilizando Kopia como transportador de datos, puede especificar opcionalmente anotaciones en la CR o usar la CLI para controlar el comportamiento del almacenamiento temporal utilizado por Kopia. Consulte el "Documentación de Kopia" Para obtener más información sobre las opciones que puede configurar. Utilice el `tridentctl-protect create --help` Comando para obtener más información sobre cómo especificar anotaciones con la CLI de Trident Protect.

Utilice un CR

Pasos

1. Crea el archivo de recursos personalizados (CR) y nómbralo. `trident-protect-backup-ipr-cr.yaml` .
2. En el archivo que creó, configure los siguientes atributos:
   - **metadata.name**: (*Obligatorio*) El nombre de este recurso personalizado; elija un nombre único y adecuado para su entorno.
   - **spec.appArchivePath**: La ruta dentro de AppVault donde se almacenan los contenidos de la copia de seguridad. Puedes utilizar el siguiente comando para encontrar esta ruta:

     ```
     kubectl get backups <BACKUP_NAME> -n my-app-namespace -o jsonpath='{.status.appArchivePath}'
     ```

   - **spec.appVaultRef**: (*Obligatorio*) El nombre del AppVault donde se almacenan los contenidos de la copia de seguridad.

     Por ejemplo:

```
---
apiVersion: protect.trident.netapp.io/v1
kind: BackupInplaceRestore
metadata:
  name: my-cr-name
  namespace: my-app-namespace
spec:
  appArchivePath: my-backup-path
  appVaultRef: appvault-name
```

3. (Opcional) Si necesita seleccionar solo ciertos recursos de la aplicación para restaurar, agregue un filtro que incluya o excluya los recursos marcados con etiquetas particulares:

   > Trident Protect selecciona algunos recursos automáticamente debido a su relación con los recursos que usted selecciona. Por ejemplo, si selecciona un recurso de reclamo de volumen persistente y tiene un pod asociado, Trident Protect también restaurará el pod asociado.

   - **resourceFilter.resourceSelectionCriteria**: (Obligatorio para filtrar) Usar `Include` o `Exclude` para incluir o excluir un recurso definido en resourceMatchers. Agregue los siguientes parámetros resourceMatchers para definir los recursos que se incluirán o excluirán:
     - **resourceFilter.resourceMatchers**: Una matriz de objetos resourceMatcher. Si defines varios elementos en esta matriz, coinciden como una operación OR, y los campos dentro de cada elemento (grupo, tipo, versión) coinciden como una operación AND.
       - **resourceMatchers[].group**: (*Opcional*) Grupo del recurso a filtrar.
       - **resourceMatchers[].kind**: (*Opcional*) Tipo de recurso a filtrar.

- **resourceMatchers[].version**: (*Opcional*) Versión del recurso a filtrar.
- **resourceMatchers[].names**: (*Opcional*) Nombres en el campo metadata.name de Kubernetes del recurso que se va a filtrar.
- **resourceMatchers[].namespaces**: (*Opcional*) Espacios de nombres en el campo metadata.name de Kubernetes del recurso que se va a filtrar.
- **resourceMatchers[].labelSelectors**: (*Opcional*) Cadena de selector de etiquetas en el campo metadata.name de Kubernetes del recurso, tal como se define en el "Documentación de Kubernetes" . Por ejemplo: `"trident.netapp.io/os=linux"` .

  Por ejemplo:

```
spec:
  resourceFilter:
    resourceSelectionCriteria: "Include"
    resourceMatchers:
      - group: my-resource-group-1
        kind: my-resource-kind-1
        version: my-resource-version-1
        names: ["my-resource-names"]
        namespaces: ["my-resource-namespaces"]
        labelSelectors: ["trident.netapp.io/os=linux"]
      - group: my-resource-group-2
        kind: my-resource-kind-2
        version: my-resource-version-2
        names: ["my-resource-names"]
        namespaces: ["my-resource-namespaces"]
        labelSelectors: ["trident.netapp.io/os=linux"]
```

4. Después de rellenar el `trident-protect-backup-ipr-cr.yaml` Archivo con los valores correctos, aplicar el CR:

```
kubectl apply -f trident-protect-backup-ipr-cr.yaml
```

Utilice la CLI

Pasos

1. Restaure la copia de seguridad al espacio de nombres original, reemplazando los valores entre corchetes con información de su entorno. El `backup` El argumento utiliza un espacio de nombres y un nombre de respaldo en el formato `<namespace>/<name>` . Por ejemplo:

```
tridentctl-protect create backupinplacerestore <my_restore_name> \
--backup <namespace/backup_to_restore> \
-n <application_namespace>
```

Restaurar desde una copia de seguridad a un clúster diferente

Puedes restaurar una copia de seguridad en un clúster diferente si hay algún problema con el clúster original.

> Al restaurar copias de seguridad utilizando Kopia como transportador de datos, puede especificar opcionalmente anotaciones en la CR o usar la CLI para controlar el comportamiento del almacenamiento temporal utilizado por Kopia. Consulte el "Documentación de Kopia" Para obtener más información sobre las opciones que puede configurar. Utilice el `tridentctl-protect create --help` Comando para obtener más información sobre cómo especificar anotaciones con la CLI de Trident Protect.

Antes de empezar

Asegúrese de que se cumplan los siguientes requisitos previos:

- El clúster de destino tiene Trident Protect instalado.
- El clúster de destino tiene acceso a la ruta del bucket del mismo AppVault que el clúster de origen, donde se almacena la copia de seguridad.
- Asegúrese de que su entorno local pueda conectarse al bucket de almacenamiento de objetos definido en el CR de AppVault al ejecutar el proceso. `tridentctl-protect get appvaultcontent` dominio. Si las restricciones de red impiden el acceso, ejecute la CLI de Trident Protect desde dentro de un pod en el clúster de destino.
- Asegúrese de que la expiración del token de sesión de AWS sea suficiente para cualquier operación de restauración de larga duración. Si el token caduca durante la operación de restauración, la operación puede fallar.
  - Consulte el "Documentación de AWS API" Para obtener más información sobre cómo comprobar la caducidad del token de sesión actual.
  - Consulte el "Documentación de AWS" Para obtener más información sobre las credenciales con los recursos de AWS.

Pasos

1. Verifique la disponibilidad de AppVault CR en el clúster de destino mediante el complemento CLI de Trident Protect:

```
tridentctl-protect get appvault --context <destination_cluster_name>
```

> Asegúrese de que el espacio de nombres previsto para la restauración de la aplicación exista en el clúster de destino.

2. Vea el contenido de la copia de seguridad del AppVault disponible desde el clúster de destino:

```
tridentctl-protect get appvaultcontent <appvault_name> \
--show-resources backup \
--show-paths \
--context <destination_cluster_name>
```

Al ejecutar este comando se muestran las copias de seguridad disponibles en AppVault, incluidos sus clústeres de origen, los nombres de las aplicaciones correspondientes, las marcas de tiempo y las rutas

de archivo.

Ejemplo de salida:

```
+-------------+-----------+--------+-----------------
+--------------------------+-------------+
|   CLUSTER   |    APP    |  TYPE  |      NAME       |        TIMESTAMP
|    PATH     |
+-------------+-----------+--------+-----------------
+--------------------------+-------------+
| production1 | wordpress | backup | wordpress-bkup-1| 2024-10-30
08:37:40 (UTC)| backuppath1 |
| production1 | wordpress | backup | wordpress-bkup-2| 2024-10-30
08:37:40 (UTC)| backuppath2 |
+-------------+-----------+--------+-----------------
+--------------------------+-------------+
```

3. Restaure la aplicación en el clúster de destino utilizando el nombre de AppVault y la ruta de archivo:

Utilice un CR

1. Crea el archivo de recursos personalizados (CR) y nómbralo. `trident-protect-backup-restore-cr.yaml` .
2. En el archivo que creó, configure los siguientes atributos:
   - **metadata.name**: (*Obligatorio*) El nombre de este recurso personalizado; elija un nombre único y adecuado para su entorno.
   - **spec.appVaultRef**: (*Obligatorio*) El nombre del AppVault donde se almacenan los contenidos de la copia de seguridad.
   - **spec.appArchivePath**: La ruta dentro de AppVault donde se almacenan los contenidos de la copia de seguridad. Puedes utilizar el siguiente comando para encontrar esta ruta:

     ```
     kubectl get backups <BACKUP_NAME> -n my-app-namespace -o
     jsonpath='{.status.appArchivePath}'
     ```

     Si BackupRestore CR no está disponible, puede utilizar el comando mencionado en el paso 2 para ver el contenido de la copia de seguridad.

   - **spec.namespaceMapping**: La asignación del espacio de nombres de origen de la operación de restauración al espacio de nombres de destino. Reemplazar `my-source-namespace` y `my-destination-namespace` con información de tu entorno.

     Por ejemplo:

```
apiVersion: protect.trident.netapp.io/v1
kind: BackupRestore
metadata:
  name: my-cr-name
  namespace: my-destination-namespace
spec:
  appVaultRef: appvault-name
  appArchivePath: my-backup-path
  namespaceMapping: [{"source": "my-source-namespace", "
destination": "my-destination-namespace"}]
```

3. Después de rellenar el `trident-protect-backup-restore-cr.yaml` Archivo con los valores correctos, aplicar el CR:

```
kubectl apply -f trident-protect-backup-restore-cr.yaml
```

Utilice la CLI

1. Utilice el siguiente comando para restaurar la aplicación, reemplazando los valores entre corchetes con información de su entorno. El argumento de asignación de espacios de nombres utiliza espacios

de nombres separados por dos puntos para asignar los espacios de nombres de origen a los espacios de nombres de destino correctos en el formato source1:dest1,source2:dest2. Por ejemplo:

```
tridentctl-protect create backuprestore <restore_name> \
--namespace-mapping <source_to_destination_namespace_mapping> \
--appvault <appvault_name> \
--path <backup_path> \
--context <destination_cluster_name> \
-n <application_namespace>
```

Restaurar desde una instantánea a un espacio de nombres diferente

Puede restaurar datos desde una instantánea utilizando un archivo de recursos personalizados (CR) ya sea en un espacio de nombres diferente o en el espacio de nombres de origen original. Cuando restaura una instantánea a un espacio de nombres diferente mediante un CR de SnapshotRestore, Trident Protect restaura la aplicación en un nuevo espacio de nombres y crea un CR de aplicación para la aplicación restaurada. Para proteger la aplicación restaurada, cree copias de seguridad o instantáneas bajo demanda, o establezca un programa de protección.

> ⓘ SnapshotRestore admite `spec.storageClassMapping` atributo, pero solo cuando las clases de almacenamiento de origen y destino utilizan el mismo backend de almacenamiento. Si intenta restaurar a un `StorageClass` Si utiliza un sistema de almacenamiento diferente, la operación de restauración fallará.

Antes de empezar

Asegúrese de que la expiración del token de sesión de AWS sea suficiente para cualquier operación de restauración de S3 de larga duración. Si el token caduca durante la operación de restauración, la operación puede fallar.

- Consulte el "Documentación de AWS API" Para obtener más información sobre cómo comprobar la caducidad del token de sesión actual.
- Consulte el "Documentación de AWS IAM" Para obtener más información sobre las credenciales con los recursos de AWS.

Utilice un CR

Pasos

1. Crea el archivo de recursos personalizados (CR) y nómbralo. `trident-protect-snapshot-restore-cr.yaml` .
2. En el archivo que creó, configure los siguientes atributos:
   - **metadata.name**: (*Obligatorio*) El nombre de este recurso personalizado; elija un nombre único y adecuado para su entorno.
   - **spec.appVaultRef**: (*Obligatorio*) El nombre del AppVault donde se almacenan los contenidos de la instantánea.
   - **spec.appArchivePath**: La ruta dentro de AppVault donde se almacenan los contenidos de las instantáneas. Puedes utilizar el siguiente comando para encontrar esta ruta:

     ```
     kubectl get snapshots <SNAPHOT_NAME> -n my-app-namespace -o
     jsonpath='{.status.appArchivePath}'
     ```

   - **spec.namespaceMapping**: La asignación del espacio de nombres de origen de la operación de restauración al espacio de nombres de destino. Reemplazar `my-source-namespace` y `my-destination-namespace` con información de tu entorno.

     ```
     ---
     apiVersion: protect.trident.netapp.io/v1
     kind: SnapshotRestore
     metadata:
       name: my-cr-name
       namespace: my-app-namespace
     spec:
       appVaultRef: appvault-name
       appArchivePath: my-snapshot-path
       namespaceMapping: [{"source": "my-source-namespace",
     "destination": "my-destination-namespace"}]
     ```

3. (Opcional) Si necesita seleccionar solo ciertos recursos de la aplicación para restaurar, agregue un filtro que incluya o excluya los recursos marcados con etiquetas particulares:

   > Trident Protect selecciona algunos recursos automáticamente debido a su relación con los recursos que usted selecciona. Por ejemplo, si selecciona un recurso de reclamo de volumen persistente y tiene un pod asociado, Trident Protect también restaurará el pod asociado.

   - **resourceFilter.resourceSelectionCriteria**: (Obligatorio para filtrar) Usar `Include` o `Exclude` para incluir o excluir un recurso definido en resourceMatchers. Agregue los siguientes parámetros resourceMatchers para definir los recursos que se incluirán o excluirán:
     - **resourceFilter.resourceMatchers**: Una matriz de objetos resourceMatcher. Si defines varios elementos en esta matriz, coinciden como una operación OR, y los campos dentro de cada

elemento (grupo, tipo, versión) coinciden como una operación AND.

- **resourceMatchers[].group**: (*Opcional*) Grupo del recurso a filtrar.
- **resourceMatchers[].kind**: (*Opcional*) Tipo de recurso a filtrar.
- **resourceMatchers[].version**: (*Opcional*) Versión del recurso a filtrar.
- **resourceMatchers[].names**: (*Opcional*) Nombres en el campo metadata.name de Kubernetes del recurso que se va a filtrar.
- **resourceMatchers[].namespaces**: (*Opcional*) Espacios de nombres en el campo metadata.name de Kubernetes del recurso que se va a filtrar.
- **resourceMatchers[].labelSelectors**: (*Opcional*) Cadena de selector de etiquetas en el campo metadata.name de Kubernetes del recurso, tal como se define en el "Documentación de Kubernetes" . Por ejemplo: `"trident.netapp.io/os=linux"` .

  Por ejemplo:

```
spec:
  resourceFilter:
    resourceSelectionCriteria: "Include"
    resourceMatchers:
      - group: my-resource-group-1
        kind: my-resource-kind-1
        version: my-resource-version-1
        names: ["my-resource-names"]
        namespaces: ["my-resource-namespaces"]
        labelSelectors: ["trident.netapp.io/os=linux"]
      - group: my-resource-group-2
        kind: my-resource-kind-2
        version: my-resource-version-2
        names: ["my-resource-names"]
        namespaces: ["my-resource-namespaces"]
        labelSelectors: ["trident.netapp.io/os=linux"]
```

4. Después de rellenar el `trident-protect-snapshot-restore-cr.yaml` Archivo con los valores correctos, aplicar el CR:

```
kubectl apply -f trident-protect-snapshot-restore-cr.yaml
```

Utilice la CLI

Pasos

1. Restaure la instantánea en un espacio de nombres diferente, reemplazando los valores entre corchetes con información de su entorno.

   - El `snapshot` El argumento utiliza un espacio de nombres y un nombre de instantánea en el formato `<namespace>/<name>` .
   - El `namespace-mapping` El argumento utiliza espacios de nombres separados por dos puntos

para asignar los espacios de nombres de origen a los espacios de nombres de destino correctos en el formato `source1:dest1,source2:dest2` .

Por ejemplo:

```
tridentctl-protect create snapshotrestore <my_restore_name> \
--snapshot <namespace/snapshot_to_restore> \
--namespace-mapping <source_to_destination_namespace_mapping> \
-n <application_namespace>
```

Restaurar desde una instantánea al espacio de nombres original

Puedes restaurar una instantánea al espacio de nombres original en cualquier momento.

Antes de empezar

Asegúrese de que la expiración del token de sesión de AWS sea suficiente para cualquier operación de restauración de S3 de larga duración. Si el token caduca durante la operación de restauración, la operación puede fallar.

- Consulte el "Documentación de AWS API" Para obtener más información sobre cómo comprobar la caducidad del token de sesión actual.
- Consulte el "Documentación de AWS IAM" Para obtener más información sobre las credenciales con los recursos de AWS.

Utilice un CR

Pasos

1. Crea el archivo de recursos personalizados (CR) y nómbralo. `trident-protect-snapshot-ipr-cr.yaml` .
2. En el archivo que creó, configure los siguientes atributos:
   - **metadata.name**: (*Obligatorio*) El nombre de este recurso personalizado; elija un nombre único y adecuado para su entorno.
   - **spec.appVaultRef**: (*Obligatorio*) El nombre del AppVault donde se almacenan los contenidos de la instantánea.
   - **spec.appArchivePath**: La ruta dentro de AppVault donde se almacenan los contenidos de las instantáneas. Puedes utilizar el siguiente comando para encontrar esta ruta:

     ```
     kubectl get snapshots <SNAPSHOT_NAME> -n my-app-namespace -o
     jsonpath='{.status.appArchivePath}'
     ```

     ```
     ---
     apiVersion: protect.trident.netapp.io/v1
     kind: SnapshotInplaceRestore
     metadata:
       name: my-cr-name
       namespace: my-app-namespace
     spec:
       appVaultRef: appvault-name
         appArchivePath: my-snapshot-path
     ```

3. (Opcional) Si necesita seleccionar solo ciertos recursos de la aplicación para restaurar, agregue un filtro que incluya o excluya los recursos marcados con etiquetas particulares:

   > Trident Protect selecciona algunos recursos automáticamente debido a su relación con los recursos que usted selecciona. Por ejemplo, si selecciona un recurso de reclamo de volumen persistente y tiene un pod asociado, Trident Protect también restaurará el pod asociado.

   - **resourceFilter.resourceSelectionCriteria**: (Obligatorio para filtrar) Usar `Include` o `Exclude` para incluir o excluir un recurso definido en resourceMatchers. Agregue los siguientes parámetros resourceMatchers para definir los recursos que se incluirán o excluirán:
     - **resourceFilter.resourceMatchers**: Una matriz de objetos resourceMatcher. Si defines varios elementos en esta matriz, coinciden como una operación OR, y los campos dentro de cada elemento (grupo, tipo, versión) coinciden como una operación AND.
       - **resourceMatchers[].group**: (*Opcional*) Grupo del recurso a filtrar.
       - **resourceMatchers[].kind**: (*Opcional*) Tipo de recurso a filtrar.
       - **resourceMatchers[].version**: (*Opcional*) Versión del recurso a filtrar.
       - **resourceMatchers[].names**: (*Opcional*) Nombres en el campo metadata.name de

Kubernetes del recurso que se va a filtrar.

- **resourceMatchers[].namespaces**: (*Opcional*) Espacios de nombres en el campo metadata.name de Kubernetes del recurso que se va a filtrar.
- **resourceMatchers[].labelSelectors**: (*Opcional*) Cadena de selector de etiquetas en el campo metadata.name de Kubernetes del recurso, tal como se define en el "Documentación de Kubernetes" . Por ejemplo: `"trident.netapp.io/os=linux"` .

  Por ejemplo:

```
spec:
  resourceFilter:
    resourceSelectionCriteria: "Include"
    resourceMatchers:
      - group: my-resource-group-1
        kind: my-resource-kind-1
        version: my-resource-version-1
        names: ["my-resource-names"]
        namespaces: ["my-resource-namespaces"]
        labelSelectors: ["trident.netapp.io/os=linux"]
      - group: my-resource-group-2
        kind: my-resource-kind-2
        version: my-resource-version-2
        names: ["my-resource-names"]
        namespaces: ["my-resource-namespaces"]
        labelSelectors: ["trident.netapp.io/os=linux"]
```

4. Después de rellenar el `trident-protect-snapshot-ipr-cr.yaml` Archivo con los valores correctos, aplicar el CR:

```
kubectl apply -f trident-protect-snapshot-ipr-cr.yaml
```

Utilice la CLI

Pasos

1. Restaure la instantánea al espacio de nombres original, reemplazando los valores entre corchetes con información de su entorno. Por ejemplo:

```
tridentctl-protect create snapshotinplacerestore <my_restore_name> \
--snapshot <snapshot_to_restore> \
-n <application_namespace>
```

Comprobar el estado de una operación de restauración

Puede utilizar la línea de comandos para comprobar el estado de una operación de restauración que esté en curso, haya finalizado o haya fallado.

Pasos

1. Utilice el siguiente comando para recuperar el estado de la operación de restauración, reemplazando los valores entre corchetes con información de su entorno:

```
kubectl get backuprestore -n <namespace_name> <my_restore_cr_name> -o
jsonpath='{.status}'
```

Utilice la configuración de restauración avanzada de Trident Protect

Puede personalizar las operaciones de restauración utilizando configuraciones avanzadas, como anotaciones, configuraciones de espacios de nombres y opciones de almacenamiento para satisfacer sus requisitos específicos.

Anotaciones y etiquetas de espacio de nombres durante las operaciones de restauración y conmutación por error

Durante las operaciones de restauración y conmutación por error, las etiquetas y anotaciones del espacio de nombres de destino se ajustan para que coincidan con las etiquetas y anotaciones del espacio de nombres de origen. Las etiquetas o anotaciones del espacio de nombres de origen que no existen en el espacio de nombres de destino se agregan, y las etiquetas o anotaciones que ya existen se sobrescriben para que coincidan con el valor del espacio de nombres de origen. Las etiquetas o anotaciones que existen únicamente en el espacio de nombres de destino permanecen sin cambios.

> Si utiliza Red Hat OpenShift, es importante tener en cuenta el papel fundamental que desempeñan las anotaciones de espacios de nombres en los entornos OpenShift. Las anotaciones de espacio de nombres garantizan que los pods restaurados cumplan con los permisos y las configuraciones de seguridad adecuados definidos por las restricciones de contexto de seguridad (SCC) de OpenShift y puedan acceder a los volúmenes sin problemas de permisos. Para obtener más información, consulte la "Documentación sobre las restricciones del contexto de seguridad de OpenShift" .

Puedes evitar que se sobrescriban anotaciones específicas en el espacio de nombres de destino configurando la variable de entorno de Kubernetes. `RESTORE_SKIP_NAMESPACE_ANNOTATIONS` antes de realizar la operación de restauración o conmutación por error. Por ejemplo:

```
helm upgrade trident-protect --set
restoreSkipNamespaceAnnotations=<annotation_key_to_skip_1>,<annotation_key
_to_skip_2> --reuse-values
```

Al realizar una operación de restauración o conmutación por error, se tendrán en cuenta las anotaciones y etiquetas de espacio de nombres especificadas en `restoreSkipNamespaceAnnotations` y `restoreSkipNamespaceLabels` quedan excluidos de la operación de restauración o conmutación por error. Asegúrese de que estos ajustes se configuren durante la instalación inicial de Helm. Para obtener más información, consulte "Configurar las opciones de filtrado de espacios de nombres y AutoSupport".

Si instalaste la aplicación de origen usando Helm con el `--create-namespace` bandera, se le da un trato especial a la `name` Clave de etiqueta. Durante el proceso de restauración o conmutación por error, Trident Protect copia esta etiqueta en el espacio de nombres de destino, pero actualiza el valor al valor del espacio de nombres de destino si el valor del origen coincide con el espacio de nombres de origen. Si este valor no coincide con el espacio de nombres de origen, se copia al espacio de nombres de destino sin cambios.

Ejemplo

El siguiente ejemplo presenta un espacio de nombres de origen y destino, cada uno con diferentes anotaciones y etiquetas. Puedes ver el estado del espacio de nombres de destino antes y después de la operación, y cómo se combinan o sobrescriben las anotaciones y etiquetas en el espacio de nombres de destino.

Antes de la operación de restauración o conmutación por error

La siguiente tabla ilustra el estado de los espacios de nombres de origen y destino de ejemplo antes de la operación de restauración o conmutación por error:

Espacio de nombres	Anotaciones	Etiquetas
Espacio de nombres ns-1 (fuente)	• annotation.one/key: "valor actualizado" • annotation.two/key: "true"	• entorno=producción • cumplimiento=HIPAA • nombre=ns-1
Espacio de nombres ns-2 (destino)	• annotation.one/key: "true" • anotación.tres/clave: "falso"	• rol=base de datos

Después de la operación de restauración

La siguiente tabla ilustra el estado del espacio de nombres de destino de ejemplo después de la operación de restauración o conmutación por error. Se han añadido algunas claves, se han sobrescrito otras y la `name` La etiqueta se ha actualizado para que coincida con el espacio de nombres de destino:

Espacio de nombres	Anotaciones	Etiquetas
Espacio de nombres ns-2 (destino)	• annotation.one/key: "valor actualizado" • annotation.two/key: "true" • anotación.tres/clave: "falso"	• nombre=ns-2 • cumplimiento=HIPAA • entorno=producción • rol=base de datos

Campos admitidos

Esta sección describe campos adicionales disponibles para operaciones de restauración.

Mapeo de clases de almacenamiento

El `spec.storageClassMapping` El atributo define una asignación desde una clase de almacenamiento presente en la aplicación de origen a una nueva clase de almacenamiento en el clúster de destino. Puede usar esto al migrar aplicaciones entre clústeres con diferentes clases de almacenamiento o al cambiar el backend de almacenamiento para operaciones de BackupRestore.

Ejemplo:

```
storageClassMapping:
  - destination: "destinationStorageClass1"
    source: "sourceStorageClass1"
  - destination: "destinationStorageClass2"
    source: "sourceStorageClass2"
```

Anotaciones admitidas

Esta sección enumera las anotaciones admitidas para configurar diversos comportamientos en el sistema. Si el usuario no establece explícitamente una anotación, el sistema utilizará el valor predeterminado.

Anotación	Tipo	Descripción	Valor predeterminado
protect.trident.netapp.io/data-mover-timeout-sec	cadena	Tiempo máximo (en segundos) permitido para que se detenga la operación de transferencia de datos.	"300"
protect.trident.netapp.io/kopia-content-cache-size-limit-mb	cadena	El límite de tamaño máximo (en megabytes) para la caché de contenido de Kopia.	"1000"

Replicar aplicaciones utilizando NetApp SnapMirror y Trident Protect

Con Trident Protect, puede utilizar las capacidades de replicación asincrónica de la tecnología SnapMirror de NetApp para replicar cambios de datos y aplicaciones de un backend de almacenamiento a otro, en el mismo clúster o entre diferentes clústeres.

Anotaciones y etiquetas de espacio de nombres durante las operaciones de restauración y conmutación por error

Durante las operaciones de restauración y conmutación por error, las etiquetas y anotaciones del espacio de nombres de destino se ajustan para que coincidan con las etiquetas y anotaciones del espacio de nombres de origen. Las etiquetas o anotaciones del espacio de nombres de origen que no existen en el espacio de nombres de destino se agregan, y las etiquetas o anotaciones que ya existen se sobrescriben para que coincidan con el valor del espacio de nombres de origen. Las etiquetas o anotaciones que existen únicamente en el espacio de nombres de destino permanecen sin cambios.

Si utiliza Red Hat OpenShift, es importante tener en cuenta el papel fundamental que desempeñan las anotaciones de espacios de nombres en los entornos OpenShift. Las anotaciones de espacio de nombres garantizan que los pods restaurados cumplan con los permisos y las configuraciones de seguridad adecuados definidos por las restricciones de contexto de seguridad (SCC) de OpenShift y puedan acceder a los volúmenes sin problemas de permisos. Para obtener más información, consulte la "Documentación sobre las restricciones del contexto de seguridad de OpenShift" .

Puedes evitar que se sobrescriban anotaciones específicas en el espacio de nombres de destino configurando la variable de entorno de Kubernetes. `RESTORE_SKIP_NAMESPACE_ANNOTATIONS` antes de realizar la operación de restauración o conmutación por error. Por ejemplo:

```
helm upgrade trident-protect --set
restoreSkipNamespaceAnnotations=<annotation_key_to_skip_1>,<annotation_key
_to_skip_2> --reuse-values
```

Al realizar una operación de restauración o conmutación por error, se tendrán en cuenta las anotaciones y etiquetas de espacio de nombres especificadas en `restoreSkipNamespaceAnnotations` y `restoreSkipNamespaceLabels` quedan excluidos de la operación de restauración o conmutación por error. Asegúrese de que estos ajustes se configuren durante la instalación inicial de Helm. Para obtener más información, consulte "Configurar las opciones de filtrado de espacios de nombres y AutoSupport".

Si instalaste la aplicación de origen usando Helm con el `--create-namespace` bandera, se le da un trato especial a la `name` Clave de etiqueta. Durante el proceso de restauración o conmutación por error, Trident Protect copia esta etiqueta en el espacio de nombres de destino, pero actualiza el valor al valor del espacio de nombres de destino si el valor del origen coincide con el espacio de nombres de origen. Si este valor no coincide con el espacio de nombres de origen, se copia al espacio de nombres de destino sin cambios.

Ejemplo

El siguiente ejemplo presenta un espacio de nombres de origen y destino, cada uno con diferentes anotaciones y etiquetas. Puedes ver el estado del espacio de nombres de destino antes y después de la operación, y cómo se combinan o sobrescriben las anotaciones y etiquetas en el espacio de nombres de destino.

Antes de la operación de restauración o conmutación por error

La siguiente tabla ilustra el estado de los espacios de nombres de origen y destino de ejemplo antes de la operación de restauración o conmutación por error:

Espacio de nombres	Anotaciones	Etiquetas
Espacio de nombres ns-1 (fuente)	• annotation.one/key: "valor actualizado" • annotation.two/key: "true"	• entorno=producción • cumplimiento=HIPAA • nombre=ns-1

Espacio de nombres	Anotaciones	Etiquetas
Espacio de nombres ns-2 (destino)	• annotation.one/key: "true" • anotación.tres/clave: "falso"	• rol=base de datos

Después de la operación de restauración

La siguiente tabla ilustra el estado del espacio de nombres de destino de ejemplo después de la operación de restauración o conmutación por error. Se han añadido algunas claves, se han sobrescrito otras y la `name` La etiqueta se ha actualizado para que coincida con el espacio de nombres de destino:

Espacio de nombres	Anotaciones	Etiquetas
Espacio de nombres ns-2 (destino)	• annotation.one/key: "valor actualizado" • annotation.two/key: "true" • anotación.tres/clave: "falso"	• nombre=ns-2 • cumplimiento=HIPAA • entorno=producción • rol=base de datos

Puede configurar Trident Protect para congelar y descongelar sistemas de archivos durante operaciones de protección de datos. "Obtenga más información sobre cómo configurar la congelación del sistema de archivos con Trident Protect".

Puntos de ejecución durante las operaciones de conmutación por error y reversa

Al utilizar la relación AppMirror para proteger su aplicación, existen comportamientos específicos relacionados con los ganchos de ejecución que debe tener en cuenta durante las operaciones de conmutación por error y reversión.

- Durante la conmutación por error, los ganchos de ejecución se copian automáticamente del clúster de origen al clúster de destino. No es necesario recrearlos manualmente. Tras la conmutación por error, existen ganchos de ejecución en la aplicación que se ejecutarán durante cualquier acción relevante.
- Durante la resincronización inversa o inversa, se eliminan todos los ganchos de ejecución existentes en la aplicación. Cuando la aplicación de origen se convierte en la aplicación de destino, estos ganchos de ejecución dejan de ser válidos y se eliminan para evitar su ejecución.

Para obtener más información sobre los ganchos de ejecución, consulte"Administrar los ganchos de ejecución de Trident Protect" .

Establecer una relación de replicación

Establecer una relación de replicación implica lo siguiente:

- Elegir la frecuencia con la que desea que Trident Protect tome una instantánea de la aplicación (que incluye los recursos de Kubernetes de la aplicación, así como las instantáneas de volumen para cada uno de los volúmenes de la aplicación)
- Elegir el plan de replicación (incluye recursos de Kubernetes y datos de volúmenes persistentes).
- Configurar la hora para tomar la instantánea

Pasos

1. En el clúster de origen, cree un AppVault para la aplicación de origen. Dependiendo de su proveedor de almacenamiento, modifique un ejemplo en"recursos personalizados de AppVault" para adaptarse a su entorno:

Cree un AppVault usando un CR

a. Cree el archivo de recursos personalizados (CR) y asígnele un nombre (por ejemplo, `trident-protect-appvault-primary-source.yaml`).

b. Configure los siguientes atributos:

- **metadata.name**: (*Obligatorio*) El nombre del recurso personalizado de AppVault. Tome nota del nombre que elija, ya que otros archivos CR necesarios para una relación de replicación hacen referencia a este valor.
- **spec.providerConfig**: (*Obligatorio*) Almacena la configuración necesaria para acceder a AppVault utilizando el proveedor especificado. Elige un nombre para el bucket y cualquier otro detalle necesario para tu proveedor. Tome nota de los valores que elija, ya que otros archivos CR necesarios para una relación de replicación hacen referencia a estos valores. Referirse a"recursos personalizados de AppVault" Para ver ejemplos de solicitudes de cambio de AppVault con otros proveedores.
- **spec.providerCredentials**: (*Obligatorio*) Almacena referencias a cualquier credencial requerida para acceder a AppVault utilizando el proveedor especificado.
  - **spec.providerCredentials.valueFromSecret**: (*Requerido*) Indica que el valor de la credencial debe provenir de un secreto.
    - **clave**: (*Obligatorio*) La clave válida del secreto a seleccionar.
    - **nombre**: (*Obligatorio*) Nombre del secreto que contiene el valor de este campo. Debe estar en el mismo espacio de nombres.
  - **spec.providerCredentials.secretAccessKey**: (*Obligatorio*) La clave de acceso utilizada para acceder al proveedor. El **nombre** debe coincidir con **spec.providerCredentials.valueFromSecret.name**.
- **spec.providerType**: (*Obligatorio*) Determina qué proporciona la copia de seguridad; por ejemplo, NetApp ONTAP S3, S3 genérico, Google Cloud o Microsoft Azure. Valores posibles:
  - AWS
  - azur
  - gcp
  - genérico-s3
  - ontap-s3
  - almacenamiento-s3

c. Después de rellenar el `trident-protect-appvault-primary-source.yaml` Archivo con los valores correctos, aplicar el CR:

```
kubectl apply -f trident-protect-appvault-primary-source.yaml -n
trident-protect
```

Crea un AppVault mediante la CLI.

a. Crea el AppVault, reemplazando los valores entre corchetes con información de tu entorno:

```
tridentctl-protect create vault Azure <vault-name> --account
<account-name> --bucket <bucket-name> --secret <secret-name> -n
trident-protect
```

2. En el clúster de origen, cree la aplicación de origen CR:

Cree la aplicación de origen utilizando un CR.

a. Cree el archivo de recursos personalizados (CR) y asígnele un nombre (por ejemplo, `trident-protect-app-source.yaml`).

b. Configure los siguientes atributos:

- **metadata.name**: (*Obligatorio*) El nombre del recurso personalizado de la aplicación. Tome nota del nombre que elija, ya que otros archivos CR necesarios para una relación de replicación hacen referencia a este valor.
- **spec.includedNamespaces**: (*Obligatorio*) Una matriz de espacios de nombres y etiquetas asociadas. Utilice nombres de espacios de nombres y, opcionalmente, restrinja el alcance de los espacios de nombres con etiquetas para especificar los recursos que existen en los espacios de nombres enumerados aquí. El espacio de nombres de la aplicación debe formar parte de esta matriz.

  Ejemplo de YAML:

```
---
apiVersion: protect.trident.netapp.io/v1
kind: Application
metadata:
  name: my-app-name
  namespace: my-app-namespace
spec:
  includedNamespaces:
    - namespace: my-app-namespace
      labelSelector: {}
```

c. Después de rellenar el `trident-protect-app-source.yaml` Archivo con los valores correctos, aplicar el CR:

```
kubectl apply -f trident-protect-app-source.yaml -n my-app-namespace
```

Cree la aplicación de origen utilizando la CLI.

a. Crea la aplicación de origen. Por ejemplo:

```
tridentctl-protect create app <my-app-name> --namespaces <namespaces-to-be-included> -n <my-app-namespace>
```

3. Opcionalmente, en el clúster de origen, tome una instantánea de la aplicación de origen. Esta instantánea se utiliza como base para la aplicación en el clúster de destino. Si omite este paso, deberá esperar a que se ejecute la siguiente instantánea programada para tener una instantánea reciente. Para crear una instantánea bajo demanda, consulte "Crea una instantánea bajo demanda".

4. En el clúster de origen, cree la programación de replicación CR:

Además del cronograma que se proporciona a continuación, se recomienda crear un cronograma de instantáneas diarias separado con un período de retención de 7 días para mantener una instantánea común entre los clústeres ONTAP emparejados. Esto garantiza que las instantáneas estén disponibles durante un máximo de 7 días, pero el período de retención se puede personalizar según las necesidades del usuario.

Si se produce una conmutación por error, el sistema puede utilizar estas instantáneas durante un máximo de 7 días para operaciones inversas. Este enfoque hace que el proceso inverso sea más rápido y eficiente porque solo se transferirán los cambios realizados desde la última instantánea, no todos los datos.

Si el cronograma existente para la aplicación ya cumple con los requisitos de retención deseados, no se requieren cronogramas adicionales.

Cree la programación de replicación utilizando un CR.

a. Cree una planificación de replicación para la aplicación de origen:

   i. Cree el archivo de recursos personalizados (CR) y asígnele un nombre (por ejemplo, `trident-protect-schedule.yaml`).

   ii. Configure los siguientes atributos:

   - **metadata.name**: (*Obligatorio*) El nombre del recurso personalizado de programación.
   - **spec.appVaultRef**: (*Obligatorio*) Este valor debe coincidir con el campo metadata.name del AppVault para la aplicación de origen.
   - **spec.applicationRef**: (*Obligatorio*) Este valor debe coincidir con el campo metadata.name del CR de la aplicación de origen.
   - **spec.backupRetention**: (*Obligatorio*) Este campo es obligatorio y el valor debe establecerse en 0.
   - **spec.enabled**: Debe establecerse en verdadero.
   - **spec.granularity**: Debe configurarse en `Custom` .
   - **spec.recurrenceRule**: Define una fecha de inicio en tiempo UTC y un intervalo de recurrencia.
   - **spec.snapshotRetention**: Debe establecerse en 2.

     Ejemplo de YAML:

```
---
apiVersion: protect.trident.netapp.io/v1
kind: Schedule
metadata:
  name: appmirror-schedule
  namespace: my-app-namespace
spec:
  appVaultRef: my-appvault-name
  applicationRef: my-app-name
  backupRetention: "0"
  enabled: true
  granularity: Custom
  recurrenceRule: |-
    DTSTART:20220101T000200Z
    RRULE:FREQ=MINUTELY;INTERVAL=5
  snapshotRetention: "2"
```

   i. Después de rellenar el `trident-protect-schedule.yaml` Archivo con los valores correctos, aplicar el CR:

```
kubectl apply -f trident-protect-schedule.yaml -n my-app-
namespace
```

Cree la programación de replicación mediante la CLI.

a. Cree la planificación de replicación, reemplazando los valores entre corchetes con información de su entorno:

```
tridentctl-protect create schedule --name appmirror-schedule
--app <my_app_name> --appvault <my_app_vault> --granularity
Custom --recurrence-rule <rule> --snapshot-retention
<snapshot_retention_count> -n <my_app_namespace>
```

Ejemplo:

```
tridentctl-protect create schedule --name appmirror-schedule
--app <my_app_name> --appvault <my_app_vault> --granularity
Custom --recurrence-rule  "DTSTART:20220101T000200Z
\nRRULE:FREQ=MINUTELY;INTERVAL=5" --snapshot-retention 2 -n
<my_app_namespace>
```

5. En el clúster de destino, cree un CR de AppVault de aplicación de origen que sea idéntico al CR de AppVault que aplicó en el clúster de origen y asígnele un nombre (por ejemplo, `trident-protect-appvault-primary-destination.yaml`).
6. Aplicar la CR:

```
kubectl apply -f trident-protect-appvault-primary-destination.yaml -n
trident-protect
```

7. Cree un CR de AppVault de destino para la aplicación de destino en el clúster de destino. Dependiendo de su proveedor de almacenamiento, modifique un ejemplo en"recursos personalizados de AppVault" para adaptarse a su entorno:
   a. Cree el archivo de recursos personalizados (CR) y asígnele un nombre (por ejemplo, `trident-protect-appvault-secondary-destination.yaml`).
   b. Configure los siguientes atributos:
      - **metadata.name**: (*Obligatorio*) El nombre del recurso personalizado de AppVault. Tome nota del nombre que elija, ya que otros archivos CR necesarios para una relación de replicación hacen referencia a este valor.
      - **spec.providerConfig**: (*Obligatorio*) Almacena la configuración necesaria para acceder a AppVault utilizando el proveedor especificado. Elige uno `bucketName` y cualquier otro detalle necesario para su proveedor. Tome nota de los valores que elija, ya que otros archivos CR necesarios para una relación de replicación hacen referencia a estos valores. Referirse a"recursos personalizados

de AppVault" Para ver ejemplos de solicitudes de cambio de AppVault con otros proveedores.

- **spec.providerCredentials**: (*Obligatorio*) Almacena referencias a cualquier credencial requerida para acceder a AppVault utilizando el proveedor especificado.
  - **spec.providerCredentials.valueFromSecret**: (*Requerido*) Indica que el valor de la credencial debe provenir de un secreto.
    - **clave**: (*Obligatorio*) La clave válida del secreto a seleccionar.
    - **nombre**: (*Obligatorio*) Nombre del secreto que contiene el valor de este campo. Debe estar en el mismo espacio de nombres.
  - **spec.providerCredentials.secretAccessKey**: (*Obligatorio*) La clave de acceso utilizada para acceder al proveedor. El **nombre** debe coincidir con **spec.providerCredentials.valueFromSecret.name**.
- **spec.providerType**: (*Obligatorio*) Determina qué proporciona la copia de seguridad; por ejemplo, NetApp ONTAP S3, S3 genérico, Google Cloud o Microsoft Azure. Valores posibles:
  - AWS
  - azur
  - gcp
  - genérico-s3
  - ontap-s3
  - almacenamiento-s3

c. Después de rellenar el `trident-protect-appvault-secondary-destination.yaml` Archivo con los valores correctos, aplicar el CR:

```
kubectl apply -f trident-protect-appvault-secondary-destination.yaml
-n trident-protect
```

8. En el clúster de destino, cree un archivo CR AppMirrorRelationship:

Crea una relación AppMirrorRelationship usando un CR

a. Cree el archivo de recursos personalizados (CR) y asígnele un nombre (por ejemplo, `trident-protect-relationship.yaml`).

b. Configure los siguientes atributos:

- **metadata.name:** (Obligatorio) El nombre del recurso personalizado AppMirrorRelationship.
- **spec.destinationAppVaultRef**: (*Obligatorio*) Este valor debe coincidir con el nombre del AppVault para la aplicación de destino en el clúster de destino.
- **spec.namespaceMapping**: (*Obligatorio*) Los espacios de nombres de destino y origen deben coincidir con el espacio de nombres de la aplicación definido en el CR de la aplicación correspondiente.
- **spec.sourceAppVaultRef**: (*Obligatorio*) Este valor debe coincidir con el nombre del AppVault de la aplicación de origen.
- **spec.sourceApplicationName**: (*Obligatorio*) Este valor debe coincidir con el nombre de la aplicación de origen que definió en el CR de la aplicación de origen.
- **spec.sourceApplicationUID**: (Obligatorio) Este valor debe coincidir con el UID de la aplicación de origen que definió en el CR de la aplicación de origen.
- **spec.storageClassName**: (*Opcional*) Elija el nombre de una clase de almacenamiento válida en el clúster. La clase de almacenamiento debe estar vinculada a una máquina virtual de almacenamiento ONTAP que esté emparejada con el entorno de origen. Si no se proporciona la clase de almacenamiento, se utilizará por defecto la clase de almacenamiento predeterminada del clúster.
- **spec.recurrenceRule**: Define una fecha de inicio en tiempo UTC y un intervalo de recurrencia.

  Ejemplo de YAML:

```
---
apiVersion: protect.trident.netapp.io/v1
kind: AppMirrorRelationship
metadata:
  name: amr-16061e80-1b05-4e80-9d26-d326dc1953d8
  namespace: my-app-namespace
spec:
  desiredState: Established
  destinationAppVaultRef: generic-s3-trident-protect-dst-bucket-
8fe0b902-f369-4317-93d1-ad7f2edc02b5
  namespaceMapping:
    - destination: my-app-namespace
      source: my-app-namespace
  recurrenceRule: |-
    DTSTART:20220101T000200Z
    RRULE:FREQ=MINUTELY;INTERVAL=5
  sourceAppVaultRef: generic-s3-trident-protect-src-bucket-
b643cc50-0429-4ad5-971f-ac4a83621922
  sourceApplicationName: my-app-name
  sourceApplicationUID: 7498d32c-328e-4ddd-9029-122540866aeb
  storageClassName: sc-vsim-2
```

c. Después de rellenar el `trident-protect-relationship.yaml` Archivo con los valores correctos, aplicar el CR:

```
kubectl apply -f trident-protect-relationship.yaml -n my-app-
namespace
```

Crea una relación AppMirrorRelationship usando la CLI

a. Crea y aplica el objeto AppMirrorRelationship, reemplazando los valores entre corchetes con información de tu entorno:

```
tridentctl-protect create appmirrorrelationship
<name_of_appmirorrelationship> --destination-app-vault
<my_vault_name> --source-app-vault <my_vault_name> --recurrence
-rule <rule> --namespace-mapping <ns_mapping> --source-app-id
<source_app_UID> --source-app <my_source_app_name> --storage
-class <storage_class_name> -n <application_namespace>
```

Ejemplo:

```
tridentctl-protect create appmirrorrelationship my-amr
--destination-app-vault appvault2 --source-app-vault appvault1
--recurrence-rule
"DTSTART:20220101T000200Z\nRRULE:FREQ=MINUTELY;INTERVAL=5"
--source-app my-app --namespace-mapping "my-source-ns1:my-dest-
ns1,my-source-ns2:my-dest-ns2" --source-app-id 373f24c1-5769-
404c-93c3-5538af6ccc36 --storage-class my-storage-class -n my-
dest-ns1
```

9. (Opcional) En el clúster de destino, compruebe el estado y la situación de la relación de replicación:

```
kubectl get amr -n my-app-namespace <relationship name> -o=jsonpath
='{.status}' | jq
```

Conmutación por error al clúster de destino

Con Trident Protect, puede conmutar por error las aplicaciones replicadas a un clúster de destino. Este procedimiento detiene la relación de replicación y pone la aplicación en línea en el clúster de destino. Trident Protect no detiene la aplicación en el clúster de origen si estaba operativa.

Pasos

1. En el clúster de destino, edite el archivo CR AppMirrorRelationship (por ejemplo, `trident-protect-relationship.yaml`) y cambiar el valor de **spec.desiredState** a `Promoted` .
2. Guarde el archivo CR.
3. Aplicar la CR:

```
kubectl apply -f trident-protect-relationship.yaml -n my-app-namespace
```

4. (*Opcional*) Cree los planes de protección que necesite en la aplicación en caso de fallo.
5. (Opcional) Compruebe el estado y la situación de la relación de replicación:

```
kubectl get amr -n my-app-namespace <relationship name> -o=jsonpath
='{.status}' | jq
```

Resincronizar una relación de replicación fallida

La operación de resincronización restablece la relación de replicación. Después de realizar una operación de resincronización, la aplicación de origen original se convierte en la aplicación en ejecución y se descartan todos los cambios realizados en la aplicación en ejecución en el clúster de destino.

El proceso detiene la aplicación en el clúster de destino antes de restablecer la replicación.

Cualquier dato escrito en la aplicación de destino durante la conmutación por error se perderá.

Pasos

1. Opcional: En el clúster de origen, cree una instantánea de la aplicación de origen. Esto garantiza que se capturen los últimos cambios del clúster de origen.
2. En el clúster de destino, edite el archivo CR AppMirrorRelationship (por ejemplo, `trident-protect-relationship.yaml`) y cambiar el valor de spec.desiredState a `Established` .
3. Guarde el archivo CR.
4. Aplicar la CR:

```
kubectl apply -f trident-protect-relationship.yaml -n my-app-namespace
```

5. Si creó alguna programación de protección en el clúster de destino para proteger la aplicación en caso de fallo, elimínela. Cualquier programación que permanezca provoca fallos en las instantáneas de volumen.

Resincronización inversa de una relación de replicación fallida

Cuando se resincroniza de forma inversa una relación de replicación por conmutación por error, la aplicación de destino se convierte en la aplicación de origen y el origen se convierte en el destino. Los cambios realizados en la aplicación de destino durante la conmutación por error se conservan.

Pasos

1. En el clúster de destino original, elimine el CR AppMirrorRelationship. Esto provoca que el destino se convierta en la fuente. Si quedan programaciones de protección en el nuevo clúster de destino, elimínelas.
2. Establezca una relación de replicación aplicando los archivos CR que utilizó originalmente para establecer la relación a los clústeres opuestos.
3. Asegúrese de que el nuevo destino (clúster de origen original) esté configurado con ambos CR de AppVault.
4. Establezca una relación de replicación en el clúster opuesto, configurando los valores para la dirección inversa.

Dirección de replicación de la aplicación inversa

Cuando invierte la dirección de replicación, Trident Protect mueve la aplicación al backend de almacenamiento de destino mientras continúa replicando al backend de almacenamiento de origen original. Trident Protect detiene la aplicación de origen y replica los datos en el destino antes de conmutar por error a la aplicación de destino.

En esta situación, estás intercambiando el origen y el destino.

Pasos

1. En el clúster de origen, cree una instantánea de apagado:

Cree una instantánea de apagado mediante un CR.

a. Deshabilite las programaciones de políticas de protección para la aplicación de origen.

b. Crear un archivo CR ShutdownSnapshot:

   i. Cree el archivo de recursos personalizados (CR) y asígnele un nombre (por ejemplo, `trident-protect-shutdownsnapshot.yaml`).

   ii. Configure los siguientes atributos:

   - **metadata.name**: (*Obligatorio*) El nombre del recurso personalizado.
   - **spec.AppVaultRef**: (*Obligatorio*) Este valor debe coincidir con el campo metadata.name del AppVault para la aplicación de origen.
   - **spec.ApplicationRef**: (*Obligatorio*) Este valor debe coincidir con el campo metadata.name del archivo CR de la aplicación de origen.

     Ejemplo de YAML:

```
---
apiVersion: protect.trident.netapp.io/v1
kind: ShutdownSnapshot
metadata:
  name: replication-shutdown-snapshot-afc4c564-e700-4b72-86c3-
c08a5dbe844e
  namespace: my-app-namespace
spec:
  appVaultRef: generic-s3-trident-protect-src-bucket-04b6b4ec-
46a3-420a-b351-45795e1b5e34
  applicationRef: my-app-name
```

c. Después de rellenar el `trident-protect-shutdownsnapshot.yaml` Archivo con los valores correctos, aplicar el CR:

```
kubectl apply -f trident-protect-shutdownsnapshot.yaml -n my-app-
namespace
```

Cree una instantánea de apagado mediante la CLI.

a. Cree la instantánea de apagado, reemplazando los valores entre corchetes con información de su entorno. Por ejemplo:

```
tridentctl-protect create shutdownsnapshot <my_shutdown_snapshot>
--appvault <my_vault> --app <app_to_snapshot> -n
<application_namespace>
```

2. En el clúster de origen, una vez completada la instantánea de apagado, obtenga el estado de la instantánea de apagado:

```
kubectl get shutdownsnapshot -n my-app-namespace
<shutdown_snapshot_name> -o yaml
```

3. En el clúster de origen, busque el valor de **shutdownsnapshot.status.appArchivePath** utilizando el siguiente comando y registre la última parte de la ruta del archivo (también llamada nombre base; esto será todo lo que esté después de la última barra):

```
k get shutdownsnapshot -n my-app-namespace <shutdown_snapshot_name> -o
jsonpath='{.status.appArchivePath}'
```

4. Realizar una conmutación por error desde el nuevo clúster de destino al nuevo clúster de origen, con el siguiente cambio:

> ⓘ En el paso 2 del procedimiento de conmutación por error, incluya el `spec.promotedSnapshot` campo en el archivo CR AppMirrorRelationship, y establezca su valor en el nombre base que registró en el paso 3 anterior.

5. Realice los pasos de resincronización inversa enResincronización inversa de una relación de replicación fallida .
6. Habilite las programaciones de protección en el nuevo clúster de origen.

Resultado

Las siguientes acciones se producen debido a la replicación inversa:

- Se toma una instantánea de los recursos de Kubernetes de la aplicación fuente original.
- Los pods de la aplicación fuente original se detienen correctamente eliminando los recursos de Kubernetes de la aplicación (dejando los PVC y PV en su lugar).
- Una vez apagados los pods, se toman instantáneas de los volúmenes de la aplicación y se replican.
- Las relaciones SnapMirror se han roto, dejando los volúmenes de destino listos para lectura/escritura.
- Los recursos de Kubernetes de la aplicación se restauran a partir de la instantánea previa al apagado, utilizando los datos de volumen replicados después de que se apagara la aplicación de origen original.
- La replicación se restablece en la dirección inversa.

Revertir las aplicaciones al clúster de origen original

Al utilizar Trident Protect, puede lograr una "conmutación por error" después de una operación de conmutación por error mediante la siguiente secuencia de operaciones. En este flujo de trabajo para restaurar la dirección de replicación original, Trident Protect replica (resincroniza) cualquier cambio de aplicación a la aplicación de origen original antes de revertir la dirección de replicación.

Este proceso comienza a partir de una relación que ha completado una conmutación por error a un destino e implica los siguientes pasos:

- Comience con un estado de conmutación por error.
- Resincronizar de forma inversa la relación de replicación.

> No realice una operación de resincronización normal, ya que esto descartará los datos escritos en el clúster de destino durante el procedimiento de conmutación por error.

- Invierta la dirección de replicación.

Pasos

1. Realizar elResincronización inversa de una relación de replicación fallida pasos.
2. Realizar elDirección de replicación de la aplicación inversa pasos.

Eliminar una relación de replicación

Puedes eliminar una relación de replicación en cualquier momento. Al eliminar la relación de replicación de la aplicación, se generan dos aplicaciones separadas sin ninguna relación entre ellas.

Pasos

1. En el clúster de destino actual, elimine el CR AppMirrorRelationship:

```
kubectl delete -f trident-protect-relationship.yaml -n my-app-namespace
```

Migrar aplicaciones usando Trident Protect

Puede migrar sus aplicaciones entre clústeres o a diferentes clases de almacenamiento restaurando datos de respaldo.

> Al migrar una aplicación, todos los ganchos de ejecución configurados para la aplicación se migran con la aplicación. Si existe un gancho de ejecución posterior a la restauración, se ejecuta automáticamente como parte de la operación de restauración.

Operaciones de copia de seguridad y restauración

Para realizar operaciones de copia de seguridad y restauración en los siguientes escenarios, puede automatizar tareas específicas de copia de seguridad y restauración.

Clonar al mismo clúster

Para clonar una aplicación en el mismo clúster, cree una instantánea o copia de seguridad y restaure los datos en el mismo clúster.

Pasos

1. Debe realizar una de las siguientes acciones:
   a. "Crear una instantánea".
   b. "Crea una copia de seguridad".
2. En el mismo clúster, realice una de las siguientes acciones, dependiendo de si creó una instantánea o una copia de seguridad:

a. "Restaura tus datos desde la instantánea.".

b. "Restaura tus datos desde la copia de seguridad.".

Clonar a un clúster diferente

Para clonar una aplicación en un clúster diferente (realizar una clonación entre clústeres), cree una copia de seguridad en el clúster de origen y, a continuación, restaure la copia de seguridad en un clúster diferente. Asegúrese de que Trident Protect esté instalado en el clúster de destino.

Puedes replicar una aplicación entre diferentes clústeres usando"Replicación de SnapMirror" .

Pasos

1. "Crea una copia de seguridad".
2. Asegúrese de que el CR de AppVault para el bucket de almacenamiento de objetos que contiene la copia de seguridad se haya configurado en el clúster de destino.
3. En el clúster de destino,"Restaura tus datos desde la copia de seguridad" .

Migrar aplicaciones de una clase de almacenamiento a otra clase de almacenamiento

Puede migrar aplicaciones de una clase de almacenamiento a una clase de almacenamiento diferente restaurando una copia de seguridad en la clase de almacenamiento de destino.

Por ejemplo (excluyendo los secretos de la restauración CR):

```
apiVersion: protect.trident.netapp.io/v1
kind: SnapshotRestore
metadata:
  name: "${snapshotRestoreCRName}"
spec:
  appArchivePath: "${snapshotArchivePath}"
  appVaultRef: "${appVaultCRName}"
  namespaceMapping:
    - destination: "${destinationNamespace}"
      source: "${sourceNamespace}"
  storageClassMapping:
    - destination: "${destinationStorageClass}"
      source: "${sourceStorageClass}"
  resourceFilter:
    resourceMatchers:
      kind: Secret
      version: v1
    resourceSelectionCriteria: exclude
```

Restaure la instantánea mediante un CR

Pasos

1. Crea el archivo de recursos personalizados (CR) y nómbralo. `trident-protect-snapshot-restore-cr.yaml` .
2. En el archivo que creó, configure los siguientes atributos:
   - **metadata.name**: (*Obligatorio*) El nombre de este recurso personalizado; elija un nombre único y adecuado para su entorno.
   - **spec.appArchivePath**: La ruta dentro de AppVault donde se almacenan los contenidos de las instantáneas. Puedes utilizar el siguiente comando para encontrar esta ruta:

     ```
     kubectl get snapshots <my-snapshot-name> -n trident-protect -o jsonpath='{.status.appArchivePath}'
     ```

   - **spec.appVaultRef**: (*Obligatorio*) El nombre del AppVault donde se almacenan los contenidos de la instantánea.
   - **spec.namespaceMapping**: La asignación del espacio de nombres de origen de la operación de restauración al espacio de nombres de destino. Reemplazar `my-source-namespace` y `my-destination-namespace` con información de tu entorno.

     ```
     ---
     apiVersion: protect.trident.netapp.io/v1
     kind: SnapshotRestore
     metadata:
       name: my-cr-name
       namespace: trident-protect
     spec:
       appArchivePath: my-snapshot-path
       appVaultRef: appvault-name
       namespaceMapping: [{"source": "my-source-namespace",
     "destination": "my-destination-namespace"}]
     ```

3. Opcionalmente, si necesita seleccionar solo ciertos recursos de la aplicación para restaurar, agregue un filtro que incluya o excluya los recursos marcados con etiquetas particulares:
   - **resourceFilter.resourceSelectionCriteria**: (Obligatorio para filtrar) Usar `include or exclude` para incluir o excluir un recurso definido en resourceMatchers. Agregue los siguientes parámetros resourceMatchers para definir los recursos que se incluirán o excluirán:
     - **resourceFilter.resourceMatchers**: Una matriz de objetos resourceMatcher. Si defines varios elementos en esta matriz, coinciden como una operación OR, y los campos dentro de cada elemento (grupo, tipo, versión) coinciden como una operación AND.
       - **resourceMatchers[].group**: (*Opcional*) Grupo del recurso a filtrar.
       - **resourceMatchers[].kind**: (*Opcional*) Tipo de recurso a filtrar.
       - **resourceMatchers[].version**: (*Opcional*) Versión del recurso a filtrar.

- **resourceMatchers[].names**: (*Opcional*) Nombres en el campo metadata.name de Kubernetes del recurso que se va a filtrar.
- **resourceMatchers[].namespaces**: (*Opcional*) Espacios de nombres en el campo metadata.name de Kubernetes del recurso que se va a filtrar.
- **resourceMatchers[].labelSelectors**: (*Opcional*) Cadena de selector de etiquetas en el campo metadata.name de Kubernetes del recurso, tal como se define en el "Documentación de Kubernetes" . Por ejemplo: `"trident.netapp.io/os=linux"` .

  Por ejemplo:

```
spec:
  resourceFilter:
    resourceSelectionCriteria: "include"
    resourceMatchers:
      - group: my-resource-group-1
        kind: my-resource-kind-1
        version: my-resource-version-1
        names: ["my-resource-names"]
        namespaces: ["my-resource-namespaces"]
        labelSelectors: ["trident.netapp.io/os=linux"]
      - group: my-resource-group-2
        kind: my-resource-kind-2
        version: my-resource-version-2
        names: ["my-resource-names"]
        namespaces: ["my-resource-namespaces"]
        labelSelectors: ["trident.netapp.io/os=linux"]
```

4. Después de rellenar el `trident-protect-snapshot-restore-cr.yaml` Archivo con los valores correctos, aplicar el CR:

```
kubectl apply -f trident-protect-snapshot-restore-cr.yaml
```

Restaure la instantánea mediante la CLI.

Pasos

1. Restaure la instantánea en un espacio de nombres diferente, reemplazando los valores entre corchetes con información de su entorno.

   - El `snapshot` El argumento utiliza un espacio de nombres y un nombre de instantánea en el formato `<namespace>/<name>` .
   - El `namespace-mapping` El argumento utiliza espacios de nombres separados por dos puntos para asignar los espacios de nombres de origen a los espacios de nombres de destino correctos en el formato `source1:dest1,source2:dest2` .

     Por ejemplo:

```
tridentctl-protect create snapshotrestore <my_restore_name>
--snapshot <namespace/snapshot_to_restore> --namespace-mapping
<source_to_destination_namespace_mapping>
```

Administrar los ganchos de ejecución de Trident Protect

Un gancho de ejecución es una acción personalizada que puede configurar para que se ejecute junto con una operación de protección de datos de una aplicación administrada. Por ejemplo, si tiene una aplicación de base de datos, puede usar un gancho de ejecución para pausar todas las transacciones de la base de datos antes de una instantánea y reanudarlas una vez que se complete la instantánea. Esto garantiza instantáneas consistentes con la aplicación.

Tipos de ganchos de ejecución

Trident Protect admite los siguientes tipos de ganchos de ejecución, según cuándo se puedan ejecutar:

- Pre-instantánea
- Post-instantánea
- Copia de seguridad previa
- Post-copia de seguridad
- Post-restauración
- Post-conmutación por error

Orden de ejecución

Cuando se ejecuta una operación de protección de datos, los eventos de enlace de ejecución tienen lugar en el siguiente orden:

1. Todos los ganchos de ejecución de preoperación personalizados aplicables se ejecutan en los contenedores apropiados. Puede crear y ejecutar tantos ganchos previos a la operación personalizados como necesite, pero el orden de ejecución de estos ganchos antes de la operación no está garantizado ni es configurable.
2. Si corresponde, se producen bloqueos del sistema de archivos. "Obtenga más información sobre cómo configurar la congelación del sistema de archivos con Trident Protect".
3. Se realiza la operación de protección de datos.
4. Los sistemas de archivos congelados se descongelan, si corresponde.
5. Cualquier gancho de ejecución posterior a la operación personalizado aplicable se ejecuta en los contenedores apropiados. Puede crear y ejecutar tantos ganchos posteriores a la operación personalizados como necesite, pero el orden de ejecución de estos ganchos después de la operación no está garantizado ni es configurable.

Si crea varios ganchos de ejecución del mismo tipo (por ejemplo, pre-instantánea), no se garantiza el orden de ejecución de esos ganchos. Sin embargo, el orden de ejecución de los ganchos de diferentes tipos está garantizado. Por ejemplo, el siguiente es el orden de ejecución de una configuración que tiene todos los

diferentes tipos de ganchos:

1. Ganchos previos a la instantánea ejecutados
2. Ganchos posteriores a la instantánea ejecutados
3. Ganchos de pre-copia de seguridad ejecutados
4. Ganchos posteriores a la copia de seguridad ejecutados

El ejemplo de orden anterior solo se aplica cuando se ejecuta una copia de seguridad que no utiliza una instantánea existente.

Siempre debe probar sus scripts de gancho de ejecución antes de habilitarlos en un entorno de producción. Puede utilizar el comando 'kubectl exec' para probar los scripts cómodamente. Después de habilitar los ganchos de ejecución en un entorno de producción, pruebe las instantáneas y las copias de seguridad resultantes para asegurarse de que sean consistentes. Puede hacerlo clonando la aplicación en un espacio de nombres temporal, restaurando la instantánea o la copia de seguridad y luego probando la aplicación.

Si un gancho de ejecución previo a la instantánea agrega, cambia o elimina recursos de Kubernetes, esos cambios se incluyen en la instantánea o la copia de seguridad y en cualquier operación de restauración posterior.

Notas importantes sobre los ganchos de ejecución personalizados

Tenga en cuenta lo siguiente al planificar ganchos de ejecución para sus aplicaciones.

- Un gancho de ejecución debe utilizar un script para realizar acciones. Muchos ganchos de ejecución pueden hacer referencia al mismo script.
- Trident Protect requiere que los scripts que utilizan los ganchos de ejecución estén escritos en el formato de scripts de shell ejecutables.
- El tamaño del script está limitado a 96 KB.
- Trident Protect utiliza configuraciones de gancho de ejecución y cualquier criterio coincidente para determinar qué ganchos son aplicables a una operación de instantánea, copia de seguridad o restauración.

Debido a que los ganchos de ejecución a menudo reducen o deshabilitan por completo la funcionalidad de la aplicación en la que se ejecutan, siempre debe intentar minimizar el tiempo que tardan en ejecutarse sus ganchos de ejecución personalizados. Si inicia una operación de copia de seguridad o instantánea con ganchos de ejecución asociados pero luego la cancela, los ganchos aún podrán ejecutarse si la operación de copia de seguridad o instantánea ya ha comenzado. Esto significa que la lógica utilizada en un gancho de ejecución posterior a una copia de seguridad no puede asumir que la copia de seguridad se completó.

Filtros de gancho de ejecución

Cuando agrega o edita un gancho de ejecución para una aplicación, puede agregar filtros al gancho de ejecución para administrar con qué contenedores coincidirá el gancho. Los filtros son útiles para las aplicaciones que utilizan la misma imagen de contenedor en todos los contenedores, pero pueden usar cada imagen para un propósito diferente (como Elasticsearch). Los filtros le permiten crear escenarios en los que los ganchos de ejecución se ejecutan en algunos, pero no necesariamente en todos los contenedores idénticos. Si crea varios filtros para un único gancho de ejecución, se combinan con un operador AND lógico.

Puede tener hasta 10 filtros activos por gancho de ejecución.

Cada filtro que agrega a un gancho de ejecución utiliza una expresión regular para que coincida con los contenedores en su clúster. Cuando un gancho coincide con un contenedor, el gancho ejecutará su script asociado en ese contenedor. Las expresiones regulares para filtros utilizan la sintaxis de Expresión regular 2 (RE2), que no admite la creación de un filtro que excluya contenedores de la lista de coincidencias. Para obtener información sobre la sintaxis que Trident Protect admite para expresiones regulares en filtros de gancho de ejecución, consulte "Compatibilidad con la sintaxis de expresiones regulares 2 (RE2)" .

> Si agrega un filtro de espacio de nombres a un gancho de ejecución que se ejecuta después de una operación de restauración o clonación y el origen y el destino de la restauración o clonación están en espacios de nombres diferentes, el filtro de espacio de nombres solo se aplica al espacio de nombres de destino.

Ejemplos de ganchos de ejecución

Visita el "Proyecto NetApp Verda en GitHub" para descargar ganchos de ejecución reales para aplicaciones populares como Apache Cassandra y Elasticsearch. También puede ver ejemplos y obtener ideas para estructurar sus propios ganchos de ejecución personalizados.

Crea un gancho de ejecución

Puede crear un gancho de ejecución personalizado para una aplicación usando . Necesitas tener permisos de Propietario, Administrador o Miembro para crear ganchos de ejecución.

Utilice un CR

Pasos

1. Crea el archivo de recursos personalizados (CR) y nómbralo. `trident-protect-hook.yaml` .
2. Configure los siguientes atributos para que coincidan con su entorno de Trident Protect y la configuración del clúster:
   - **metadata.name**: (*Obligatorio*) El nombre de este recurso personalizado; elija un nombre único y adecuado para su entorno.
   - **spec.applicationRef**: (*Obligatorio*) El nombre de Kubernetes de la aplicación para la cual se ejecutará el gancho de ejecución.
   - **spec.stage**: (*Obligatorio*) Una cadena que indica en qué etapa de la acción debe ejecutarse el gancho de ejecución. Valores posibles:
     - Pre
     - Correo
   - **spec.action**: (*Obligatorio*) Una cadena que indica qué acción tomará el gancho de ejecución, suponiendo que coincidan los filtros del gancho de ejecución especificados. Valores posibles:
     - Snapshot
     - Respaldo
     - Restaurar
     - Conmutación por error
   - **spec.enabled**: (*Opcional*) Indica si este gancho de ejecución está habilitado o deshabilitado. Si no se especifica, el valor predeterminado es verdadero.
   - **spec.hookSource**: (*Obligatorio*) Una cadena que contiene el script de hook codificado en base64.
   - **spec.timeout**: (*Opcional*) Un número que define cuánto tiempo en minutos se le permite ejecutar el gancho de ejecución. El valor mínimo es de 1 minuto, y el valor predeterminado es de 25 minutos si no se especifica.
   - **spec.arguments**: (*Opcional*) Una lista YAML de argumentos que puede especificar para el gancho de ejecución.
   - **spec.matchingCriteria**: (*Opcional*) Una lista opcional de pares clave-valor de criterios, cada par conforma un filtro de gancho de ejecución. Puedes agregar hasta 10 filtros por gancho de ejecución.
   - **spec.matchingCriteria.type**: (*Opcional*) Una cadena que identifica el tipo de filtro del gancho de ejecución. Valores posibles:
     - Imagen de contenedor
     - Nombre del contenedor
     - Nombre del pod
     - PodLabel
     - Nombre del espacio de nombres
   - **spec.matchingCriteria.value**: (*Opcional*) Una cadena o expresión regular que identifica el valor del filtro del gancho de ejecución.

     Ejemplo de YAML:

```
apiVersion: protect.trident.netapp.io/v1
kind: ExecHook
metadata:
  name: example-hook-cr
  namespace: my-app-namespace
  annotations:
    astra.netapp.io/astra-control-hook-source-id:
/account/test/hookSource/id
spec:
  applicationRef: my-app-name
  stage: Pre
  action: Snapshot
  enabled: true
  hookSource: IyEvYmluL2Jhc2gKZWNobyAiZXhhbXBsZSBzY3JpcHQiCg==
  timeout: 10
  arguments:
    - FirstExampleArg
    - SecondExampleArg
  matchingCriteria:
    - type: containerName
      value: mysql
    - type: containerImage
      value: bitnami/mysql
    - type: podName
      value: mysql
    - type: namespaceName
      value: mysql-a
    - type: podLabel
      value: app.kubernetes.io/component=primary
    - type: podLabel
      value: helm.sh/chart=mysql-10.1.0
    - type: podLabel
      value: deployment-type=production
```

3. Después de completar el archivo CR con los valores correctos, aplique el CR:

```
kubectl apply -f trident-protect-hook.yaml
```

Utilice la CLI

Pasos

1. Crea el gancho de ejecución, reemplazando los valores entre corchetes con información de tu entorno. Por ejemplo:

```
tridentctl-protect create exechook <my_exec_hook_name> --action
<action_type> --app <app_to_use_hook> --stage <pre_or_post_stage>
--source-file <script-file> -n <application_namespace>
```

Ejecutar manualmente un gancho de ejecución

Puede ejecutar manualmente un gancho de ejecución con fines de prueba o si necesita volver a ejecutar el gancho manualmente después de un fallo. Necesitas tener permisos de Propietario, Administrador o Miembro para ejecutar manualmente los ganchos de ejecución.

La ejecución manual de un gancho de ejecución consta de dos pasos básicos:

1. Crea una copia de seguridad de recursos, que recopila los recursos y crea una copia de seguridad de ellos, determinando dónde se ejecutará el hook.
2. Ejecutar el gancho de ejecución contra la copia de seguridad

Paso 1: Crear una copia de seguridad del recurso

Utilice un CR

Pasos

1. Crea el archivo de recursos personalizados (CR) y nómbralo. `trident-protect-resource-backup.yaml` .
2. Configure los siguientes atributos para que coincidan con su entorno de Trident Protect y la configuración del clúster:
   - **metadata.name**: (*Obligatorio*) El nombre de este recurso personalizado; elija un nombre único y adecuado para su entorno.
   - **spec.applicationRef**: (*Obligatorio*) El nombre de Kubernetes de la aplicación para la cual se creará la copia de seguridad del recurso.
   - **spec.appVaultRef**: (*Obligatorio*) El nombre del AppVault donde se almacenan los contenidos de la copia de seguridad.
   - **spec.appArchivePath**: La ruta dentro de AppVault donde se almacenan los contenidos de la copia de seguridad. Puedes utilizar el siguiente comando para encontrar esta ruta:

     ```
     kubectl get backups <BACKUP_NAME> -n my-app-namespace -o jsonpath='{.status.appArchivePath}'
     ```

     Ejemplo de YAML:

```
---
apiVersion: protect.trident.netapp.io/v1
kind: ResourceBackup
metadata:
  name: example-resource-backup
spec:
  applicationRef: my-app-name
  appVaultRef: my-appvault-name
  appArchivePath: example-resource-backup
```

3. Después de completar el archivo CR con los valores correctos, aplique el CR:

```
kubectl apply -f trident-protect-resource-backup.yaml
```

Utilice la CLI

Pasos

1. Crea la copia de seguridad, reemplazando los valores entre corchetes con información de tu entorno. Por ejemplo:

```
tridentctl protect create resourcebackup <my_backup_name> --app
<my_app_name> --appvault <my_appvault_name> -n
<my_app_namespace> --app-archive-path <app_archive_path>
```

2. Consulta el estado de la copia de seguridad. Puedes utilizar este comando de ejemplo repetidamente hasta que la operación se complete:

```
tridentctl protect get resourcebackup -n <my_app_namespace>
<my_backup_name>
```

3. Verifique que la copia de seguridad se haya realizado correctamente:

```
kubectl describe resourcebackup <my_backup_name>
```

Paso 2: Ejecutar el gancho de ejecución

Utilice un CR

Pasos

1. Crea el archivo de recursos personalizados (CR) y nómbralo. `trident-protect-hook-run.yaml` .
2. Configure los siguientes atributos para que coincidan con su entorno de Trident Protect y la configuración del clúster:
   - **metadata.name**: (*Obligatorio*) El nombre de este recurso personalizado; elija un nombre único y adecuado para su entorno.
   - **spec.applicationRef**: (*Obligatorio*) Asegúrese de que este valor coincida con el nombre de la aplicación del CR de ResourceBackup que creó en el paso 1.
   - **spec.appVaultRef**: (*Obligatorio*) Asegúrese de que este valor coincida con el appVaultRef del CR de ResourceBackup que creó en el paso 1.
   - **spec.appArchivePath**: Asegúrese de que este valor coincida con el appArchivePath del CR ResourceBackup que creó en el paso 1.

     ```
     kubectl get backups <BACKUP_NAME> -n my-app-namespace -o
     jsonpath='{.status.appArchivePath}'
     ```

   - **spec.action**: (*Obligatorio*) Una cadena que indica qué acción tomará el gancho de ejecución, suponiendo que coincidan los filtros del gancho de ejecución especificados. Valores posibles:
     - Snapshot
     - Respaldo
     - Restaurar
     - Conmutación por error
   - **spec.stage**: (*Obligatorio*) Una cadena que indica en qué etapa de la acción debe ejecutarse el gancho de ejecución. Esta tanda de anzuelos no utilizará anzuelos en ninguna otra etapa. Valores posibles:
     - Pre
     - Correo

     Ejemplo de YAML:

```
---
apiVersion: protect.trident.netapp.io/v1
kind: ExecHooksRun
metadata:
  name: example-hook-run
spec:
  applicationRef: my-app-name
  appVaultRef: my-appvault-name
  appArchivePath: example-resource-backup
  stage: Post
  action: Failover
```

3. Después de completar el archivo CR con los valores correctos, aplique el CR:

```
kubectl apply -f trident-protect-hook-run.yaml
```

Utilice la CLI

Pasos

1. Cree la solicitud de ejecución manual del gancho:

```
tridentctl protect create exechooksrun <my_exec_hook_run_name>
-n <my_app_namespace> --action snapshot --stage <pre_or_post>
--app <my_app_name> --appvault <my_appvault_name> --path
<my_backup_name>
```

2. Compruebe el estado de la ejecución del gancho. Puedes ejecutar este comando repetidamente hasta que la operación se complete:

```
tridentctl protect get exechooksrun -n <my_app_namespace>
<my_exec_hook_run_name>
```

3. Describa el objeto exechooksrun para ver los detalles finales y el estado:

```
kubectl -n <my_app_namespace> describe exechooksrun
<my_exec_hook_run_name>
```

Desinstalar Trident Protect

Es posible que necesites eliminar los componentes de Trident Protect si estás actualizando desde una versión de prueba a una versión completa del producto.

Para eliminar Trident Protect, realice los siguientes pasos.

Pasos

1. Eliminar los archivos CR de Trident Protect:

   Este paso no es necesario para la versión 25.06 y posteriores.

   ```
   helm uninstall -n trident-protect trident-protect-crds
   ```

2. Eliminar Trident Protect:

   ```
   helm uninstall -n trident-protect trident-protect
   ```

3. Eliminar el espacio de nombres Trident Protect:

   ```
   kubectl delete ns trident-protect
   ```

Blogs de Trident y Trident Protect

Puede encontrar algunos excelentes blogs NetApp Trident y Trident Protect aquí:

Blogs de Trident

- 9 de mayo de 2025:"Configuración automática del backend Trident para FSx para ONTAP con el complemento Amazon EKS"
- 19 de agosto de 2025:"Mejora de la consistencia de los datos: instantáneas de grupos de volúmenes en la virtualización de OpenShift con Trident"
- 15 de abril de 2025:"NetApp Trident con Google Cloud NetApp Volumes para protocolo SMB"
- 14 de abril de 2025:"Aprovechamiento del protocolo de canal de fibra con Trident 25.02 para el almacenamiento persistente en Kubernetes"
- 14 de abril de 2025:"Desbloqueando el potencial de los sistemas NetApp ASA r2 para el almacenamiento en bloque de Kubernetes"
- 31 de marzo de 2025:"Simplificación de la instalación de Trident en Red Hat OpenShift con el nuevo operador certificado"
- 27 de marzo de 2025:"Aprovisionamiento de Trident para pymes con Google Cloud NetApp Volumes"
- 5 de marzo de 2025:"Desbloquea la integración perfecta del almacenamiento iSCSI: una guía para FSxN en clústeres ROSA para AWS"
- 27 de febrero de 2025:"Implementación de identidad en la nube con Trident, GKE y Google Cloud NetApp Volumes"
- 12 de diciembre de 2024:"Presentamos la compatibilidad con Fibre Channel en Trident"
- 26 de noviembre de 2024:"Trident 25.01: Mejora de la experiencia de almacenamiento de Kubernetes con nuevas funciones y mejoras"
- 11 de noviembre de 2024:"NetApp Trident con Google Cloud NetApp Volumes"
- 29 de octubre de 2024:"Amazon FSx for NetApp ONTAP con Red Hat OpenShift Service en AWS (ROSA) mediante Trident"
- 29 de octubre de 2024:"Migración en vivo de máquinas virtuales con OpenShift Virtualization en ROSA y Amazon FSx for NetApp ONTAP"
- 8 de julio de 2024:"Uso de NVMe/TCP para consumir almacenamiento ONTAP para sus aplicaciones modernas en contenedores en Amazon EKS"
- 1 de julio de 2024:"Almacenamiento de Kubernetes sin interrupciones con Google Cloud NetApp Volumes Flex y Astra Trident"
- 11 de junio de 2024:"ONTAP como almacenamiento backend para el registro de imágenes integrado en OpenShift"

Blogs de Trident Protect

- 16 de mayo de 2025:"Automatización de la conmutación por error del registro para la recuperación ante desastres con los ganchos posteriores a la restauración de Trident Protect"
- 16 de mayo de 2025:"Recuperación ante desastres de virtualización de OpenShift con NetApp Trident Protect"

- 13 de mayo de 2025:"Migración de clases de almacenamiento con copia de seguridad y restauración de Trident Protect"
- 9 de mayo de 2025:"Reescalar aplicaciones de Kubernetes con los ganchos posteriores a la restauración de Trident Protect"
- 3 de abril de 2025:"Trident Protect Power Up: Replicación de Kubernetes para protección y recuperación ante desastres"
- 13 de marzo de 2025:"Operaciones de copia de seguridad y restauración coherentes ante fallos para máquinas virtuales de virtualización de OpenShift"
- 11 de marzo de 2025:"Ampliación de los patrones de GitOps a la protección de datos de aplicaciones con NetApp Trident"
- 3 de marzo de 2025:"Trident 25.02: Mejorando la experiencia de Red Hat OpenShift con nuevas e interesantes funciones"
- 15 de enero de 2025:"Presentamos el control de acceso basado en roles de Trident Protect"
- 11 de noviembre de 2024: "Presentamos tridentctl protect: la potente CLI para Trident Protect"
- 11 de noviembre de 2024:"Gestión de datos impulsada por Kubernetes: la nueva era con Trident Protect"

Conocimiento y apoyo

Preguntas frecuentes

Encuentre respuestas a las preguntas más frecuentes sobre la instalación, configuración, actualización y solución de problemas de Trident.

Preguntas generales

¿Con qué frecuencia se lanza Trident ?

A partir del lanzamiento del 24 de febrero, Trident se lanza cada cuatro meses: febrero, junio y octubre.

¿Es compatible Trident con todas las características que se lanzan en una versión específica de Kubernetes?

Trident normalmente no admite funciones alfa en Kubernetes. Trident podría admitir funciones beta en las dos versiones de Trident que sigan a la versión beta de Kubernetes.

¿Depende Trident de algún otro producto de NetApp para su funcionamiento?

Trident no tiene dependencias de otros productos de software de NetApp y funciona como una aplicación independiente. Sin embargo, deberá disponer de un dispositivo de almacenamiento backend de NetApp .

¿Cómo puedo obtener los detalles completos de la configuración de Trident ?

Utilice el `tridentctl get` comando para obtener más información sobre su configuración de Trident .

¿Puedo obtener métricas sobre cómo Trident aprovisiona el almacenamiento?

Sí. Puntos de conexión de Prometheus que se pueden utilizar para recopilar información sobre el funcionamiento de Trident , como el número de backends gestionados, el número de volúmenes aprovisionados, los bytes consumidos, etc. También puedes usar"Cloud Insights" para el seguimiento y análisis.

¿Cambia la experiencia del usuario al utilizar Trident como aprovisionador CSI?

No. No hay cambios en lo que respecta a la experiencia del usuario y las funcionalidades. El nombre del proveedor utilizado es `csi.trident.netapp.io` . Se recomienda este método de instalación de Trident si desea utilizar todas las nuevas funciones que ofrecen las versiones actuales y futuras.

Instala y usa Trident en un clúster de Kubernetes.

¿Admite Trident la instalación sin conexión desde un registro privado?

Sí, Trident se puede instalar sin conexión a internet. Referirse a"Aprenda sobre la instalación de Trident" .

¿Puedo instalar Trident de forma remota?

Sí. Trident 18.10 y versiones posteriores admiten la capacidad de instalación remota desde cualquier máquina que tenga `kubectl` acceso al clúster. Después `kubectl` Se verifica el acceso (por ejemplo, al iniciar un

`kubectl get nodes` Comando desde la máquina remota para verificar), siga las instrucciones de instalación.

¿Puedo configurar la alta disponibilidad con Trident?

Trident se instala como un Deployment de Kubernetes (ReplicaSet) con una sola instancia, por lo que cuenta con alta disponibilidad integrada. No se recomienda aumentar el número de réplicas en el deployment. Si se pierde el nodo donde está instalado Trident o el pod resulta inaccesible por algún otro motivo, Kubernetes vuelve a implementar automáticamente el pod en un nodo en buen estado de su clúster. Trident solo funciona en el plano de control, por lo que las cápsulas actualmente instaladas no se verán afectadas si se vuelve a desplegar Trident .

¿Necesita Trident acceso al espacio de nombres kube-system?

Trident lee del servidor de la API de Kubernetes para determinar cuándo las aplicaciones solicitan nuevos PVC, por lo que necesita acceso a kube-system.

¿Cuáles son los roles y privilegios utilizados por Trident?

El instalador de Trident crea un Kubernetes ClusterRole, que tiene acceso específico a los recursos PersistentVolume, PersistentVolumeClaim, StorageClass y Secret del clúster de Kubernetes. Referirse a"Personalizar la instalación de tridentctl" .

¿Puedo generar localmente los archivos de manifiesto exactos que Trident utiliza para la instalación?

Si es necesario, puede generar y modificar localmente los archivos de manifiesto exactos que Trident utiliza para la instalación. Referirse a"Personalizar la instalación de tridentctl" .

¿Puedo compartir la misma SVM de backend de ONTAP para dos instancias de Trident separadas en dos clústeres de Kubernetes separados?

Aunque no es recomendable, puede utilizar el mismo SVM de backend para dos instancias de Trident . Especifique un nombre de volumen único para cada instancia durante la instalación o especifique un único `StoragePrefix` parámetro en el `setup/backend.json` archivo. Esto es para garantizar que no se utilice el mismo FlexVol volume para ambas instancias.

¿Es posible instalar Trident en ContainerLinux (anteriormente CoreOS)?

Trident es simplemente un pod de Kubernetes y se puede instalar dondequiera que se esté ejecutando Kubernetes.

¿Puedo usar Trident con NetApp Cloud Volumes ONTAP?

Sí, Trident es compatible con AWS, Google Cloud y Azure.

¿Es compatible Trident con Cloud Volumes Services?

Sí, Trident es compatible con el servicio Azure NetApp Files en Azure, así como con el Cloud Volumes Service en GCP.

Solución de problemas y soporte

¿ NetApp ofrece soporte Trident?

Aunque Trident es de código abierto y se proporciona de forma gratuita, NetApp lo admite completamente siempre que su backend de NetApp sea compatible.

¿Cómo puedo presentar una solicitud de asistencia?

Para abrir un caso de soporte, realice una de las siguientes acciones:

1. Contacta con tu gestor de cuenta de soporte y obtén ayuda para abrir una incidencia.
2. Abra un caso de soporte poniéndose en contacto con "Soporte de NetApp" .

¿Cómo genero un paquete de registros de soporte?

Puedes crear un paquete de soporte ejecutando `tridentctl logs -a` . Además de los registros capturados en el paquete, capture el registro de kubelet para diagnosticar los problemas de montaje en el lado de Kubernetes. Las instrucciones para obtener el registro de kubelet varían según cómo esté instalado Kubernetes.

¿Qué debo hacer si necesito solicitar una nueva funcionalidad?

Crea un problema en "Trident Github" y mencione **RFE** en el asunto y la descripción del problema.

¿Dónde puedo reportar un defecto?

Crea un problema en "Trident Github" . Asegúrese de incluir toda la información y los registros necesarios relacionados con el problema.

¿Qué ocurre si tengo una pregunta rápida sobre Trident que necesito aclarar? ¿Existe alguna comunidad o foro?

Si tiene alguna pregunta, problema o solicitud, póngase en contacto con nosotros a través de nuestro Trident."Canal de Discord" o GitHub.

La contraseña de mi sistema de almacenamiento ha cambiado y Trident ya no funciona, ¿cómo puedo recuperarlo?

Actualiza la contraseña del backend con `tridentctl update backend myBackend -f </path/to_new_backend.json> -n trident` . Reemplazar `myBackend` en el ejemplo con el nombre de su backend, y `` `/path/to_new_backend.json `` con el camino correcto `backend.json` archivo.

Trident no puede encontrar mi nodo de Kubernetes. ¿Cómo puedo solucionar esto?

Existen dos posibles escenarios por los que Trident no puede encontrar un nodo de Kubernetes. Puede deberse a un problema de red dentro de Kubernetes o a un problema de DNS. El daemonset del nodo Trident que se ejecuta en cada nodo de Kubernetes debe poder comunicarse con el controlador Trident para registrar el nodo con Trident. Si se produjeron cambios en la red después de instalar Trident , este problema solo se presentará con los nuevos nodos de Kubernetes que se agreguen al clúster.

Si la cápsula Trident se destruye, ¿perderé los datos?

Los datos no se perderán si la cápsula Trident es destruida. Los metadatos de Trident se almacenan en objetos CRD. Todos los PV que hayan sido aprovisionados por Trident funcionarán con normalidad.

Mejora el Trident

¿Puedo actualizar directamente de una versión anterior a una versión más reciente (saltándome algunas versiones)?

NetApp admite la actualización de Trident de una versión principal a la siguiente versión principal inmediata. Puedes actualizar de la versión 18.xx a la 19.xx, de la 19.xx a la 20.xx, y así sucesivamente. Deberías probar la actualización en un entorno de laboratorio antes de su implementación en producción.

¿Es posible degradar Trident a una versión anterior?

Si necesita solucionar errores detectados tras una actualización, problemas de dependencias o una actualización fallida o incompleta, debería"Desinstalar Trident" y reinstale la versión anterior siguiendo las instrucciones específicas para esa versión. Esta es la única forma recomendada de volver a una versión anterior.

Gestionar backends y volúmenes

¿Necesito definir tanto ManagementLIF como DataLIF en un archivo de definición de backend de ONTAP ?

La gestión del LIF es obligatoria. DataLIF varía:

- ONTAP SAN: No especificar para iSCSI. Trident utiliza"Mapa selectivo de LUN de ONTAP" para descubrir los LIF de iSCI necesarios para establecer una sesión de múltiples rutas. Se genera una advertencia si `dataLIF` está definido explícitamente. Referirse a "Opciones y ejemplos de configuración de SAN de ONTAP" Para más detalles.
- NAS ONTAP : NetApp recomienda especificar `dataLIF` . Si no se proporcionan, Trident obtiene los dataLIF del SVM. Puede especificar un nombre de dominio completo (FQDN) para usarlo en las operaciones de montaje NFS, lo que le permite crear un DNS round-robin para equilibrar la carga entre varios dataLIF. Referirse a"Opciones y ejemplos de configuración de ONTAP NAS" para más detalles

¿Puede Trident configurar CHAP para backends ONTAP ?

Sí. Trident admite CHAP bidireccional para backends ONTAP . Esto requiere configurar `useCHAP=true` en la configuración de tu backend.

¿Cómo gestiono las políticas de exportación con Trident?

Trident puede crear y gestionar políticas de exportación de forma dinámica a partir de la versión 20.04. Esto permite al administrador de almacenamiento proporcionar uno o más bloques CIDR en su configuración de backend y hacer que Trident agregue las direcciones IP de los nodos que se encuentren dentro de estos rangos a una política de exportación que cree. De esta forma, Trident gestiona automáticamente la adición y eliminación de reglas para nodos con direcciones IP dentro de los CIDR dados.

¿Se pueden utilizar direcciones IPv6 para las LIF de gestión y datos?

Trident permite definir direcciones IPv6 para:

- `managementLIF`y `dataLIF` para backends NAS de ONTAP .
- `managementLIF`para backends SAN de ONTAP . No puedes especificar `dataLIF` en un backend SAN ONTAP .

Trident debe instalarse usando el flag `--use-ipv6` (para `tridentctl` instalación), `IPv6` (para el operador Trident), o `tridentTPv6` (para la instalación de Helm) para que funcione sobre IPv6.

¿Es posible actualizar el LIF de gestión en el backend?

Sí, es posible actualizar el LIF de gestión del backend utilizando el `tridentctl update backend` dominio.

¿Es posible actualizar DataLIF en el backend?

Puedes actualizar DataLIF en `ontap-nas` y `ontap-nas-economy` solo.

¿Puedo crear múltiples backends en Trident para Kubernetes?

Trident puede admitir muchos backends simultáneamente, ya sea con el mismo controlador o con controladores diferentes.

¿Cómo almacena Trident las credenciales del backend?

Trident almacena las credenciales del backend como secretos de Kubernetes.

¿Cómo selecciona Trident un backend específico?

Si los atributos del backend no se pueden usar para seleccionar automáticamente los pools adecuados para una clase, el `storagePools` y `additionalStoragePools` Los parámetros se utilizan para seleccionar un conjunto específico de pools.

¿Cómo puedo asegurarme de que Trident no realice el aprovisionamiento desde un backend específico?

El `excludeStoragePools` Este parámetro se utiliza para filtrar el conjunto de pools que Trident utiliza para el aprovisionamiento y eliminará cualquier pool que coincida.

Si existen varios backends del mismo tipo, ¿cómo selecciona Trident cuál backend utilizar?

Si existen varios backends configurados del mismo tipo, Trident selecciona el backend apropiado en función de los parámetros presentes en `StorageClass` y `PersistentVolumeClaim` . Por ejemplo, si existen varios backends de controladores ontap-nas, Trident intenta hacer coincidir los parámetros en el `StorageClass` y `PersistentVolumeClaim` combinados y compatibles con un backend que pueda cumplir con los requisitos enumerados en `StorageClass` y `PersistentVolumeClaim` . Si existen varios servidores backend que coinciden con la solicitud, Trident selecciona uno de ellos al azar.

¿Admite Trident CHAP bidireccional con Element/ SolidFire?

Sí.

¿Cómo implementa Trident Qtrees en un volumen ONTAP ? ¿Cuántos Qtrees se pueden implementar en un solo volumen?

El `ontap-nas-economy` El controlador crea hasta 200 Qtrees en el mismo FlexVol volume (configurable entre 50 y 300), 100.000 Qtrees por nodo de clúster y 2,4 millones por clúster. Cuando entras en un nuevo `PersistentVolumeClaim` Ese servicio lo proporciona el controlador de economía; el controlador comprueba si ya existe un FlexVol volume que pueda dar servicio al nuevo Qtree. Si no existe un FlexVol volume que pueda dar servicio al Qtree, se crea un nuevo FlexVol volume .

¿Cómo puedo configurar los permisos Unix para los volúmenes aprovisionados en ONTAP NAS?

Puede configurar los permisos Unix en el volumen aprovisionado por Trident estableciendo un parámetro en el archivo de definición del backend.

¿Cómo puedo configurar un conjunto explícito de opciones de montaje NFS de ONTAP al aprovisionar un volumen?

Por defecto, Trident no establece ninguna opción de montaje con Kubernetes. Para especificar las opciones de montaje en la clase de almacenamiento de Kubernetes, siga el ejemplo proporcionado."aquí" .

¿Cómo configuro los volúmenes aprovisionados con una política de exportación específica?

Para permitir que los hosts apropiados accedan a un volumen, utilice el `exportPolicy` parámetro configurado en el archivo de definición del backend.

¿Cómo configuro el cifrado de volumen a través de Trident con ONTAP?

Puede configurar el cifrado en el volumen aprovisionado por Trident utilizando el parámetro de cifrado en el archivo de definición del backend. Para obtener más información, consulte:"Cómo funciona Trident con NVE y NAE"

¿Cuál es la mejor manera de implementar QoS para ONTAP a través de Trident?

Usar `StorageClasses` Implementar QoS para ONTAP.

¿Cómo puedo especificar el aprovisionamiento ligero o grueso a través de Trident?

Los controladores ONTAP admiten aprovisionamiento ligero o grueso. Los controladores ONTAP utilizan por defecto el aprovisionamiento ligero. Si se desea un aprovisionamiento grueso, debe configurar el archivo de definición del backend o el `StorageClass` . Si ambos están configurados, `StorageClass` tiene prioridad. Configure lo siguiente para ONTAP:

1. En `StorageClass` , establecer el `provisioningType` atributo como grueso.
2. En el archivo de definición del backend, habilite los volúmenes gruesos configurando `backend spaceReserve parameter` como volumen.

¿Cómo puedo asegurarme de que los volúmenes en uso no se eliminen incluso si borro accidentalmente el PVC?

La protección PVC se habilita automáticamente en Kubernetes a partir de la versión 1.10.

¿Puedo cultivar PVC NFS creados con Trident?

Sí. Puedes expandir un tubo de PVC que haya sido creado por Trident. Tenga en cuenta que el crecimiento automático de volumen es una característica de ONTAP que no es aplicable a Trident.

¿Puedo importar un volumen mientras está en modo de protección de datos (DP) de SnapMirror o en modo sin conexión?

La importación de volumen falla si el volumen externo está en modo DP o está fuera de línea. Recibes el siguiente mensaje de error:

```
Error: could not import volume: volume import failed to get size of
volume: volume <name> was not found (400 Bad Request) command terminated
with exit code 1.
Make sure to remove the DP mode or put the volume online before importing
the volume.
```

¿Cómo se traduce la cuota de recursos a un clúster NetApp ?

La cuota de recursos de almacenamiento de Kubernetes debería funcionar siempre que el almacenamiento de NetApp tenga capacidad. Cuando el almacenamiento de NetApp no puede respetar la configuración de cuota de Kubernetes debido a la falta de capacidad, Trident intenta realizar el aprovisionamiento, pero se produce un error.

¿Puedo crear instantáneas de volumen usando Trident?

Sí. Trident admite la creación de instantáneas de volumen bajo demanda y volúmenes persistentes a partir de instantáneas. Para crear volúmenes persistentes a partir de instantáneas, asegúrese de que `VolumeSnapshotDataSource` La puerta de funciones se ha habilitado.

¿Qué controladores admiten instantáneas de volumen de Trident ?

A partir de hoy, el soporte de instantáneas bajo demanda está disponible para nuestros `ontap-nas` , `ontap-nas-flexgroup` , `ontap-san` , `ontap-san-economy` , `solidfire-san` , `gcp-cvs` , y `azure-netapp-files` Controladores de backend.

¿Cómo puedo realizar una copia de seguridad instantánea de un volumen aprovisionado por Trident con ONTAP?

Esto está disponible en `ontap-nas` , `ontap-san` , y `ontap-nas-flexgroup` conductores. También puedes especificar un `snapshotPolicy` Para el `ontap-san-economy` Controlador a nivel FlexVol .

Esto también está disponible en el `ontap-nas-economy` controladores, pero con una granularidad a nivel de FlexVol volume y no a nivel de qtree. Para habilitar la capacidad de crear instantáneas de volúmenes aprovisionados por Trident, configure la opción de parámetro de backend. `snapshotPolicy` a la política de instantáneas deseada tal como se define en el backend de ONTAP . Trident desconoce cualquier instantánea tomada por el controlador de almacenamiento.

¿Puedo establecer un porcentaje de reserva de instantáneas para un volumen aprovisionado a través de Trident?

Sí, puedes reservar un porcentaje específico de espacio en disco para almacenar las copias de instantáneas a través de Trident configurando la opción `snapshotReserve` atributo en el archivo de definición del backend. Si lo has configurado `snapshotPolicy` y `snapshotReserve` En el archivo de definición del backend, el porcentaje de reserva de instantáneas se establece de acuerdo con `snapshotReserve` porcentaje mencionado en el archivo backend. Si el `snapshotReserve` No se menciona el porcentaje; ONTAP, por defecto, toma el porcentaje de reserva de instantáneas como 5. Si el `snapshotPolicy` Si la opción está configurada en ninguna, el porcentaje de reserva de instantáneas se establece en 0.

¿Puedo acceder directamente al directorio de instantáneas de volumen y copiar archivos?

Sí, puedes acceder al directorio de instantáneas en el volumen aprovisionado por Trident configurando

`snapshotDir` parámetro en el archivo de definición del backend.

¿Puedo configurar SnapMirror para volúmenes a través de Trident?

Actualmente, SnapMirror debe configurarse externamente utilizando ONTAP CLI o OnCommand System Manager.

¿Cómo puedo restaurar volúmenes persistentes a una instantánea ONTAP específica?

Para restaurar un volumen a una instantánea de ONTAP , siga los siguientes pasos:

1. Quitar el pod de la aplicación que está utilizando el volumen persistente.
2. Restaure la instantánea requerida a través de ONTAP CLI o OnCommand System Manager.
3. Reinicie el pod de la aplicación.

¿Puede Trident aprovisionar volúmenes en SVM que tienen configurado un espejo de reparto de carga?

Se pueden crear réplicas de reparto de carga para los volúmenes raíz de las SVM que sirven datos a través de NFS. ONTAP actualiza automáticamente los espejos de reparto de carga para los volúmenes que han sido creados por Trident. Esto puede ocasionar retrasos en el aumento de los volúmenes. Cuando se crean varios volúmenes utilizando Trident, el aprovisionamiento de un volumen depende de que ONTAP actualice el espejo de reparto de carga.

¿Cómo puedo separar el uso de la clase de almacenamiento para cada cliente/inquilino?

Kubernetes no permite clases de almacenamiento en espacios de nombres. Sin embargo, puede usar Kubernetes para limitar el uso de una clase de almacenamiento específica por espacio de nombres mediante el uso de cuotas de recursos de almacenamiento, que son por espacio de nombres. Para denegar el acceso de un espacio de nombres específico a un almacenamiento específico, establezca la cuota de recursos en 0 para esa clase de almacenamiento.

Solución de problemas

Utilice las indicaciones que se proporcionan aquí para solucionar los problemas que pueda encontrar al instalar y usar Trident.

> Para obtener ayuda con Trident, cree un paquete de soporte usando `tridentctl logs -a -n trident` y envíelo al soporte de NetApp .

Solución de problemas generales

- Si la cápsula Trident no se eleva correctamente (por ejemplo, cuando la cápsula Trident se queda atascada en el `ContainerCreating` fase con menos de dos contenedores listos), en ejecución `kubectl -n trident describe deployment trident` y `kubectl -n trident describe pod trident-**` puede proporcionar información adicional. Obtención de registros de kubelet (por ejemplo, mediante `journalctl -xeu kubelet`) también puede ser útil.
- Si los registros de Trident no contienen suficiente información, puede intentar habilitar el modo de depuración para Trident pasando el parámetro correspondiente. `-d` Se añade una bandera al parámetro de instalación según la opción de instalación seleccionada.

A continuación, confirme que la depuración está configurada mediante `./tridentctl logs -n trident` y buscando `level=debug msg` en el registro.

Instalado con el operador

```
kubectl patch torc trident -n <namespace> --type=merge -p
'{"spec":{"debug":true}}'
```

Esto reiniciará todos los pods de Trident , lo que puede tardar varios segundos. Puedes comprobarlo observando la columna 'EDAD' en la salida de `kubectl get pod -n trident` .

Para Trident 20.07 y 20.10 utilice `tprov` en lugar de `torc` .

Instalado con Helm

```
helm upgrade <name> trident-operator-21.07.1-custom.tgz --set
tridentDebug=true`
```

Instalado con tridentctl

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

- También puedes obtener registros de depuración para cada backend incluyendo `debugTraceFlags` en la definición de tu backend. Por ejemplo, incluir `debugTraceFlags: {"api":true, "method":true,}` para obtener las llamadas a la API y los recorridos de métodos en los registros de Trident . Los sistemas backend existentes pueden tener `debugTraceFlags` configurado con un `tridentctl backend update` .
- Al utilizar Red Hat Enterprise Linux CoreOS (RHCOS), asegúrese de que `iscsid` Está habilitado en los nodos de trabajo y se inicia de forma predeterminada. Esto se puede hacer utilizando OpenShift MachineConfigs o modificando las plantillas de Ignition.
- Un problema común que podría encontrar al usar Trident con "Azure NetApp Files" Se produce cuando los secretos del inquilino y del cliente provienen de un registro de aplicación con permisos insuficientes. Para obtener una lista completa de los requisitos de Trident , consulte"Azure NetApp Files" configuración.
- Si hay problemas al montar un panel fotovoltaico en un contenedor, asegúrese de que `rpcbind` Está instalado y en funcionamiento. Utilice el gestor de paquetes requerido para el sistema operativo anfitrión y compruebe si `rpcbind` está en funcionamiento. Puedes comprobar el estado de `rpcbind` servicio mediante la ejecución de un `systemctl status rpcbind` o su equivalente.
- Si un backend de Trident informa que está en el `failed` Aunque este estado haya funcionado anteriormente, es probable que se deba a un cambio en las credenciales de SVM/administrador asociadas con el backend. Actualizando la información del backend usando `tridentctl update backend` o bien, hacer rebotar el pod Trident solucionará este problema.
- Si encuentra problemas de permisos al instalar Trident con Docker como entorno de ejecución de contenedores, intente la instalación de Trident con `--in cluster=false` bandera. Esto no utilizará un pod de instalación y evitará los problemas de permisos observados debido a `trident-installer` usuario.
- Utilice el `uninstall parameter <Uninstalling Trident>` para limpiar después de una ejecución

fallida. Por defecto, el script no elimina los CRD que han sido creados por Trident, lo que hace que sea seguro desinstalarlo y volverlo a instalar incluso en una implementación en ejecución.

- Si quieres volver a una versión anterior de Trident, primero ejecuta el `tridentctl uninstall` comando para eliminar Trident. Descarga el archivo deseado "Versión Trident" e instalar usando el `tridentctl install` dominio.
- Tras una instalación correcta, si un tubo de PVC queda atascado en el `Pending` fase, corriendo `kubectl describe pvc` puede proporcionar información adicional sobre por qué Trident no pudo aprovisionar un PV para este PVC.

Despliegue fallido de Trident utilizando el operador

Si está implementando Trident utilizando el operador, el estado de `TridentOrchestrator` cambios de `Installing` a `Installed` . Si observas el `Failed` Si el operador no puede recuperarse por sí mismo, revise los registros del operador ejecutando el siguiente comando:

```
tridentctl logs -l trident-operator
```

El seguimiento de los registros del contenedor trident-operator puede indicar dónde reside el problema. Por ejemplo, uno de esos problemas podría ser la incapacidad de obtener las imágenes de contenedor necesarias de los registros ascendentes en un entorno aislado.

Para comprender por qué fracasó la instalación de Trident , debería echar un vistazo a… `TridentOrchestrator` estado.

```
kubectl describe torc trident-2
Name:         trident-2
Namespace:
Labels:       <none>
Annotations:  <none>
API Version:  trident.netapp.io/v1
Kind:         TridentOrchestrator
...
Status:
  Current Installation Params:
    IPv6:
    Autosupport Hostname:
    Autosupport Image:
    Autosupport Proxy:
    Autosupport Serial Number:
    Debug:
    Image Pull Secrets:         <nil>
    Image Registry:
    k8sTimeout:
    Kubelet Dir:
    Log Format:
    Silence Autosupport:
    Trident Image:
  Message:                      Trident is bound to another CR 'trident'
  Namespace:                    trident-2
  Status:                       Error
  Version:
Events:
  Type     Reason  Age                From                          Message
  ----     ------  ----               ----                          -------
  Warning  Error   16s (x2 over 16s)  trident-operator.netapp.io  Trident
is bound to another CR 'trident'
```

Este error indica que ya existe un `TridentOrchestrator` que se utilizó para instalar Trident. Dado que cada clúster de Kubernetes solo puede tener una instancia de Trident, el operador garantiza que en cualquier momento dado solo exista una instancia activa. `TridentOrchestrator` que puede crear.

Además, observar el estado de las cápsulas Trident a menudo puede indicar si algo no funciona correctamente.

```
kubectl get pods -n trident

NAME                                READY   STATUS             RESTARTS
AGE
trident-csi-4p5kq                   1/2     ImagePullBackOff   0
5m18s
trident-csi-6f45bfd8b6-vfrkw        4/5     ImagePullBackOff   0
5m19s
trident-csi-9q5xc                   1/2     ImagePullBackOff   0
5m18s
trident-csi-9v95z                   1/2     ImagePullBackOff   0
5m18s
trident-operator-766f7b8658-ldzsv   1/1     Running            0
8m17s
```

Se puede observar claramente que los pods no pueden inicializarse completamente porque no se han descargado una o más imágenes de contenedor.

Para solucionar el problema, debes editar el `TridentOrchestrator` CR. Alternativamente, puede eliminar `TridentOrchestrator` y crear una nueva con la definición modificada y precisa.

Despliegue fallido de Trident utilizando `tridentctl`

Para ayudar a averiguar qué salió mal, podría volver a ejecutar el instalador usando el`-d` argumento, que activará el modo de depuración y le ayudará a comprender cuál es el problema:

```
./tridentctl install -n trident -d
```

Tras solucionar el problema, puede limpiar la instalación del siguiente modo y, a continuación, ejecutar el programa. `tridentctl install` comando de nuevo:

```
./tridentctl uninstall -n trident
INFO Deleted Trident deployment.
INFO Deleted cluster role binding.
INFO Deleted cluster role.
INFO Deleted service account.
INFO Removed Trident user from security context constraint.
INFO Trident uninstallation succeeded.
```

Eliminar completamente Trident y CRD

Puedes eliminar completamente Trident y todos los CRD creados y los recursos personalizados asociados.

Esto no se puede deshacer. No haga esto a menos que desee una instalación completamente nueva de Trident. Para desinstalar Trident sin eliminar los CRD, consulte"Desinstalar Trident" .

Operador de Trident

Para desinstalar Trident y eliminar por completo los CRD utilizando el operador Trident :

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

Timón

Para desinstalar Trident y eliminar completamente los CRD usando Helm:

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

<code>tridentctl</code>

Para eliminar completamente los CRD después de desinstalar Trident, utilice `tridentctl`

```
tridentctl obliviate crd
```

Fallo en la desorganización del nodo NVMe con espacios de nombres de bloques sin formato RWX en Kubernetes 1.26

Si está ejecutando Kubernetes 1.26, la desinstalación de nodos podría fallar al usar NVMe/TCP con espacios de nombres de bloques sin formato RWX. Los siguientes escenarios ofrecen soluciones alternativas al fallo. Como alternativa, puede actualizar Kubernetes a la versión 1.27.

Se eliminó el espacio de nombres y el pod.

Considere un escenario en el que tiene un espacio de nombres administrado Trident (volumen persistente NVMe) conectado a un pod. Si elimina el espacio de nombres directamente desde el backend de ONTAP , el proceso de desensamblaje se bloquea después de intentar eliminar el pod. Este escenario no afecta al clúster de Kubernetes ni a otros funcionamientos.

Solución alternativa

Desmonte el volumen persistente (correspondiente a ese espacio de nombres) del nodo respectivo y elimínelo.

LIF de datos bloqueados

```
 If you block (or bring down) all the dataLIFs of the NVMe Trident
backend, the unstaging process gets stuck when you attempt to delete the
pod. In this scenario, you cannot run any NVMe CLI commands on the
Kubernetes node.
.Solución alternativa
Active dataLIFS para restaurar la funcionalidad completa.
```

Asignación de espacio de nombres eliminada

```
 If you remove the `hostNQN` of the worker node from the corresponding
subsystem, the unstaging process gets stuck when you attempt to delete the
pod. In this scenario, you cannot run any NVMe CLI commands on the
Kubernetes node.
.Solución alternativa
Añade el `hostNQN` Volvamos al subsistema.
```

Los clientes NFSv4.2 informan "argumento no válido" después de actualizar ONTAP cuando esperaban que "v4.2-xattrs" estuviera habilitado

Después de actualizar ONTAP, los clientes NFSv4.2 pueden informar errores de "argumento inválido" al intentar montar exportaciones NFSv4.2. Este problema ocurre cuando el `v4.2-xattrs` La opción no está habilitada en el SVM. .Solución alternativa Habilite la `v4.2-xattrs` opción en SVM o actualizar a ONTAP 9.12.1 o posterior, donde esta opción está habilitada de forma predeterminada.

Soporte

NetApp ofrece soporte para Trident de diversas maneras. Dispone de amplias opciones de autoayuda gratuitas las 24 horas del día, los 7 días de la semana, como artículos de la base de conocimientos (KB) y un canal de Discord.

Soporte Trident

Trident ofrece tres niveles de soporte según su versión. Referirse a"Compatibilidad de versiones de software de NetApp con definiciones" .

Soporte completo
: Trident ofrece soporte completo durante doce meses a partir de la fecha de lanzamiento.

Apoyo limitado
: Trident ofrece soporte limitado durante los meses 13 a 24 a partir de la fecha de lanzamiento.

Autosuficiencia
: La documentación de Trident está disponible para los meses 25 a 36 a partir de la fecha de lanzamiento.

Versión	Soporte completo	Apoyo limitado	Autosuficiencia

"25,06"	junio de 2026	junio de 2027	junio de 2028
"25,02"	febrero de 2026	febrero de 2027	febrero de 2028
"24,10"	octubre de 2025	octubre de 2026	octubre de 2027
"24,06"	junio de 2025	junio de 2026	junio de 2027
"24,02"	febrero de 2025	febrero de 2026	febrero de 2027
"23,10"	—	octubre de 2025	octubre de 2026
"23,07"	—	julio de 2025	julio de 2026
"23,04"	—	abril de 2025	abril de 2026
"23,01"	—	—	enero de 2026
"22,10"	—	—	octubre de 2025

Autosuficiencia

Para obtener una lista completa de artículos de solución de problemas, consulte "Base de conocimientos de NetApp (se requiere iniciar sesión)" .

Apoyo comunitario

Existe una activa comunidad pública de usuarios de contenedores (incluidos los desarrolladores de Trident) en nuestra plataforma."Canal de Discord" . Este es un excelente lugar para hacer preguntas generales sobre el proyecto y debatir temas relacionados con compañeros con intereses similares.

Soporte técnico de NetApp

Para obtener ayuda con Trident, cree un paquete de soporte usando `tridentctl logs -a -n trident` y envíalo a `NetApp Support <Getting Help>` .

Para más información

- "Recursos de Trident"
- "Centro de Kubernetes"

Referencia

Puertos Trident

Obtén más información sobre los puertos que Trident utiliza para la comunicación.

Puertos Trident

Trident utiliza los siguientes puertos para la comunicación dentro de Kubernetes:

Puerto	Objetivo
8443	HTTPS de canal de retorno
8001	punto de conexión de métricas de Prometheus
8000	Servidor REST Trident
17546	Puerto de sonda de actividad/preparación utilizado por los pods del daemonset de Trident

El puerto de la sonda de actividad/preparación se puede cambiar durante la instalación utilizando el `--probe-port` bandera. Es importante asegurarse de que este puerto no esté siendo utilizado por otro proceso en los nodos de trabajo.

API REST de Trident

Mientras"Comandos y opciones de tridentctl" son la forma más sencilla de interactuar con la API REST de Trident ; si lo prefiere, puede utilizar el punto de conexión REST directamente.

Cuándo usar la API REST

La API REST es útil para instalaciones avanzadas que utilizan Trident como binario independiente en implementaciones que no son de Kubernetes.

Para mayor seguridad, el Trident `REST API` Por defecto, está restringido a localhost cuando se ejecuta dentro de un pod. Para cambiar este comportamiento, debes configurar Trident `-address` argumento en su configuración de pod.

Utilizando la API REST

Para ver ejemplos de cómo se llaman estas API, pasa el paquete de depuración.(`-d`) bandera. Para obtener más información, consulte"Gestiona Trident usando tridentctl" .

La API funciona de la siguiente manera:

CONSEGUIR

`GET <trident-address>/trident/v1/<object-type>`

Enumera todos los objetos de ese tipo.

`GET <trident-address>/trident/v1/<object-type>/<object-name>`

Obtiene los detalles del objeto nombrado.

CORREO

`POST <trident-address>/trident/v1/<object-type>`

Crea un objeto del tipo especificado.

- Requiere una configuración JSON para que se cree el objeto. Para la especificación de cada tipo de objeto, consulte"Gestiona Trident usando tridentctl" .
- Si el objeto ya existe, el comportamiento varía: los backends actualizan el objeto existente, mientras que todos los demás tipos de objetos fallarán la operación.

BORRAR

`DELETE <trident-address>/trident/v1/<object-type>/<object-name>`

Elimina el recurso especificado.

> Los volúmenes asociados con los sistemas backend o las clases de almacenamiento seguirán existiendo; estos deben eliminarse por separado. Para obtener más información, consulte"Gestiona Trident usando tridentctl" .

Opciones de línea de comandos

Trident expone varias opciones de línea de comandos para el orquestador Trident . Puedes utilizar estas opciones para modificar tu implementación.

Explotación florestal

`-debug`

Habilita la salida de depuración.

`-loglevel <level>`

Establece el nivel de registro (depuración, información, advertencia, error, fatal). Valor predeterminado: información.

Kubernetes

`-k8s_pod`

Utilice esta opción o `-k8s_api_server` para habilitar la compatibilidad con Kubernetes. Al configurar esto, Trident utilizará las credenciales de la cuenta de servicio de Kubernetes del pod que lo contiene para contactar con el servidor de la API. Esto solo funciona cuando Trident se ejecuta como un pod en un clúster de Kubernetes con cuentas de servicio habilitadas.

-k8s_api_server <insecure-address:insecure-port>

Utilice esta opción o `-k8s_pod` para habilitar la compatibilidad con Kubernetes. Cuando se especifica, Trident se conecta al servidor de la API de Kubernetes utilizando la dirección y el puerto no seguros proporcionados. Esto permite que Trident se implemente fuera de un pod; sin embargo, solo admite conexiones no seguras al servidor de la API. Para conectarse de forma segura, implemente Trident en un pod con el `-k8s_pod` opción.

Docker

-volume_driver <name>

Nombre del controlador utilizado al registrar el complemento de Docker. Valor predeterminado `netapp` .

-driver_port <port-number>

Escuche en este puerto en lugar de en un socket de dominio UNIX.

-config <file>

Obligatorio; debe especificar esta ruta a un archivo de configuración del backend.

DESCANSAR

-address <ip-or-host>

Especifica la dirección en la que debe escuchar el servidor REST de Trident. Por defecto, se utiliza localhost. Al escuchar en localhost y ejecutarse dentro de un pod de Kubernetes, la interfaz REST no es directamente accesible desde fuera del pod. Usar `-address ""` para que la interfaz REST sea accesible desde la dirección IP del pod.

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

-port <port-number>

Especifica el puerto en el que debe escuchar el servidor REST de Trident. Valor predeterminado: 8000.

-rest

Habilita la interfaz REST. Valor predeterminado: verdadero.

Objetos de Kubernetes y Trident

Puedes interactuar con Kubernetes y Trident utilizando API REST para leer y escribir objetos de recursos. Existen varios objetos de recursos que dictan la relación entre Kubernetes y Trident, Trident y el almacenamiento, y Kubernetes y el almacenamiento. Algunos de estos objetos se gestionan a través de Kubernetes y otros a través de Trident.

¿Cómo interactúan los objetos entre sí?

Quizás la forma más sencilla de comprender los objetos, para qué sirven y cómo interactúan, sea seguir una única solicitud de almacenamiento de un usuario de Kubernetes:

1. Un usuario crea un `PersistentVolumeClaim` solicitando un nuevo `PersistentVolume` de un tamaño determinado de un Kubernetes `StorageClass` que fue configurado previamente por el administrador.
2. Kubernetes `StorageClass` identifica a Trident como su proveedor e incluye parámetros que le indican a Trident cómo aprovisionar un volumen para la clase solicitada.
3. Trident analiza su propia situación. `StorageClass` con el mismo nombre que identifica la coincidencia `Backends` y `StoragePools` que puede utilizar para aprovisionar volúmenes para la clase.
4. Trident proporciona almacenamiento en un backend coincidente y crea dos objetos: un `PersistentVolume` en Kubernetes que le indica a Kubernetes cómo encontrar, montar y tratar el volumen, y un volumen en Trident que mantiene la relación entre el `PersistentVolume` y el almacenamiento real.
5. Kubernetes enlaza el `PersistentVolumeClaim` a lo nuevo `PersistentVolume` . Cápsulas que incluyen `PersistentVolumeClaim` Monta ese PersistentVolume en cualquier host en el que se ejecute.
6. Un usuario crea un `VolumeSnapshot` de un PVC existente, utilizando un `VolumeSnapshotClass` eso apunta a Trident.
7. Trident identifica el volumen asociado al PVC y crea una instantánea del volumen en su sistema backend. También crea un `VolumeSnapshotContent` que le indica a Kubernetes cómo identificar la instantánea.
8. Un usuario puede crear un `PersistentVolumeClaim` usando `VolumeSnapshot` como fuente.
9. Trident identifica la instantánea requerida y realiza el mismo conjunto de pasos involucrados en la creación de una `PersistentVolume` y un `Volume` .

Para obtener más información sobre los objetos de Kubernetes, le recomendamos encarecidamente que lea el "Volúmenes persistentes" sección de la documentación de Kubernetes.

Kubernetes `PersistentVolumeClaim` objetos

Kubernetes `PersistentVolumeClaim` El objeto es una solicitud de almacenamiento realizada por un usuario de un clúster de Kubernetes.

Además de la especificación estándar, Trident permite a los usuarios especificar las siguientes anotaciones específicas del volumen si desean anular los valores predeterminados que se establecen en la configuración del backend:

Anotación	Opción de volumen	Controladores compatibles
trident.netapp.io/sistema de archivos	sistema de archivos	ontap-san, solidfire-san, economía ontap-san
trident.netapp.io/cloneFromPVC	clonar volumen de origen	ontap-nas, ontap-san, solidfire-san, azure-netapp-files, gcp-cvs, ontap-san-economy
trident.netapp.io/splitOnClone	dividirEnClon	ontap-nas, ontap-san
trident.netapp.io/protocolo	protocolo	cualquier
trident.netapp.io/exportPolicy	Política de exportación	ontap-nas, ontap-nas-economy, ontap-nas-flexgroup

Anotación	Opción de volumen	Controladores compatibles
trident.netapp.io/snapshotPolicy	política de instantáneas	ontap-nas, ontap-nas-economy, ontap-nas-flexgroup, ontap-san
trident.netapp.io/snapshotReserve	snapshotReserve	ontap-nas, ontap-nas-flexgroup, ontap-san, gcp-cvs
trident.netapp.io/directorio de instantáneas	directorio de instantáneas	ontap-nas, ontap-nas-economy, ontap-nas-flexgroup
trident.netapp.io/unixPermissions	permisos unix	ontap-nas, ontap-nas-economy, ontap-nas-flexgroup
trident.netapp.io/tamaño de bloque	tamaño del bloque	solidfire-san

Si el PV creado tiene el `Delete` Según la política de recuperación, Trident elimina tanto el PV como el volumen de respaldo cuando se libera el PV (es decir, cuando el usuario elimina el PVC). Si la acción de eliminación falla, Trident marca el PV como tal y reintenta periódicamente la operación hasta que tenga éxito o el PV se elimine manualmente. Si el PV utiliza el `Retain` Según la política de Trident , esto se ignora y se asume que el administrador lo eliminará de Kubernetes y del backend, lo que permite realizar una copia de seguridad o inspeccionar el volumen antes de su eliminación. Tenga en cuenta que eliminar el PV no provoca que Trident elimine el volumen de respaldo. Debes eliminarlo usando la API REST.(`tridentctl`).

Trident admite la creación de instantáneas de volumen mediante la especificación CSI: puede crear una instantánea de volumen y usarla como fuente de datos para clonar PVC existentes. De esta forma, se pueden exponer a Kubernetes copias puntuales de los PV en forma de instantáneas. Las instantáneas se pueden utilizar posteriormente para crear nuevos PV. Echa un vistazo a `On-Demand Volume Snapshots` para ver cómo funcionaría esto.

Trident también proporciona `cloneFromPVC` y `splitOnClone` Anotaciones para la creación de clones. Puedes usar estas anotaciones para clonar un PVC sin tener que usar la implementación CSI.

Aquí hay un ejemplo: Si un usuario ya tiene un PVC llamado `mysql` , el usuario puede crear un nuevo PVC llamado `mysqlclone` mediante el uso de la anotación, como por ejemplo `trident.netapp.io/cloneFromPVC: mysql` . Con este conjunto de anotaciones, Trident clona el volumen correspondiente al PVC de MySQL, en lugar de aprovisionar un volumen desde cero.

Considere los siguientes puntos:

- NetApp recomienda clonar un volumen inactivo.
- Un PVC y su clon deben estar en el mismo espacio de nombres de Kubernetes y tener la misma clase de almacenamiento.
- Con el `ontap-nas` y `ontap-san` Para los controladores, podría ser conveniente configurar la anotación PVC. `trident.netapp.io/splitOnClone` en conjunto con `trident.netapp.io/cloneFromPVC` . Con `trident.netapp.io/splitOnClone` empezar a `true` Trident separa el volumen clonado del volumen original y, por lo tanto, desacopla completamente el ciclo de vida del volumen clonado de su original a costa de perder cierta eficiencia de almacenamiento. No se configura `trident.netapp.io/splitOnClone` o configurarlo para `false` Esto da como resultado un menor consumo de espacio en el backend, a costa de crear dependencias entre los volúmenes padre y clon, de modo que el volumen padre no se puede eliminar a menos que primero se elimine el clon. Un escenario donde tiene sentido dividir el clon es clonar un volumen de base de datos vacío donde se espera que el volumen y su clon diverjan mucho y no se beneficien de las eficiencias de almacenamiento que ofrece ONTAP.

El `sample-input` El directorio contiene ejemplos de definiciones de PVC para usar con Trident. Referirse a Para obtener una descripción completa de los parámetros y la configuración asociados con los volúmenes de Trident .

Kubernetes `PersistentVolume` objetos

Kubernetes `PersistentVolume` El objeto representa una porción de almacenamiento que se pone a disposición del clúster de Kubernetes. Tiene un ciclo de vida independiente del pod que lo utiliza.

> Trident crea `PersistentVolume` objetos y los registra automáticamente en el clúster de Kubernetes en función de los volúmenes que aprovisiona. No se espera que usted los gestione.

Cuando creas un PVC que hace referencia a un Trident basado en `StorageClass` Trident aprovisiona un nuevo volumen utilizando la clase de almacenamiento correspondiente y registra un nuevo PV para ese volumen. Al configurar el volumen aprovisionado y el PV correspondiente, Trident sigue las siguientes reglas:

- Trident genera un nombre de PV para Kubernetes y un nombre interno que utiliza para aprovisionar el almacenamiento. En ambos casos, resulta tranquilizador que los nombres sean únicos en su ámbito.
- El tamaño del volumen coincide lo más fielmente posible con el tamaño solicitado en el PVC, aunque podría redondearse al alza a la cantidad asignable más cercana, dependiendo de la plataforma.

Kubernetes `StorageClass` objetos

Kubernetes `StorageClass` Los objetos se especifican por nombre en `PersistentVolumeClaims` para aprovisionar el almacenamiento con un conjunto de propiedades. La propia clase de almacenamiento identifica el aprovisionador que se utilizará y define ese conjunto de propiedades en términos que el aprovisionador entiende.

Es uno de los dos objetos básicos que debe crear y gestionar el administrador. El otro es el objeto backend de Trident .

Kubernetes `StorageClass` Un objeto que utiliza Trident tiene este aspecto:

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: <Name>
provisioner: csi.trident.netapp.io
mountOptions: <Mount Options>
parameters: <Trident Parameters>
allowVolumeExpansion: true
volumeBindingMode: Immediate
```

Estos parámetros son específicos de Trident e indican a Trident cómo aprovisionar volúmenes para la clase.

Los parámetros de la clase de almacenamiento son:

Atributo	Tipo	Requerido	Descripción
atributos	map[cadena]cadena	No	Consulte la sección de atributos a continuación.
Grupos de almacenamiento	map[string]StringList	No	Mapa de nombres de backend a listas de grupos de almacenamiento dentro de
Grupos de almacenamiento adicionales	map[string]StringList	No	Mapa de nombres de backend a listas de grupos de almacenamiento dentro de
excluirStoragePools	map[string]StringList	No	Mapa de nombres de backend a listas de grupos de almacenamiento dentro de

Los atributos de almacenamiento y sus posibles valores se pueden clasificar en atributos de selección de grupo de almacenamiento y atributos de Kubernetes.

atributos de selección del grupo de almacenamiento

Estos parámetros determinan qué pools de almacenamiento gestionados por Trident deben utilizarse para aprovisionar volúmenes de un tipo determinado.

Atributo	Tipo	Valores	Oferta	Pedido	Con el apoyo de
medios[1]	cadena	disco duro, híbrido, SSD	La piscina contiene medios de este tipo; híbrido significa ambos	Tipo de medio especificado	ontap-nas, ontap-nas-economy, ontap-nas-flexgroup, ontap-san, solidfire-san
tipo de aprovisionamiento	cadena	delgado, grueso	Pool admite este método de aprovisionamiento.	Método de aprovisionamiento especificado	Espeso: todo Ontap; fino: todo Ontap y Solidfire-san

Atributo	Tipo	Valores	Oferta	Pedido	Con el apoyo de
Tipo de backend	cadena	ontap-nas, ontap-nas-economy, ontap-nas-flexgroup, ontap-san, solidfire-san, gcp-cvs, azure-netapp-files, ontap-san-economy	Pool pertenece a este tipo de backend.	Backend especificado	Todos los conductores
instantáneas	bool	verdadero, falso	El pool admite volúmenes con instantáneas.	Volumen con instantáneas habilitadas	ontap-nas, ontap-san, solidfire-san, gcp-cvs
clones	bool	verdadero, falso	Pool admite la clonación de volúmenes.	Volumen con clones habilitado	ontap-nas, ontap-san, solidfire-san, gcp-cvs
cifrado	bool	verdadero, falso	Pool admite volúmenes cifrados	Volumen con cifrado habilitado	ontap-nas, ontap-nas-economy, ontap-nas-flexgroups, ontap-san
IOPS	int	entero positivo	Pool es capaz de garantizar IOPS en este rango.	El volumen garantizaba estas IOPS	solidfire-san

[1]: No compatible con los sistemas ONTAP Select

En la mayoría de los casos, los valores solicitados influyen directamente en el aprovisionamiento; por ejemplo, solicitar un aprovisionamiento grueso da como resultado un volumen con aprovisionamiento grueso. Sin embargo, un grupo de almacenamiento de Element utiliza sus valores mínimos y máximos de IOPS ofrecidos para establecer los valores de QoS, en lugar del valor solicitado. En este caso, el valor solicitado se utiliza únicamente para seleccionar el grupo de almacenamiento.

Idealmente, puedes usar `attributes` por sí solo para modelar las cualidades del almacenamiento que necesita para satisfacer las necesidades de una clase en particular. Trident descubre y selecciona automáticamente los grupos de almacenamiento que coinciden con *todos* los requisitos. `attributes` que usted especifique.

Si no puede usar `attributes` Para seleccionar automáticamente los grupos adecuados para una clase, puede utilizar el `storagePools` y `additionalStoragePools` parámetros para refinar aún más los grupos o incluso para seleccionar un conjunto específico de grupos.

Puedes utilizar el `storagePools` parámetro para restringir aún más el conjunto de pools que coincidan con cualquier especificado `attributes` . En otras palabras, Trident utiliza la intersección de pools identificada por el `attributes` y `storagePools` Parámetros para el aprovisionamiento. Puede utilizar cualquiera de los parámetros por separado o ambos juntos.

Puedes utilizar el `additionalStoragePools` parámetro para ampliar el conjunto de pools que Trident utiliza para el aprovisionamiento, independientemente de los pools seleccionados por el `attributes` y `storagePools` parámetros.

Puedes utilizar el `excludeStoragePools` Parámetro para filtrar el conjunto de pools que Trident utiliza para el aprovisionamiento. El uso de este parámetro elimina cualquier grupo que coincida.

En el `storagePools` y `additionalStoragePools` parámetros, cada entrada toma la forma `<backend>:<storagePoolList>` , dónde `<storagePoolList>` es una lista de grupos de almacenamiento separados por comas para el backend especificado. Por ejemplo, un valor para `additionalStoragePools` podría verse como `ontapnas_192.168.1.100:aggr1,aggr2;solidfire_192.168.1.101:bronze` . Estas listas aceptan valores de expresiones regulares tanto para el backend como para los valores de la lista. Puedes utilizar `tridentctl get backend` para obtener la lista de backends y sus pools.

atributos de Kubernetes

Estos atributos no influyen en la selección de pools/backends de almacenamiento por parte de Trident durante el aprovisionamiento dinámico. En cambio, estos atributos simplemente proporcionan parámetros compatibles con los volúmenes persistentes de Kubernetes. Los nodos de trabajo son responsables de las operaciones de creación del sistema de archivos y pueden requerir utilidades del sistema de archivos, como xfsprogs.

Atributo	Tipo	Valores	Descripción	Factores relevantes	Versión de Kubernetes
fsType	cadena	ext4, ext3, xfs	El tipo de sistema de archivos para volúmenes de bloques	Solidfire-san, ontap-nas, ontap-nas-economy, ontap-nas-flexgroup, ontap-san, ontap-san-economy	Todo
permitirExpansiónDeVolumen	booleano	verdadero, falso	Habilitar o deshabilitar la compatibilidad con el aumento del tamaño del PVC	ontap-nas, ontap-nas-economy, ontap-nas-flexgroup, ontap-san, ontap-san-economy, solidfire-san, gcp-cvs, azure-netapp-files	1.11+
modo de enlace de volumen	cadena	Inmediato, Esperar al primer consumidor	Elija cuándo se produce el enlace de volúmenes y el aprovisionamiento dinámico	Todo	1,19 - 1,26

- El `fsType` Este parámetro se utiliza para controlar el tipo de sistema de archivos deseado para las LUN SAN. Además, Kubernetes también utiliza la presencia de `fsType` en una clase de almacenamiento para indicar que existe un sistema de archivos. La propiedad del volumen se puede controlar mediante el `fsGroup` contexto de seguridad de un pod solo si `fsType` está listo. Referirse a"Kubernetes: Configurar un contexto de seguridad para un pod o contenedor" Para obtener una descripción general sobre cómo configurar la propiedad del volumen utilizando `fsGroup` contexto. Kubernetes aplicará `fsGroup` valor solo si:

  - `fsType`se establece en la clase de almacenamiento.
  - El modo de acceso al PVC es RWO.

  Para los controladores de almacenamiento NFS, ya existe un sistema de archivos como parte de la exportación NFS. Para usar `fsGroup` La clase de almacenamiento aún necesita especificar un `fsType` Puedes configurarlo para `nfs` o cualquier valor no nulo.

- Referirse a"Ampliar volúmenes" Para obtener más detalles sobre la ampliación de volumen.
- El paquete de instalación de Trident proporciona varias definiciones de clases de almacenamiento de ejemplo para usar con Trident en`sample-input/storage-class-*.yaml` . Eliminar una clase de almacenamiento de Kubernetes provoca que también se elimine la clase de almacenamiento de Trident correspondiente.

Kubernetes `VolumeSnapshotClass` objetos

Kubernetes `VolumeSnapshotClass` Los objetos son análogos a `StorageClasses` . Ayudan a definir múltiples clases de almacenamiento y las instantáneas de volumen las utilizan como referencia para asociar la instantánea con la clase de instantánea requerida. Cada instantánea de volumen está asociada a una única clase de instantánea de volumen.

A `VolumeSnapshotClass` Debe ser definido por un administrador para poder crear instantáneas. Se crea una clase de instantánea de volumen con la siguiente definición:

```
apiVersion: snapshot.storage.k8s.io/v1
kind: VolumeSnapshotClass
metadata:
  name: csi-snapclass
driver: csi.trident.netapp.io
deletionPolicy: Delete
```

El `driver` especifica a Kubernetes que las solicitudes de instantáneas de volumen de `csi-snapclass` Las clases son gestionadas por Trident. El `deletionPolicy` Especifica la acción que se debe realizar cuando se deba eliminar una instantánea. Cuando `deletionPolicy` está configurado para `Delete` Cuando se elimina una instantánea, se eliminan tanto los objetos de instantánea de volumen como la instantánea subyacente en el clúster de almacenamiento. Alternativamente, configurándolo a `Retain` significa que `VolumeSnapshotContent` y se conserva la instantánea física.

Kubernetes `VolumeSnapshot` objetos

Kubernetes `VolumeSnapshot` El objeto es una solicitud para crear una instantánea de un volumen. Así como

un PVC representa una solicitud realizada por un usuario para obtener un volumen, una instantánea de volumen es una solicitud realizada por un usuario para crear una instantánea de un PVC existente.

Cuando se recibe una solicitud de instantánea de volumen, Trident gestiona automáticamente la creación de la instantánea para el volumen en el backend y la expone creando un identificador único. `VolumeSnapshotContent` objeto. Puede crear instantáneas a partir de PVC existentes y utilizarlas como fuente de datos al crear nuevos PVC.

> El ciclo de vida de un VolumeSnapshot es independiente del PVC de origen: una instantánea persiste incluso después de que se elimine el PVC de origen. Al eliminar un PVC que tiene instantáneas asociadas, Trident marca el volumen de respaldo de este PVC en estado **Eliminando**, pero no lo elimina por completo. El volumen se elimina cuando se borran todas las instantáneas asociadas.

Kubernetes `VolumeSnapshotContent` objetos

Kubernetes `VolumeSnapshotContent` El objeto representa una instantánea tomada de un volumen ya aprovisionado. Es análogo a un `PersistentVolume` y significa una instantánea aprovisionada en el clúster de almacenamiento. Similar a `PersistentVolumeClaim` y `PersistentVolume` objetos, cuando se crea una instantánea, `VolumeSnapshotContent` el objeto mantiene una correspondencia uno a uno con el `VolumeSnapshot` objeto, que había solicitado la creación de la instantánea.

El `VolumeSnapshotContent` El objeto contiene detalles que identifican de forma única la instantánea, como por ejemplo: `snapshotHandle` . Este `snapshotHandle` es una combinación única del nombre del PV y el nombre del `VolumeSnapshotContent` objeto.

Cuando llega una solicitud de instantánea, Trident crea la instantánea en el backend. Después de crear la instantánea, Trident configura un `VolumeSnapshotContent` objeto y, por lo tanto, expone la instantánea a la API de Kubernetes.

> Normalmente, no necesitas gestionar el `VolumeSnapshotContent` objeto. Una excepción a esto es cuando quieres"Importar una instantánea de volumen" Creado fuera de Trident.

Kubernetes `VolumeGroupSnapshotClass` objetos

Kubernetes `VolumeGroupSnapshotClass` Los objetos son análogos a `VolumeSnapshotClass` . Ayudan a definir múltiples clases de almacenamiento y las instantáneas de grupos de volúmenes las utilizan para asociar la instantánea con la clase de instantánea requerida. Cada instantánea de grupo de volúmenes está asociada a una única clase de instantánea de grupo de volúmenes.

A `VolumeGroupSnapshotClass` Debe ser definido por un administrador para crear un grupo de instantáneas. Se crea una clase de instantánea de grupo de volúmenes con la siguiente definición:

```
apiVersion: groupsnapshot.storage.k8s.io/v1beta1
kind: VolumeGroupSnapshotClass
metadata:
  name: csi-group-snap-class
  annotations:
    kubernetes.io/description: "Trident group snapshot class"
driver: csi.trident.netapp.io
deletionPolicy: Delete
```

El `driver` especifica a Kubernetes que las solicitudes de instantáneas de grupos de volúmenes de `csi-group-snap-class` Las clases son gestionadas por Trident. El `deletionPolicy` Especifica la acción que se debe realizar cuando se deba eliminar una instantánea de grupo. Cuando `deletionPolicy` está configurado para `Delete` Cuando se elimina una instantánea, se eliminan tanto los objetos de instantánea del grupo de volúmenes como la instantánea subyacente en el clúster de almacenamiento. Alternativamente, configurándolo a `Retain` significa que `VolumeGroupSnapshotContent` y se conserva la instantánea física.

Kubernetes `VolumeGroupSnapshot` objetos

Kubernetes `VolumeGroupSnapshot` El objeto es una solicitud para crear una instantánea de múltiples volúmenes. Así como un PVC representa una solicitud realizada por un usuario para obtener un volumen, una instantánea de grupo de volúmenes es una solicitud realizada por un usuario para crear una instantánea de un PVC existente.

Cuando se recibe una solicitud de instantánea de grupo de volúmenes, Trident gestiona automáticamente la creación de la instantánea de grupo para los volúmenes en el backend y la expone creando un identificador único. `VolumeGroupSnapshotContent` objeto. Puede crear instantáneas a partir de PVC existentes y utilizarlas como fuente de datos al crear nuevos PVC.

> El ciclo de vida de un VolumeGroupSnapshot es independiente del PVC de origen: una instantánea persiste incluso después de que se elimine el PVC de origen. Al eliminar un PVC que tiene instantáneas asociadas, Trident marca el volumen de respaldo de este PVC en estado **Eliminando**, pero no lo elimina por completo. La instantánea del grupo de volúmenes se elimina cuando se borran todas las instantáneas asociadas.

Kubernetes `VolumeGroupSnapshotContent` objetos

Kubernetes `VolumeGroupSnapshotContent` El objeto representa una instantánea de grupo tomada de un volumen ya aprovisionado. Es análogo a un `PersistentVolume` y significa una instantánea aprovisionada en el clúster de almacenamiento. Similar a `PersistentVolumeClaim` y `PersistentVolume` objetos, cuando se crea una instantánea, `VolumeSnapshotContent` el objeto mantiene una correspondencia uno a uno con el `VolumeSnapshot` objeto, que había solicitado la creación de la instantánea.

El `VolumeGroupSnapshotContent` El objeto contiene detalles que identifican el grupo de instantáneas, como por ejemplo: `volumeGroupSnapshotHandle` y los identificadores de instantáneas de volumen individuales existentes en el sistema de almacenamiento.

Cuando llega una solicitud de instantánea, Trident crea la instantánea del grupo de volúmenes en el backend. Después de crear la instantánea del grupo de volúmenes, Trident configura un `VolumeGroupSnapshotContent` objeto y, por lo tanto, expone la instantánea a la API de Kubernetes.

Kubernetes `CustomResourceDefinition` objetos

Los recursos personalizados de Kubernetes son puntos de conexión en la API de Kubernetes que son definidos por el administrador y se utilizan para agrupar objetos similares. Kubernetes admite la creación de recursos personalizados para almacenar una colección de objetos. Puede obtener estas definiciones de recursos ejecutando `kubectl get crds` .

Kubernetes almacena las definiciones de recursos personalizados (CRD) y sus metadatos de objetos asociados en su almacén de metadatos. Esto elimina la necesidad de una tienda separada para Trident.

Trident utiliza `CustomResourceDefinition` objetos para preservar la identidad de los objetos Trident , como backends de Trident , clases de almacenamiento de Trident y volúmenes de Trident . Estos objetos son gestionados por Trident. Además, el marco de instantáneas de volumen CSI introduce algunos CRD que son necesarios para definir las instantáneas de volumen.

Los CRD son una construcción de Kubernetes. Los objetos de los recursos definidos anteriormente son creados por Trident. Como ejemplo sencillo, cuando se crea un backend utilizando `tridentctl` , un correspondiente `tridentbackends` El objeto CRD se crea para su uso por Kubernetes.

Aquí hay algunos puntos a tener en cuenta sobre los CRD de Trident:

- Cuando se instala Trident , se crea un conjunto de CRD que se pueden usar como cualquier otro tipo de recurso.
- Al desinstalar Trident utilizando el `tridentctl uninstall` Al ejecutar el comando, los pods de Trident se eliminan, pero los CRD creados no se limpian. Referirse a"Desinstalar Trident" para comprender cómo se puede eliminar completamente Trident y reconfigurarlo desde cero.

Trident `StorageClass` objetos

Trident crea clases de almacenamiento coincidentes para Kubernetes. `StorageClass` objetos que especifican `csi.trident.netapp.io` en su campo de aprovisionamiento. El nombre de la clase de almacenamiento coincide con el de Kubernetes. `StorageClass` objeto que representa.

> Con Kubernetes, estos objetos se crean automáticamente cuando se inicia una red Kubernetes. `StorageClass` que utiliza Trident como proveedor está registrado.

Las clases de almacenamiento comprenden un conjunto de requisitos para los volúmenes. Trident compara estos requisitos con los atributos presentes en cada grupo de almacenamiento; si coinciden, ese grupo de almacenamiento es un destino válido para el aprovisionamiento de volúmenes utilizando esa clase de almacenamiento.

Puede crear configuraciones de clases de almacenamiento para definir directamente las clases de almacenamiento mediante la API REST. Sin embargo, para las implementaciones de Kubernetes, esperamos que se creen al registrar un nuevo Kubernetes. `StorageClass` objetos.

objetos de backend de Trident

Los backends representan los proveedores de almacenamiento sobre los cuales Trident aprovisiona volúmenes; una sola instancia de Trident puede administrar cualquier número de backends.

Este es uno de los dos tipos de objetos que usted crea y administra usted mismo. El otro es Kubernetes. `StorageClass` objeto.

Para obtener más información sobre cómo construir estos objetos, consulte"configuración de backends" .

Trident `StoragePool` objetos

Los grupos de almacenamiento representan las distintas ubicaciones disponibles para el aprovisionamiento en cada backend. Para ONTAP, estos corresponden a agregados en SVM. Para NetApp HCI/ SolidFire, estos corresponden a las bandas QoS especificadas por el administrador. Para el Cloud Volumes Service, estos corresponden a las regiones del proveedor de la nube. Cada grupo de almacenamiento tiene un conjunto de atributos de almacenamiento distintos, que definen sus características de rendimiento y de protección de datos.

A diferencia de los demás objetos aquí presentes, los candidatos a grupo de almacenamiento siempre se descubren y gestionan automáticamente.

Trident `Volume` objetos

Los volúmenes son la unidad básica de aprovisionamiento, que comprende puntos de conexión de backend, como recursos compartidos NFS y LUN iSCSI y FC. En Kubernetes, estos se corresponden directamente con `PersistentVolumes` . Cuando cree un volumen, asegúrese de que tenga una clase de almacenamiento, que determina dónde se puede aprovisionar ese volumen, junto con un tamaño.

- En Kubernetes, estos objetos se gestionan automáticamente. Puedes consultarlos para ver qué provisionó Trident .
- Al eliminar un PV con instantáneas asociadas, el volumen Trident correspondiente se actualiza al estado **Eliminando**. Para eliminar el volumen de Trident , debe eliminar las instantáneas del volumen.

La configuración de un volumen define las propiedades que debe tener un volumen aprovisionado.

Atributo	Tipo	Requerido	Descripción
versión	cadena	No	Versión de la API de Trident ("1")
nombre	cadena	Sí	Nombre del volumen a crear
clase de almacenamiento	cadena	Sí	Clase de almacenamiento que se utilizará al aprovisionar el volumen
tamaño	cadena	Sí	Tamaño del volumen a aprovisionar en bytes
protocolo	cadena	No	Tipo de protocolo a utilizar: "archivo" o "bloque".

Atributo	Tipo	Requerido	Descripción
nombre interno	cadena	No	Nombre del objeto en el sistema de almacenamiento; generado por Trident
clonar volumen de origen	cadena	No	ontap (nas, san) y solidfire-*: Nombre del volumen a clonar
dividirEnClon	cadena	No	ontap (nas, san): Separa el clon de su padre
política de instantáneas	cadena	No	ontap-*: Política de instantánea a utilizar
snapshotReserve	cadena	No	ontap-*: Porcentaje de volumen reservado para instantáneas
Política de exportación	cadena	No	ontap-nas*: Política de exportación a utilizar
directorio de instantáneas	bool	No	ontap-nas*: Indica si el directorio de instantáneas es visible.
permisos unix	cadena	No	ontap-nas*: Permisos UNIX iniciales
tamaño del bloque	cadena	No	solidfire-*: Tamaño del bloque/sector
sistema de archivos	cadena	No	tipo de sistema de archivos

Trident genera `internalName` al crear el volumen. Esto consta de dos pasos. Primero, antepone el prefijo de almacenamiento (ya sea el predeterminado). `trident` o el prefijo en la configuración del backend) al nombre del volumen, lo que da como resultado un nombre de la forma `<prefix>-<volume-name>` . A continuación, procede a sanear el nombre, reemplazando los caracteres no permitidos en el backend. Para los backends de ONTAP , reemplaza los guiones con guiones bajos (por lo tanto, el nombre interno se convierte en `<prefix>_<volume-name>`). Para los backends de Element, reemplaza los guiones bajos con guiones.

Puedes usar configuraciones de volumen para aprovisionar volúmenes directamente mediante la API REST, pero en implementaciones de Kubernetes esperamos que la mayoría de los usuarios utilicen la configuración estándar de Kubernetes. `PersistentVolumeClaim` método. Trident crea este objeto de volumen automáticamente como parte del proceso de aprovisionamiento.

Trident `Snapshot` objetos

Las instantáneas son una copia puntual de los volúmenes, que se puede utilizar para aprovisionar nuevos volúmenes o restaurar el estado. En Kubernetes, estos se corresponden directamente con `VolumeSnapshotContent` objetos. Cada instantánea está asociada a un volumen, que es la fuente de los datos para la instantánea.

Cada `Snapshot` El objeto incluye las propiedades que se enumeran a continuación:

Atributo	Tipo	Requerido	Descripción
versión	Cadena	Sí	Versión de la API de Trident ("1")
nombre	Cadena	Sí	Nombre del objeto de instantánea de Trident
nombre interno	Cadena	Sí	Nombre del objeto de instantánea de Trident en el sistema de almacenamiento
nombre del volumen	Cadena	Sí	Nombre del volumen persistente para el que se crea la instantánea
nombre interno del volumen	Cadena	Sí	Nombre del objeto de volumen Trident asociado en el sistema de almacenamiento

En Kubernetes, estos objetos se gestionan automáticamente. Puedes consultarlos para ver qué provisionó Trident .

Cuando un Kubernetes `VolumeSnapshot` Se crea una solicitud de objeto; Trident funciona creando un objeto de instantánea en el sistema de almacenamiento de respaldo. El `internalName` Este objeto de instantánea se genera combinando el prefijo `snapshot-` con el `UID` del `VolumeSnapshot` objeto (por ejemplo, `snapshot-e8d8a0ca-9826-11e9-9807-525400f3f660`). `volumeName` y `volumeInternalName` se completan obteniendo los detalles del volumen de respaldo.

Trident `ResourceQuota` objeto

El conjunto de demonios Trident consume un `system-node-critical` Clase de prioridad: la clase de prioridad más alta disponible en Kubernetes, para garantizar que Trident pueda identificar y limpiar volúmenes durante el apagado ordenado de nodos y permitir que los pods del conjunto de demonios de Trident prevalezcan sobre las cargas de trabajo con una prioridad más baja en clústeres donde hay una alta presión de recursos.

Para lograrlo, Trident emplea un `ResourceQuota` objeto para garantizar que se cumpla una clase de prioridad "crítica del nodo del sistema" en el conjunto de demonios Trident . Antes del despliegue y la creación del DaemonSet, Trident busca el `ResourceQuota` objeto y, si no lo descubre, lo aplica.

Si necesita un mayor control sobre la cuota de recursos y la clase de prioridad predeterminadas, puede generar un `custom.yaml` o configurar el `ResourceQuota` objeto que utiliza el gráfico Helm.

El siguiente es un ejemplo de un objeto `ResourceQuota` que prioriza el daemonset Trident .

```
apiVersion: <version>
kind: ResourceQuota
metadata:
  name: trident-csi
  labels:
    app: node.csi.trident.netapp.io
spec:
  scopeSelector:
    matchExpressions:
      - operator: In
        scopeName: PriorityClass
        values:
          - system-node-critical
```

Para obtener más información sobre las cuotas de recursos, consulte"Kubernetes: Cuotas de recursos" .

Limpiar `ResourceQuota` si falla la instalación

En el raro caso de que la instalación falle después de la `ResourceQuota` Se crea el objeto, primer intento"desinstalación" y luego reinstalar.

Si eso no funciona, elimine manualmente el `ResourceQuota` objeto.

Eliminar `ResourceQuota`

Si prefieres controlar tu propia asignación de recursos, puedes eliminar el Trident. `ResourceQuota` objeto usando el comando:

```
kubectl delete quota trident-csi -n trident
```

Estándares de seguridad de pods (PSS) y restricciones de contexto de seguridad (SCC)

Los estándares de seguridad de pods (PSS) y las políticas de seguridad de pods (PSP) de Kubernetes definen los niveles de permisos y restringen el comportamiento de los pods. De manera similar, las restricciones del contexto de seguridad (SCC) de OpenShift definen restricciones de pods específicas del motor de Kubernetes de OpenShift. Para ofrecer esta personalización, Trident habilita ciertos permisos durante la instalación. Las siguientes secciones detallan los permisos establecidos por Trident.

PSS reemplaza a las Políticas de Seguridad de Pod (PSP). PSP quedó obsoleto en Kubernetes v1.21 y se eliminará en la versión v1.25. Para obtener más información, consulte"Kubernetes: Seguridad" .

Contexto de seguridad de Kubernetes requerido y campos relacionados

Permiso	Descripción
Privilegiado	CSI requiere que los puntos de montaje sean bidireccionales, lo que significa que el pod del nodo Trident debe ejecutar un contenedor privilegiado. Para obtener más información, consulte"Kubernetes: Propagación de montaje" .
Redes de host	Requerido para el demonio iSCSI. `iscsiadm` Gestiona los montajes iSCSI y utiliza la red del host para comunicarse con el demonio iSCSI.
IPC del host	NFS utiliza comunicación entre procesos (IPC) para comunicarse con NFSD.
PID del host	Requisitos para comenzar `rpc-statd` para NFS. Trident consulta los procesos del host para determinar si `rpc-statd` se ejecuta antes de montar los volúmenes NFS.

Capacidades	El `SYS_ADMIN` Esta capacidad se proporciona como parte de las capacidades predeterminadas para contenedores privilegiados. Por ejemplo, Docker establece estas capacidades para los contenedores privilegiados:
	`CapPrm: 0000003fffffffff`
	`CapEff: 0000003fffffffff`

Seccomp	El perfil Seccomp siempre está "Sin restricciones" en contenedores privilegiados; por lo tanto, no se puede habilitar en Trident.
SELinux	En OpenShift, los contenedores privilegiados se ejecutan en el `spc_t` («Contenedor con privilegios superiores»), y los contenedores sin privilegios se ejecutan en el dominio … sin privilegios»). `container_t` dominio. En `containerd` , con `container-selinux` instalados, todos los contenedores se ejecutan en el `spc_t` dominio, lo que efectivamente deshabilita SELinux. Por lo tanto, Trident no agrega `seLinuxOptions` a los contenedores.
DAC	Los contenedores privilegiados deben ejecutarse como root. Los contenedores sin privilegios se ejecutan como root para acceder a los sockets Unix requeridos por CSI.

Estándares de seguridad de pods (PSS)

Etiqueta	Descripción	Por defecto
`pod-security.kubernetes.io/enforce pod-security.kubernetes.io/enforce-version`	Permite que el controlador Trident y los nodos sean admitidos en el espacio de nombres de instalación. No cambie la etiqueta del espacio de nombres.	`enforce: privileged enforce-version: <version of the current cluster or highest version of PSS tested.>`

> Cambiar las etiquetas del espacio de nombres puede provocar que los pods no se programen, un "Error al crear: …" o "Advertencia: trident-csi-…". Si esto ocurre, compruebe si la etiqueta del espacio de nombres para `privileged` fue cambiado En ese caso, reinstale Trident.

Políticas de seguridad de pods (PSP)

Campo	Descripción	Por defecto
`allowPrivilegeEscalation`	Los contenedores privilegiados deben permitir la escalada de privilegios.	`true`
`allowedCSIDrivers`	Trident no utiliza volúmenes efímeros CSI en línea.	Vacío
`allowedCapabilities`	Los contenedores Trident no privilegiados no requieren más capacidades que el conjunto predeterminado, y a los contenedores privilegiados se les otorgan todas las capacidades posibles.	Vacío
`allowedFlexVolumes`	Trident no utiliza un"Controlador FlexVolume" Por lo tanto, no están incluidos en la lista de volúmenes permitidos.	Vacío
`allowedHostPaths`	El pod del nodo Trident monta el sistema de archivos raíz del nodo, por lo tanto, no hay ningún beneficio en configurar esta lista.	Vacío
`allowedProcMountTypes`	Trident no utiliza ningún `ProcMountTypes` .	Vacío
`allowedUnsafeSysctls`	Trident no requiere ningún método inseguro. `sysctls` .	Vacío
`defaultAddCapabilities`	No es necesario agregar capacidades a los contenedores privilegiados.	Vacío
`defaultAllowPrivilegeEscalation`	La gestión de la escalada de privilegios se realiza en cada pod de Trident .	`false`
`forbiddenSysctls`	No `sysctls` están permitidos.	Vacío

Campo	Descripción	Por defecto
`fsGroup`	Los contenedores Trident se ejecutan como root.	`RunAsAny`
`hostIPC`	El montaje de volúmenes NFS requiere que el IPC del host se comunique con `nfsd`	`true`
`hostNetwork`	iscsiadm requiere que la red del host se comunique con el demonio iSCSI.	`true`
`hostPID`	Se requiere el PID del host para comprobar si `rpc-statd` se está ejecutando en el nodo.	`true`
`hostPorts`	Trident no utiliza ningún puerto del host.	Vacío
`privileged`	Los pods de nodos Trident deben ejecutar un contenedor con privilegios para poder montar volúmenes.	`true`
`readOnlyRootFilesystem`	Los pods de nodo Trident deben escribir en el sistema de archivos del nodo.	`false`
`requiredDropCapabilities`	Los pods de nodos Trident ejecutan un contenedor privilegiado y no pueden descartar capacidades.	`none`
`runAsGroup`	Los contenedores Trident se ejecutan como root.	`RunAsAny`
`runAsUser`	Los contenedores Trident se ejecutan como root.	`runAsAny`
`runtimeClass`	Trident no utiliza `RuntimeClasses` .	Vacío
`seLinux`	Trident no se configura `seLinuxOptions` porque actualmente existen diferencias en la forma en que los entornos de ejecución de contenedores y las distribuciones de Kubernetes gestionan SELinux.	Vacío
`supplementalGroups`	Los contenedores Trident se ejecutan como root.	`RunAsAny`
`volumes`	Los pods Trident requieren estos plugins de volumen.	`hostPath, projected, emptyDir`

Restricciones del contexto de seguridad (SCC)

Etiquetas	Descripción	Por defecto
`allowHostDirVolumePlugin`	Los pods de nodo Trident montan el sistema de archivos raíz del nodo.	`true`
`allowHostIPC`	El montaje de volúmenes NFS requiere que el IPC del host se comunique con `nfsd` .	`true`
`allowHostNetwork`	iscsiadm requiere que la red del host se comunique con el demonio iSCSI.	`true`
`allowHostPID`	Se requiere el PID del host para comprobar si `rpc-statd` se está ejecutando en el nodo.	`true`
`allowHostPorts`	Trident no utiliza ningún puerto del host.	`false`
`allowPrivilegeEscalation`	Los contenedores privilegiados deben permitir la escalada de privilegios.	`true`
`allowPrivilegedContainer`	Los pods de nodos Trident deben ejecutar un contenedor con privilegios para poder montar volúmenes.	`true`
`allowedUnsafeSysctls`	Trident no requiere ningún método inseguro. `sysctls` .	`none`
`allowedCapabilities`	Los contenedores Trident no privilegiados no requieren más capacidades que el conjunto predeterminado, y a los contenedores privilegiados se les otorgan todas las capacidades posibles.	Vacío
`defaultAddCapabilities`	No es necesario agregar capacidades a los contenedores privilegiados.	Vacío
`fsGroup`	Los contenedores Trident se ejecutan como root.	`RunAsAny`
`groups`	Este SCC es específico de Trident y está vinculado a su usuario.	Vacío
`readOnlyRootFilesystem`	Los pods de nodo Trident deben escribir en el sistema de archivos del nodo.	`false`
`requiredDropCapabilities`	Los pods de nodos Trident ejecutan un contenedor privilegiado y no pueden descartar capacidades.	`none`

Etiquetas	Descripción	Por defecto
`runAsUser`	Los contenedores Trident se ejecutan como root.	`RunAsAny`
`seLinuxContext`	Trident no se configura `seLinuxOptions` porque actualmente existen diferencias en la forma en que los entornos de ejecución de contenedores y las distribuciones de Kubernetes gestionan SELinux.	Vacío
`seccompProfiles`	Los contenedores privilegiados siempre se ejecutan "sin restricciones".	Vacío
`supplementalGroups`	Los contenedores Trident se ejecutan como root.	`RunAsAny`
`users`	Se proporciona una entrada para vincular este SCC al usuario Trident en el espacio de nombres Trident .	n / A
`volumes`	Los pods Trident requieren estos plugins de volumen.	`hostPath, downwardAPI, projected, emptyDir`

Avisos legales

Los avisos legales proporcionan acceso a declaraciones de derechos de autor, marcas comerciales, patentes y más.

Copyright

"https://www.netapp.com/company/legal/copyright/"

Marcas comerciales

NETAPP, el logotipo de NETAPP y las marcas enumeradas en la página de Marcas comerciales de NetApp son marcas comerciales de NetApp, Inc. Otros nombres de empresas y productos pueden ser marcas comerciales de sus respectivos propietarios.

"https://www.netapp.com/company/legal/trademarks/"

Patentes

Puede encontrar una lista actualizada de las patentes propiedad de NetApp en:

https://www.netapp.com/pdf.html?item=/media/11887-patentspage.pdf

Política de privacidad

"https://www.netapp.com/company/legal/privacy-policy/"

Código abierto

Puedes consultar los derechos de autor y las licencias de terceros utilizados en el software NetApp para Trident en el archivo de avisos de cada versión en https://github.com/NetApp/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.