Use el aprovisionador de Astra Control

Astra Control Center

NetApp
April 25, 2024

Tabla de contenidos

Use el aprovisionador de Astra Control

Configurar el cifrado de backend de almacenamiento

Con Astra Control Provisioning, puede mejorar la seguridad de acceso a los datos al habilitar el cifrado del tráfico entre su clúster gestionado y el back-end de almacenamiento.

Astra Control Provisioning admite el cifrado Kerberos para dos tipos de back-ends de almacenamiento:

- **ONTAP en las instalaciones** - El aprovisionador de control de Astra admite el cifrado de Kerberos a través de conexiones NFSv3 y NFSv4 desde Red Hat OpenShift y los clústeres de Kubernetes ascendentes a volúmenes ONTAP locales.
- **Azure NetApp Files** - El aprovisionador de control de Astra admite el cifrado de Kerberos a través de conexiones NFSv4,1 desde clústeres de Kubernetes anteriores a volúmenes de Azure NetApp Files.

Puede crear, eliminar, cambiar el tamaño, copiar, clonar, Clone de solo lectura e importe volúmenes que usen cifrado NFS.

Configure el cifrado de Kerberos en tránsito con volúmenes de ONTAP en las instalaciones

Puede habilitar el cifrado de Kerberos en el tráfico de almacenamiento entre su clúster gestionado y un back-end de almacenamiento de ONTAP en las instalaciones.

> El cifrado de Kerberos para el tráfico NFS con back-ends de almacenamiento de ONTAP en las instalaciones solo se admite mediante el `ontap-nas` controlador de almacenamiento.

Antes de empezar

- Asegúrese de que tiene "Habilitado Astra Control Provisioning" en el clúster gestionado.
- Asegúrese de tener acceso al `tridentctl` utilidad.
- Asegúrese de tener acceso de administrador al back-end de almacenamiento de ONTAP.
- Asegúrese de conocer el nombre del volumen o los volúmenes que compartirá desde el back-end de almacenamiento de ONTAP.
- Asegúrese de haber preparado la máquina virtual de almacenamiento de ONTAP para admitir el cifrado de Kerberos para los volúmenes de NFS. Consulte "Habilite Kerberos en una LIF de datos" si desea obtener instrucciones.
- Asegúrese de que los volúmenes de NFSv4 GB que utilice con el cifrado de Kerberos se hayan configurado correctamente. Consulte la sección Configuración de dominio de NetApp NFSv4 (página 13) del "Guía de mejoras y prácticas recomendadas de NetApp NFSv4".

Añada o modifique las políticas de exportación de ONTAP

Tiene que agregar reglas a políticas de exportación de ONTAP existentes o crear nuevas políticas de exportación que sean compatibles con el cifrado de 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 políticas de exportación que añada, o las nuevas políticas de exportación que cree, deben admitir los siguientes protocolos de acceso y permisos de acceso:

Protocolos de acceso

Configure la directiva de exportación con los protocolos de acceso NFS, NFSv3 y NFSv4.

Detalles de acceso

Puede configurar una de tres versiones diferentes de cifrado de Kerberos, según las necesidades del volumen:

- **Kerberos 5** - (autenticación y cifrado)
- **Kerberos 5i** - (autenticación y encriptación con protección de identidad)
- **Kerberos 5p** - (autenticación y encriptación 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 montarán los volúmenes NFS con una combinación de Kerberos 5i y cifrado Kerberos 5p, utilice los siguientes ajustes de acceso:

Tipo	Acceso de solo lectura	Acceso de lectura/escritura	Acceso de superusuario
UNIX	Activado	Activado	Activado
Kerberos 5i	Activado	Activado	Activado
Kerberos 5p	Activado	Activado	Activado

Consulte la siguiente documentación para saber cómo crear políticas de exportación de ONTAP y reglas de políticas de exportación:

- "Cree una política de exportación"
- "Añada una regla a una política de exportación"

Cree un back-end de almacenamiento

Puede crear una configuración de back-end de almacenamiento de Astra Control Provisioner que incluya la funcionalidad de cifrado Kerberos.

Acerca de esta tarea

Al crear un archivo de configuración de backend de almacenamiento que configure el cifrado Kerberos, puede especificar una de las tres versiones diferentes del 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 encriptación con protección de identidad y privacidad)

Especifique solo un nivel de Kerberos. Si especifica más de un nivel de cifrado de Kerberos en la lista de parámetros, sólo se utilizará la primera opción.

Pasos

1. En el clúster gestionado, cree un archivo de configuración de back-end de almacenamiento utilizando el ejemplo siguiente. Sustituya los valores entre paréntesis <> por información de su entorno:

```yaml
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 la creación del back-end falla, algo está mal con la configuración del back-end. Puede ver 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 ejecutar de nuevo el comando create.

Cree una clase de almacenamiento

Puede crear una clase de almacenamiento para aprovisionar volúmenes con el cifrado de Kerberos.

Acerca de esta tarea

Al crear un objeto de clase de almacenamiento, puede especificar una de las tres versiones diferentes del cifrado de 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 encriptación con protección de identidad y privacidad)

Especifique solo un nivel de Kerberos. Si especifica más de un nivel de cifrado de Kerberos en la lista de parámetros, sólo se utilizará la primera opción. Si el nivel de cifrado especificado en la configuración de backend de almacenamiento es diferente al nivel especificado en el objeto de clase de almacenamiento, el objeto de clase de almacenamiento tiene prioridad.

Pasos

1. Cree un objeto de Kubernetes StorageClass, mediante 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. Cree la clase de almacenamiento:

```
kubectl create -f sample-input/storage-class-ontap-nas-sc.yaml
```

3. Asegúrese de que se ha creado la clase de almacenamiento:

```
kubectl get sc ontap-nas-sc
```

   Debería ver una salida similar a la siguiente:

```
NAME            PROVISIONER             AGE
ontap-nas-sc    csi.trident.netapp.io   15h
```

Aprovisione los volúmenes

Después de crear un back-end de almacenamiento y una clase de almacenamiento, ahora puede aprovisionar

un volumen. Consulte estas instrucciones para "aprovisionamiento de un volumen".

Configure el cifrado de Kerberos en tránsito con volúmenes Azure NetApp Files

Puede habilitar el cifrado de Kerberos en el tráfico de almacenamiento entre su clúster gestionado y un solo back-end de almacenamiento de Azure NetApp Files o un pool virtual de back-ends de almacenamiento de Azure NetApp Files.

Antes de empezar

- Asegúrese de haber habilitado el aprovisionador de Astra Control en el clúster Red Hat OpenShift gestionado. Consulte "Habilita el aprovisionador de Astra Control" si desea obtener instrucciones.
- Asegúrese de tener acceso al `tridentctl` utilidad.
- Asegúrese de haber preparado el back-end de almacenamiento de Azure NetApp Files para cifrado Kerberos siguiendo los requisitos y siguiendo las instrucciones de "Documentación de Azure NetApp Files".
- Asegúrese de que los volúmenes de NFSv4 GB que utilice con el cifrado de Kerberos se hayan configurado correctamente. Consulte la sección Configuración de dominio de NetApp NFSv4 (página 13) del "Guía de mejoras y prácticas recomendadas de NetApp NFSv4".

Cree un back-end de almacenamiento

Puede crear una configuración de back-end de almacenamiento de Azure NetApp Files que incluya la funcionalidad de cifrado de 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 los dos niveles posibles:

- El **storage backend level** usando el `spec.kerberos` campo
- El **nivel de grupo virtual** usando el `spec.storage.kerberos` campo

Cuando se define la configuración en el nivel del pool virtual, el pool se selecciona con la etiqueta de la clase de almacenamiento.

En cualquier nivel, puede especificar una de las tres versiones diferentes del cifrado Kerberos:

- `kerberos: sec=krb5` (autenticación y cifrado)
- `kerberos: sec=krb5i` (autenticación y cifrado con protección de identidad)
- `kerberos: sec=krb5p` (autenticación y encriptación con protección de identidad y privacidad)

Pasos

1. En el clúster gestionado, cree un archivo de configuración de back-end de almacenamiento mediante uno de los siguientes ejemplos, en función del lugar donde necesite definir el back-end de almacenamiento (nivel de back-end de almacenamiento o nivel de pool virtual). Sustituya los valores entre paréntesis <> por información de su entorno:

Ejemplo de nivel de back-end de almacenamiento

```yaml
apiVersion: v1
kind: Secret
metadata:
  name: backend-tbc-anf-secret
type: Opaque
stringData:
  clientID: <CLIENT_ID>
  clientSecret: <CLIENT_SECRET>
---
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-anf
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-anf-secret
```

Ejemplo de nivel de pool virtual

```yaml
apiVersion: v1
kind: Secret
metadata:
  name: backend-tbc-anf-secret
type: Opaque
stringData:
  clientID: <CLIENT_ID>
  clientSecret: <CLIENT_SECRET>
---
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-anf
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-anf-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 la creación del back-end falla, algo está mal con la configuración del back-end. Puede ver 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 ejecutar de nuevo el comando create.

Cree una clase de almacenamiento

Puede crear una clase de almacenamiento para aprovisionar volúmenes con el cifrado de Kerberos.

Pasos

1. Cree un objeto de Kubernetes StorageClass, mediante el siguiente ejemplo:

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: anf-sc-nfs
provisioner: csi.trident.netapp.io
parameters:
  backendType: "azure-netapp-files"
  trident.netapp.io/nasType: "nfs"
  selector: "type=encryption"
```

2. Cree la clase de almacenamiento:

```
kubectl create -f sample-input/storage-class-anf-sc-nfs.yaml
```

3. Asegúrese de que se ha creado la clase de almacenamiento:

```
kubectl get sc anf-sc-nfs
```

Debería ver una salida similar a la siguiente:

```
NAME          PROVISIONER              AGE
anf-sc-nfs     csi.trident.netapp.io   15h
```

Aprovisione los volúmenes

Después de crear un back-end de almacenamiento y una clase de almacenamiento, ahora puede aprovisionar un volumen. Consulte estas instrucciones para "aprovisionamiento de un volumen".

Recuperar datos de volumen mediante una copia Snapshot

Astra Control Provisioning permite restaurar volúmenes rápidamente sin movimiento a partir de una copia Snapshot mediante el `TridentActionSnapshotRestore` (TASR) CR. Esta CR funciona como una acción imprescindible de Kubernetes y no persiste una

vez que finaliza la operación.

Astra Control Provisioner admite la restauración de copias Snapshot en el `ontap-san`, `ontap-san-economy`, `ontap-nas`, `ontap-nas-flexgroup`, `azure-netapp-files`, `gcp-cvs`, y. `solidfire-san` de windows

Antes de empezar

Debe tener una snapshot de volumen disponible y la RVP vinculada.

- Compruebe que el estado de la RVP es de enlace.

```
kubectl get pvc
```

- Compruebe que la copia de Snapshot de volumen esté lista para utilizarse.

```
kubectl get vs
```

Pasos

1. Cree el CR de TASR. En este ejemplo se crea una CR para la RVP `pvc1` y copia de snapshot de volumen `pvc1-snapshot`.

```
cat tasr-pvc1-snapshot.yaml

apiVersion: trident.netapp.io/v1
kind: TridentActionSnapshotRestore
metadata:
  name: this-doesnt-matter
  namespace: trident
spec:
  pvcName: pvc1
  volumeSnapshotName: pvc1-snapshot
```

2. Aplique el CR para restaurar a partir de la instantánea. En este ejemplo se restaura a partir de una copia Snapshot `pvc1`.

```
kubectl create -f tasr-pvc1-snapshot.yaml

tridentactionsnapshotrestore.trident.netapp.io/this-doesnt-matter
created
```

Resultados

El aprovisionador de Astra Control restaura los datos a partir de la snapshot. Es posible verificar el estado de restauración de la Snapshot.

```
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: this-doesnt-matter
    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, el aprovisionador de Astra Control no volverá a intentar automáticamente la operación en caso de fallo. Deberá realizar la operación de nuevo.
- Es posible que el administrador deba conceder permiso al usuario de Kubernetes sin acceso de administrador para crear una CR TASR en su espacio de nombres de la aplicación.

Replicar volúmenes mediante SnapMirror

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

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

Requisitos previos de replicación

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

Clústeres ONTAP

- **Astra Control Provisionador**: Astra Control Provisionador versión 23,10 o posterior o A "Astra Trident compatible" Debe existir en los clústeres de Kubernetes de origen y de destino que utilicen ONTAP como back-end.
- **Licencias**: Las licencias asíncronas de SnapMirror de ONTAP que utilizan el paquete de protección de datos deben estar habilitadas en los clústeres de ONTAP de origen y de destino. Consulte "Información general sobre las licencias de SnapMirror en ONTAP" si quiere más información.

Interconexión

- **Cluster y SVM**: Los back-ends de almacenamiento ONTAP deben ser peered. Consulte "Información general sobre relaciones entre iguales de clústeres y SVM" si quiere más información.

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

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

Controladores compatibles

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

Cree una RVP reflejada

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

Pasos

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

      Ejemplo

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

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

Ejemplo

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

c. Cree un CR de MirrorRelationship con información local.

Ejemplo

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

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

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

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

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

2. Realice los siguientes pasos en el clúster de Kubernetes secundario:
   a. Cree una StorageClass con el parámetro trident.netapp.io/replication: true.

      Ejemplo

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

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

      Ejemplo

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

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

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

Ejemplo

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

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

Estados de replicación de volúmenes

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

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

Promocione la RVP secundaria durante una conmutación al respaldo no planificada

Realice el siguiente paso en el clúster de Kubernetes secundario:

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

Promocione la RVP secundaria durante una conmutación al respaldo planificada

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

Pasos

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

   Ejemplo

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

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

Restaure una relación de mirroring después de una conmutación al nodo de respaldo

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

Pasos

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

Operaciones adicionales

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

Replica la PVC primaria a una nueva PVC secundaria

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

Pasos

1. Elimine los CRD de PersistentVolumeClaim y TridentMirrorRelationship del clúster secundario (destino) establecido.
2. Elimine el CRD de TridentMirrorRelationship del clúster primario (origen).

3. Cree un nuevo CRD de TridentMirrorRelationship en el clúster primario (de origen) para la nueva PVC secundaria (de destino) que desea establecer.

Cambie el tamaño de una RVP reflejada, primaria o secundaria

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

Elimine la replicación de una RVP

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

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

Eliminar una RVP (que se había duplicado previamente)

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

Eliminar un TMR

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

Actualice las relaciones de reflejo cuando el ONTAP esté en línea

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

Actualice las relaciones de reflejo cuando la ONTAP esté sin conexión

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

Ejemplo

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

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