



Replique volumes usando o SnapMirror

Astra Trident

NetApp
March 11, 2025

Índice

Replique volumes usando o SnapMirror	1
Pré-requisitos de replicação	1
Crie um PVC espelhado	1
Estados de replicação de volume	4
Promover PVC secundário durante um failover não planejado	5
Promover PVC secundário durante um failover planejado	5
Restaurar uma relação de espelhamento após um failover	5
Operações adicionais	6
Replique PVC primário para um novo PVC secundário	6
Redimensione um PVC espelhado, primário ou secundário	6
Remova a replicação de um PVC	6
Excluir um PVC (que foi anteriormente espelhado)	6
Eliminar um TMR	6
Atualizar relações de espelho quando o ONTAP estiver online	6
Atualizar relações de espelho quando o ONTAP estiver offline	7
Habilite o Astra Control Provisioner	7

Replique volumes usando o SnapMirror

Com o Astra Control Provisioner, você pode criar relacionamentos de espelhamento entre um volume de origem em um cluster e o volume de destino no cluster peered para replicação de dados para recuperação de desastres. Você pode usar uma Definição de recursos personalizados (CRD) para executar as seguintes operações:

- Criar relações de espelhamento entre volumes (PVCs)
- Remova as relações de espelho entre volumes
- Quebre as relações do espelho
- Promover o volume secundário durante as condições de desastre (failovers)
- Realizar a transição sem perda de aplicativos do cluster para o cluster (durante failovers planejados ou migrações)

Pré-requisitos de replicação

Certifique-se de que os seguintes pré-requisitos sejam atendidos antes de começar:

Clusters de ONTAP

- **Provisioner:** O Astra Control Provisioner versão 23,10 ou posterior deve existir nos clusters do Kubernetes de origem e destino que utilizam o ONTAP como um back-end.
- **Licenças:** As licenças assíncronas do ONTAP SnapMirror usando o pacote proteção de dados devem estar ativadas nos clusters ONTAP de origem e destino. ["Visão geral do licenciamento do SnapMirror no ONTAP"](#) Consulte para obter mais informações.

Peering

- **Cluster e SVM:** Os backends de storage do ONTAP devem ser colocados em Contato. ["Visão geral do peering de cluster e SVM"](#) Consulte para obter mais informações.



Certifique-se de que os nomes do SVM usados na relação de replicação entre dois clusters ONTAP sejam exclusivos.

- **Astra Control Provisioner e SVM:** Os SVMs remotas com peering devem estar disponíveis para o Astra Control Provisioner no cluster de destino.

Drivers suportados

- A replicação de volume é compatível com os drivers ONTAP-nas e ONTAP-san.

Crie um PVC espelhado

Siga estas etapas e use os exemplos CRD para criar relação de espelhamento entre volumes primário e secundário.

Passos

1. Execute as etapas a seguir no cluster primário do Kubernetes:
 - a. Crie um objeto StorageClass com o `trident.netapp.io/replication: true` parâmetro.

Exemplo

```
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. Crie um PVC com StorageClass criado anteriormente.

Exemplo

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

- c. Crie um MirrorRelationship CR com informações locais.

Exemplo

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

O Astra Control Provisioner obtém as informações internas do volume e do estado atual de proteção de dados (DP) do volume e, em seguida, preenche o campo de status do MirrorRelationship.

- d. Obtenha o tridentMirrorRelationship CR para obter o nome interno e SVM do 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. Execute as etapas a seguir no cluster secundário do Kubernetes:
 - a. Crie um StorageClass com o parâmetro Trident.NetApp.io/replicação: True.

Exemplo

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

- b. Crie um MirrorRelationship CR com informações de destino e origem.

Exemplo

```
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"
```

O Provisioner criará um relacionamento SnapMirror com o nome da política de relacionamento configurado (ou padrão para ONTAP) e inicializará-o.

- c. Crie um PVC com StorageClass criado anteriormente para atuar como secundário (destino SnapMirror).

Exemplo

```
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
```

O Astra Control Provisioner verificará o CRD de relacionamento do tridentMirrorRelationship e falhará em criar o volume se o relacionamento não existir. Se o relacionamento existir, o Supervisor de Controle Astra garantirá que o novo FlexVol volume seja colocado em um SVM que seja emparelhado com o SVM remoto definido no espelhamento.

Estados de replicação de volume

Um relacionamento de espelhamento do Trident (TMR) é um CRD que representa um fim de uma relação de replicação entre PVCs. O TMR de destino tem um estado, que diz ao Astra Control Provisioner qual é o estado desejado. O TMR de destino tem os seguintes estados:

- *** Estabelecido***: O PVC local é o volume de destino de uma relação de espelho, e esta é uma nova relação.

- **Promovido:** O PVC local é ReadWrite e montável, sem relação de espelho atualmente em vigor.
- *** Restabelecido*:** O PVC local é o volume de destino de uma relação de espelho e também estava anteriormente nessa relação de espelho.
 - O estado restabelecido deve ser usado se o volume de destino estiver em uma relação com o volume de origem, porque ele sobrescreve o conteúdo do volume de destino.
 - O estado restabelecido falhará se o volume não estiver previamente em uma relação com a fonte.

Promover PVC secundário durante um failover não planejado

Execute a seguinte etapa no cluster secundário do Kubernetes:

- Atualize o campo `spec.State` do `TridentMirrorRelationship` para `promoted`.

Promover PVC secundário durante um failover planejado

Durante um failover planejado (migração), execute as seguintes etapas para promover o PVC secundário:

Passos

1. No cluster primário do Kubernetes, crie um snapshot do PVC e aguarde até que o snapshot seja criado.
2. No cluster principal do Kubernetes, crie o `SnapshotInfo` CR para obter detalhes internos.

Exemplo

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

3. No cluster secundário do Kubernetes, atualize o campo `spec.State` do `tridentMirrorRelationship` CR para `promoted` e `spec.promotedSnapshotHandle` para ser o `internalName` do snapshot.
4. No cluster secundário do Kubernetes, confirme o status (campo `status.State`) do `TridentMirrorRelationship` para promovido.

Restaurar uma relação de espelhamento após um failover

Antes de restaurar uma relação de espelho, escolha o lado que você deseja fazer como o novo primário.

Passos

1. No cluster secundário do Kubernetes, certifique-se de que os valores do campo `spec.remoteVolumeHandle` no `TridentMirrorRelationship` sejam atualizados.
2. No cluster secundário do Kubernetes, atualize o campo `spec.mirror` do `TridentMirrorRelationship` para `reestablished`.

Operações adicionais

O Astra Control Provisioner dá suporte às seguintes operações nos volumes primário e secundário:

Replique PVC primário para um novo PVC secundário

Certifique-se de que você já tem um PVC primário e um PVC secundário.

Passos

1. Exclua as CRDs `PersistentVolumeClaim` e `TridentMirrorRelationship` do cluster secundário (destino) estabelecido.
2. Exclua o CRD do `tridentMirrorRelationship` do cluster primário (de origem).
3. Crie um novo CRD de `TridentMirrorRelationship` no cluster primário (de origem) para o novo PVC secundário (de destino) que você deseja estabelecer.

Redimensione um PVC espelhado, primário ou secundário

O PVC pode ser redimensionado como normal, o ONTAP irá expandir automaticamente qualquer destino flexvols se a quantidade de dados exceder o tamanho atual.

Remova a replicação de um PVC

Para remover a replicação, execute uma das seguintes operações no volume secundário atual:

- Exclua o `MirrorRelationship` no PVC secundário. Isso quebra a relação de replicação.
- Ou atualize o campo `spec.State` para *promovido*.

Excluir um PVC (que foi anteriormente espelhado)

O Astra Control Provisioner verifica se há PVCs replicados e libera a relação de replicação antes de tentar excluir o volume.

Eliminar um TMR

A exclusão de um TMR em um lado de um relacionamento espelhado faz com que o TMR restante passe para o estado *promovido* antes que o Astra Control Provisioner conclua a exclusão. Se o TMR selecionado para exclusão já estiver no estado *promovido*, não há relacionamento de espelhamento existente e o TMR será removido e o Astra Control Provisioner promoverá o PVC local para *ReadWrite*. Essa exclusão libera os metadados do `SnapMirror` para o volume local no ONTAP. Se este volume for usado em uma relação de espelho no futuro, ele deve usar um novo TMR com um estado de replicação de volume *established* ao criar a nova relação de espelho.

Atualizar relações de espelho quando o ONTAP estiver online

As relações de espelho podem ser atualizadas a qualquer momento depois que são estabelecidas. Pode utilizar os `state: promoted` campos ou `state: reestablished` para atualizar as relações. Ao promover um volume de destino para um volume `ReadWrite` regular, você pode usar *promotedSnapshotHandle* para especificar um snapshot específico para restaurar o volume atual.

Atualizar relações de espelho quando o ONTAP estiver offline

Você pode usar um CRD para executar uma atualização do SnapMirror sem que o Astra Control tenha conectividade direta com o cluster do ONTAP. Consulte o seguinte formato de exemplo do `TridentActionMirrorUpdate`:

Exemplo

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

`status.state` Reflete o estado do CRD do `TridentActionMirrorUpdate`. Ele pode tomar um valor de *successful*, *in progress* ou *Failed*.

Habilite o Astra Control Provisioner

As versões 23,10 e posteriores do Trident incluem a opção de usar o Astra Control Provisioner, que permite que usuários licenciados do Astra Control acessem o recurso avançado de provisionamento de storage. O Astra Control Provisioner fornece essa funcionalidade estendida, além da funcionalidade padrão baseada em CSI Astra Trident. Você pode usar este procedimento para ativar e instalar o Astra Control Provisioner.

Sua assinatura do Astra Control Service inclui automaticamente a licença para uso do Astra Control Provisioner.

Nas próximas atualizações do Astra Control, o parceiro Astra Control substituirá o Astra Trident como provisionador de storage e orquestrador e será obrigatório para uso do Astra Control. Por causa disso, é altamente recomendável que os usuários do Astra Control ativem o Astra Control Provisioner. O Astra Trident continuará a ser de código aberto e será lançado, mantido, suportado e atualizado com o novo CSI e outros recursos do NetApp.

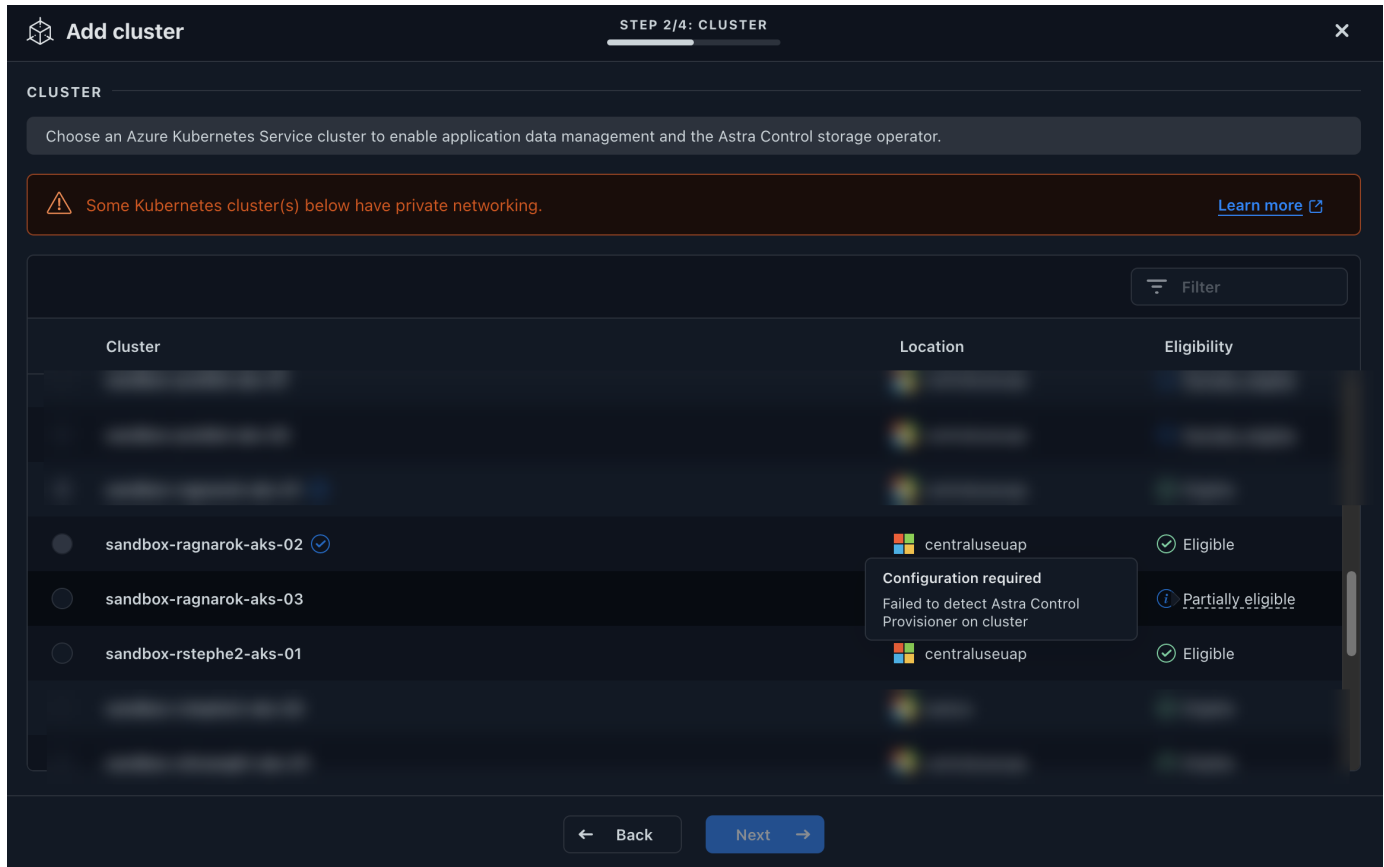
Como sei se preciso habilitar o Astra Control Provisioner?

Se você adicionar um cluster ao Astra Control Service que não tenha o Astra Trident instalado anteriormente, o cluster será marcado como `Eligible`. Depois de "[Adicione o cluster ao Astra Control](#)" você , o Astra Control Provisioner será ativado automaticamente.

Se o cluster não estiver marcado `Eligible`, ele será marcado `Partially eligible` por uma das seguintes opções:

- Ele está usando uma versão mais antiga do Astra Trident
- Ele está usando um Astra Trident 23,10 que ainda não tem a opção de provisionador habilitada
- É um tipo de cluster que não permite a ativação automática

Para `Partially eligible` casos, use estas instruções para ativar manualmente o Astra Control Provisioner para seu cluster.



The screenshot shows the 'Add cluster' wizard in the Astra Control console, specifically the 'STEP 2/4: CLUSTER' screen. The interface is dark-themed. At the top, there's a header 'Add cluster' and a close button. Below the header, there's a sub-header 'CLUSTER' and a instruction: 'Choose an Azure Kubernetes Service cluster to enable application data management and the Astra Control storage operator.' A warning banner indicates 'Some Kubernetes cluster(s) below have private networking.' with a 'Learn more' link. A table lists several clusters with columns for 'Cluster', 'Location', and 'Eligibility'. One cluster, 'sandbox-ragnarok-aks-03', is marked as 'Partially eligible' with an information icon. A tooltip for this cluster reads 'Configuration required Failed to detect Astra Control Provisioner on cluster'. At the bottom, there are 'Back' and 'Next' navigation buttons.

Cluster	Location	Eligibility
<input type="radio"/> sandbox-ragnarok-aks-02	centraluseuap	Eligible
<input type="radio"/> sandbox-ragnarok-aks-03	centraluseuap	Partially eligible
<input type="radio"/> sandbox-rstephe2-aks-01	centraluseuap	Eligible

Antes de ativar o Astra Control Provisioner

Se você já tiver um Astra Trident sem o parceiro Astra Control e quiser habilitar o parceiro Astra Control, faça o seguinte primeiro:

- **Se você tiver o Astra Trident instalado, confirme que sua versão está dentro de uma janela de quatro versões:** Você pode fazer uma atualização direta para o Astra Trident 24,02 com o Astra Control Provisioner se o seu Astra Trident estiver dentro de uma janela de quatro versões da versão 24,02. Por exemplo, você pode fazer o upgrade diretamente do Astra Trident 23,04 para o 24,02.
- **Confirme que seu cluster tem uma arquitetura de sistema AMD64:** A imagem Astra Control Provisioner é fornecida em arquiteturas de CPU AMD64 e ARM64, mas apenas AMD64 é compatível com Astra Control.

Passos

1. Acesse o Registro de imagem do NetApp Astra Control:
 - a. Faça login na IU do Astra Control Service e Registre sua ID de conta do Astra Control.
 - i. Selecione o ícone de figura no canto superior direito da página.
 - ii. Selecione **Acesso à API**.
 - iii. Anote o seu ID de conta.
 - b. Na mesma página, selecione **Generate API token** e copie a cadeia de token da API para a área de transferência e salve-a no seu editor.
 - c. Faça login no Registro Astra Control usando seu método preferido:

```
docker login cr.astra.netapp.io -u <account-id> -p <api-token>
```

```
crane auth login cr.astra.netapp.io -u <account-id> -p <api-token>
```

2. (Apenas registros personalizados) siga estes passos para mover a imagem para o seu registro personalizado. Se você não estiver usando um Registro, siga as etapas do operador Trident no [próxima seção](#).



Você pode usar Podman em vez de Docker para os seguintes comandos. Se você estiver usando um ambiente Windows, o PowerShell é recomendado.

Docker

- a. Extraia a imagem Astra Control Provisioner do Registro:



A imagem puxada não suportará múltiplas plataformas e só suportará a mesma plataforma que o host que puxou a imagem, como o Linux AMD64.

```
docker pull cr.astra.netapp.io/astra/trident-acp:24.02.0
--platform <cluster platform>
```

Exemplo:

```
docker pull cr.astra.netapp.io/astra/trident-acp:24.02.0
--platform linux/amd64
```

- b. Marque a imagem:

```
docker tag cr.astra.netapp.io/astra/trident-acp:24.02.0
<my_custom_registry>/trident-acp:24.02.0
```

- c. Envie a imagem para o seu registo personalizado:

```
docker push <my_custom_registry>/trident-acp:24.02.0
```

Grua

- a. Copie o manifesto Astra Control Provisioner para o seu Registro personalizado:

```
crane copy cr.astra.netapp.io/astra/trident-acp:24.02.0
<my_custom_registry>/trident-acp:24.02.0
```

3. Determine se o método de instalação original do Astra Trident usou um.
4. Ative o Astra Control Provisioner no Astra Trident usando o método de instalação que você usou originalmente:

Operador do Astra Trident

- a. "Baixe o instalador do Astra Trident e extraia-o."
- b. Siga estas etapas se você ainda não tiver instalado o Astra Trident ou se tiver removido o operador da sua implantação original do Astra Trident:
 - i. Crie o CRD:

```
kubectl create -f
deploy/crds/trident.netapp.io_tridentorchestrators_crd_post1.1
6.yaml
```

- ii. Crie o namespace Trident (`kubectl create namespace trident`) ou confirme se o namespace Trident ainda existe (`kubectl get all -n trident`). Se o namespace tiver sido removido, crie-o novamente.
- c. Atualize o Astra Trident para 24.06.0:



Para clusters que executam o Kubernetes 1,24 ou anterior, `bundle_pre_1_25.yaml` use o `.`. Para clusters que executam o Kubernetes 1,25 ou posterior, `bundle_post_1_25.yaml` use o `.`.

```
kubectl -n trident apply -f trident-installer/deploy/<bundle-
name.yaml>
```

- d. Verifique se o Astra Trident está em execução:

```
kubectl get torc -n trident
```

Resposta:

```
NAME          AGE
trident       21m
```

- e. se você tem um Registro que usa segredos, crie um segredo para usar para puxar a imagem Astra Control Provisioner:

```
kubectl create secret docker-registry <secret_name> -n trident
--docker-server=<my_custom_registry> --docker-username=<username>
--docker-password=<token>
```

- f. Edite o TridentOrchestrator CR e faça as seguintes edições:

```
kubectl edit torc trident -n trident
```

- i. Defina um local de Registro personalizado para a imagem Astra Trident ou extraia-a do Registro Astra Control (`tridentImage: <my_custom_registry>/trident:24.02.0`ou `tridentImage: netapp/trident:24.06.0`).
- ii. Ative o Astra Control Provisioner (`enableACP: true`).
- iii. Defina o local de Registro personalizado para a imagem Astra Control Provisioner ou extraia-a do Registro Astra Control (`acpImage: <my_custom_registry>/trident-acp:24.02.0`ou `acpImage: cr.astra.netapp.io/astra/trident-acp:24.02.0`).
- iv. Se tiver estabelecido [a imagem puxa segredos](#) anteriormente neste procedimento, pode defini-los aqui (`imagePullSecrets: - <secret_name>`). Use o mesmo nome secreto que você estabeleceu nas etapas anteriores.

```
apiVersion: trident.netapp.io/v1
kind: TridentOrchestrator
metadata:
  name: trident
spec:
  debug: true
  namespace: trident
  tridentImage: <registry>/trident:24.06.0
  enableACP: true
  acpImage: <registry>/trident-acp:24.06.0
  imagePullSecrets:
  - <secret_name>
```

- g. Salve e saia do arquivo. O processo de implantação começará automaticamente.
- h. Verifique se o operador, a implantação e as replicaset são criados.

```
kubectl get all -n trident
```



Deve haver apenas **uma instância** do operador em um cluster do Kubernetes. Não crie várias implantações do operador Astra Trident.

- i. Verifique se o `trident-acp` contentor está em execução e se `acpVersion` está `24.02.0` com um status de `Installed`:

```
kubectl get torc -o yaml
```

Resposta:

```
status:
  acpVersion: 24.02.0
  currentInstallationParams:
    ...
    acpImage: <registry>/trident-acp:24.02.0
    enableACP: "true"
    ...
  ...
status: Installed
```

tridentctl

- "Baixe o instalador do Astra Trident e extraia-o."
- "Se você tiver um Astra Trident existente, desinstale-o do cluster que o hospeda".
- Instalar o Astra Trident com a previsão de controle Astra ativada (`--enable-acp=true`):

```
./tridentctl -n trident install --enable-acp=true --acp
-image=mycustomregistry/trident-acp:24.02
```

- Confirme se o Astra Control Provisioner foi ativado:

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

Resposta:

```
+-----+-----+-----+ | SERVER
VERSION | CLIENT VERSION | ACP VERSION | +-----+
+-----+-----+-----+ | 24.02.0 | 24.02.0 | 24.02.0. |
+-----+-----+-----+
```

Leme

- Se tiver o Astra Trident 23.07.1 ou anterior instalado, "**desinstalar**" o operador e outros componentes.
- Se o cluster do Kubernetes estiver executando o 1,24 ou anterior, exclua a psp:

```
kubectl delete psp tridentoperatorpod
```

- Adicione o repositório Astra Trident Helm:

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

d. Atualize o gráfico Helm:

```
helm repo update netapp-trident
```

Resposta:

```
Hang tight while we grab the latest from your chart
repositories...
...Successfully got an update from the "netapp-trident" chart
repository
Update Complete. ☐Happy Helming!☐
```

e. Liste as imagens:

```
./tridentctl images -n trident
```

Resposta:

```
| v1.28.0 | netapp/trident:24.06.0|
| | docker.io/netapp/trident-
autosupport:24.06|
| | registry.k8s.io/sig-storage/csi-
provisioner:v4.0.0|
| | registry.k8s.io/sig-storage/csi-
attacher:v4.5.0|
| | registry.k8s.io/sig-storage/csi-
resizer:v1.9.3|
| | registry.k8s.io/sig-storage/csi-
snapshotter:v6.3.3|
| | registry.k8s.io/sig-storage/csi-node-
driver-registrar:v2.10.0 |
| | netapp/trident-operator:24.06.0 (optional)
```

f. Certifique-se de que o Trident-Operator 24.06.0 está disponível:

```
helm search repo netapp-trident/trident-operator --versions
```

Resposta:

NAME	CHART VERSION	APP VERSION	
DESCRIPTION			
netapp-trident/trident-operator	100.2406.0	24.06.0	A

g. Utilize `helm install` e execute uma das seguintes opções que incluem estas definições:

- Um nome para o local de implantação
- A versão Astra Trident
- O nome da imagem Astra Control Provisioner
- A bandeira para habilitar o provisionador
- (Opcional) Um caminho de Registro local. Se você estiver usando um Registro local, o "Imagens de Trident" pode estar localizado em um Registro ou Registros diferentes, mas todas as imagens CSI devem estar localizadas no mesmo Registro.
- O namespace Trident

Opções

- Imagens sem registo

```
helm install trident netapp-trident/trident-operator --version
100.2402.0 --set acpImage=cr.astra.netapp.io/astra/trident-
acp:24.06.0 --set enableACP=true --set operatorImage=netapp/trident-
operator:24.06.0 --set
tridentAutosupportImage=docker.io/netapp/trident-autosupport:24.06
--set tridentImage=netapp/trident:24.06.0 --namespace trident
```

- Imagens em um ou mais Registros

```
helm install trident netapp-trident/trident-operator --version
100.2402.0 --set acpImage=<your-registry>:<acp image> --set
enableACP=true --set imageRegistry=<your-registry>/sig-storage --set
operatorImage=netapp/trident-operator:24.06.0 --set
tridentAutosupportImage=docker.io/netapp/trident-autosupport:24.06
--set tridentImage=netapp/trident:24.06.0 --namespace trident
```

Você pode usar `helm list` para revisar detalhes de instalação, como nome, namespace, gráfico, status, versão do aplicativo e número de revisão.

Se você tiver algum problema na implantação do Trident usando o Helm, execute este comando para desinstalar completamente o Astra Trident:


```
./tridentctl uninstall -n trident
```

Não ["Remova completamente CRDS Astra Trident"](#) como parte da sua desinstalação antes de tentar ativar o Astra Control Provisioner novamente.



Resultado

A funcionalidade Astra Control Provisioner está ativada e você pode usar todos os recursos disponíveis para a versão em execução.

Depois que o Astra Control Provisioner for instalado, o cluster que hospeda o provisionador na IU do Astra Control mostrará um `ACP version` número de versão instalado em vez `Trident version` de campo e atual.

 **CLUSTER STATUS**

✔ Available

Version v1.24.9+rke2r2	Managed 2024/03/15 17:32 UTC	Kube-system namespace UID <div style="background-color: #ccc; height: 15px; width: 100%;"></div>	ACP Version <div style="background-color: #ccc; height: 15px; width: 100%;"></div>
Private route identifier <div style="background-color: #ccc; height: 15px; width: 100%;"></div>	Cloud instance private 	Default bucket astra-bucket1 (inherited) 	

[Overview](#)[Namespaces](#)[Storage](#)[Activity](#)

Para mais informações

- ["O Astra Trident atualiza a documentação"](#)

Informações sobre direitos autorais

Copyright © 2025 NetApp, Inc. Todos os direitos reservados. Impresso nos EUA. Nenhuma parte deste documento protegida por direitos autorais pode ser reproduzida de qualquer forma ou por qualquer meio — gráfico, eletrônico ou mecânico, incluindo fotocópia, gravação, gravação em fita ou storage em um sistema de recuperação eletrônica — sem permissão prévia, por escrito, do proprietário dos direitos autorais.

O software derivado do material da NetApp protegido por direitos autorais está sujeito à seguinte licença e isenção de responsabilidade:

ESTE SOFTWARE É FORNECIDO PELA NETAPP "NO PRESENTE ESTADO" E SEM QUAISQUER GARANTIAS EXPRESSAS OU IMPLÍCITAS, INCLUINDO, SEM LIMITAÇÕES, GARANTIAS IMPLÍCITAS DE COMERCIALIZAÇÃO E ADEQUAÇÃO A UM DETERMINADO PROPÓSITO, CONFORME A ISENÇÃO DE RESPONSABILIDADE DESTES DOCUMENTOS. EM HIPÓTESE ALGUMA A NETAPP SERÁ RESPONSÁVEL POR QUALQUER DANO DIRETO, INDIRETO, INCIDENTAL, ESPECIAL, EXEMPLAR OU CONSEQUENCIAL (INCLUINDO, SEM LIMITAÇÕES, AQUISIÇÃO DE PRODUTOS OU SERVIÇOS SOBRESSALIENTES; PERDA DE USO, DADOS OU LUCROS; OU INTERRUPTÃO DOS NEGÓCIOS), INDEPENDENTEMENTE DA CAUSA E DO PRINCÍPIO DE RESPONSABILIDADE, SEJA EM CONTRATO, POR RESPONSABILIDADE OBJETIVA OU PREJUÍZO (INCLUINDO NEGLIGÊNCIA OU DE OUTRO MODO), RESULTANTE DO USO DESTES SOFTWARES, MESMO SE ADVERTIDA DA RESPONSABILIDADE DE TAL DANO.

A NetApp reserva-se o direito de alterar quaisquer produtos descritos neste documento, a qualquer momento e sem aviso. A NetApp não assume nenhuma responsabilidade nem obrigação decorrentes do uso dos produtos descritos neste documento, exceto conforme expressamente acordado por escrito pela NetApp. O uso ou a compra deste produto não representam uma licença sob quaisquer direitos de patente, direitos de marca comercial ou quaisquer outros direitos de propriedade intelectual da NetApp.

O produto descrito neste manual pode estar protegido por uma ou mais patentes dos EUA, patentes estrangeiras ou pedidos pendentes.

LEGENDA DE DIREITOS LIMITADOS: o uso, a duplicação ou a divulgação pelo governo estão sujeitos a restrições conforme estabelecido no subparágrafo (b)(3) dos Direitos em Dados Técnicos - Itens Não Comerciais no DFARS 252.227-7013 (fevereiro de 2014) e no FAR 52.227- 19 (dezembro de 2007).

Os dados aqui contidos pertencem a um produto comercial e/ou serviço comercial (conforme definido no FAR 2.101) e são de propriedade da NetApp, Inc. Todos os dados técnicos e software de computador da NetApp fornecidos sob este Contrato são de natureza comercial e desenvolvidos exclusivamente com despesas privadas. O Governo dos EUA tem uma licença mundial limitada, irrevogável, não exclusiva, intransferível e não sublicenciável para usar os Dados que estão relacionados apenas com o suporte e para cumprir os contratos governamentais desse país que determinam o fornecimento de tais Dados. Salvo disposição em contrário no presente documento, não é permitido usar, divulgar, reproduzir, modificar, executar ou exibir os dados sem a aprovação prévia por escrito da NetApp, Inc. Os direitos de licença pertencentes ao governo dos Estados Unidos para o Departamento de Defesa estão limitados aos direitos identificados na cláusula 252.227-7015(b) (fevereiro de 2014) do DFARS.

Informações sobre marcas comerciais

NETAPP, o logotipo NETAPP e as marcas listadas em <http://www.netapp.com/TM> são marcas comerciais da NetApp, Inc. Outros nomes de produtos e empresas podem ser marcas comerciais de seus respectivos proprietários.