



Gerenciar o Astra Trident

Astra Trident

NetApp
January 31, 2025

Índice

- Gerenciar o Astra Trident 1
 - Atualizar o Astra Trident 1
 - Atualize com o operador 2
 - Atualize com o tridentctl 10
- Desinstale o Astra Trident 13
- Downgrade Astra Trident 15

Gerenciar o Astra Trident

Atualizar o Astra Trident

O Astra Trident segue uma cadência de lançamento trimestral, entregando quatro grandes lançamentos todos os anos. Cada nova versão é construída sobre as versões anteriores, fornecendo novos recursos e melhorias de desempenho, bem como correções de bugs e melhorias. Recomendamos que você atualize pelo menos uma vez por ano para aproveitar os novos recursos do Astra Trident.



Atualizar para uma versão com cinco versões anteriores exigirá que você execute uma atualização em várias etapas.

Determine a versão para a qual atualizar

- Você pode atualizar para a YY.MM versão a partir YY-1.MM da versão e quaisquer versões intermédias. Por exemplo, você pode executar uma atualização direta para 20,07 a partir de 19,07 e posterior (incluindo lançamentos de pontos, como 19.07.1).
- Se você tiver uma versão anterior, você deve executar uma atualização em várias etapas. Isso requer que você primeiro atualize para a versão mais recente que se encaixa na janela de quatro versões. Por exemplo, se você estiver executando o 18,07 e quiser atualizar para a versão 20,07, siga o processo de atualização em várias etapas conforme indicado abaixo:
 - Primeiro upgrade de 18,07 para 19,07. Consulte a documentação da respectiva versão para obter instruções específicas para atualização.
 - Em seguida, atualize de 19,07 para 20,07.



Todas as atualizações para as versões 19,04 e anteriores exigem a migração dos metadados do Astra Trident `etcd` para objetos CRD. Verifique a documentação da versão para entender como a atualização funciona.



Ao atualizar, é importante que você forneça `parameter.fsType` o `StorageClasses` Astra Trident usado. Você pode excluir e recriar `StorageClasses` sem interromper volumes pré-existent. Este é um **requisito** para aplicação de "contextos de segurança" volumes SAN. O "entrada de amostra" diretório contém exemplos, tais como `[storage-class-basic.yaml, templ:[storage-class-bronze-default.yaml` Para obter mais informações, "Problemas conhecidos" consulte .

Qual caminho de atualização devo escolher?

Você pode atualizar usando um dos seguintes caminhos:

- Utilizando o operador Trident.
- Usando ``tridentctl`` o .



O CSI volume Snapshots agora é um recurso que é o GA, começando com o Kubernetes 1,20. Ao atualizar o Astra Trident, todos os CRS e CRDs de instantâneo alfa anteriores (classes de instantâneo de volume, instantâneos de volume e conteúdo de instantâneo de volume) devem ser removidos antes que a atualização seja executada. "este blog"Consulte para compreender os passos envolvidos na migração de instantâneos alfa para a especificação beta/GA.

Você pode usar o operador Trident para atualizar se as seguintes condições forem atendidas:

- Você está executando o CSI Trident (19,07 e posterior).
- Você tem uma versão Trident baseada em CRD (19,07 e posterior).
- Você não está** executando uma instalação personalizada (usando YAMLS personalizados).



Não utilize o operador para atualizar o Trident se estiver a utilizar uma `etcd` versão Trident baseada em `-(19,04` ou anterior).

Se não pretender utilizar o operador ou se tiver uma instalação personalizada que não possa ser suportada pelo operador, pode atualizar utilizando ``tridentctl``o . Este é o método preferido de atualizações para as versões 19,04 e anteriores do Trident.

Alterações ao operador

O lançamento de 21,01 do Astra Trident introduz algumas mudanças importantes na arquitetura do operador, nomeadamente as seguintes:

- O operador está agora **com escopo de cluster**. Instâncias anteriores do operador Trident (versões 20,04 a 20,10) eram **com escopo de namespace**. Um operador com escopo de cluster é vantajoso pelas seguintes razões:
 - Responsabilidade dos recursos: A operadora agora gerencia recursos associados à instalação do Astra Trident no nível do cluster. Como parte da instalação do Astra Trident, o operador cria e mantém vários recursos `ownerReferences` usando o `.` Manter `ownerReferences` recursos com escopo de cluster pode lançar erros em certos distribuidores do Kubernetes, como o OpenShift. Isso é atenuado com um operador com escopo de cluster. Para auto-cura e correção de recursos do Trident, este é um requisito essencial.
 - Limpeza durante a desinstalação: Uma remoção completa do Astra Trident exigiria que todos os recursos associados fossem excluídos. Um operador com escopo de namespace pode ter problemas com a remoção de recursos com escopo de cluster (como o `clusterRole`, `ClusterRoleBinding` e `PodSecurityPolicy`) e levar a uma limpeza incompleta. Um operador com escopo de cluster elimina esse problema. Os usuários podem desinstalar completamente o Astra Trident e instalar novamente, se necessário.
- `TridentProvisioner` Agora é substituído pelo `TridentOrchestrator` como recurso personalizado usado para instalar e gerenciar o Astra Trident. Além disso, um novo campo é introduzido à `TridentOrchestrator` especificação. Os usuários podem especificar que o namespace Trident deve ser instalado/atualizado usando o `spec.namespace` campo. Você pode dar uma olhada em um exemplo ["aqui"](#).

Encontre mais informações

- ["Atualize utilizando o operador Trident"](#)

*

Atualize com o operador

Você pode atualizar facilmente uma instalação existente do Astra Trident usando o operador.

O que você vai precisar

Para atualizar usando o operador, as seguintes condições devem ser atendidas:

- Você deve ter uma instalação do Astra Trident baseada em CSI. Para verificar se você está executando o CSI Trident, examine os pods no namespace do Trident. Se eles seguirem o `trident-csi-*` padrão de nomenclatura, você está executando o CSI Trident.
- Você deve ter uma instalação Trident baseada em CRD. Isso representa todas as versões do 19,07 e posteriores. Se você tiver uma instalação baseada em CSI, provavelmente terá uma instalação baseada em CRD.
- Se tiver desinstalado o CSI Trident e os metadados da instalação persistirem, pode atualizar utilizando o operador.
- Somente uma instalação do Astra Trident deve existir em todos os namespaces em um determinado cluster do Kubernetes.
- Você deve estar usando um cluster do Kubernetes que executa "[versão 1,18 - 1,24](#)".
- Se CRDs de snapshot alfa estiverem presentes, você deve removê-los com `tridentctl obliviate alpha-snapshot-crd`. Isso exclui as CRDs para a especificação do instantâneo alfa. Para obter instantâneos existentes que devem ser excluídos/migrados, "[este blog](#)" consulte .



Ao atualizar o Trident usando o operador na Plataforma de contêiner OpenShift, você deve atualizar para o Trident 21.01.1 ou posterior. O operador Trident lançado com 21.01.0 contém um problema conhecido que foi corrigido no 21.01.1. Para obter mais detalhes, consulte "[Detalhes do problema no GitHub](#)".

Atualizar uma instalação de operador com escopo de cluster

Para atualizar de **Trident 21,01 e posterior**, aqui está o conjunto de etapas a serem seguidas.

Passos

1. Exclua o operador Trident que foi usado para instalar a instância atual do Astra Trident. Por exemplo, se você estiver atualizando do 21,01, execute o seguinte comando:

```
kubectl delete -f 21.01/trident-installer/deploy/bundle.yaml -n trident
```

2. (Opcional) se você quiser modificar os parâmetros de instalação, edite o `TridentOrchestrator` objeto que você criou ao instalar o Trident. Isso pode incluir alterações, como modificar a imagem Trident personalizada, Registro de imagem privado para extrair imagens de contentor, ativar logs de depuração ou especificar segredos de recebimento de imagens.
3. Instale o Astra Trident usando o `bundle.yaml` arquivo que configura o operador Trident para a nova versão. Execute o seguinte comando:

```
kubectl create -f 21.10.0/trident-installer/deploy/bundle.yaml -n trident
```

Como parte desta etapa, o operador do 21.10.0 Trident identificará uma instalação existente do Astra Trident e a atualizará para a mesma versão do operador.

Atualize uma instalação de operador com escopo de namespace

Para atualizar de uma instância do Astra Trident instalada usando o operador com escopo de namespace (versões 20,07 a 20,10), aqui está o conjunto de etapas a serem seguidas:

Passos

1. Verifique o status da instalação existente do Trident. Para fazer isso, verifique o **Status** de `TridentProvisioner`. O estado deve ser `Installed`.

```
$ kubectl describe tprov trident -n trident | grep Message: -A 3
Message:  Trident installed
Status:   Installed
Version:  v20.10.1
```



Se o status for exibido `Updating`, certifique-se de resolvê-lo antes de prosseguir. Para obter uma lista de valores de estado possíveis, ["aqui"](#) consulte .

2. Crie o `TridentOrchestrator` CRD usando o manifesto fornecido com o instalador do Trident.

```
# Download the release required [21.01]
$ mkdir 21.07.1
$ cd 21.07.1
$ wget
https://github.com/NetApp/trident/releases/download/v21.07.1/trident-
installer-21.07.1.tar.gz
$ tar -xf trident-installer-21.07.1.tar.gz
$ cd trident-installer
$ kubectl create -f
deploy/crds/trident.netapp.io_tridentorchestrators_crd_post1.16.yaml
```

3. Exclua o operador com escopo de namespace usando seu manifesto. Para concluir esta etapa, você precisa do `bundle.yaml` arquivo usado para implantar o operador com escopo de namespace. Pode `bundle.yaml` obter a partir do ["Repositório Trident"](#). Certifique-se de usar a ramificação apropriada.



Você deve fazer as alterações necessárias nos parâmetros de instalação do Trident (por exemplo, alterando os valores para `tridentImage` `autosupportImage` , , repositório de imagens privado e fornecendo `imagePullSecrets`) depois de excluir o operador com escopo de namespace e antes de instalar o operador com escopo de cluster. Para obter uma lista completa de parâmetros que podem ser atualizados, consulte ["lista de parâmetros"](#).

```

#Ensure you are in the right directory
$ pwd
$ /root/20.10.1/trident-installer

#Delete the namespace-scoped operator
$ kubectl delete -f deploy/bundle.yaml
serviceaccount "trident-operator" deleted
clusterrole.rbac.authorization.k8s.io "trident-operator" deleted
clusterrolebinding.rbac.authorization.k8s.io "trident-operator" deleted
deployment.apps "trident-operator" deleted
podsecuritypolicy.policy "tridentoperatorpods" deleted

#Confirm the Trident operator was removed
$ kubectl get all -n trident
NAME                                READY   STATUS    RESTARTS   AGE
pod/trident-csi-68d979fb85-dsrmn    6/6    Running   12         99d
pod/trident-csi-8jfhf               2/2    Running   6          105d
pod/trident-csi-jtnjz               2/2    Running   6          105d
pod/trident-csi-lcxvh               2/2    Running   8          105d

NAME                                TYPE          CLUSTER-IP      EXTERNAL-IP    PORT(S)
AGE
service/trident-csi                 ClusterIP     10.108.174.125  <none>
34571/TCP,9220/TCP                 105d

NAME                                DESIRED   CURRENT   READY   UP-TO-DATE   AGE
AVAILABLE   NODE SELECTOR
daemonset.apps/trident-csi          3         3         3       3            3
kubernetes.io/arch=amd64,kubernetes.io/os=linux  105d

NAME                                READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/trident-csi         1/1     1             1           105d

NAME                                DESIRED   CURRENT   READY   AGE
replicaset.apps/trident-csi-68d979fb85  1         1         1       105d

```

Nesta fase, o trident-operator-xxxxxxxxxx-xxxxxx pod é excluído.

- (Opcional) se os parâmetros de instalação precisarem ser modificados, atualize a TridentProvisioner especificação. Essas alterações podem ser alterações, como modificar o Registro de imagens privadas para extrair imagens de contentor, ativar logs de depuração ou especificar segredos de recebimento de imagens.

```
$ kubectl patch tprov <trident-provisioner-name> -n <trident-namespace>
--type=merge -p '{"spec":{"debug":true}}'
```

5. Instale o operador com escopo de cluster.



A instalação do operador com escopo de cluster inicia a migração `TridentProvisioner` de objetos para `TridentOrchestrator` objetos, exclui `TridentProvisioner` objetos e `tridentprovisioner` CRD e atualiza o Astra Trident para a versão do operador com escopo de cluster que está sendo usado. No exemplo a seguir, o Trident é atualizado para 21.07.1.



A atualização do Astra Trident usando o operador com escopo de cluster resulta na migração de `tridentProvisioner` para um `tridentOrchestrator` objeto com o mesmo nome. Este procedimento é automaticamente gerido pelo operador. A atualização também terá o Astra Trident instalado no mesmo namespace que antes.


```

#Ensure you are in the correct directory
$ pwd
$ /root/21.07.1/trident-installer

#Install the cluster-scoped operator in the **same namespace**
$ kubectl create -f deploy/bundle.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

#All tridentProvisioners will be removed, including the CRD itself
$ kubectl get tprov -n trident
Error from server (NotFound): Unable to list "trident.netapp.io/v1,
Resource=tridentprovisioners": the server could not find the requested
resource (get tridentprovisioners.trident.netapp.io)

#tridentProvisioners are replaced by tridentOrchestrator
$ kubectl get torc
NAME          AGE
trident       13s

#Examine Trident pods in the namespace
$ kubectl get pods -n trident
NAME                                                    READY   STATUS    RESTARTS   AGE
trident-csi-79df798bdc-m79dc                          6/6     Running   0           1m41s
trident-csi-xrst8                                       2/2     Running   0           1m41s
trident-operator-5574dbbc68-nthjv                    1/1     Running   0           1m52s

#Confirm Trident has been updated to the desired version
$ kubectl describe torc trident | grep Message -A 3
Message:                Trident installed
Namespace:              trident
Status:                 Installed
Version:                v21.07.1

```

Atualize uma instalação de operador baseada em Helm

Execute as etapas a seguir para atualizar uma instalação do operador baseada em Helm.

Passos

1. Baixe o mais recente lançamento do Astra Trident.
2. Use o `helm upgrade` comando. Veja o exemplo a seguir:

```
$ helm upgrade <name> trident-operator-21.07.1.tgz
```

onde `trident-operator-21.07.1.tgz` reflete a versão para a qual você deseja atualizar.

3. Execute `helm list` para verificar se o gráfico e a versão do aplicativo foram atualizados.



Para passar dados de configuração durante a atualização, use `--set`.

Por exemplo, para alterar o valor padrão `tridentDebug` do , execute o seguinte comando:

```
$ helm upgrade <name> trident-operator-21.07.1-custom.tgz --set  
tridentDebug=true
```

Se você executar ``$ tridentctl logs`` , você poderá ver as mensagens de depuração.



Se você definir qualquer opção não padrão durante a instalação inicial, certifique-se de que as opções estão incluídas no comando `upgrade`, ou então, os valores serão redefinidos para seus padrões.

Atualize a partir de uma instalação que não seja do operador

Se você tiver uma instância do CSI Trident que atenda aos pré-requisitos listados acima, poderá atualizar para a versão mais recente do operador Trident.

Passos

1. Baixe o mais recente lançamento do Astra Trident.

```
# Download the release required [21.07.1]  
$ mkdir 21.07.1  
$ cd 21.07.1  
$ wget  
https://github.com/NetApp/trident/releases/download/v21.07.1/trident-  
installer-21.07.1.tar.gz  
$ tar -xf trident-installer-21.07.1.tar.gz  
$ cd trident-installer
```

2. Crie o `tridentorchestrator` CRD a partir do manifesto.

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

3. Implante o operador.

```
#Install the cluster-scoped operator in the **same namespace**
$ kubectl create -f deploy/bundle.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-csi-79df798bdc-m79dc        6/6    Running   0          150d
trident-csi-xrst8                    2/2    Running   0          150d
trident-operator-5574dbbc68-nthjv   1/1    Running   0          1m30s
```

4. Crie TridentOrchestrator um CR para a instalação do Astra Trident.

```
#Create a tridentOrchestrator to initiate a Trident install
$ 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

#Confirm Trident was upgraded to the desired version
$ kubectl describe torc trident | grep Message -A 3
Message:          Trident installed
Namespace:       trident
Status:          Installed
Version:         v21.07.1
```

Os backends e PVCs existentes estão disponíveis automaticamente.

Atualize com o tridentctl

Você pode atualizar facilmente uma instalação existente do Astra Trident usando `tridentctl`.

Considerações

Ao atualizar para a versão mais recente do Astra Trident, considere o seguinte:

- A partir do Trident 20,01, apenas a versão beta do ["instantâneos de volume"](#) é suportada. Os administradores do Kubernetes devem cuidar para fazer backup ou converter com segurança os objetos de snapshot alfa em beta para reter os snapshots alfa legados.
- A versão beta de snapshots de volume introduz um conjunto modificado de CRDs e um controlador de snapshot, que devem ser configurados antes de instalar o Astra Trident.



"Este blog" discute as etapas envolvidas na migração de snapshots de volume alfa para o formato beta.

Sobre esta tarefa

Desinstalar e reinstalar o Astra Trident funciona como uma atualização. Quando você desinstalar o Trident, a reivindicação de volume persistente (PVC) e o volume persistente (PV) usados pela implantação do Astra Trident não são excluídos. Os PVS que já tiverem sido provisionados permanecerão disponíveis enquanto o Astra Trident estiver offline, e o Astra Trident provisionará volumes para quaisquer PVCs que forem criados nesse período, uma vez que estiverem novamente online.



Ao atualizar o Astra Trident, não interrompa o processo de atualização. Certifique-se de que o instalador é executado até a conclusão.

Próximos passos após a atualização

Para fazer uso do rico conjunto de recursos que estão disponíveis em versões mais recentes do Trident (como snapshots de volume sob demanda), você pode atualizar os volumes usando o `tridentctl upgrade` comando.

Se houver volumes legados, você deve atualizá-los de um tipo NFS/iSCSI para o tipo CSI para poder usar o conjunto completo de novos recursos no Astra Trident. Um PV legado que foi provisionado pelo Trident suporta o conjunto tradicional de recursos.

Considere o seguinte ao decidir atualizar volumes para o tipo CSI:

- Talvez você não precise atualizar todos os volumes. Os volumes criados anteriormente continuarão a ser acessíveis e a funcionar normalmente.
- Um PV pode ser montado como parte de um deployment/StatefulSet ao atualizar. Não é necessário derrubar o deployment/StatefulSet.
- Você **não pode** anexar um PV a um pod autônomo ao atualizar. Você deve desligar o pod antes de atualizar o volume.
- Você pode atualizar apenas um volume que esteja vinculado a um PVC. Os volumes que não estão vinculados a PVCs devem ser removidos e importados antes da atualização.

Exemplo de atualização de volume

Aqui está um exemplo que mostra como uma atualização de volume é realizada.

1. Execute `kubectl get pv` para listar os PVS.

```
$ kubectl get pv
NAME                                CAPACITY    ACCESS MODES    RECLAIM POLICY
STATUS    CLAIM                                STORAGECLASS    REASON    AGE
default-pvc-1-a8475                1073741824    RWO              Delete
Bound    default/pvc-1                        standard
default-pvc-2-a8486                1073741824    RWO              Delete
Bound    default/pvc-2                        standard
default-pvc-3-a849e                1073741824    RWO              Delete
Bound    default/pvc-3                        standard
default-pvc-4-a84de                1073741824    RWO              Delete
Bound    default/pvc-4                        standard
trident                             2Gi         RWO              Retain
Bound    trident/trident                      19h
```

Atualmente, existem quatro PVS que foram criados pelo Trident 20,07, usando o `netapp.io/trident` provisionador.

2. Corra `kubectl describe pv` para obter os detalhes do PV.

```
$ kubectl describe pv default-pvc-2-a8486

Name:                default-pvc-2-a8486
Labels:              <none>
Annotations:         pv.kubernetes.io/provisioned-by: netapp.io/trident
                    volume.beta.kubernetes.io/storage-class: standard
Finalizers:          [kubernetes.io/pv-protection]
StorageClass:        standard
Status:              Bound
Claim:               default/pvc-2
Reclaim Policy:      Delete
Access Modes:        RWO
VolumeMode:          Filesystem
Capacity:            1073741824
Node Affinity:       <none>
Message:
Source:
  Type:              NFS (an NFS mount that lasts the lifetime of a pod)
  Server:            10.xx.xx.xx
  Path:              /trid_1907_alpha_default_pvc_2_a8486
  ReadOnly:          false
```

O PV foi criado usando o `netapp.io/trident` provisionador e é do tipo NFS. Para suportar todos os novos recursos fornecidos pelo Astra Trident, este PV deve ser atualizado para o tipo CSI.

3. Execute o `tridentctl upgrade volume <name-of-trident-volume>` comando para atualizar um volume legado Astra Trident para a especificação CSI.

```

$ ./tridentctl get volumes -n trident
+-----+-----+-----+-----+
+-----+-----+-----+-----+
|          NAME          |  SIZE  | STORAGE CLASS | PROTOCOL |
BACKEND UUID            | STATE  | MANAGED      |          |
+-----+-----+-----+-----+
+-----+-----+-----+-----+
| default-pvc-2-a8486 | 1.0 GiB | standard      | file     | c5a6f6a4-
b052-423b-80d4-8fb491a14a22 | online | true         |          |
| default-pvc-3-a849e | 1.0 GiB | standard      | file     | c5a6f6a4-
b052-423b-80d4-8fb491a14a22 | online | true         |          |
| default-pvc-1-a8475 | 1.0 GiB | standard      | file     | c5a6f6a4-
b052-423b-80d4-8fb491a14a22 | online | true         |          |
| default-pvc-4-a84de | 1.0 GiB | standard      | file     | c5a6f6a4-
b052-423b-80d4-8fb491a14a22 | online | true         |          |
+-----+-----+-----+-----+
+-----+-----+-----+-----+

$ ./tridentctl upgrade volume default-pvc-2-a8486 -n trident
+-----+-----+-----+-----+
+-----+-----+-----+-----+
|          NAME          |  SIZE  | STORAGE CLASS | PROTOCOL |
BACKEND UUID            | STATE  | MANAGED      |          |
+-----+-----+-----+-----+
+-----+-----+-----+-----+
| default-pvc-2-a8486 | 1.0 GiB | standard      | file     | c5a6f6a4-
b052-423b-80d4-8fb491a14a22 | online | true         |          |
+-----+-----+-----+-----+
+-----+-----+-----+-----+

```

4. Execute um `kubectl describe pv` para verificar se o volume é um volume CSI.

```

$ kubectl describe pv default-pvc-2-a8486
Name:                default-pvc-2-a8486
Labels:              <none>
Annotations:         pv.kubernetes.io/provisioned-by: csi.trident.netapp.io
                    volume.beta.kubernetes.io/storage-class: standard
Finalizers:          [kubernetes.io/pv-protection]
StorageClass:        standard
Status:              Bound
Claim:               default/pvc-2
Reclaim Policy:      Delete
Access Modes:        RWO
VolumeMode:          Filesystem
Capacity:            1073741824
Node Affinity:       <none>
Message:
Source:
  Type:               CSI (a Container Storage Interface (CSI) volume
source)
  Driver:              csi.trident.netapp.io
  VolumeHandle:        default-pvc-2-a8486
  ReadOnly:            false
  VolumeAttributes:    backendUUID=c5a6f6a4-b052-423b-80d4-
8fb491a14a22

internalName=trid_1907_alpha_default_pvc_2_a8486
                    name=default-pvc-2-a8486
                    protocol=file
Events:               <none>

```

Dessa forma, você pode atualizar volumes do tipo NFS/iSCSI criados pelo Astra Trident para o tipo CSI, por volume.

Desinstale o Astra Trident

Dependendo de como o Astra Trident é instalado, há várias opções para desinstalá-lo.

Desinstalar usando Helm

Se você instalou o Astra Trident usando o Helm, você pode desinstalá-lo usando `helm uninstall`.

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

Desinstale utilizando o operador Trident

Se você instalou o Astra Trident usando o operador, você pode desinstalá-lo fazendo um dos seguintes procedimentos:

- **Editar `TridentOrchestrator` para definir o sinalizador de desinstalação:** você pode editar `TridentOrchestrator` e definir `spec.uninstall=true`. Edite o `TridentOrchestrator` CR e defina a `uninstall` bandeira como mostrado abaixo:

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

Quando o `uninstall` sinalizador está definido como `true`, o operador Trident desinstala o Trident, mas não remove o próprio `TridentOrchestrator`. Você deve limpar o `TridentOrchestrator` e criar um novo se quiser instalar o Trident novamente.

- **Excluir `TridentOrchestrator`:** removendo o `TridentOrchestrator` CR que foi usado para implantar o Astra Trident, você instrui o operador a desinstalar o Trident. O operador processa a remoção `TridentOrchestrator` e remove a implantação do Astra Trident e o `daemonset`, excluindo os pods do Trident que ele criou como parte da instalação. Para remover completamente o Astra Trident (incluindo as CRDs que ele cria) e limpar efetivamente a ardósia limpa, você pode editar `TridentOrchestrator` para passar a `wipeout` opção. Veja o exemplo a seguir:

```
$ kubectl patch torc <trident-orchestrator-name> --type=merge -p
'{"spec":{"wipeout":["crds"],"uninstall":true}}'
```

Isso desinstala completamente o Astra Trident e limpa todos os metadados relacionados aos back-ends e volumes que ele gerencia. As instalações subsequentes são tratadas como instalações frescas.



Você só deve considerar limpar as CRDs ao executar uma desinstalação completa. Isso não pode ser desfeito. **Não limpe as CRDs a menos que você esteja olhando para começar de novo e criar uma nova instalação do Astra Trident.**

Desinstalar usando `tridentctl`

Execute o `uninstall` comando `tridentctl` da seguinte forma para remover todos os recursos associados ao Astra Trident, exceto para os CRDs e objetos relacionados, facilitando assim a execução do instalador novamente para atualizar para uma versão mais recente.

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

Para realizar uma remoção completa do Astra Trident, você deve remover os localizadores das CRDs criadas pelo Astra Trident e excluir os CRDs.

Downgrade Astra Trident

Saiba mais sobre os passos envolvidos na redução para uma versão anterior do Astra Trident.

Você pode considerar downgrade por vários motivos, como o seguinte:

- Planejamento de contingência
- Correção imediata para bugs observados como resultado de uma atualização
- Problemas de dependência, atualizações sem sucesso e incompletas

Quando fazer downgrade

Você deve considerar um downgrade ao mudar para uma versão do Astra Trident que usa CRDs. Como o Astra Trident agora usa CRDs para manter o estado, todas as entidades de storage criadas (backends, classes de storage, PV e snapshots de volume) têm objetos CRD associados em vez de dados gravados `trident` no PV (usados pela versão instalada anterior do Astra Trident). PVS, backends e classes de armazenamento recém-criados são todos mantidos como objetos CRD. Se você precisar fazer o downgrade, isso só deve ser tentado para uma versão do Astra Trident que seja executada usando CRDs (19,07 e posterior). Isso garante que todas as operações realizadas no atual lançamento do Astra Trident sejam visíveis após o downgrade ocorrer.

Quando não para downgrade

Você não deve fazer o downgrade para uma versão do Trident que usa `etcd` para manter o estado (19,04 e anterior). Todas as operações executadas com o lançamento atual do Astra Trident não se refletem após o downgrade. PVS recém-criados não são utilizáveis ao voltar para uma versão anterior. As alterações feitas em objetos como backends, PVS, classes de armazenamento e snapshots de volume (criados/atualizados/excluídos) não são visíveis para o Astra Trident ao voltar para uma versão anterior. Voltar para uma versão anterior não interrompe o acesso para PVS que já foram criados usando a versão mais antiga, a menos que tenham sido atualizados.

Processo de downgrade quando o Astra Trident é instalado usando o operador

Para instalações realizadas usando o Operador Trident, o processo de downgrade é diferente e não requer o uso ``tridentctl`` do .

Para instalações feitas usando o operador Trident, o Astra Trident pode ser rebaixado para uma das seguintes opções:

- Uma versão que é instalada usando o operador com escopo de namespace (20,07 - 20,10).
- Uma versão que é instalada usando o operador com escopo de cluster (21,01 e posterior).

Downgrade para operador com escopo de cluster

Para fazer o downgrade do Astra Trident para uma versão que usa o operador com escopo de cluster, siga as etapas mencionadas abaixo.

Passos

1. **"Desinstale o Astra Trident". Não use os CRDs a menos que você queira remover completamente uma instalação existente.**
2. Exclua o operador com escopo de cluster. Para fazer isso, você precisará do manifesto usado para implantar o operador. Você pode obtê-lo a partir do ["Trident repo"](#). Certifique-se de mudar para a filial pretendida.
3. Continue a fazer o downgrade instalando a versão desejada do Astra Trident. Siga a documentação para a versão desejada.

Downgrade para operador com escopo de namespace

Esta seção resume as etapas envolvidas na redução para uma versão do Astra Trident que cai no intervalo de 20,07 a 20,10, que será instalada usando o operador com escopo de namespace.

Passos

1. **"Desinstale o Astra Trident". Não use os CRDs a menos que você queira remover completamente uma instalação existente.** Certifique-se de que o `tridentorchestrator` é eliminado.

```
#Check to see if there are any tridentorchestrators present
$ kubectl get torc
NAME          AGE
trident       20h

#Looks like there is a tridentorchestrator that needs deleting
$ kubectl delete torc trident
tridentorchestrator.trident.netapp.io "trident" deleted
```

2. Exclua o operador com escopo de cluster. Para fazer isso, você precisará do manifesto usado para implantar o operador. Você pode obtê-lo aqui a partir do ["Trident repo"](#). Certifique-se de mudar para a filial pretendida.
3. Eliminar o `tridentorchestrator` CRD.

```
#Check to see if ``tridentorchestrators.trident.netapp.io`` CRD is
present and delete it.
$ kubectl get crd tridentorchestrators.trident.netapp.io
NAME                                CREATED AT
tridentorchestrators.trident.netapp.io  2021-01-21T21:11:37Z
$ kubectl delete crd tridentorchestrators.trident.netapp.io
customresourcedefinition.apiextensions.k8s.io
"tridentorchestrators.trident.netapp.io" deleted
```

O Astra Trident foi desinstalado.

4. Continue a baixar instalando a versão desejada. Siga a documentação para a versão desejada.

Downgrade usando Helm

Para fazer o downgrade, use o `helm rollback` comando. Veja o exemplo a seguir:

```
$ helm rollback trident [revision #]
```

Processo de downgrade quando o Astra Trident é instalado usando `tridentctl`

Se você instalou o Astra Trident usando `tridentctl`, o processo de downgrade envolve as etapas a seguir. Esta sequência orienta você pelo processo de downgrade para passar do Astra Trident 21,07 para 20,07.



Antes de começar o downgrade, você deve tirar um snapshot dos clusters do Kubernetes `etcd`. Isso permite fazer backup do estado atual das CRDs do Astra Trident.

Passos

1. Certifique-se de que o Trident esteja instalado `tridentctl` usando o `tridentctl`. Se você não tiver certeza sobre como o Astra Trident está instalado, execute este teste simples:
 - a. Liste os pods presentes no namespace Trident.
 - b. Identifique a versão do Astra Trident em execução no cluster. Você pode `tridentctl` usar ou dar uma olhada na imagem usada nos pods do Trident.
 - c. **Se não vir** a `tridentOrchestrator`, (ou) a `tridentprovisioner`, (ou) um pod chamado `trident-operator-xxxxxxxx-xxxxx`, o Astra Trident **está instalado** com `tridentctl`.
2. Desinstale o Astra Trident com o binário existente `tridentctl`. Neste caso, você irá desinstalar com o binário 21,07.

```

$ tridentctl version -n trident
+-----+-----+
| SERVER VERSION | CLIENT VERSION |
+-----+-----+
| 21.07.0        | 21.07.0        |
+-----+-----+

$ tridentctl uninstall -n trident
INFO Deleted Trident deployment.
INFO Deleted Trident daemonset.
INFO Deleted Trident service.
INFO Deleted Trident secret.
INFO Deleted cluster role binding.
INFO Deleted cluster role.
INFO Deleted service account.
INFO Deleted pod security policy.
podSecurityPolicy=tridentpods
INFO The uninstaller did not delete Trident's namespace in case it is
going to be reused.
INFO Trident uninstallation succeeded.

```

3. Depois que isso for concluído, obtenha o binário Trident para a versão desejada (neste exemplo, 20,07) e use-o para instalar o Astra Trident. Você pode gerar YAMLs personalizados para um ["instalação personalizada"](#), se necessário.

```

$ cd 20.07/trident-installer/
$ ./tridentctl install -n trident-ns
INFO Created installer service account.
serviceaccount=trident-installer
INFO Created installer cluster role.                clusterrole=trident-
installer
INFO Created installer cluster role binding.
clusterrolebinding=trident-installer
INFO Created installer configmap.                   configmap=trident-
installer
...
...
INFO Deleted installer cluster role binding.
INFO Deleted installer cluster role.
INFO Deleted installer service account.

```

O processo de downgrade está concluído.

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 INTERRUPÇÃO 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.