



# Gerencie e monitore o Astra Trident

## Astra Trident

NetApp  
January 31, 2025

# Índice

- Gerencie e monitore o Astra Trident ..... 1
  - Atualizar o Astra Trident ..... 1
  - Monitore o Astra Trident ..... 7
  - Desinstale o Astra Trident ..... 11

# Gerencie e monitore o Astra Trident

## Atualizar 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 baseia-se nas versões anteriores e fornece novos recursos, melhorias de desempenho, correções de bugs e melhorias. Recomendamos que você atualize pelo menos uma vez por ano para aproveitar os novos recursos do Astra Trident.

### Considerações antes da atualização

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

- Deve haver apenas uma instância do Astra Trident instalada em todos os namespaces em um determinado cluster do Kubernetes.
- O Astra Trident 23,07 e posterior requer snapshots de volume v1 e não é mais compatível com snapshots alfa ou beta.
- Se você criou o Cloud Volumes Service para Google Cloud no "[Tipo de serviço CVS](#)", atualize a configuração de back-end para usar o `standardsw` nível de serviço ou `zoneredundantstandardsw` ao atualizar a partir do Astra Trident 23,01. A falha ao atualizar o `serviceLevel` no back-end pode causar falha de volumes. "[Amostras do tipo de serviço CVS](#)" Consulte para obter detalhes.
- Ao atualizar, é importante que você forneça `parameter.fsType` o `StorageClasses` Astra Trident usado. Você pode excluir e recriar `StorageClasses` sem interromper volumes pré-existentes.
  - Este é um **requisito** para aplicação de "[contextos de segurança](#)" volumes SAN.
  - O diretório <https://github.com/NetApp/Trident> `Trident/tree/master/Trident-Installer/sample-input[sample input]` contém exemplos, como <https://github.com/NetApp/Trident/blob/master/Trident-installer/sample-input/storage-class-samples/storage-class-bronze-default.yaml> `basic[storage-class-basic.yaml.template]`
  - Para obter mais informações, "[Problemas conhecidos](#)" consulte .

### Passo 1: Selecione uma versão

As versões do Astra Trident seguem uma convenção de nomenclatura baseada em data `YY.MM`, onde "YY" é os últimos dois dígitos do ano e "MM" é o mês. Os lançamentos de ponto seguem uma `YY.MM.X` convenção, onde "X" é o nível de patch. Você selecionará a versão para a qual atualizar com base na versão da qual você está atualizando.

- Você pode fazer uma atualização direta para qualquer versão de destino que esteja dentro de uma janela de quatro versões da versão instalada. Por exemplo, você pode atualizar diretamente de 22,07 (ou qualquer lançamento de 22,07 pontos) para 23,07.
- Se você estiver atualizando de uma versão fora da janela de quatro versões, execute uma atualização em várias etapas. Use as instruções de atualização do "[versão anterior](#)" para atualizar para a versão mais recente que se encaixa na janela de quatro versões. Por exemplo, se você estiver executando o 21,07 e quiser atualizar para o 23,07:

- a. Primeiro upgrade de 21,07 para 22,07.
- b. Em seguida, atualize de 22,07 para 23,07.



Ao atualizar usando o operador Trident 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"](#).

## Passo 2: Determine o método de instalação original

Para determinar qual versão você usou para instalar originalmente o Astra Trident:

1. Use `kubectl get pods -n trident` para examinar os pods.
  - Se não houver nenhum pod do operador, o Astra Trident foi instalado usando `tridentctl`.
  - Se houver um pod do operador, o Astra Trident foi instalado usando o operador Trident manualmente ou usando o Helm.
2. Se houver um pod do operador, use `kubectl describe tproc trident` para determinar se o Astra Trident foi instalado usando o Helm.
  - Se houver uma etiqueta Helm, o Astra Trident foi instalado usando Helm.
  - Se não houver nenhuma etiqueta Helm, o Astra Trident foi instalado manualmente usando o operador Trident.

## Passo 3: Selecione um método de atualização

Geralmente, você deve atualizar usando o mesmo método usado para a instalação inicial, no entanto, você pode ["mova entre os métodos de instalação"](#). Há duas opções para atualizar o Astra Trident.

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



Sugerimos que você revise ["Compreender o fluxo de trabalho de atualização do operador"](#) antes de atualizar com o operador.

\*

## Atualize com o operador

### Compreender o fluxo de trabalho de atualização do operador

Antes de usar o operador Trident para atualizar o Astra Trident, você deve entender os processos em segundo plano que ocorrem durante a atualização. Isso inclui alterações no controlador Trident, no pod de nó e no pod de nó e no DaemonSet que permitem atualizações contínuas.

### Manuseio de atualização do operador Trident

Um dos muitos ["Benefícios de usar o operador Trident"](#) que instalar e atualizar o Astra Trident é o manuseio automático de objetos Kubernetes e Astra Trident sem interromper os volumes montados existentes. Dessa forma, o Astra Trident pode dar suporte a atualizações sem inatividade, ou ["atualizações contínuas"](#). Em particular, o operador do Trident se comunica com o cluster do Kubernetes para:

- Exclua e recrie a implantação do controlador Trident e o nó DaemonSet.
- Substitua o pod de nó Trident e o pod de nó Trident por novas versões.
  - Se um nó não for atualizado, ele não impedirá que os nós restantes sejam atualizados.
  - Somente os nós com um pod de nó Trident em execução podem montar volumes.



Para obter mais informações sobre a arquitetura Astra Trident no cluster Kubernetes, "[A arquitetura do Astra Trident](#)" consulte .

### Fluxo de trabalho de atualização do operador

Quando você inicia uma atualização usando o operador Trident:

1. O operador **Trident**:
  - a. Detecta a versão atualmente instalada do Astra Trident (versão  $n$ ).
  - b. Atualiza todos os objetos Kubernetes, incluindo CRDs, RBAC e Trident SVC.
  - c. Exclui a implantação do controlador Trident para a versão  $n$ .
  - d. Cria a implantação do controlador Trident para a versão  $n-1$ .
2. O Kubernetes\* cria o pod de controlador Trident para  $n-1$ .
3. O operador **Trident**:
  - a. Exclui o nó Trident DaemonSet para  $n$ . O operador não espera o encerramento do Node Pod.
  - b. Cria o nó Trident Daemonset para  $n-1$ .
4. **Kubernetes** cria pods de nós do Trident em nós que não executam o Pod de nó do Trident  $n$ . Isso garante que nunca mais de um pod de nó Trident, de qualquer versão, em um nó.

### Atualize a instalação de um operador Trident

Você pode atualizar o Astra Trident usando o operador Trident manualmente ou usando o Helm. Você pode atualizar de uma instalação de operador Trident para outra instalação de operador Trident ou atualizar de uma `tridentctl` instalação para uma versão de operador Trident. Reveja "[Selecione um método de atualização](#)" antes de atualizar a instalação de um operador Trident.

#### Atualize uma instalação manual

Você pode atualizar de uma instalação de operador Trident com escopo de cluster para outra instalação de operador Trident com escopo de cluster. Todos os Astra Trident versões 21,01 e superiores usam um operador com escopo de cluster.



Para atualizar do Astra Trident que foi instalado usando o operador com escopo de namespace (versões 20,07 a 20,10), use as instruções de atualização do "[sua versão instalada](#)" Astra Trident.

#### Sobre esta tarefa

O Trident fornece um arquivo de pacote que você pode usar para instalar o operador e criar objetos associados para sua versão do Kubernetes.

- 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 .

### Antes de começar

Verifique se você está usando um cluster do Kubernetes executando "[Uma versão compatível do Kubernetes](#)" o .

### Passos

1. Verifique sua versão do Astra Trident:

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

2. Exclua o operador Trident que foi usado para instalar a instância atual do Astra Trident. Por exemplo, se você estiver atualizando do 23,04, execute o seguinte comando:

```
kubectl delete -f 23.04/trident-installer/deploy/<bundle.yaml> -n trident
```

3. Se você personalizou sua instalação inicial usando `TridentOrchestrator` atributos, você pode editar o `TridentOrchestrator` objeto para modificar os parâmetros de instalação. Isso pode incluir alterações feitas para especificar Registros de imagens Trident e CSI espelhados para o modo offline, habilitar logs de depuração ou especificar segredos de recebimento de imagens.
4. Instale o Astra Trident usando o arquivo YAML do pacote correto para o seu ambiente, onde `<bundle.yaml>` é `bundle_pre_1_25.yaml` ou `bundle_post_1_25.yaml` baseado na sua versão do Kubernetes. Por exemplo, se você estiver instalando o Astra Trident 23,07, execute o seguinte comando:

```
kubectl create -f 23.07.1/trident-installer/deploy/<bundle.yaml> -n trident
```

### Atualize uma instalação do Helm

Você pode atualizar uma instalação do Astra Trident Helm.



Ao atualizar um cluster do Kubernetes do 1,24 para o 1,25 ou posterior que tenha o Astra Trident instalado, você deve atualizar o `Values.yaml` para definir `excludePodSecurityPolicy true` ou adicionar `--set excludePodSecurityPolicy=true` helm upgrade ao comando antes de atualizar o cluster.

### Passos

1. Baixe o mais recente lançamento do Astra Trident.
2. Use o `helm upgrade` comando onde `trident-operator-23.07.1.tgz` reflete a versão para a qual você deseja atualizar.

```
helm upgrade <name> trident-operator-23.07.1.tgz
```

Se você definir opções não padrão durante a instalação inicial (como especificar Registros privados espelhados para imagens Trident e CSI), use `--set` para garantir que essas opções estejam incluídas no comando `upgrade`, caso contrário, os valores serão redefinidos para padrão.



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

```
helm upgrade <name> trident-operator-23.07.1-custom.tgz --set  
tridentDebug=true
```

3. Execute `helm list` para verificar se o gráfico e a versão do aplicativo foram atualizados. Execute `tridentctl logs` para rever todas as mensagens de depuração.

### Atualize de uma `tridentctl` instalação para o operador Trident

Pode atualizar para a versão mais recente do operador Trident a partir de uma `tridentctl` instalação. Os backends e PVCs existentes estarão automaticamente disponíveis.



Antes de alternar entre os métodos de instalação, reveja "[Movendo-se entre os métodos de instalação](#)".

### Passos

1. Baixe o mais recente lançamento do Astra Trident.

```
# Download the release required [23.07.1]  
mkdir 23.07.1  
cd 23.07.1  
wget  
https://github.com/NetApp/trident/releases/download/v22.01.1/trident-  
installer-23.07.1.tar.gz  
tar -xf trident-installer-23.07.1.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 com escopo de cluster no mesmo namespace.

```
kubectl create -f deploy/<bundle-name.yaml>

serviceaccount/trident-operator created
clusterrole.rbac.authorization.k8s.io/trident-operator created
clusterrolebinding.rbac.authorization.k8s.io/trident-operator created
deployment.apps/trident-operator created
podsecuritypolicy.policy/tridentoperatorpods created

#Examine the pods in the Trident namespace
```

NAME	READY	STATUS	RESTARTS	AGE
trident-controller-79df798bdc-m79dc	6/6	Running	0	150d
trident-node-linux-xrst8	2/2	Running	0	150d
trident-operator-5574dbbc68-nthjv	1/1	Running	0	1m30s

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

```
cat deploy/crds/tridentorchestrator_cr.yaml
apiVersion: trident.netapp.io/v1
kind: TridentOrchestrator
metadata:
  name: trident
spec:
  debug: true
  namespace: trident

kubectl create -f deploy/crds/tridentorchestrator_cr.yaml

#Examine the pods in the Trident namespace
```

NAME	READY	STATUS	RESTARTS	AGE
trident-csi-79df798bdc-m79dc	6/6	Running	0	1m
trident-csi-xrst8	2/2	Running	0	1m
trident-operator-5574dbbc68-nthjv	1/1	Running	0	5m41s

#### 5. Confirme se o Trident foi atualizado para a versão pretendida.

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

Message:          Trident installed
Namespace:        trident
Status:           Installed
Version:          v23.07.1
```

## Atualize com o tridentctl

É possível atualizar facilmente uma instalação existente do Astra Trident usando `tridentctl`.

### 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.

### Antes de começar

Revise "[Selecione um método de atualização](#)" antes de atualizar usando `tridentctl`.

### Passos

1. Execute o comando `uninstall tridentctl` para remover todos os recursos associados ao Astra Trident, exceto para CRDs e objetos relacionados.

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

2. Reinstale o Astra Trident. "[Instale o Astra Trident usando o tridentctl](#)" Consulte a .



Não interrompa o processo de atualização. Certifique-se de que o instalador é executado até a conclusão.

## Monitore o Astra Trident

O Astra Trident fornece um conjunto de pontos de extremidade de métricas Prometheus que você pode usar para monitorar a performance do Astra Trident.

### Visão geral

As métricas fornecidas pelo Astra Trident permitem que você faça o seguinte:

- Acompanhe a integridade e a configuração do Astra Trident. Você pode examinar como as operações são bem-sucedidas e se elas podem se comunicar com os backends como esperado.
- Examine as informações de uso do back-end e entenda quantos volumes são provisionados em um back-end e a quantidade de espaço consumido, etc.
- Mantenha um mapeamento da quantidade de volumes provisionados em backends disponíveis.
- Acompanhe o desempenho. Você pode ver quanto tempo leva para que o Astra Trident se comunique com back-ends e realize operações.



Por padrão, as métricas do Trident são expostas na porta de destino 8001 no `/metrics` endpoint. Essas métricas são **ativadas por padrão** quando o Trident está instalado.

### O que você vai precisar

- Um cluster Kubernetes com Astra Trident instalado.
- Uma instância Prometheus. Isso pode ser um ["Implantação do Prometheus em contêiner"](#) ou você pode optar por executar Prometheus como um ["aplicação nativa"](#).

## Passo 1: Defina um alvo Prometheus

Você deve definir um alvo Prometheus para reunir as métricas e obter informações sobre os back-ends que o Astra Trident gerencia, os volumes que ele cria e assim por diante. ["blog"](#) Isso explica como você pode usar Prometheus e Grafana com o Astra Trident para recuperar métricas. O blog explica como você pode executar o Prometheus como um operador no cluster Kubernetes e a criação de um ServiceMonitor para obter métricas do Astra Trident.

## Passo 2: Crie um Prometheus ServiceMonitor

Para consumir as métricas do Trident, você deve criar um Prometheus ServiceMonitor que vigia `trident-csi` o serviço e escuta na `metrics` porta. Um exemplo de ServiceMonitor se parece com isso:

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

Essa definição do ServiceMonitor recupera as métricas retornadas pelo `trident-csi` serviço e procura especificamente o `metrics` ponto final do serviço. Como resultado, prometheus agora está configurado para entender as métricas do Astra Trident.

Além das métricas disponíveis diretamente do Astra Trident, o kubelet expõe muitas `kubelet_volume_*` métricas por meio do seu próprio ponto de extremidade de métricas. O Kubelet pode fornecer informações sobre os volumes anexados e pods e outras operações internas que ele manipula. ["aqui"](#) Consulte .

## Passo 3: Consultar métricas do Trident com PromQL

PromQL é bom para criar expressões que retornam dados de séries temporais ou tabulares.

Aqui estão algumas consultas PromQL que você pode usar:

### Obtenha informações de saúde do Trident

- **Porcentagem de respostas HTTP 2XX do Astra Trident**

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

- **Porcentagem de RESPOSTAS REST do Astra Trident via código de status**

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

- **Duração média em ms das operações realizadas pelo Astra Trident**

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

### Obtenha informações de uso do Astra Trident

- **Tamanho médio do volume**

```
trident_volume_allocated_bytes/trident_volume_count
```

- **Espaço total de volume provisionado por cada back-end**

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

### Obtenha uso de volume individual



Isso é ativado somente se as métricas do kubelet também forem coletadas.

- **Porcentagem de espaço usado para cada volume**

```
kubelet_volume_stats_used_bytes / kubelet_volume_stats_capacity_bytes *  
100
```

## Saiba mais sobre a telemetria do Astra Trident AutoSupport

Por padrão, o Astra Trident envia métricas de Prometheus e informações básicas de back-end para o NetApp em uma cadência diária.

- Para impedir que o Astra Trident envie métricas e informações básicas de back-end para o NetApp, passe a `--silence-autosupport` bandeira durante a instalação do Astra Trident.
- O Astra Trident também pode enviar logs de contêiner para o suporte do NetApp sob demanda por meio ``tridentctl send autosupport`` do . Você precisará acionar o Astra Trident para fazer o upload dos seus logs. Antes de enviar logs, você deve aceitar o NetApp "[política de privacidade](#)"s .
- A menos que especificado, o Astra Trident obtém os logs das últimas 24 horas.
- Você pode especificar o período de tempo de retenção do log com o `--since` sinalizador. Por exemplo `tridentctl send autosupport --since=1h`. Essas informações são coletadas e enviadas por meio `trident-autosupport` de um contêiner que é instalado ao lado do Astra Trident. Pode obter a imagem do contentor em "[Trident AutoSupport](#)".
- A Trident AutoSupport não coleta nem transmite informações de identificação pessoal (PII) ou informações pessoais. Ele vem com um "[EULA](#)" que não é aplicável à própria imagem de contentor Trident. Você pode saber mais sobre o compromisso da NetApp com a segurança e a confiança dos dados "[aqui](#)" .

Um exemplo de payload enviado pelo Astra Trident é parecido com este:

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

- As mensagens do AutoSupport são enviadas para o ponto de extremidade do AutoSupport do NetApp. Se você estiver usando um Registro privado para armazenar imagens de contentor, você pode usar o `--image-registry` sinalizador.
- Você também pode configurar URLs de proxy gerando os arquivos YAML de instalação. Isso pode ser feito usando `tridentctl install --generate-custom-yaml` para criar os arquivos YAML e adicionar o `--proxy-url` argumento para o `trident-autosupport` contentor no `trident-deployment.yaml`.

## Desativar métricas do Astra Trident

Para **desabilitar métricas** de serem reportadas, você deve gerar YAMLS personalizados (usando o

--generate-custom-yaml sinalizador) e editá-los para remover o --metrics sinalizador de ser invocado para o trident-main contentor.

## Desinstale o Astra Trident

Você deve usar o mesmo método para desinstalar o Astra Trident usado para instalar o Astra Trident.

### Sobre esta tarefa

- Se você precisar de uma correção para bugs observados após uma atualização, problemas de dependência ou uma atualização mal sucedida ou incompleta, você deve desinstalar o Astra Trident e reinstalar a versão anterior usando as instruções específicas para isso "[versão](#)". Esta é a única maneira recomendada de *downgrade* para uma versão anterior.
- Para facilitar a atualização e reinstalação, a desinstalação do Astra Trident não remove os CRDs ou objetos relacionados criados pelo Astra Trident. Se você precisar remover completamente o Astra Trident e todos os seus dados, "[Remova completamente o Astra Trident e CRDs](#)" consulte .

### Antes de começar

Se você está desativando clusters do Kubernetes, exclua todas as aplicações que usam volumes criados pelo Astra Trident antes da desinstalação. Isso garante que os PVCs sejam inéditos nos nós do Kubernetes antes que sejam excluídos.

## Determine o método de instalação original

Você deve usar o mesmo método para desinstalar o Astra Trident que você usou para instalá-lo. Antes de desinstalar, verifique qual versão você usou para instalar originalmente o Astra Trident.

1. Use `kubectl get pods -n trident` para examinar os pods.
  - Se não houver nenhum pod do operador, o Astra Trident foi instalado usando ``tridentctl`` .
  - Se houver um pod do operador, o Astra Trident foi instalado usando o operador Trident manualmente ou usando o Helm.
2. Se houver um pod do operador, use `kubectl describe tproc trident` para determinar se o Astra Trident foi instalado usando o Helm.
  - Se houver uma etiqueta Helm, o Astra Trident foi instalado usando Helm.
  - Se não houver nenhuma etiqueta Helm, o Astra Trident foi instalado manualmente usando o operador Trident.

## Desinstale a instalação de um operador Trident

Você pode desinstalar manualmente uma instalação do operador do Trident ou usando o Helm.

### Desinstalar a instalação manual

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

1. **Editar `TridentOrchestrator` CR e definir o sinalizador de desinstalação:**

```
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.

2. **Excluir `TridentOrchestrator`:** Ao remover 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.

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

## Desinstale a instalação do 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 uma `tridentctl` instalação

Use o `uninstall` comando in `tridentctl` para remover todos os recursos associados ao Astra Trident, exceto para CRDs e objetos relacionados:

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

## **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.