



Gerenciar e monitorar o Trident

Trident

NetApp
January 15, 2026

Índice

Gerenciar e monitorar o Trident	1
Trident aprimorado	1
Trident aprimorado	1
Atualize com a operadora	2
Atualize com tridentctl	7
Gerencie o Trident usando o tridentctl	8
Comandos e bandeiras globais	8
Opções e parâmetros de comando	10
Suporte a plugins	16
Monitor Trident	16
Visão geral	16
Etapa 1: Defina um alvo do Prometheus	16
Etapa 2: Criar um ServiceMonitor do Prometheus	17
Etapa 3: Consultar métricas do Trident com PromQL	17
Saiba mais sobre a telemetria do Trident AutoSupport.. ..	18
Desativar métricas do Trident	19
Desinstale o Trident	19
Determine o método de instalação original.. ..	20
Desinstalar uma instalação do operador Trident	20
Desinstale um tridentctl instalação	21

Gerenciar e monitorar o Trident

Trident aprimorado

Trident aprimorado

A partir da versão 24.02, o Trident segue um ciclo de lançamentos de quatro meses, disponibilizando três versões principais a cada ano civil. Cada nova versão se baseia nas versões anteriores e oferece novos recursos, melhorias de desempenho, correções de bugs e aprimoramentos. Recomendamos que você atualize pelo menos uma vez por ano para aproveitar os novos recursos do Trident.

Considerações antes de atualizar

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

- Deve haver apenas uma instância do Trident instalada em todos os namespaces de um determinado cluster Kubernetes.
- O Trident 23.07 e versões posteriores exigem snapshots de volume v1 e não são mais compatíveis com snapshots alfa ou beta.
- Se você criou o Cloud Volumes Service para o Google Cloud no "[Tipo de serviço CVS](#)" Você precisa atualizar a configuração do backend para usar o `standardsw` ou `zoneredundantstandardsw` Nível de serviço ao atualizar do Trident 23.01. A falha na atualização do `serviceLevel` Problemas no backend podem causar falhas nos volumes. Consulte "[Exemplos de tipos de serviço da CVS](#)" para mais detalhes.
- Ao fazer o upgrade, é importante que você forneça `parameter.fsType` em `StorageClasses` Usado pela Trident. Você pode excluir e recriar `StorageClasses` sem interromper os volumes preexistentes.
 - Este é um **requisito** para a aplicação da lei. "[contextos de segurança](#)" para volumes SAN.
 - O diretório <https://github.com/NetApp/trident/tree/master/trident-installer/sample-input> [sample input^] contém exemplos, como `storage-class-basic.yaml.templ` e `xref:/trident-managing-k8s/storage-class-bronze-default.yaml`.
 - Para obter mais informações, consulte "[Problemas conhecidos](#)".

Passo 1: Selecione uma versão

As versões do Trident seguem um padrão baseado em datas. `YY.MM` convenção de nomenclatura, onde "AA" são os dois últimos dígitos do ano e "MM" é o mês. Os lançamentos do Dot seguem um `YY.MM.X` convenção, onde "X" é o nível de patch. Você selecionará a versão para a qual deseja atualizar com base na versão atual.

- Você pode realizar uma atualização direta para qualquer versão de destino que esteja dentro de um intervalo de quatro versões da sua versão instalada. Por exemplo, você pode atualizar diretamente da versão 24.06 (ou qualquer versão 24.06 com ponto) para a versão 25.06.
- Se você estiver atualizando de uma versão fora do período de quatro versões, execute uma atualização em várias etapas. Use as instruções de atualização para o "[versão anterior](#)" Você está atualizando para a versão mais recente que se encaixa no período de quatro versões. Por exemplo, se você estiver executando a versão 23.07 e quiser atualizar para a versão 25.06:
 - a. Primeira atualização da versão 23.07 para a 24.06.
 - b. Em seguida, atualize da versão 24.06 para a 25.06.



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

Etapla 2: Determine o método de instalação original.

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

1. Usar `kubectl get pods -n trident` para examinar as cápsulas.
 - Se não houver um módulo de operador, o Trident foi instalado usando `tridentctl` .
 - Caso exista um pod de operador, o Trident foi instalado usando o operador Trident, seja manualmente ou através do Helm.
2. Se houver um módulo de operador, use-o. `kubectl describe torc` Para determinar se o Trident foi instalado usando o Helm.
 - Se houver uma etiqueta "Helm", o Trident foi instalado usando o Helm.
 - Se não houver nenhuma etiqueta "Helm", o Trident foi instalado manualmente usando o operador Trident .

Etapla 3: Selecione um método de atualização

Geralmente, você deve atualizar usando o mesmo método que usou para a instalação inicial; no entanto, você pode ["alternar entre métodos de instalação"](#) . Existem duas opções para aprimorar o Trident.

- ["Faça o upgrade usando o operador Trident ."](#)



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

*

Atualize com a operadora

Entenda o fluxo de trabalho de atualização do operador.

Antes de usar o operador Trident para atualizar o Trident, você deve entender os processos em segundo plano que ocorrem durante a atualização. Isso inclui alterações no controlador Trident , no Pod do controlador e nos Pods dos nós, bem como no DaemonSet dos nós, que permitem atualizações contínuas.

Gestão de atualização do operador Trident

Um dos muitos ["Benefícios de usar o operador Trident"](#) A instalação e atualização do Trident consiste no gerenciamento automático de objetos Trident e Kubernetes sem interromper os volumes montados existentes. Dessa forma, a Trident pode suportar atualizações sem tempo de inatividade, ou ["atualizações contínuas"](#) . Em particular, o operador Trident comunica-se com o cluster Kubernetes para:

- Exclua e recrie a implantação do Trident Controller e o DaemonSet do nó.

- Substitua os pods do controlador Trident e do nó Trident pelas novas versões.
 - Se um nó não for atualizado, isso não impede que os nós restantes sejam atualizados.
 - Somente nós com um Trident Node Pod em execução podem montar volumes.



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

Fluxo de trabalho de atualização do operador

Ao iniciar uma atualização usando o operador Trident :

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

Atualize uma instalação do Trident usando o operador Trident ou o Helm.

Você pode atualizar o Trident usando o operador Trident, seja manualmente ou usando o Helm. Você pode atualizar de uma instalação do operador Trident para outra instalação do operador Trident ou atualizar de uma para outra. `tridentctl` Instalação em uma versão de operador Trident . Analise "[Selecione um método de atualização](#)" antes de atualizar a instalação do operador Trident .

Atualize uma instalação manual.

Você pode atualizar de uma instalação do operador Trident com escopo de cluster para outra instalação do operador Trident com escopo de cluster. Todas as versões do Trident usam um operador com escopo de cluster.



Para atualizar a partir do Trident instalado usando o operador com escopo de namespace (versões 20.07 a 20.10), utilize as instruções de atualização para "[sua versão instalada](#)" do 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 executando Kubernetes 1.24, use "[bundle_pre_1_25.yaml](#)".

- Para clusters executando Kubernetes 1.25 ou posterior, use "[bundle_post_1_25.yaml](#)".

Antes de começar

Certifique-se de estar usando um cluster Kubernetes em execução. "[uma versão do Kubernetes compatível](#)".

Passos

1. Verifique sua versão do Trident :

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

2. Atualizar o `operator.yaml`, `tridentorchestrator_cr.yaml`, e `post_1_25_bundle.yaml` com o registro e os caminhos das imagens da versão para a qual você está atualizando (ex: 25.06) e a chave secreta correta.
3. Exclua o operador Trident que foi usado para instalar a instância atual do Trident . Por exemplo, se você estiver atualizando da versão 25.02, execute o seguinte comando:

```
kubectl delete -f 25.02.0/trident-installer/deploy/<bundle.yaml> -n trident
```

4. 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 espelhados do Trident e do CSI para o modo offline, ativar logs de depuração ou especificar segredos de extração de imagens.
5. Instale o Trident usando o arquivo YAML correto para o seu ambiente, onde `<bundle.yaml>` é `bundle_pre_1_25.yaml` ou `bundle_post_1_25.yaml` com base na sua versão do Kubernetes. Por exemplo, se você estiver instalando o Trident 25.06.0, execute o seguinte comando:

```
kubectl create -f 25.06.0/trident-installer/deploy/<bundle.yaml> -n trident
```

6. Edite o torque do tridente para incluir a imagem 25.06.0.

Atualize uma instalação do Helm

Você pode atualizar uma instalação do Trident Helm.



Ao atualizar um cluster Kubernetes da versão 1.24 para a 1.25 ou posterior que tenha o Trident instalado, você deve atualizar o arquivo `values.yaml` para configurar `excludePodSecurityPolicy` para `true` ou adicione `--set excludePodSecurityPolicy=true` para o `helm upgrade`. Execute o comando antes de atualizar o cluster.

Se você já atualizou seu cluster Kubernetes da versão 1.24 para a 1.25 sem atualizar o Trident Helm, a atualização do Helm falhará. Para que a atualização do leme seja concluída, execute estas etapas como pré-requisitos:

1. Instale o plugin helm-mapkubeapis a partir de <https://github.com/helm/helm-mapkubeapis> .
2. Execute um teste de lançamento do Trident no namespace onde o Trident está instalado. Esta lista contém os recursos que serão limpos.

```
helm mapkubeapis --dry-run trident --namespace trident
```

3. Execute uma corrida completa com o leme para fazer a limpeza.

```
helm mapkubeapis trident --namespace trident
```

Passos

1. Se você "[Instalei o Trident usando o Helm.](#)", você pode usar `helm upgrade trident netapp-trident/trident-operator --version 100.2506.0` Para atualizar em uma única etapa. Se você não adicionou o repositório Helm ou não consegue usá-lo para atualizar:
 - a. Baixe a versão mais recente do Trident em "[a seção Assets no GitHub](#)".
 - b. Use o `helm upgrade` comando onde `trident-operator-25.06.0.tgz` Reflete a versão para a qual você deseja atualizar.



Se você definir opções personalizadas durante a instalação inicial (como especificar registros privados e espelhados para imagens Trident e CSI), adicione o `helm upgrade` comando usando `--set` Para garantir que essas opções estejam incluídas no comando de atualização, caso contrário, os valores serão redefinidos para o padrão.

2. Correr `helm list` Para verificar se o gráfico e a versão do aplicativo foram atualizados. Correr `tridentctl logs` Para analisar quaisquer mensagens de depuração.

Atualize de um `tridentctl` instalação para o operador Trident

Você pode atualizar para a versão mais recente do operador Trident a partir de um `tridentctl` instalação. Os sistemas de backend e PVCs existentes estarão automaticamente disponíveis.



Antes de alternar entre os métodos de instalação, revise "[Alternando entre métodos de instalação](#)".

Passos

1. Baixe a versão mais recente do Trident .

```
# Download the release required [25.06.0]
mkdir 25.06.0
cd 25.06.0
wget
https://github.com/NetApp/trident/releases/download/v25.06.0/trident-
installer-25.06.0.tar.gz
tar -xf trident-installer-25.06.0.tar.gz
cd trident-installer
```

2. Crie o tridentorchestrator CRD 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. Criar um TridentOrchestrator CR para instalação do 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:          v25.06.0
```

Atualize com tridentctl

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

Sobre esta tarefa

Desinstalar e reinstalar o Trident funciona como uma atualização. Ao desinstalar o Trident, o Persistent Volume Claim (PVC) e o Persistent Volume (PV) usados pela implementação do Trident não são excluídos. Os PVs que já foram provisionados permanecerão disponíveis enquanto o Trident estiver offline, e o Trident provisionará volumes para quaisquer PVCs que forem criados nesse período, após o seu retorno à internet.

Antes de começar

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

Passos

1. Execute o comando de desinstalação em `tridentctl` Remover todos os recursos associados ao Trident, exceto os CRDs e objetos relacionados.

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

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



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

Gerencie o Trident usando o tridentctl

O ["Pacote de instalação Trident"](#) inclui o `tridentctl` Utilitário de linha de comando para fornecer acesso simples ao Trident. Usuários do Kubernetes com privilégios suficientes podem usá-lo para instalar o Trident ou gerenciar o namespace que contém o pod do Trident .

Comandos e bandeiras globais

Você pode correr `tridentctl help` para obter uma lista dos comandos disponíveis para `tridentctl` ou acrescente o `--help` Adicione a flag a qualquer comando para obter uma lista de opções e parâmetros para esse comando específico.

```
tridentctl [command] [--optional-flag]
```

O Trident `tridentctl` O utilitário suporta os seguintes comandos e opções globais.

Comandos

create

Adicionar um recurso ao Trident.

delete

Remova um ou mais recursos do Trident.

get

Obtenha um ou mais recursos da Trident.

help

Ajuda com qualquer comando.

images

Imprima uma tabela com as imagens dos contêineres que o Trident precisa.

import

Importe um recurso existente para o Trident.

install

Instale o Trident.

logs

Imprima os registros do Trident.

send

Enviar um recurso do Trident.

uninstall

Desinstale o Trident.

update

Modificar um recurso no Trident.

update backend state

Suspender temporariamente as operações de backend.

upgrade

Aprimore um recurso no Trident.

version

Imprima a versão do Trident.

Bandeiras globais

-d, --debug

Saída de depuração.

-h, --help

Ajuda para `tridentctl`.

-k, --kubeconfig string

Especifique o KUBECONFIG Caminho para executar comandos localmente ou de um cluster Kubernetes para outro.



Alternativamente, você pode exportar o KUBECONFIG variável para apontar para um cluster Kubernetes específico e problema `tridentctl` comandos para esse cluster.

-n, --namespace string

Espaço de nomes da implantação do Trident.

-o, --output string

Formato de saída. Uma das opções: `json|yaml|name|wide|ps` (padrão).

-s, --server string

Endereço/porta da interface REST do Trident.



A interface REST do Trident pode ser configurada para escutar e servir apenas em 127.0.0.1 (para IPv4) ou `:::1` (para IPv6).

Opções e parâmetros de comando

criar

Use o `create` comando para adicionar um recurso ao Trident.

```
tridentctl create [option]
```

Opções

``backend`` Adicionar um backend ao Trident.

excluir

Use o `delete` comando para remover um ou mais recursos do Trident.

```
tridentctl delete [option]
```

Opções

`backend`` Excluir um ou mais servidores de armazenamento do Trident.

``snapshot`` Excluir um ou mais snapshots de volume do Trident.

`storageclass`` Excluir uma ou mais classes de armazenamento do Trident.

`volume` Excluir um ou mais volumes de armazenamento do Trident.

pegar

Use o `get` comando para obter um ou mais recursos do Trident.

```
tridentctl get [option]
```

Opções

`backend`` Obtenha um ou mais backends de armazenamento da Trident.

``snapshot` Obtenha uma ou mais fotos instantâneas do Trident.

`storageclass` Obtenha uma ou mais classes de armazenamento da Trident.

`volume` Adquira um ou mais volumes da Trident.

Bandeiras

`-h, --help` Ajuda para volumes.

`--parentOfSubordinate string`: Limitar a consulta ao volume de origem subordinado.

`--subordinateOf string`: Limitar a consulta aos subordinados do volume.

imagens

Usar `images` Bandeiras para imprimir uma tabela com as imagens dos contêineres que o Trident precisa.

```
tridentctl images [flags]
```

Bandeiras

`-h, --help` Ajuda para imagens.

`-v, --k8s-version string` Versão semântica do cluster Kubernetes.

volume de importação

Use o `import volume` comando para importar um volume existente para o Trident.

```
tridentctl import volume <backendName> <volumeName> [flags]
```

Pseudônimos

`volume, v`

Bandeiras

`-f, --filename string` Caminho para o arquivo PVC em YAML ou JSON.

`-h, --help` Ajuda para o volume.

`--no-manage` Criar apenas PV/PVC. Não assuma a gestão do ciclo de vida do volume.

instalar

Use o `install` sinalizadores para instalar o Trident.

```
tridentctl install [flags]
```

Bandeiras

`--autosupport-image string`: A imagem do contêiner para Telemetria de Suporte Automático (padrão "netapp/trident autosupport:<versão-atual>").

`--autosupport-proxy string` Endereço/porta de um proxy para envio de telemetria do Autosupport.

`--enable-node-prep` Tentativa de instalação dos pacotes necessários nos nós.

`--generate-custom-yaml` Gere arquivos YAML sem instalar nada.

`-h, --help` Ajuda para instalação.

`--http-request-timeout` : Substitui o tempo limite de requisição HTTP para a API REST do controlador Trident (padrão de 1m30s).

`--image-registry string` : O endereço/porta de um registro de imagens interno.

`--k8s-timeout duration` : O tempo limite para todas as operações do Kubernetes (padrão: 3m0s).

`--kubelet-dir string` : O local do host onde o estado interno do kubelet está armazenado (o padrão é "/var/lib/kubelet").

`--log-format string` : O formato de registro do Trident (texto, json) (padrão "texto").

`--node-prep` Permite que o Trident prepare os nós do cluster Kubernetes para gerenciar volumes usando o protocolo de armazenamento de dados especificado. **Atualmente, `iscsi` é o único valor suportado. A partir do OpenShift 4.19, a versão mínima do Trident compatível com este recurso é a 25.06.1.**

`--pv string` : O nome do PV legado usado pelo Trident; certifique-se de que ele não exista (padrão "trident").

`--pvc string` : O nome do PVC legado usado pelo Trident; certifique-se de que ele não exista (padrão "trident").

`--silence-autosupport` : Não enviar pacotes de suporte automático para a NetApp automaticamente (padrão: verdadeiro).

`--silent` Desative a maior parte das saídas durante a instalação.

`--trident-image string` : A imagem do Trident para instalar.

`--k8s-api-qps` : O limite de consultas por segundo (QPS) para solicitações da API do Kubernetes (padrão 100; opcional).

`--use-custom-yaml` Utilize quaisquer arquivos YAML existentes no diretório de configuração.

`--use-ipv6` Utilize IPv6 para a comunicação do Trident.

toras

Usar logs sinalizadores para imprimir os registros do Trident.

```
tridentctl logs [flags]
```

Bandeiras

`-a, --archive` Criar um arquivo de suporte com todos os registros, a menos que especificado de outra forma.

`-h, --help` Ajuda para registros.

`-l, --log string` : Registro do Trident para exibir. Uma das opções: trident|auto|trident-operator|all (padrão "auto").

`--node string` : O nome do nó Kubernetes do qual coletar os logs dos pods.

`-p, --previous` Obtenha os registros da instância de contêiner anterior, se existirem.

`--sidecars` Obtenha os registros dos contêineres auxiliares.

enviar

Use o `send` comando para enviar um recurso do Trident.

```
tridentctl send [option]
```

Opções

``autosupport`` Envie um arquivo de suporte automático para a NetApp.

desinstalar

Usar `uninstall` sinalizadores para desinstalar o Trident.

```
tridentctl uninstall [flags]
```

Bandeiras

`-h, --help`` Ajuda para desinstalar.

``--silent`` Desative a maior parte das saídas durante a desinstalação.

atualizar

Use o `update` comando para modificar um recurso no Trident.

```
tridentctl update [option]
```

Opções

``backend`` Atualizar um backend no Trident.

atualizar estado do backend

Use o `update backend state` Comando para suspender ou retomar operações de backend.

```
tridentctl update backend state <backend-name> [flag]
```

Pontos a considerar

- Se um backend for criado usando um `TridentBackendConfig` (tbc), o backend não poderá ser atualizado usando um `backend.json` arquivo.
- Se o `userState` foi definido em um tbc, não pode ser modificado usando o `tridentctl update backend state <backend-name> --user-state suspended/normal` comando.
- Para recuperar a capacidade de definir o `userState` via `tridentctl` após ter sido configurado via tbc, o `userState` O campo deve ser removido do tbc. Isso pode ser feito usando o `kubectrl edit tbc` comando. Depois do `userState` O campo foi removido, você pode usar o `tridentctl update backend state` comando para alterar o `userState` de um backend.
- Use o `tridentctl update backend state` para mudar o `userState`. Você também pode atualizar o `userState` usando `TridentBackendConfig` ou `backend.json` arquivo; isso desencadeia uma reinicialização completa do backend e pode ser demorado.

Bandeiras

`-h, --help`` Ajuda para o estado do backend.

`--user-state`` Definir para `suspended` Para pausar as operações de backend. Definir para `normal` para retomar as operações de back-end. Quando definido para `suspended``:

- `AddVolume`` e ``Import Volume` estão em pausa.
- `CloneVolume`, `ResizeVolume`, `PublishVolume`, `UnPublishVolume`, `CreateSnapshot`, `GetSnapshot`, `RestoreSnapshot`, `DeleteSnapshot`, `RemoveVolume`, `GetVolumeExternal`,

`ReconcileNodeAccess` permanecer disponível.

Você também pode atualizar o estado do backend usando `userState` campo no arquivo de configuração do backend `TridentBackendConfig` ou `backend.json`. Para obter mais informações, consulte ["Opções para gerenciar back-ends"](#) e ["Realize o gerenciamento de backend com kubectl"](#).

Exemplo:

JSON

Siga estes passos para atualizar o `userState` usando o `backend.json` arquivo:

1. Editar o `backend.json` arquivo para incluir o `userState` campo com o valor definido como `'suspenseo'`.
2. Atualize o backend usando o `tridentctl update backend` comando e o caminho para a versão atualizada `backend.json` arquivo.

Exemplo: `tridentctl update backend -f /<path to backend JSON file>/backend.json -n trident`

```
{
  "version": 1,
  "storageDriverName": "ontap-nas",
  "managementLIF": "<redacted>",
  "svm": "nas-svm",
  "backendName": "customBackend",
  "username": "<redacted>",
  "password": "<redacted>",
  "userState": "suspended"
}
```

YAML

Você pode editar o TBC depois que ele for aplicado usando o `kubectl edit <tbc-name> -n <namespace>` comando. O exemplo a seguir atualiza o estado do backend para suspender usando o `userState: suspended` opção:

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-ontap-nas
spec:
  version: 1
  backendName: customBackend
  storageDriverName: ontap-nas
  managementLIF: <redacted>
  svm: nas-svm
  userState: suspended
  credentials:
    name: backend-tbc-ontap-nas-secret
```

versão

Usar `version` bandeiras para imprimir a versão de `tridentctl` e o serviço Trident em funcionamento.

```
tridentctl version [flags]
```

Bandeiras

- `--client`: Somente versão cliente (não requer servidor).
- `-h`, `--help` Ajuda para a versão.

Suporte a plugins

O `Tridentctl` suporta plugins semelhantes ao `kubectl`. O `Tridentctl` detecta um plugin se o nome do arquivo binário do plugin seguir o esquema "`tridentctl-<plugin>`" e o binário estiver localizado em uma pasta listada na variável de ambiente `PATH`. Todos os plugins detectados estão listados na seção de plugins da ajuda do `tridentctl`. Opcionalmente, você também pode limitar a pesquisa especificando uma pasta de plugins na variável de ambiente `TRIDENTCTL_PLUGIN_PATH` (Exemplo: `TRIDENTCTL_PLUGIN_PATH=~/.tridentctl-plugins/`). Se a variável for usada, o `tridentctl` pesquisará apenas na pasta especificada.

Monitor Trident

O Trident fornece um conjunto de endpoints de métricas do Prometheus que você pode usar para monitorar o desempenho do Trident .

Visão geral

As métricas fornecidas pela Trident permitem que você faça o seguinte:

- Monitore o estado de saúde e a configuração do Trident. Você pode analisar o sucesso das operações e verificar se a comunicação com os sistemas de backend ocorre conforme o esperado.
- Analise as informações de utilização do backend e compreenda quantos volumes estão provisionados em um backend, a quantidade de espaço consumido e assim por diante.
- Mantenha um mapeamento da quantidade de volumes provisionados nos backends disponíveis.
- Acompanhe o desempenho. Você pode verificar quanto tempo o Trident leva para se comunicar com os servidores de backend e executar operações.



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

O que você vai precisar

- Um cluster Kubernetes com o Trident instalado.
- Uma instância do Prometheus. Isso pode ser um ["implantação do Prometheus em contêineres"](#) ou você pode optar por executar o Prometheus como um ["aplicativo nativo"](#) .

Etapa 1: Defina um alvo do Prometheus

Você deve definir um alvo do Prometheus para coletar as métricas e obter informações sobre os backends gerenciados Trident , os volumes que ele cria e assim por diante. Esse ["blog"](#) Explica como você pode usar o Prometheus e o Grafana com o Trident para obter métricas. O blog explica como você pode executar o

Prometheus como um operador em seu cluster Kubernetes e a criação de um ServiceMonitor para obter métricas do Trident .

Etapa 2: Criar um ServiceMonitor do Prometheus

Para consumir as métricas do Trident , você deve criar um ServiceMonitor do Prometheus que monitore o `trident-csi` serviço e escuta no `metrics` porta. Um exemplo de ServiceMonitor tem a seguinte aparência:

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

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

Além das métricas disponíveis diretamente no Trident, o kubelet expõe muitas outras. `kubelet_volume_*` métricas através de seu próprio endpoint de métricas. O Kubelet pode fornecer informações sobre os volumes que estão anexados, bem como sobre os pods e outras operações internas que ele gerencia. Consulte ["aqui"](#) .

Etapa 3: Consultar métricas do Trident com PromQL

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

Aqui estão algumas consultas PromQL que você pode usar:

Obtenha informações sobre saúde da Trident

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

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

- **Percentagem de respostas REST do 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 Trident**

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

Obtenha informações sobre o uso do Trident

- **Tamanho médio do volume**

```
trident_volume_allocated_bytes/trident_volume_count
```

- **Espaço total provisionado por cada backend**

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

Obtenha o consumo de volume individual.



Isso só é habilitado se as métricas do kubelet também forem coletadas.

- **Percentagem de espaço utilizado para cada volume**

```
kubelet_volume_stats_used_bytes / kubelet_volume_stats_capacity_bytes *
100
```

Saiba mais sobre a telemetria do Trident AutoSupport.

Por padrão, o Trident envia métricas do Prometheus e informações básicas de backend para a NetApp diariamente.

- Para impedir que o Trident envie métricas do Prometheus e informações básicas de backend para a NetApp, passe o seguinte: `--silence-autosupport` bandeira durante a instalação do Trident .
- O Trident também pode enviar logs de contêineres para o Suporte da NetApp sob demanda via `tridentctl send autosupport` . Você precisará acionar o Trident para que ele envie seus registros. Antes de enviar os registros, você deve aceitar os termos da NetApp <https://www.netapp.com/company/legal/privacy-policy/> ["política de Privacidade"] .

- A menos que especificado de outra forma, o Trident busca os registros das últimas 24 horas.
- Você pode especificar o período de retenção de logs com o `--since` bandeira. Por exemplo: `tridentctl send autosupport --since=1h`. Essas informações são coletadas e enviadas por meio de um `trident-autosupport` contêiner que é instalado ao lado do Trident. Você pode obter a imagem do contêiner em "[AutoSupport Trident](#)".
- A Trident AutoSupport não coleta nem transmite informações de identificação pessoal (PII) ou informações pessoais. Ele vem com um "[Contrato de licença de usuário final \(EULA\)](#)". Isso não se aplica à própria imagem do contêiner Trident. Você pode saber mais sobre o compromisso da NetApp com a segurança e a confiabilidade dos dados. "[aqui](#)".

Um exemplo de carga útil enviada pelo Trident é semelhante a esta:

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

- As mensagens do AutoSupport são enviadas para o endpoint do AutoSupport da NetApp. Se você estiver usando um registro privado para armazenar imagens de contêiner, poderá usar o `--image-registry` bandeira.
- 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 a favor do `trident-autosupport` contêiner em `trident-deployment.yaml`.

Desativar métricas do Trident

Para **desativar** a exibição de métricas, você deve gerar arquivos YAML personalizados (usando o `--generate-custom-yaml` flag) e edite-os para remover o `--metrics` impedir que a flag seja invocada para o `trident-main` recipiente.

Desinstale o Trident

Você deve usar o mesmo método para desinstalar o Trident que usou para instalá-lo.

Sobre esta tarefa

- Se precisar corrigir erros observados após uma atualização, problemas de dependência ou uma atualização incompleta ou malsucedida, desinstale o Trident e reinstale a versão anterior seguindo as instruções específicas para essa versão. "[versão](#)". Esta é a única maneira recomendada de *fazer o downgrade* para uma versão anterior.
- Para facilitar a atualização e reinstalação, desinstalar o Trident não remove os CRDs ou objetos relacionados criados pelo Trident. Se precisar remover completamente o Trident e todos os seus dados, consulte "[Remova completamente o Trident e os CRDs](#)".

Antes de começar

Se você estiver desativando clusters Kubernetes, deverá excluir todos os aplicativos que usam volumes criados pelo Trident antes de desinstalá-lo. Isso garante que os PVCs sejam despublicados nos nós do Kubernetes antes de serem excluídos.

Determine o método de instalação original.

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

1. Usar `kubectl get pods -n trident` para examinar as cápsulas.
 - Se não houver um módulo de operador, o Trident foi instalado usando `tridentctl`.
 - Caso exista um pod de operador, o Trident foi instalado usando o operador Trident, seja manualmente ou através do Helm.
2. Se houver um módulo de operador, use-o. `kubectl describe tproc trident` Para determinar se o Trident foi instalado usando o Helm.
 - Se houver uma etiqueta "Helm", o Trident foi instalado usando o Helm.
 - Se não houver nenhuma etiqueta "Helm", o Trident foi instalado manualmente usando o operador Trident.

Desinstalar uma instalação do operador Trident

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

Desinstalar instalação manual

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

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

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

Quando o `uninstall` A flag está definida como `true` O operador Trident desinstala o Trident, mas não remove o próprio `TridentOrchestrator`. Você deve limpar a pasta `TridentOrchestrator` e criar uma nova se quiser instalar o Trident novamente.

2. **Excluir `TridentOrchestrator`** : Removendo o `TridentOrchestrator` CR que foi usado para implantar o Trident, você instrui o operador a desinstalar o Trident. O operador processa a remoção de

TridentOrchestrator e procede à remoção da implantação e do daemonset do Trident , excluindo os pods do Trident que havia criado como parte da instalação.

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

Desinstalar a instalação do Helm

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

```
#List the Helm release corresponding to the Trident install.
helm ls -n trident
NAME                NAMESPACE          REVISION          UPDATED
STATUS              CHART               APP VERSION
trident             trident             1                2021-04-20
00:26:42.417764794 +0000 UTC deployed    trident-operator-21.07.1
21.07.1

#Uninstall Helm release to remove Trident
helm uninstall trident -n trident
release "trident" uninstalled
```

Desinstale um tridentctl instalação

Use o `uninstall` comando em `tridentctl` Remover todos os recursos associados ao Trident, exceto os CRDs e objetos relacionados:

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

Informações sobre direitos autorais

Copyright © 2026 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 DOCUMENTOS, 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.