



Gérer et surveiller Trident

Trident

NetApp
March 05, 2026

Sommaire

Gérer et surveiller Trident	1
Amélioration du Trident	1
Amélioration du Trident	1
Mise à niveau avec l'opérateur	2
Mise à niveau avec tridentctl	7
Gérez Trident à l'aide de tridentctl	8
Commandes et drapeaux globaux	8
Options de commande et drapeaux	10
Prise en charge des plugins	16
Monitor Trident	16
Aperçu	16
Étape 1 : Définir une cible Prometheus	16
Étape 2 : Créer un moniteur de service Prometheus	17
Étape 3 : Interroger les métriques Trident avec PromQL	17
Découvrez la télémétrie de Trident AutoSupport	18
Désactiver les métriques Trident	19
Désinstaller Trident	20
Déterminer la méthode d'installation d'origine	20
Désinstaller une installation d'opérateur Trident	20
Désinstaller un tridentctl installation	21

Gérer et surveiller Trident

Amélioration du Trident

Amélioration du Trident

À compter de la version 24.02, Trident suit un rythme de sortie de quatre mois, proposant trois mises à jour majeures chaque année civile. Chaque nouvelle version s'appuie sur les versions précédentes et propose de nouvelles fonctionnalités, des améliorations de performances, des corrections de bugs et des améliorations générales. Nous vous encourageons à effectuer la mise à niveau au moins une fois par an pour profiter des nouvelles fonctionnalités de Trident.

Considérations avant la mise à niveau

Lors de la mise à niveau vers la dernière version de Trident, tenez compte des points suivants :

- Une seule instance de Trident doit être installée dans tous les espaces de noms d'un cluster Kubernetes donné.
- Trident 23.07 et versions ultérieures nécessitent des instantanés de volume v1 et ne prennent plus en charge les instantanés alpha ou bêta.
- Si vous avez créé un Cloud Volumes Service pour Google Cloud dans le ["type de service CVS"](#), vous devez mettre à jour la configuration du backend pour utiliser le `standardsw` ou `zoneredundantstandardsw` niveau de service lors de la mise à niveau depuis Trident 23.01. Défaut de mise à jour `serviceLevel` Des problèmes au niveau du système dorsal pourraient entraîner des défaillances de volumes. Se référer à ["Exemples de type de service CVS"](#) pour plus de détails.
- Lors de la mise à niveau, il est important que vous fournissiez `parameter.fsType` dans `StorageClasses` utilisé par Trident. Vous pouvez supprimer et recréer `StorageClasses` sans perturber les volumes préexistants.
 - Il s'agit d'une **condition** pour faire respecter ["contextes de sécurité"](#) pour les volumes SAN.
 - Le répertoire <https://github.com/NetApp/trident/tree/master/trident-installer/sample-input> [exemple d'entrée^] contient des exemples, tels que `storage-class-basic.yaml` et `storage-class-bronze-default.yaml`.
 - Pour plus d'informations, veuillez consulter ["Problèmes connus"](#).

Étape 1 : Sélectionnez une version

Les versions de Trident suivent un système basé sur la date. `YY.MM` convention d'appellation, où « AAAA » représente les deux derniers chiffres de l'année et « MM » le mois. Les sorties de Dot suivent un `YY.MM.X` convention, où « X » représente le niveau du patch. Vous sélectionnez la version vers laquelle effectuer la mise à niveau en fonction de la version actuelle.

- Vous pouvez effectuer une mise à niveau directe vers n'importe quelle version cible se trouvant dans un intervalle de quatre versions autour de votre version installée. Par exemple, vous pouvez effectuer une mise à niveau directe de la version 24.06 (ou de toute version 24.06) vers la version 25.06.
- Si vous effectuez une mise à niveau à partir d'une version antérieure à la période de quatre mises à niveau, procédez à une mise à niveau en plusieurs étapes. Utilisez les instructions de mise à niveau pour le ["version antérieure"](#) Vous effectuez une mise à niveau depuis une version antérieure afin de passer à la

version la plus récente compatible avec la fenêtre de quatre versions. Par exemple, si vous utilisez la version 23.07 et que vous souhaitez passer à la version 25.06 :

- a. Première mise à jour du 23.07 au 24.06.
- b. Ensuite, passez de la version 24.06 à la version 25.06.



Lors de la mise à niveau à l'aide de l'opérateur Trident sur OpenShift Container Platform, vous devez effectuer une mise à niveau vers Trident 21.01.1 ou une version ultérieure. L'opérateur Trident publié avec la version 21.01.0 contient un problème connu qui a été corrigé dans la version 21.01.1. Pour plus de détails, veuillez vous référer au ["Détails du problème sur GitHub"](#).

Étape 2 : Déterminer la méthode d'installation d'origine

Pour déterminer la version que vous avez utilisée pour installer Trident à l'origine :

1. Utiliser `kubectl get pods -n trident` pour examiner les gousses.
 - En l'absence de nacelle opérateur, Trident a été installé via `tridentctl`.
 - S'il existe un module opérateur, Trident a été installé à l'aide de l'opérateur Trident, soit manuellement, soit via Helm.
2. S'il y a une cabine d'opérateur, utilisez-la `kubectl describe torc` pour déterminer si Trident a été installé à l'aide de Helm.
 - Si une étiquette Helm est présente, Trident a été installé à l'aide de Helm.
 - S'il n'y a pas d'étiquette Helm, Trident a été installé manuellement à l'aide de l'opérateur Trident.

Étape 3 : Sélectionnez une méthode de mise à niveau

En règle générale, vous devez effectuer la mise à niveau en utilisant la même méthode que celle utilisée pour l'installation initiale, mais vous pouvez ["passer d'une méthode d'installation à une autre"](#). Il existe deux options pour mettre à niveau Trident.

- ["Mise à niveau via l'opérateur Trident"](#)



Nous vous suggérons de consulter ["Comprendre le flux de travail de mise à niveau de l'opérateur"](#) avant de procéder à la mise à niveau auprès de l'opérateur.

*

Mise à niveau avec l'opérateur

Comprendre le flux de travail de mise à niveau de l'opérateur

Avant d'utiliser l'opérateur Trident pour mettre à niveau Trident, vous devez comprendre les processus d'arrière-plan qui se produisent pendant la mise à niveau. Cela inclut des modifications apportées au contrôleur Trident, au Pod du contrôleur et aux Pods du nœud, ainsi qu'au DaemonSet du nœud, qui permettent des mises à jour progressives.

Gestion de la mise à niveau de l'opérateur Trident

L'un des nombreux ["avantages de l'utilisation de l'opérateur Trident"](#) L'installation et la mise à niveau de Trident

consistent en la gestion automatique des objets Trident et Kubernetes sans perturber les volumes montés existants. De cette manière, Trident peut prendre en charge les mises à niveau sans aucune interruption de service, ou "[mises à jour continues](#)". Plus précisément, l'opérateur Trident communique avec le cluster Kubernetes pour :

- Supprimez et recréez le déploiement du contrôleur Trident et le DaemonSet du nœud.
- Remplacez le module de contrôleur Trident et les modules de nœud Trident par les nouvelles versions.
 - Le fait qu'un nœud ne soit pas mis à jour n'empêche pas la mise à jour des autres nœuds.
 - Seuls les nœuds disposant d'un pod de nœud Trident en cours d'exécution peuvent monter des volumes.



Pour plus d'informations sur l'architecture Trident sur le cluster Kubernetes, consultez la documentation. "[Architecture Trident](#)".

Flux de travail de mise à niveau de l'opérateur

Lorsque vous lancez une mise à niveau à l'aide de l'opérateur Trident :

1. L'opérateur * Trident * :
 - a. Détecte la version actuellement installée de Trident (version n).
 - b. Met à jour tous les objets Kubernetes, y compris les CRD, RBAC et Trident SVC.
 - c. Supprime le déploiement du contrôleur Trident pour la version n .
 - d. Crée le déploiement du contrôleur Trident pour la version $n+1$.
2. **Kubernetes** crée un pod de contrôleur Trident pour $n+1$.
3. L'opérateur * Trident * :
 - a. Supprime le DaemonSet de nœud Trident pour n . L'opérateur n'attend pas la fin du Node Pod.
 - b. Crée le Daemonset de nœud Trident pour $n+1$.
4. **Kubernetes** crée des pods de nœuds Trident sur les nœuds n'exécutant pas le pod de nœud Trident n . Cela garantit qu'il n'y a jamais plus d'un Trident Node Pod, quelle que soit sa version, sur un nœud.

Mettez à niveau une installation Trident à l'aide de l'opérateur Trident ou de Helm.

Vous pouvez mettre à niveau Trident à l'aide de l'opérateur Trident , soit manuellement, soit via Helm. Vous pouvez effectuer une mise à niveau d'une installation d'opérateur Trident vers une autre installation d'opérateur Trident ou effectuer une mise à niveau depuis une `tridentctl` installation sur une version opérateur Trident .

Revoir "[Sélectionnez une méthode de mise à niveau](#)" avant de mettre à niveau une installation d'opérateur Trident .

Mise à niveau d'une installation manuelle

Vous pouvez effectuer une mise à niveau d'une installation d'opérateur Trident à l'échelle d'un cluster vers une autre installation d'opérateur Trident à l'échelle d'un cluster. Toutes les versions de Trident utilisent un opérateur à portée de cluster.



Pour mettre à niveau une version de Trident installée à l'aide de l'opérateur à portée d'espace de noms (versions 20.07 à 20.10), suivez les instructions de mise à niveau pour ["votre version installée"](#) de Trident.

À propos de cette tâche

Trident fournit un fichier bundle que vous pouvez utiliser pour installer l'opérateur et créer les objets associés pour votre version de Kubernetes.

- Pour les clusters exécutant Kubernetes 1.24, utilisez ["bundle_pre_1_25.yaml"](#).
- Pour les clusters exécutant Kubernetes 1.25 ou une version ultérieure, utilisez ["bundle_post_1_25.yaml"](#).

Avant de commencer

Assurez-vous d'utiliser un cluster Kubernetes en cours d'exécution. ["une version Kubernetes prise en charge"](#).

Étapes

1. Vérifiez votre version de Trident :

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

2. Mettre à jour `operator.yaml`, `tridentorchestrator_cr.yaml`, et `post_1_25_bundle.yaml` avec le registre et les chemins d'accès aux images pour la version vers laquelle vous effectuez la mise à niveau (par exemple 25.06), et le secret correct.
3. Supprimez l'opérateur Trident qui a été utilisé pour installer l'instance Trident actuelle. Par exemple, si vous effectuez une mise à niveau à partir de la version 25.02, exécutez la commande suivante :

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

4. Si vous avez personnalisé votre installation initiale en utilisant `TridentOrchestrator` les attributs, vous pouvez les modifier `TridentOrchestrator` objet permettant de modifier les paramètres d'installation. Cela peut inclure des modifications apportées pour spécifier les registres d'images Trident et CSI en miroir pour le mode hors ligne, activer les journaux de débogage ou spécifier les secrets d'extraction d'images.
5. Installez Trident en utilisant le fichier YAML de bundle approprié à votre environnement, où `<bundle.yaml>` est `bundle_pre_1_25.yaml` ou `bundle_post_1_25.yaml` en fonction de votre version de Kubernetes. Par exemple, si vous installez Trident 25.06.0, exécutez la commande suivante :

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

6. Modifiez le `torc trident` pour inclure l'image 25.06.0.

Mettre à niveau une installation Helm

Vous pouvez mettre à niveau une installation de Trident Helm.



Lors de la mise à niveau d'un cluster Kubernetes de la version 1.24 à la version 1.25 ou ultérieure sur lequel Trident est installé, vous devez mettre à jour le fichier `values.yaml` pour configurer `excludePodSecurityPolicy` à `true` ou ajouter `--set excludePodSecurityPolicy=true` au `helm upgrade` commande avant de pouvoir mettre à niveau le cluster.

Si vous avez déjà mis à niveau votre cluster Kubernetes de la version 1.24 à la version 1.25 sans mettre à niveau le helm Trident, la mise à niveau du helm échoue. Pour que la mise à niveau du poste de pilotage puisse être effectuée, veuillez suivre les étapes préalables suivantes :

1. Installez le plugin `helm-mapkubeapis` depuis <https://github.com/helm/helm-mapkubeapis>.
2. Effectuez un test à blanc pour la version de Trident dans l'espace de noms où Trident est installé. Cette liste répertorie les ressources qui seront nettoyées.

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

3. Effectuez une exécution complète avec Helm pour effectuer le nettoyage.

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

Étapes

1. Si tu "Trident a été installé à l'aide de Helm.", vous pouvez utiliser `helm upgrade trident netapp-trident/trident-operator --version 100.2506.0` mettre à niveau en une seule étape. Si vous n'avez pas ajouté le dépôt Helm ou si vous ne pouvez pas l'utiliser pour effectuer la mise à niveau :
 - a. Téléchargez la dernière version de Trident depuis "la section [Assets](#) sur GitHub".
 - b. Utilisez le `helm upgrade` commande où `trident-operator-25.06.0.tgz` reflète la version vers laquelle vous souhaitez effectuer la mise à niveau.

```
helm upgrade <name> trident-operator-25.06.0.tgz
```



Si vous définissez des options personnalisées lors de l'installation initiale (comme la spécification de registres privés et en miroir pour les images Trident et CSI), ajoutez le `helm upgrade` commande utilisant `--set` afin de garantir que ces options soient incluses dans la commande de mise à niveau, sinon les valeurs seront réinitialisées à leurs valeurs par défaut.

2. Courir `helm list` vérifier que la version du graphique et celle de l'application ont bien été mises à jour. Courir `tridentctl logs` pour examiner les messages de débogage.

Mise à niveau depuis un `tridentctl` installation à l'opérateur Trident

Vous pouvez mettre à jour l'opérateur Trident vers sa dernière version à partir d'un `tridentctl` installation. Les backends et les PVC existants seront automatiquement disponibles.



Avant de changer de méthode d'installation, consultez ["Passer d'une méthode d'installation à une autre"](#).

Étapes

1. Téléchargez la dernière version de 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. Créez le tridentorchestrator CRD du manifeste.

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

3. Déployez l'opérateur à portée de cluster dans le même espace de noms.

```
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. Créer un TridentOrchestrator CR pour l'installation de 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. Confirmez que Trident a bien été mis à niveau vers la version prévue.

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

Message:          Trident installed
Namespace:        trident
Status:           Installed
Version:          v25.06.0
```

Mise à niveau avec tridentctl

Vous pouvez facilement mettre à niveau une installation Trident existante en utilisant `tridentctl`.

À propos de cette tâche

Désinstaller puis réinstaller Trident équivaut à une mise à jour. Lorsque vous désinstallez Trident, la revendication de volume persistant (PVC) et le volume persistant (PV) utilisés par le déploiement de Trident ne sont pas supprimés. Les volumes persistants (PV) déjà provisionnés resteront disponibles pendant que Trident est hors ligne, et Trident provisionnera des volumes pour tous les volumes persistants (PVC) créés entre-temps après sa remise en ligne.

Avant de commencer

Revoir ["Sélectionnez une méthode de mise à niveau"](#) avant de mettre à niveau en utilisant `tridentctl`.

Étapes

1. Exécutez la commande de désinstallation dans `tridentctl` supprimer toutes les ressources associées à Trident, à l'exception des CRD et des objets associés.

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

2. Réinstallez Trident. Se référer à ["Installez Trident à l'aide de tridentctl"](#) .



N'interrompez pas le processus de mise à niveau. Assurez-vous que l'installation se déroule jusqu'à son terme.

Gérez Trident à l'aide de tridentctl

Le ["Pack d'installation Trident"](#) comprend le `tridentctl` utilitaire en ligne de commande pour fournir un accès simple à Trident. Les utilisateurs de Kubernetes disposant de privilèges suffisants peuvent l'utiliser pour installer Trident ou gérer l'espace de noms contenant le pod Trident .

Commandes et drapeaux globaux

Vous pouvez courir `tridentctl help` pour obtenir la liste des commandes disponibles pour `tridentctl` ou ajouter le `--help` Utilisez l'option `--flag` avec n'importe quelle commande pour obtenir la liste des options et des indicateurs disponibles pour cette commande spécifique.

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

Le Trident `tridentctl` Cet utilitaire prend en charge les commandes et les options globales suivantes.

Commandes

create

Ajouter une ressource à Trident.

delete

Supprimer une ou plusieurs ressources de Trident.

get

Obtenez une ou plusieurs ressources de Trident.

help

Aide concernant n'importe quelle commande.

images

Imprimez un tableau des images conteneurs nécessaires à Trident .

import

Importer une ressource existante dans Trident.

install

Installez Trident.

logs

Imprimer les journaux de Trident.

send

Envoyer une ressource depuis Trident.

uninstall

Désinstallez Trident.

update

Modifier une ressource dans Trident.

update backend state

Suspension temporaire des opérations en arrière-plan.

upgrade

Mettre à niveau une ressource dans Trident.

version

Imprimez la version de Trident.

drapeaux du monde

-d, --debug

Sortie de débogage.

-h, --help

Aide pour `tridentctl`.

-k, --kubeconfig string

Précisez le KUBECONFIG chemin d'exécution des commandes localement ou d'un cluster Kubernetes à un autre.



Vous pouvez également exporter le KUBECONFIG variable permettant de pointer vers un cluster Kubernetes spécifique et de résoudre le problème `tridentctl` commandes à ce cluster.

-n, --namespace string

Espace de noms du déploiement Trident.

-o, --output string

Format de sortie. L'un des formats `json|yaml|name|wide|ps` (par défaut).

-s, --server string

Adresse/port de l'interface REST de Trident.



L'interface REST de Trident peut être configurée pour écouter et servir uniquement à l'adresse `127.0.0.1` (pour IPv4) ou `:::1` (pour IPv6).

Options de commande et drapeaux

créer

Utilisez le `create` commande pour ajouter une ressource à Trident.

```
tridentctl create [option]
```

Options

`backend`: Ajouter un backend à Trident.

supprimer

Utilisez le `delete` commande permettant de supprimer une ou plusieurs ressources de Trident.

```
tridentctl delete [option]
```

Options

`backend`: Supprimez un ou plusieurs backends de stockage de Trident.

`snapshot` : Supprimez un ou plusieurs instantanés de volume de Trident.

`storageclass` : Supprimez une ou plusieurs classes de stockage de Trident.
`volume` : Supprimez un ou plusieurs volumes de stockage de Trident.

obtenir

Utilisez le `get` commande pour obtenir une ou plusieurs ressources de Trident.

```
tridentctl get [option]
```

Options

`backend` : Obtenez un ou plusieurs backends de stockage auprès de Trident.
``snapshot` : Obtenez un ou plusieurs instantanés de Trident.
`storageclass` : Obtenez une ou plusieurs classes de stockage de Trident.
`volume` : Procurez-vous un ou plusieurs volumes chez Trident.

drapeaux

`-h, --help` : Aide pour les volumes.
`--parentOfSubordinate string` Limiter la requête au volume de la source subordonnée.
`--subordinateOf string` Limiter la requête aux subordonnés du volume.

images

Utiliser `images` drapeaux permettant d'imprimer un tableau des images de conteneurs dont Trident a besoin.

```
tridentctl images [flags]
```

drapeaux

`-h, --help` Aide pour les images.
`-v, --k8s-version string` : Version sémantique du cluster Kubernetes.

volume d'importation

Utilisez le `import volume` commande pour importer un volume existant dans Trident.

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

Alias

`volume, v`

drapeaux

`-f, --filename string` : Chemin d'accès au fichier PVC YAML ou JSON.
`-h, --help` : Aide pour le volume.
`--no-manage` : Créer uniquement des PV/PVC. Ne présumez pas de la gestion du cycle de vie des volumes.

installer

Utilisez le `install` options pour installer Trident.

```
tridentctl install [flags]
```

drapeaux

`--autosupport-image string`: L'image conteneur pour la télémétrie Autosupport (par défaut « `netapp/trident autosupport:<version-actuelle>` »).

`--autosupport-proxy string`: Adresse/port d'un proxy pour l'envoi des données de télémétrie Autosupport.

`--enable-node-prep` Tentative d'installation des paquets requis sur les nœuds.

`--generate-custom-yaml` Générez des fichiers YAML sans rien installer.

`-h`, `--help`: Aide à l'installation.

`--http-request-timeout`: Remplacer le délai d'expiration de la requête HTTP pour l'API REST du contrôleur Trident (par défaut 1 min 30 s).

`--image-registry string`: Adresse/port d'un registre d'images interne.

`--k8s-timeout duration`: Le délai d'expiration pour toutes les opérations Kubernetes (par défaut 3m0s).

`--kubelet-dir string`: Emplacement hôte de l'état interne de kubelet (par défaut « `/var/lib/kubelet` »).

`--log-format string`: Le format de journalisation Trident (texte, json) (par défaut "texte").

`--node-prep`: Permet à Trident de préparer les nœuds du cluster Kubernetes à gérer les volumes à l'aide du protocole de stockage de données spécifié. **Actuellement, `iscsi` est la seule valeur prise en charge. À partir d'OpenShift 4.19, la version minimale de Trident prise en charge pour cette fonctionnalité est la 25.06.1.**

`--pv string`: Le nom du PV hérité utilisé par Trident, garantit que celui-ci n'existe pas (par défaut « `trident` »).

`--pvc string`: Le nom du PVC hérité utilisé par Trident, garantit que celui-ci n'existe pas (par défaut « `trident` »).

`--silence-autosupport`: Ne pas envoyer automatiquement les bundles d'autosupport à NetApp (valeur par défaut: vrai).

`--silent` Désactiver la plupart des sorties pendant l'installation.

`--trident-image string`: L'image Trident à installer.

`--k8s-api-qps`: La limite de requêtes par seconde (QPS) pour les requêtes API Kubernetes (par défaut 100 ; optionnel).

`--use-custom-yaml` Utilisez les fichiers YAML existants dans le répertoire d'installation.

`--use-ipv6` Utilisez IPv6 pour la communication de Trident.

journaux

Utiliser `logs` options pour imprimer les journaux de Trident.

```
tridentctl logs [flags]
```

drapeaux

`-a`, `--archive`: Créer une archive de support contenant tous les journaux, sauf indication contraire.

`-h`, `--help` Aide pour les journaux.

`-l`, `--log string`: Affichage du journal Trident . L'un des `trident|auto|trident-operator|all` (par défaut "auto").

`--node string`: Nom du nœud Kubernetes à partir duquel collecter les journaux des pods.

`-p`, `--previous` Récupérez les journaux de l'instance de conteneur précédente, si elle existe.

`--sidecars` Récupérez les journaux des conteneurs sidecar.

envoyer

Utilisez le `send` commande pour envoyer une ressource depuis Trident.

```
tridentctl send [option]
```

Options

`autosupport`: Envoyer une archive Autosupport à NetApp.

désinstaller

Utiliser `uninstall` options pour désinstaller Trident.

```
tridentctl uninstall [flags]
```

drapeaux

`-h, --help` Aide à la désinstallation.
`--silent` Désactiver la plupart des sorties lors de la désinstallation.

mise à jour

Utilisez le `update` commande pour modifier une ressource dans Trident.

```
tridentctl update [option]
```

Options

`backend`: Mettre à jour un backend dans Trident.

mettre à jour l'état du backend

Utilisez le `update backend state` commande permettant de suspendre ou de reprendre les opérations en arrière-plan.

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

Points à considérer

- Si un backend est créé à l'aide d'un `TridentBackendConfig` (tbc), il ne peut pas être mis à jour à l'aide d'un `backend.json` déposer.
- Si le `userState` a été défini dans une date limite, il ne peut pas être modifié à l'aide de `tridentctl update backend state <backend-name> --user-state suspended/normal` commande.
- Pour retrouver la capacité de régler `userState` via `tridentctl` après avoir été configuré via `tbc`, le `userState` Ce champ doit être supprimé du fichier `tbc`. Cela peut être fait en utilisant le `kubectl edit tbc` commande. Après le `userState` Le champ a été supprimé, vous pouvez utiliser le `tridentctl update backend state` commande pour modifier la `userState` d'un backend.
- Utilisez le `tridentctl update backend state` pour changer le `userState`. Vous pouvez également mettre à jour le `userState` en utilisant `TridentBackendConfig` ou `backend.json` fichier ; cela déclenche une réinitialisation complète du système dorsal et peut prendre du temps.

drapeaux

`-h, --help` : Aide pour l'état du backend.
`--user-state` : Définir sur `suspended` interrompre les opérations en arrière-plan. Réglé sur `normal` pour reprendre les opérations en arrière-plan. Lorsqu'il est réglé sur `suspended` :

- `AddVolume` et `Import Volume` sont en pause.
- `CloneVolume`, `ResizeVolume`, `PublishVolume`, `UnPublishVolume`, `CreateSnapshot`,

`GetSnapshot` , `RestoreSnapshot` , `DeleteSnapshot` , `RemoveVolume` , `GetVolumeExternal` , `ReconcileNodeAccess` restent disponibles.

Vous pouvez également mettre à jour l'état du backend en utilisant `userState` champ dans le fichier de configuration du backend `TridentBackendConfig` ou `backend.json` . Pour plus d'informations, veuillez consulter ["Options pour la gestion des backends"](#) et ["Effectuez la gestion du backend avec kubectl"](#) .

Exemple:

JSON

Suivez ces étapes pour mettre à jour le `userState` en utilisant le `backend.json` déposer:

1. Modifier le `backend.json` fichier à inclure le `userState` champ dont la valeur est définie sur « `suspendu` ».
2. Mettez à jour le backend en utilisant le `tridentctl update backend` la commande et le chemin d'accès à la version mise à jour `backend.json` déposer.

Exemple: `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

Vous pouvez modifier le tbc après son application en utilisant `kubectl edit <tbc-name> -n <namespace>` commande. L'exemple suivant met à jour l'état du backend pour le suspendre en utilisant le `userState: suspended` option:

```
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
```

version

Utiliser `version` drapeaux pour imprimer la version de `tridentctl` et le service Trident en fonctionnement.

```
tridentctl version [flags]
```

drapeaux

`--client` Version client uniquement (aucun serveur requis).
`-h, --help` Aide pour la version.

Prise en charge des plugins

Tridentctl prend en charge des plugins similaires à `kubectl`. Tridentctl détecte un plugin si le nom du fichier binaire du plugin suit le schéma « `tridentctl-<plugin>` », et si le binaire est situé dans un dossier répertorié dans la variable d'environnement `PATH`. Tous les plugins détectés sont répertoriés dans la section plugins de l'aide de `tridentctl`. Vous pouvez également limiter la recherche en spécifiant un dossier de plugins dans la variable d'environnement `TRIDENTCTL_PLUGIN_PATH` (Exemple : `TRIDENTCTL_PLUGIN_PATH=~/.tridentctl-plugins/`). Si la variable est utilisée, `tridentctl` effectue la recherche uniquement dans le dossier spécifié.

Monitor Trident

Trident fournit un ensemble de points de terminaison de métriques Prometheus que vous pouvez utiliser pour surveiller les performances de Trident.

Aperçu

Les indicateurs fournis par Trident vous permettent de faire ce qui suit :

- Surveillez l'état et la configuration de Trident. Vous pouvez examiner le bon déroulement des opérations et vérifier si la communication avec les systèmes dorsaux se fait comme prévu.
- Examinez les informations d'utilisation du backend et comprenez combien de volumes sont provisionnés sur un backend et la quantité d'espace consommée, etc.
- Tenir à jour une cartographie des volumes provisionnés sur les serveurs backend disponibles.
- Suivre les performances. Vous pouvez observer combien de temps il faut à Trident pour communiquer avec les serveurs d'arrière-plan et effectuer les opérations.



Par défaut, les métriques de Trident sont exposées sur le port cible 8001 à `/metrics` point final. Ces indicateurs sont **activés par défaut** lorsque Trident est installé.

Ce dont vous aurez besoin

- Un cluster Kubernetes avec Trident installé.
- Une instance de Prometheus. Cela peut être un ["Déploiement Prometheus conteneurisé"](#) ou vous pouvez choisir d'exécuter Prometheus en tant que ["application native"](#).

Étape 1 : Définir une cible Prometheus

Vous devez définir une cible Prometheus pour collecter les métriques et obtenir des informations sur les backends gérés par Trident, les volumes qu'il crée, etc. Ce ["blog"](#) explique comment utiliser Prometheus et Grafana avec Trident pour récupérer des métriques. Ce blog explique comment exécuter Prometheus en tant

qu'opérateur dans votre cluster Kubernetes et comment créer un ServiceMonitor pour obtenir les métriques Trident .

Étape 2 : Créer un moniteur de service Prometheus

Pour exploiter les métriques Trident , vous devez créer un ServiceMonitor Prometheus qui surveille le `trident-csi` service et écoute sur le `metrics` port. Voici à quoi ressemble un exemple de ServiceMonitor :

```
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
```

Cette définition ServiceMonitor récupère les métriques renvoyées par le `trident-csi` le service et recherche spécifiquement le `metrics` point de terminaison du service. Par conséquent, Prometheus est désormais configuré pour comprendre les indicateurs de Trident.

En plus des métriques disponibles directement depuis Trident, kubelet expose de nombreuses autres. `kubelet_volume_*` métriques via son propre point de terminaison de métriques. Kubelet peut fournir des informations sur les volumes qui y sont attachés, ainsi que sur les pods et autres opérations internes qu'il gère. Se référer à ["ici"](#) .

Étape 3 : Interroger les métriques Trident avec PromQL

PromQL est idéal pour créer des expressions qui renvoient des données de séries temporelles ou des données tabulaires.

Voici quelques requêtes PromQL que vous pouvez utiliser :

Obtenez des informations sur la santé de Trident

- Pourcentage de réponses HTTP 2XX provenant de Trident

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

- **Pourcentage de réponses REST de Trident par code d'état**

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

- **Durée moyenne en ms des opérations effectuées par Trident**

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

Obtenez des informations sur l'utilisation de Trident

- **Volume moyen**

```
trident_volume_allocated_bytes/trident_volume_count
```

- **Espace total alloué par chaque serveur backend**

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

Obtenez l'utilisation individuelle du volume



Cette option n'est activée que si les métriques kubelet sont également collectées.

- **Pourcentage d'espace utilisé pour chaque volume**

```
kubelet_volume_stats_used_bytes / kubelet_volume_stats_capacity_bytes *  
100
```

Découvrez la télémétrie de Trident AutoSupport

Par défaut, Trident envoie quotidiennement à NetApp les métriques Prometheus et les informations de base du système dorsal.

- Pour empêcher Trident d'envoyer les métriques Prometheus et les informations de base du système dorsal à NetApp, transmettez le `--silence-autosupport` drapeau pendant l'installation de Trident .

- Trident peut également envoyer les journaux de conteneurs au support NetApp à la demande via `tridentctl send autosupport`. Vous devrez déclencher l'envoi des journaux de Trident. Avant de soumettre les journaux, vous devez accepter les conditions de NetApp <https://www.netapp.com/company/legal/privacy-policy/>["politique de confidentialité"].
- Sauf indication contraire, Trident récupère les journaux des dernières 24 heures.
- Vous pouvez spécifier la durée de conservation des journaux avec le `--since` drapeau. Par exemple: `tridentctl send autosupport --since=1h`. Ces informations sont collectées et envoyées via un `trident-autosupport` conteneur installé en parallèle de Trident. Vous pouvez obtenir l'image du conteneur à l'adresse suivante : "[Trident AutoSupport](#)".
- Trident AutoSupport ne recueille ni ne transmet d'informations personnelles identifiables (IPI) ou d'informations personnelles. Il est livré avec un "CLUF" Cela ne s'applique pas à l'image conteneur Trident elle-même. Vous pouvez en apprendre davantage sur l'engagement de NetApp en matière de sécurité et de confiance des données. "[ici](#)".

Voici un exemple de charge utile envoyée par Trident :

```
---
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
```

- Les messages AutoSupport sont envoyés au point de terminaison AutoSupport de NetApp. Si vous utilisez un registre privé pour stocker des images de conteneurs, vous pouvez utiliser `--image-registry` drapeau.
- Vous pouvez également configurer les URL proxy en générant les fichiers YAML d'installation. Cela peut être réalisé en utilisant `tridentctl install --generate-custom-yaml` pour créer les fichiers YAML et ajouter les `--proxy-url` argument en faveur du `trident-autosupport` conteneur dans `trident-deployment.yaml`.

Désactiver les métriques Trident

Pour **désactiver** le signalement des métriques, vous devez générer des fichiers YAML personnalisés (en utilisant le `--generate-custom-yaml` (drapeau) et modifiez-les pour supprimer le `--metrics` empêcher l'invocation de ce drapeau pour le `trident-main` récipient.

Désinstaller Trident

Vous devez utiliser la même méthode pour désinstaller Trident que celle utilisée pour l'installer.

À propos de cette tâche

- Si vous rencontrez des problèmes de bogues après une mise à niveau, des problèmes de dépendances ou une mise à niveau incomplète ou ayant échoué, vous devez désinstaller Trident et réinstaller la version précédente en suivant les instructions spécifiques. ["version"](#) . Il s'agit de la seule méthode recommandée pour revenir à une version antérieure.
- Pour faciliter la mise à niveau et la réinstallation, la désinstallation de Trident ne supprime pas les CRD ni les objets associés créés par Trident. Si vous devez supprimer complètement Trident et toutes ses données, veuillez consulter la documentation. ["Supprimer complètement Trident et CRD"](#) .

Avant de commencer

Si vous mettez hors service des clusters Kubernetes, vous devez supprimer toutes les applications qui utilisent des volumes créés par Trident avant la désinstallation. Cela garantit que les PVC sont dépubliées sur les nœuds Kubernetes avant d'être supprimées.

Déterminer la méthode d'installation d'origine

Vous devez utiliser la même méthode pour désinstaller Trident que celle utilisée pour l'installer. Avant de désinstaller, vérifiez quelle version vous avez utilisée pour installer Trident à l'origine.

1. Utiliser `kubectl get pods -n trident` pour examiner les gousses.
 - En l'absence de nacelle opérateur, Trident a été installé via `tridentctl` .
 - S'il existe un module opérateur, Trident a été installé à l'aide de l'opérateur Trident , soit manuellement, soit via Helm.
2. S'il y a une cabine d'opérateur, utilisez-la `kubectl describe tproc trident` pour déterminer si Trident a été installé à l'aide de Helm.
 - Si une étiquette Helm est présente, Trident a été installé à l'aide de Helm.
 - S'il n'y a pas d'étiquette Helm, Trident a été installé manuellement à l'aide de l'opérateur Trident .

Désinstaller une installation d'opérateur Trident

Vous pouvez désinstaller une installation de l'opérateur Trident manuellement ou à l'aide de Helm.

Désinstaller l'installation manuelle

Si vous avez installé Trident à l'aide de l'opérateur, vous pouvez le désinstaller en procédant de l'une des manières suivantes :

1. **Modifier `TridentOrchestrator` CR et définissez l'indicateur de désinstallation :**

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

Quand le `uninstall` L'indicateur est réglé sur `true` L'opérateur Trident désinstalle Trident, mais ne supprime pas `TridentOrchestrator` lui-même. Vous devriez nettoyer le `TridentOrchestrator` et en créer un nouveau si vous souhaitez réinstaller Trident .

2. **Supprimer `TridentOrchestrator`** : En supprimant le `TridentOrchestrator` CR qui a été utilisé pour déployer Trident, vous demandez à l'opérateur de désinstaller Trident. L'opérateur traite le retrait de `TridentOrchestrator` et procède à la suppression du déploiement Trident et du `daemonset`, en supprimant les pods Trident qu'il avait créés dans le cadre de l'installation.

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

Désinstaller Helm

Si vous avez installé Trident à l'aide de Helm, vous pouvez le désinstaller en utilisant `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
```

Désinstaller un `tridentctl` installation

Utilisez le `uninstall` commande dans `tridentctl` supprimer toutes les ressources associées à Trident, à l'exception des CRD et des objets associés :

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

Informations sur le copyright

Copyright © 2026 NetApp, Inc. Tous droits réservés. Imprimé aux États-Unis. Aucune partie de ce document protégé par copyright ne peut être reproduite sous quelque forme que ce soit ou selon quelque méthode que ce soit (graphique, électronique ou mécanique, notamment par photocopie, enregistrement ou stockage dans un système de récupération électronique) sans l'autorisation écrite préalable du détenteur du droit de copyright.

Les logiciels dérivés des éléments NetApp protégés par copyright sont soumis à la licence et à l'avis de non-responsabilité suivants :

CE LOGICIEL EST FOURNI PAR NETAPP « EN L'ÉTAT » ET SANS GARANTIES EXPRESSES OU TACITES, Y COMPRIS LES GARANTIES TACITES DE QUALITÉ MARCHANDE ET D'ADÉQUATION À UN USAGE PARTICULIER, QUI SONT EXCLUES PAR LES PRÉSENTES. EN AUCUN CAS NETAPP NE SERA TENU POUR RESPONSABLE DE DOMMAGES DIRECTS, INDIRECTS, ACCESSOIRES, PARTICULIERS OU EXEMPLAIRES (Y COMPRIS L'ACHAT DE BIENS ET DE SERVICES DE SUBSTITUTION, LA PERTE DE JOUISSANCE, DE DONNÉES OU DE PROFITS, OU L'INTERRUPTION D'ACTIVITÉ), QUELLES QU'EN SOIENT LA CAUSE ET LA DOCTRINE DE RESPONSABILITÉ, QU'IL S'AGISSE DE RESPONSABILITÉ CONTRACTUELLE, STRICTE OU DÉLICTELLE (Y COMPRIS LA NÉGLIGENCE OU AUTRE) DÉCOULANT DE L'UTILISATION DE CE LOGICIEL, MÊME SI LA SOCIÉTÉ A ÉTÉ INFORMÉE DE LA POSSIBILITÉ DE TELS DOMMAGES.

NetApp se réserve le droit de modifier les produits décrits dans le présent document à tout moment et sans préavis. NetApp décline toute responsabilité découlant de l'utilisation des produits décrits dans le présent document, sauf accord explicite écrit de NetApp. L'utilisation ou l'achat de ce produit ne concède pas de licence dans le cadre de droits de brevet, de droits de marque commerciale ou de tout autre droit de propriété intellectuelle de NetApp.

Le produit décrit dans ce manuel peut être protégé par un ou plusieurs brevets américains, étrangers ou par une demande en attente.

LÉGENDE DE RESTRICTION DES DROITS : L'utilisation, la duplication ou la divulgation par le gouvernement sont sujettes aux restrictions énoncées dans le sous-paragraphe (b)(3) de la clause Rights in Technical Data-Noncommercial Items du DFARS 252.227-7013 (février 2014) et du FAR 52.227-19 (décembre 2007).

Les données contenues dans les présentes se rapportent à un produit et/ou service commercial (tel que défini par la clause FAR 2.101). Il s'agit de données propriétaires de NetApp, Inc. Toutes les données techniques et tous les logiciels fournis par NetApp en vertu du présent Accord sont à caractère commercial et ont été exclusivement développés à l'aide de fonds privés. Le gouvernement des États-Unis dispose d'une licence limitée irrévocable, non exclusive, non cessible, non transférable et mondiale. Cette licence lui permet d'utiliser uniquement les données relatives au contrat du gouvernement des États-Unis d'après lequel les données lui ont été fournies ou celles qui sont nécessaires à son exécution. Sauf dispositions contraires énoncées dans les présentes, l'utilisation, la divulgation, la reproduction, la modification, l'exécution, l'affichage des données sont interdits sans avoir obtenu le consentement écrit préalable de NetApp, Inc. Les droits de licences du Département de la Défense du gouvernement des États-Unis se limitent aux droits identifiés par la clause 252.227-7015(b) du DFARS (février 2014).

Informations sur les marques commerciales

NETAPP, le logo NETAPP et les marques citées sur le site <http://www.netapp.com/TM> sont des marques déposées ou des marques commerciales de NetApp, Inc. Les autres noms de marques et de produits sont des marques commerciales de leurs propriétaires respectifs.