Gestion et contrôle d'Astra Trident

Astra Trident

NetApp
January 14, 2026

This PDF was generated from https://docs.netapp.com/fr-fr/trident-2310/trident-managing-k8s/upgrade-trident.html on January 14, 2026. Always check docs.netapp.com for the latest.

Sommaire

Gestion et contrôle d'Astra Trident . . . 1
- Mettez à niveau Astra Trident . . . 1
  - Mettez à niveau Astra Trident . . . 1
  - Mise à niveau avec l'opérateur . . . 2
  - Mise à niveau avec tridentctl . . . 7
- Gérez Astra Trident à l'aide de tridentctl . . . 7
  - Commandes et indicateurs globaux . . . 7
  - Options et indicateurs de commande . . . 9
- Contrôle d'Astra Trident . . . 12
  - Présentation . . . 12
  - Étape 1 : définir une cible Prometheus . . . 13
  - Étape 2 : créer un ServiceMonitor Prometheus . . . 13
  - Étape 3 : interroger les mesures Trident avec PromQL . . . 14
  - Découvrez la télémétrie AutoSupport d'Astra Trident . . . 15
  - Désactivation des metrics d'Astra Trident . . . 16
- Désinstaller Astra Trident . . . 16
  - Déterminez la méthode d'installation d'origine . . . 17
  - Désinstallez l'installation d'un opérateur Trident . . . 17
  - Désinstallez un `tridentctl` installation . . . 18

Gestion et contrôle d'Astra Trident

Mettez à niveau Astra Trident

Mettez à niveau Astra Trident

Astra Trident suit le rythme de sa mise à jour trimestrielle, fournissant quatre versions majeures chaque année. Chaque nouvelle version exploite les versions précédentes et fournit de nouvelles fonctionnalités, des améliorations de performances, des correctifs et des améliorations. Nous vous encourageons à effectuer une mise à niveau au moins une fois par an pour profiter des nouvelles fonctionnalités d'Astra Trident.

Considérations avant la mise à niveau

Lorsque vous mettez à niveau vers la dernière version d'Astra Trident, prenez en compte les points suivants :

- Il ne doit y avoir qu'une seule instance Astra Trident installée sur tous les namespaces d'un cluster Kubernetes donné.
- ASTRA Trident 23.07 et les versions ultérieures requièrent des snapshots de volume v1 et ne prennent plus en charge les snapshots alpha ou bêta.
- Si vous avez créé Cloud Volumes Service pour Google Cloud dans le "Type de service CVS", vous devez mettre à jour la configuration back-end pour utiliser le `standardsw` ou `zoneredundantstandardsw` Niveau de service lors de la mise à niveau à partir d'Astra Trident 23.01. Impossible de mettre à jour le `serviceLevel` dans le back-end, les volumes pourraient tomber en panne. Reportez-vous à la section "Exemples de type de service CVS" pour plus d'informations.
- Lors de la mise à niveau, il est important de fournir `parameter.fsType` dans `StorageClasses` Utilisé par Astra Trident. Vous pouvez supprimer et recréer `StorageClasses` sans interrompre les volumes existants.
  - Il s'agit d'une exigence ** pour l'application "contextes de sécurité" Pour les volumes SAN.
  - Le répertoire d'entrée sample contient des exemples, tels que `storage-class-basic.yaml.templ` et `storage-class-bronze-default.yaml`.
  - Pour plus d'informations, voir "Problèmes connus".

Étape 1 : sélectionnez une version

Les versions d'Astra Trident suivent une date `YY.MM` convention de dénomination, où "YY" est les deux derniers chiffres de l'année et "MM" est le mois. Les versions point suivent un `YY.MM.X` convention, où « X » est le niveau de patch. Vous allez sélectionner la version à mettre à niveau en fonction de la version à partir de laquelle vous effectuez la mise à niveau.

- Vous pouvez effectuer une mise à niveau directe vers n'importe quelle version cible située dans une fenêtre à quatre versions de la version installée. Par exemple, vous pouvez effectuer une mise à niveau directe de la version 22.10 (ou de toute version 22.10 points) vers la version 23.10.
- Si vous effectuez une mise à niveau à partir d'une version en dehors de la fenêtre à quatre versions, effectuez une mise à niveau en plusieurs étapes. Suivez les instructions de mise à niveau du "version antérieure" vous effectuez une mise à niveau de vers la version la plus récente qui correspond à la fenêtre à quatre versions. Par exemple, si vous exécutez 21.10 et que vous souhaitez effectuer une mise à niveau vers 23.10 :

a. Première mise à niveau de 21.10 à 22.10.

b. Puis passez de 22.10 à 23.10.

> Lorsque vous effectuez une mise à niveau avec 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 sorti avec 21.01.0 contient un problème connu qui a été résolu en 21.01.1. Pour plus de détails, consultez le "Consultez le document GitHub pour plus d'informations".

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

Pour déterminer quelle version vous avez utilisée pour l'installation initiale d'Astra Trident :

1. Utiliser `kubectl get pods -n trident` pour examiner les pods.
   - S'il n'y a pas de pod opérateur, Astra Trident a été installé avec `tridentctl`.
   - S'il existe un pod opérateur, Astra Trident a été installé à l'aide de l'opérateur Trident soit manuellement, soit à l'aide d'Helm.
2. S'il y a un boîtier opérateur, utiliser `kubectl describe tproc trident` Pour déterminer si Astra Trident a été installé à l'aide d'Helm.
   - S'il y a une étiquette Helm, Astra Trident a été installée à l'aide d'Helm.
   - S'il n'y a pas d'étiquette Helm, Astra Trident a été installé manuellement à l'aide de l'opérateur Trident.

Étape 3 : sélectionnez une méthode de mise à niveau

En général, vous devez effectuer une mise à niveau en utilisant la même méthode que celle utilisée pour l'installation initiale, mais vous pouvez le faire "passer d'une méthode d'installation à l'autre". Il existe deux options de mise à niveau d'Astra Trident.

- "Mise à niveau à l'aide de l'opérateur Trident"

> Nous vous suggérons de revoir "Comprendre le workflow de mise à niveau de l'opérateur" avant la mise à niveau avec l'opérateur.

*

Mise à niveau avec l'opérateur

Comprendre le workflow de mise à niveau de l'opérateur

Avant d'utiliser l'opérateur Trident pour mettre à niveau Astra Trident, vous devez comprendre les processus en arrière-plan qui se produisent pendant la mise à niveau. Cela inclut les modifications apportées au contrôleur Trident, au pod du contrôleur et aux pods des nœuds, ainsi qu'au jeu de démonstration des nœuds qui activent les mises à jour en continu.

Gestion des mises à niveau par l'opérateur Trident

L'un des nombreux "Avantages de l'utilisation de l'opérateur Trident" Pour installer et mettre à niveau Astra Trident, il s'agit du traitement automatique des objets Astra Trident et Kubernetes sans interrompre les

volumes montés déjà en place. De cette façon, Astra Trident peut prendre en charge les mises à niveau sans interruption, ou "*mises à jour en continu*". En particulier, l'opérateur Trident communique avec le cluster Kubernetes pour :

- Supprimez et recréez le déploiement du contrôleur Trident et le nœud DemonSet.
- Remplacez l'afficheur de contrôleur Trident et les pods de nœud Trident par de nouvelles versions.
  - Si un nœud n'est pas mis à jour, il n'empêche pas la mise à jour des nœuds restants.
  - Seuls les nœuds exécutant Trident Node Pod peuvent monter des volumes.

Pour en savoir plus sur l'architecture d'Astra Trident sur le cluster Kubernetes, reportez-vous à la section "Architecture d'Astra Trident".

Workflow de mise à niveau de l'opérateur

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

1. L'opérateur **Trident** :
   a. Détecte la version d'Astra Trident actuellement installée (version *n*).
   b. Mise à jour de tous les objets Kubernetes, y compris les CRD, RBAC et le service Trident.
   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 le pod du contrôleur Trident pour *n+1*.
3. L'opérateur **Trident** :
   a. Supprime le jeu de démonstration du nœud Trident pour *n*. L'opérateur n'attend pas la fin de Node Pod.
   b. Crée le dédéclenchement du nœud Trident pour *n+1*.
4. **Kubernetes** crée des pods de nœuds Trident sur les nœuds qui n'exécutent pas Trident Node Pod *n*. Cela permet de garantir qu'il n'y a jamais plus d'un pod de nœuds Trident, quelle que soit la version, sur un nœud.

Mettre à niveau une installation de l'opérateur Trident

Vous pouvez mettre à niveau Astra Trident à l'aide de l'opérateur Trident manuellement ou à l'aide d'Helm. Vous pouvez effectuer la mise à niveau d'une installation d'opérateur Trident vers une autre installation d'opérateur Trident ou à partir d'un `tridentctl` Installation sur une version opérateur Trident. Révision "Sélectionnez une méthode de mise à niveau" Avant de mettre à niveau l'installation d'un opérateur Trident.

Mettre à niveau une installation manuelle

Vous pouvez effectuer une mise à niveau d'une installation d'opérateur Trident dont le périmètre est défini dans le cluster vers une autre installation d'opérateur Trident dont le périmètre est défini dans le cluster. Toutes les versions 21.01 et supérieures d'Astra Trident utilisent un opérateur à périmètre de cluster.

Pour effectuer la mise à niveau à partir d'Astra Trident qui a été installée à l'aide de l'opérateur dont le périmètre d'espace de noms est compris entre les versions 20.07 et 20.10, suivez les instructions de mise à niveau de "votre version installée" D'Astra Trident.

Description de la tâche

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

- Pour les clusters exécutant Kubernetes 1.24 ou version antérieure, utilisez "bundle_pre_1_25.yaml".
- Pour les clusters exécutant Kubernetes 1.25 ou version ultérieure, utilisez "bundle_post_1_25.yaml".

Avant de commencer

Assurez-vous d'utiliser un cluster Kubernetes en cours d'exécution "Version Kubernetes prise en charge".

Étapes

1. Vérifiez votre version d'Astra Trident :

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

2. Supprimez l'opérateur Trident qui a été utilisé pour installer l'instance Astra Trident actuelle. Par exemple, si vous mettez à niveau depuis 23.07, exécutez la commande suivante :

```
kubectl delete -f 23.07.0/trident-installer/deploy/<bundle.yaml> -n
trident
```

3. Si vous avez personnalisé votre installation initiale à l'aide de `TridentOrchestrator` attributs, vous pouvez modifier le `TridentOrchestrator` objet pour modifier les paramètres d'installation. Cela peut inclure des modifications visant à spécifier les registres d'images en miroir Trident et CSI pour le mode hors ligne, à activer les journaux de débogage ou à spécifier les secrets d'extraction d'images.
4. Installez Astra Trident à l'aide du fichier YAML de bundle approprié pour votre environnement, où *<bundle.yaml>* est `bundle_pre_1_25.yaml` ou `bundle_post_1_25.yaml` Basé sur votre version de Kubernetes. Par exemple, si vous installez Astra Trident 23.10, exécutez la commande suivante :

```
kubectl create -f 23.10.0/trident-installer/deploy/<bundle.yaml> -n
trident
```

Mettre à niveau une installation Helm

Vous pouvez mettre à niveau une installation d'Astra Trident Helm.

> Lorsque vous mettez à niveau un cluster Kubernetes de 1.24 vers 1.25 ou version ultérieure sur lequel Astra Trident est installé, vous devez mettre à jour les valeurs.yaml pour les définir `excludePodSecurityPolicy` à `true` ou ajouter `--set excludePodSecurityPolicy=true` à la `helm upgrade` commande avant de pouvoir mettre à niveau le cluster.

Étapes

1. Si vous "Installez Astra Trident à l'aide d'Helm - effectué", vous pouvez utiliser `helm upgrade trident netapp-trident/trident-operator --version 23.10.0` pour mettre à niveau en une seule

étape. Si vous n'avez pas ajouté le Helm repo ou si vous ne pouvez pas l'utiliser pour mettre à niveau :

a. Téléchargez la dernière version d'Astra Trident sur le site "La section *Assets* sur GitHub".

b. Utilisez le `helm upgrade` commande où `trident-operator-23.10.0.tgz` reflète la version vers laquelle vous souhaitez effectuer la mise à niveau.

```
helm upgrade <name> trident-operator-23.10.0.tgz
```

> Si vous définissez des options personnalisées lors de l'installation initiale (par exemple, si vous spécifiez des registres privés en miroir pour les images Trident et CSI), ajoutez le `helm upgrade` commande à l'aide de `--set` pour vous assurer que ces options sont incluses dans la commande de mise à niveau, sinon les valeurs sont réinitialisées sur les valeurs par défaut.

2. Courez `helm list` pour vérifier que le graphique et la version de l'application ont tous deux été mis à niveau. Courez `tridentctl logs` pour consulter les messages de débogage.

Mise à niveau à partir d'un `tridentctl` Installation sur l'opérateur Trident

Vous pouvez effectuer la mise à niveau vers la dernière version de l'opérateur Trident à partir d'un `tridentctl` installation. Les systèmes back-end et ESV existants seront automatiquement disponibles.

> Avant de passer d'une méthode d'installation à l'autre, consultez "Passage d'une méthode d'installation à l'autre"

Étapes

1. Téléchargez la dernière version d'Astra Trident.

```
# Download the release required [23.10.0]
mkdir 23.10.0
cd 23.10.0
wget
https://github.com/NetApp/trident/releases/download/v23.10.0/trident-
installer-23.10.0.tar.gz
tar -xf trident-installer-23.10.0.tar.gz
cd trident-installer
```

2. Créer le `tridentorchestrator` CRD du manifeste.

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

3. Déployer l'opérateur cluster-scoped dans le même 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. Créer un `TridentOrchestrator` CR pour l'installation d'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. Vérifiez que Trident a été mis à niveau vers la version prévue.

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

Message:                Trident installed
Namespace:              trident
Status:                 Installed
Version:                v23.10.0
```

Mise à niveau avec tridentctl

Vous pouvez facilement mettre à niveau une installation Astra Trident existante à l'aide de `tridentctl`.

Description de la tâche

La désinstallation et la réinstallation d'Astra Trident fait office de mise à niveau. Lorsque vous désinstallez Trident, la demande de volume persistant et le volume persistant utilisés par l'Astra Trident. Les volumes persistants ayant déjà été provisionnés restent disponibles pendant la mise hors ligne d'Astra Trident, et Astra Trident provisionne les volumes pour les demandes de volume persistant créées dans l'intervalle une fois de nouveau en ligne.

Avant de commencer

Révision "Sélectionnez une méthode de mise à niveau" avant la mise à niveau avec `tridentctl`.

Étapes

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

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

2. Réinstallez Astra Trident. Reportez-vous à la section "Installation d'Astra Trident à l'aide de tridentctl".

> N'interrompez pas le processus de mise à niveau. Assurez-vous que le programme d'installation s'exécute jusqu'à la fin.

Gérez Astra Trident à l'aide de tridentctl

Le "Pack d'installation Trident" inclut le `tridentctl` Utilitaire de ligne de commande pour un accès simple à Astra Trident. Les utilisateurs Kubernetes disposant de privilèges suffisants peuvent l'utiliser pour installer Astra Trident ou gérer le namespace qui contient le pod Astra Trident.

Commandes et indicateurs globaux

Vous pouvez exécuter `tridentctl help` pour obtenir une liste des commandes disponibles pour `tridentctl` ou ajoutez le `--help` marquer n'importe quelle commande pour obtenir une liste d'options et d'indicateurs pour cette commande spécifique.

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

ASTRA Trident `tridentctl` l'utilitaire prend en charge les commandes et indicateurs globaux suivants.

Commandes

`create`
Ajoutez une ressource à Astra Trident.

`delete`
Supprimez une ou plusieurs ressources d'Astra Trident.

`get`
Obtenez une ou plusieurs ressources d'Astra Trident.

`help`
Aide sur n'importe quelle commande.

`images`
Imprimez un tableau des images de conteneur dont Astra Trident a besoin.

`import`
Importez une ressource existante dans Astra Trident.

`install`
Installer Astra Trident.

`logs`
Imprimez les journaux depuis Astra Trident.

`send`
Envoyer une ressource depuis Astra Trident.

`uninstall`
Désinstallez Astra Trident.

`update`
Modifier une ressource dans Astra Trident.

`update backend state`
Suspendre temporairement les opérations back-end.

`upgrade`
Mise à niveau d'une ressource dans Astra Trident

`version`
Imprimez la version d'Astra Trident.

Alarmes globales

-d, --debug

Sortie de débogage.

-h, --help

Aide pour `tridentctl`.

-k, --kubeconfig string

Spécifiez le `KUBECONFIG` Chemin d'accès pour exécuter des commandes localement ou d'un cluster Kubernetes vers un autre.

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

-n, --namespace string

Espace de noms du déploiement d'Astra Trident.

-o, --output string

Format de sortie. Un de json|yaml|nom|large|ps (par défaut).

-s, --server string

Adresse/port de l'interface REST d'Astra Trident.

Vous pouvez configurer l'interface REST de Trident pour écouter et utiliser l'interface 127.0.0.1 (pour IPv4) ou [::1] (pour IPv6) uniquement.

Options et indicateurs de commande

création

Utilisez le `create` Commande d'ajout d'une ressource à Astra Trident.

```
tridentctl create [option]
```

Options

`backend`: Ajouter un backend à Astra Trident.

supprimer

Utilisez le `delete` Commande de supprimer une ou plusieurs ressources d'Astra Trident.

```
tridentctl delete [option]
```

Options

`backend`: Supprimer un ou plusieurs systèmes back-end de stockage d'Astra Trident.
`snapshot`: Supprimez un ou plusieurs instantanés de volume d'Astra Trident.

`storageclass`: Supprimez une ou plusieurs classes de stockage d'Astra Trident.
`volume`: Supprimer un ou plusieurs volumes de stockage d'Astra Trident.

obtenez

Utilisez le `get` Commander pour obtenir une ou plusieurs ressources d'Astra Trident.

```
tridentctl get [option]
```

Options

`backend`: Faites passer un ou plusieurs systèmes back-end de stockage à Astra Trident.
`snapshot`: Obtenez un ou plusieurs instantanés d'Astra Trident.
`storageclass`: Obtenez une ou plusieurs classes de stockage d'Astra Trident.
`volume`: Obtenez un ou plusieurs volumes d'Astra Trident.

Alarmes

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

images

Utiliser `images` Drapeaux pour imprimer un tableau des images de conteneur dont Astra Trident a besoin.

```
tridentctl images [flags]
```

Alarmes

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

importer le volume

Utilisez le `import volume` Commande permettant d'importer un volume existant vers Astra Trident.

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

Alias

`volume`, `v`

Alarmes

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

installer

Utilisez le `install` Drapeaux pour l'installation d'Astra Trident.

```
tridentctl install [flags]
```

Alarmes

`--autosupport-image string`: Image conteneur pour AutoSupport Telemetry (par défaut « netapp/trident AutoSupport:<current-version> »).
`--autosupport-proxy string`: Adresse/port d'un proxy pour l'envoi de télémétrie AutoSupport.
`--enable-node-prep`: Tentative d'installation des paquets requis sur les nœuds.
`--generate-custom-yaml`: Générer des fichiers YAML sans rien installer.
`-h, --help`: Aide pour l'installation.
`--http-request-timeout`: Override the HTTP request timeout for Trident Controller's REST API (Default 1m30s).
`--image-registry string`: Adresse/port d'un registre d'images interne.
`--k8s-timeout duration`: Délai d'expiration pour toutes les opérations Kubernetes (par défaut 3m0s).
`--kubelet-dir string`: L'emplacement hôte de l'état interne de kubelet (par défaut "/var/lib/kubelet").
`--log-format string`: Le format de consignation Astra Trident (texte, json) (par défaut "texte").
`--pv string`: Le nom de la PV héritée utilisée par Astra Trident, s'assure que cela n'existe pas (par défaut "trident").
`--pvc string`: Le nom du PVC hérité utilisé par Astra Trident, s'assure qu'il n'existe pas (par défaut "trident").
`--silence-autosupport`: N'envoyez pas automatiquement les packs AutoSupport à NetApp (valeur par défaut vraie).
`--silent`: Désactiver la plupart des sorties lors de l'installation.
`--trident-image string`: L'image Astra Trident à installer.
`--use-custom-yaml`: Utilisez tous les fichiers YAML existants qui existent dans le répertoire de configuration.
`--use-ipv6`: Utiliser IPv6 pour la communication d'Astra Trident.

journaux

Utiliser `logs` Drapeaux pour imprimer les journaux à partir d'Astra Trident.

```
tridentctl logs [flags]
```

Alarmes

`-a, --archive`: Créez une archive de support avec tous les journaux sauf indication contraire.
`-h, --help`: Aide pour les journaux.
`-l, --log string`: Astra Trident log à afficher. Un de trident|auto|trident-operator|All (auto par défaut).
`--node string`: Le nom du nœud Kubernetes à partir duquel recueillir les journaux de pod de nœud.
`-p, --previous`: Si elle existe, procurez-vous les journaux de l'instance de conteneur précédente.
`--sidecars`: Procurez-vous les bûches pour les conteneurs de sidecar.

envoyer

Utilisez le `send` Commande permettant d'envoyer une ressource à Astra Trident.

```
tridentctl send [option]
```

Options

`autosupport`: Envoyez une archive AutoSupport à NetApp.

désinstaller

Utiliser `uninstall` Drapeaux pour désinstaller Astra Trident.

```
tridentctl uninstall [flags]
```

Alarmes

`-h, --help`: Aide pour désinstaller.
`--silent`: Désactiver la plupart des résultats lors de la désinstallation.

mise à jour

Utilisez le `update` Commande permettant de modifier une ressource dans Astra Trident.

```
tridentctl update [option]
```

Options

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

mettre à jour l'état back-end

Utilisez le `update backend state` pour suspendre ou reprendre les opérations back-end.

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

Alarmes

`-h, --help`: Aide pour l'état back-end.
`--user-state`: Défini sur `suspended` pour interrompre les opérations back-end. Réglez sur `normal` pour reprendre les opérations back-end. Lorsqu'il est réglé sur `suspended`:

- `AddVolume`, `CloneVolume`, `Import Volume`, `ResizeVolume` sont en pause.
- `PublishVolume`, `UnPublishVolume`, `CreateSnapshot`, `GetSnapshot`, `RestoreSnapshot`, `DeleteSnapshot`, `RemoveVolume`, `GetVolumeExternal`, `ReconcileNodeAccess` et restent disponibles.

version

Utiliser `version` indicateurs pour imprimer la version de `tridentctl` Et le service exécutant Trident.

```
tridentctl version [flags]
```

Alarmes

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

Contrôle d'Astra Trident

ASTRA Trident fournit un ensemble de terminaux de metrics Prometheus que vous pouvez utiliser pour contrôler les performances d'Astra Trident.

Présentation

Avec les metrics d'Astra Trident, vous pouvez :

- Surveillez l'état et la configuration d'Astra Trident. Vous avez la possibilité d'examiner la réussite des opérations et de savoir si elles peuvent communiquer avec les systèmes back-end comme prévu.
- Examiner les informations d'utilisation du système back-end et comprendre le nombre de volumes provisionnés sur un système back-end, ainsi que la quantité d'espace consommé, etc.
- Conservez un mappage de la quantité de volumes provisionnés sur les systèmes back-end disponibles.
- Suivi des performances. Découvrez le temps nécessaire pour communiquer avec Astra Trident aux systèmes back-end et effectuer les opérations.

Par défaut, les metrics de Trident sont visibles sur le port cible `8001` au `/metrics` point final. Ces mesures sont **activées par défaut** lors de l'installation de Trident.

Ce dont vous avez besoin

- Cluster Kubernetes avec Astra Trident installé.
- Instance Prometheus. Il peut s'agir d'un "Déploiement conteneurisé par Prometheus" Vous pouvez également utiliser Prometheus en tant que "application native".

Étape 1 : définir une cible Prometheus

Vous devez définir une cible Prometheus pour collecter les metrics et obtenir des informations sur les systèmes back-end gérés par Trident, les volumes qu'elle crée, etc. C'est ça "Blog" Vous apprendrez comment utiliser Prometheus et Grafana avec Astra Trident pour récupérer des metrics. Découvrez sur ce blog comment exécuter Prometheus en tant qu'opérateur dans votre cluster Kubernetes et comment créer un ServiceMonitor pour obtenir des metrics Astra Trident.

Étape 2 : créer un ServiceMonitor Prometheus

Pour consommer les metrics Trident, vous devez créer un ServiceMonitor Prometheus qui surveille la `trident-csi` service et écoute sur le `metrics` port. Un exemple de ServiceMonitor se présente comme suit :

```
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 de ServiceMonitor récupère les mesures renvoyées par le `trident-csi` et recherche spécifiquement le `metrics` point d'extrémité du service. Par conséquent, Prometheus est désormais configuré pour comprendre Astra Trident
metrics.

Outre les metrics directement disponibles par Astra Trident, kubelet expose beaucoup `kubelet_volume_*` metrics via son propre terminal de metrics Kubelet peut fournir des informations sur les volumes reliés, ainsi que sur les pods et autres opérations internes qu'elle gère. Voir "ici".

Étape 3 : interroger les mesures Trident avec PromQL

PromQL est bon pour la création d'expressions qui renvoient des séries chronologiques ou des données tabulaires.

Voici quelques questions PromQL que vous pouvez utiliser :

Accédez aux informations sur l'état de santé de Trident

- **Pourcentage de réponses HTTP 2XX d'Astra 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 d'Astra Trident par le 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 Astra Trident**

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

Découvrez les informations d'utilisation d'Astra Trident

- **Taille moyenne du volume**

```
trident_volume_allocated_bytes/trident_volume_count
```

- **Espace volume total provisionné par chaque back-end**

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

Utiliser individuellement le volume

Cette activation est uniquement possible si les indicateurs kubelet sont également collectés.

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

```
kubelet_volume_stats_used_bytes / kubelet_volume_stats_capacity_bytes *
100
```

Découvrez la télémétrie AutoSupport d'Astra Trident

Par défaut, Astra Trident envoie des metrics de Prometheus et des informations de base back-end à NetApp dans un rythme quotidien.

- Pour empêcher Astra Trident d'envoyer des metrics de Prometheus et des informations de base back-end à NetApp, passez le `--silence-autosupport` Indicateur lors de l'installation d'Astra Trident.
- Astra Trident peut également envoyer des journaux de conteneur à NetApp support à la demande via `tridentctl send autosupport`. Vous devrez déclencher Astra Trident pour télécharger ses journaux. Avant de communiquer les journaux, vous devez accepter celui de NetApp "politique de confidentialité".
- Sauf mention contraire, Astra Trident extrait les journaux des 24 dernières heures.
- Vous pouvez spécifier la période de rétention du journal avec le `--since` drapeau. Par exemple :

`tridentctl send autosupport --since=1h`. Ces informations sont collectées et envoyées via un `trident-autosupport` conteneur
Installé avec Astra Trident. Vous pouvez obtenir l'image conteneur à "AutoSupport Trident".

- Le AutoSupport Trident ne collecte pas et ne transmet pas d'informations à caractère personnel (PII) ou de données personnelles. Il est fourni avec un "CLUF" qui ne s'applique pas à l'image du conteneur Trident. Pour en savoir plus sur l'engagement de NetApp en faveur de la sécurité et de la confiance des données "ici".

Voici un exemple de charge utile envoyée par Astra Trident :

```
---
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
```

- Les messages AutoSupport sont envoyés au terminal AutoSupport de NetApp. Si vous utilisez un registre privé pour stocker des images de conteneur, vous pouvez utiliser le `--image-registry` drapeau.
- Vous pouvez également configurer des URL proxy en générant les fichiers YAML d'installation. Pour ce faire, utilisez `tridentctl install --generate-custom-yaml` Pour créer les fichiers YAML et ajouter le `--proxy-url` argument pour le `trident-autosupport` conteneur `trident-deployment.yaml`.

Désactivation des metrics d'Astra Trident

Pour désactiver** les mesures signalées, vous devez générer des YAML personnalisées (à l'aide de l' `--generate-custom-yaml` marquer) et modifiez-les pour supprimer le `--metrics` indicateur d'être appelé pour le `trident-main`
conteneur.

Désinstaller Astra Trident

Vous devez utiliser la même méthode pour désinstaller Astra Trident que pour installer Astra Trident.

Description de la tâche

- Si vous avez besoin d'un correctif pour les bugs observés après une mise à niveau, des problèmes de

dépendance ou une mise à niveau non réussie ou incomplète, désinstallez Astra Trident et réinstallez la version précédente en suivant les instructions spécifiques "version". Il s'agit de la seule méthode recommandée pour *rétrograder* vers une version antérieure.

- Pour faciliter la mise à niveau et la réinstallation, la désinstallation d'Astra Trident ne supprime pas les CRD ou les objets associés créés par Astra Trident. Si vous devez supprimer complètement Astra Trident et toutes ses données, reportez-vous à la section "Retirez complètement Astra Trident et les CRD".

Avant de commencer

Si vous désaffectez des clusters Kubernetes, vous devez supprimer toutes les applications qui utilisent des volumes créés par Astra Trident avant de procéder à la désinstallation. Cela permet de s'assurer que les ESV ne sont pas publiées sur les nœuds Kubernetes avant d'être supprimées.

Déterminez la méthode d'installation d'origine

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

1. Utiliser `kubectl get pods -n trident` pour examiner les pods.
   - S'il n'y a pas de pod opérateur, Astra Trident a été installé avec `tridentctl`.
   - S'il existe un pod opérateur, Astra Trident a été installé à l'aide de l'opérateur Trident soit manuellement, soit à l'aide d'Helm.
2. S'il y a un boîtier opérateur, utiliser `kubectl describe tproc trident` Pour déterminer si Astra Trident a été installé à l'aide d'Helm.
   - S'il y a une étiquette Helm, Astra Trident a été installée à l'aide d'Helm.
   - S'il n'y a pas d'étiquette Helm, Astra Trident a été installé manuellement à l'aide de l'opérateur Trident.

Désinstallez l'installation d'un opérateur Trident

Vous pouvez désinstaller manuellement l'installation d'un opérateur trident ou à l'aide d'Helm.

Désinstallez l'installation manuelle

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

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

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

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

2. **Supprimer `TridentOrchestrator`** : En retirant le `TridentOrchestrator` CR utilisé pour déployer Astra Trident, vous demandez à l'opérateur de désinstaller Trident. L'opérateur traite la dépose de `TridentOrchestrator` Il procède également au retrait du déploiement et de la demonset Astra Trident, en supprimant les pods Trident qu'il avait créés dans le cadre de l'installation.

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

Désinstallez l'installation d'Helm

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

Désinstallez un `tridentctl` installation

Utilisez le `uninstall` commande dans `tridentctl` Pour supprimer toutes les ressources associées à Astra 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 DÉLICTUELLE (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.