



# Gérez Astra Trident

## Astra Trident

NetApp  
April 04, 2024

# Sommaire

- Gérez Astra Trident ..... 1
- Mettez à niveau Astra Trident ..... 1
- Désinstaller Astra Trident ..... 15
- Revenir à la version antérieure d'Astra Trident ..... 17

# Gérez 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 s'appuie sur les versions précédentes, offrant de nouvelles fonctionnalités et améliorations des performances ainsi que 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.
- Depuis Trident 20.01, seule la version bêta de ["snapshots de volume"](#) est pris en charge. Les administrateurs Kubernetes doivent veiller à sauvegarder ou convertir en version bêta les objets de snapshot alpha en toute sécurité, afin de conserver les snapshots alpha hérités.
  - Les snapshots de volumes CSI sont désormais une fonctionnalité GA, qui commence par Kubernetes 1.20. Avant de procéder à la mise à niveau, vous devez supprimer les CRD alpha snapshot à l'aide de `tridentctl obliviate alpha-snapshot-crd` Pour supprimer les CRD pour la spécification alpha snapshot.
  - La version bêta des snapshots de volume introduit un ensemble modifié de CRD et un contrôleur Snapshot, qui doivent tous deux être configurés avant la mise à niveau d'Astra Trident.
  - Pour plus de détails, reportez-vous à ["Ce que vous devez savoir avant de mettre à niveau votre cluster Kubernetes"](#).
- Toutes les mises à niveau des versions 19.04 et antérieures requièrent la migration des métadonnées Astra Trident qui lui sont propres `etcd` Aux objets CRD. Vérifiez le ["Documentation spécifique à votre version d'Astra Trident"](#) pour comprendre le fonctionnement de la mise à niveau.
- 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` 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 vers 23.04 à partir de 22.04 (y compris toutes les versions de point, telles que 22.04.1) directement.

- Si vous disposez d'une version antérieure, vous devez effectuer une mise à niveau en plusieurs étapes à l'aide de la documentation de la version concernée pour obtenir des instructions spécifiques. Pour ce faire, vous devez d'abord effectuer une mise à niveau vers la version la plus récente qui correspond à votre fenêtre des quatre versions. Par exemple, si vous exécutez 18.07 et que vous souhaitez effectuer une mise à niveau vers la version 20.07, suivez la procédure de mise à niveau en plusieurs étapes comme suit :
- a. Première mise à niveau de 18.07 à 19.07.
  - b. Puis mettre à niveau de 19.07 à 20.07.



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

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](#)".

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

1. Utiliser `kubectl get pods - 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

Il existe deux méthodes pour mettre à niveau Astra Trident.

### Quand mettre à niveau à l'aide de l'opérateur

C'est possible "[Mise à niveau avec l'opérateur Trident](#)" si :

- Vous avez installé Astra Trident à l'origine à l'aide de l'opérateur ou de `tridentctl`.
- Vous avez désinstallé CSI Trident et les métadonnées de l'installation persistent.
- Vous disposez d'une installation Astra Trident basée sur CSI. Toutes les versions de 19.07 sont basées sur CSI. Vous pouvez examiner les pods de votre espace de nom Trident pour vérifier votre version.
  - La dénomination des pods dans les versions antérieures à 23.01 utilise : `trident-csi-*`
  - Le nom de pod dans la version 23.01 et ultérieures utilise :
    - `trident-controller-  
<generated id>` pour le module de contrôleur

- `trident-node-<operating system>-<generated id>` pour les pods de nœuds
- `trident-operator-<generated id>` pour le boîtier de l'opérateur



N'utilisez pas l'opérateur pour mettre à niveau Trident si vous utilisez un `etcd` Version Trident basée (19.04 ou version antérieure).

#### Quand effectuer une mise à niveau à l'aide de `tridentctl`

C'est possible Si vous avez installé Astra Trident à l'origine à l'aide de `'tridentctl'`.

`tridentctl` Est la méthode classique d'installation d'Astra Trident et fournit le plus d'options pour ceux qui nécessitent une personnalisation complexe. Pour plus de détails, reportez-vous à "[Choisissez votre méthode d'installation](#)".

#### Modifications apportées à l'opérateur

La version 21.01 d'Astra Trident a apporté des modifications architecturales au conducteur :

- L'opérateur est maintenant **cluster-scoped**. Les instances précédentes de l'opérateur Trident (versions 20.04 à 20.10) étaient **namespace-scoped**. Un opérateur à périmètre de bloc d'instruments est avantageux pour les raisons suivantes :
  - Responsabilité des ressources : l'opérateur gère désormais les ressources associées à une installation d'Astra Trident au niveau du cluster. Dans le cadre de l'installation d'Astra Trident, l'opérateur crée et gère plusieurs ressources à l'aide de `ownerReferences`. Maintenance `ownerReferences` Sur les ressources `cluster-scoped` peut générer des erreurs sur certains distributeurs Kubernetes tels qu'OpenShift. Ceci est réduit avec un opérateur à périmètre sur le cluster. Pour l'auto-rétablissement et l'application de correctifs des ressources Trident, c'est une condition essentielle.
  - Nettoyage pendant la désinstallation : une suppression complète d'Astra Trident nécessite la suppression de toutes les ressources associées. Un opérateur de type espace de noms peut rencontrer des problèmes lors de la suppression des ressources du cluster (telles que `clusterRole`, `ClusterRoleBinding` et `PodSecurityPolicy`) et entraîner un nettoyage incomplet. Un opérateur à périmètre de cluster élimine ce problème. Les utilisateurs peuvent désinstaller complètement Astra Trident et procéder à un nouveau battage si nécessaire.
- `TridentProvisioner` est maintenant remplacé par `TridentOrchestrator` Ressource personnalisée utilisée pour installer et gérer Astra Trident. En outre, un nouveau champ est introduit dans le `TridentOrchestrator` spécifications Les utilisateurs peuvent spécifier que le namespace Trident doit être installé/mis à niveau à partir du à l'aide du `spec.namespace` légale. Vous pouvez voir un exemple "[ici](#)".

#### Mise à niveau avec l'opérateur

Vous pouvez facilement mettre à niveau une installation Astra Trident existante en utilisant l'opérateur manuellement ou à l'aide d'Helm.

#### Mise à niveau à l'aide de l'opérateur Trident

Généralement, vous devez mettre à niveau Astra Trident avec la même méthode que celle utilisée pour l'installer à l'origine. Révision "[Sélectionnez une méthode de mise à niveau](#)" Avant de tenter une mise à niveau avec l'opérateur Trident.

Lors de la mise à niveau à partir d'une instance d'Astra Trident installée à l'aide de l'opérateur dont le périmètre d'espace de noms est compris entre les versions 20.07 et 20.10, l'opérateur Trident effectue automatiquement les opérations suivantes :



- Migre `tridentProvisioner` à un `tridentOrchestrator` objet du même nom
- Supprime `TridentProvisioner` objets et le `tridentprovisioner` CRD
- Mise à niveau d'Astra Trident avec la version de l'opérateur dont le périmètre est défini sur le cluster
- Installez Astra Trident dans le même espace de noms que celui où elle a été installée à l'origine

## Mettre à niveau l'installation d'un opérateur Trident du type cluster

Vous pouvez mettre à niveau l'installation d'un opérateur Trident dont le périmètre est défini sur le cluster. Toutes les versions 21.01 et supérieures d'Astra Trident utilisent un opérateur à périmètre de cluster.

### 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 effectuez une mise à niveau depuis 22.01, exécutez la commande suivante :

```
kubectl delete -f 22.01/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 correspondant à votre environnement et à la version Trident d'Astra. Par exemple, si vous installez Astra Trident 23.04 pour Kubernetes 1.27, exécutez la commande suivante :

```
kubectl create -f 23.04.0/trident-installer/deploy/bundle_post_1_25.yaml -n trident
```

Trident fournit un fichier bundle qui peut être utilisé 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"`.

## Résultats

L'opérateur de Trident identifiera une installation Astra Trident existante et la mettra à niveau vers la même version que l'opérateur.

## Mettre à niveau l'installation d'un opérateur à l'aide d'un espace de noms

Vous pouvez effectuer une mise à niveau à partir d'une instance d'Astra Trident installée en utilisant l'opérateur dont le périmètre d'espace de noms est compris entre 20.07 et 20.10, vers une installation par un opérateur dont le périmètre est défini sur le cluster.

## Avant de commencer

Vous avez besoin du fichier YAML de bundle utilisé pour déployer l'opérateur de périmètre d'espace de noms à partir de `https://github.com/NetApp/trident/tree/stable/vXX.XX/deploy/BUNDLE.YAML` où `vXX.XX` est le numéro de version et `BUNDLE.YAML` Est le nom du fichier YAML du bundle.

## Étapes

1. Vérifiez le `TridentProvisioner` L'état de l'installation Trident existante est `Installed`.

```
kubectl describe tprov trident -n trident | grep Message: -A 3

Message:  Trident installed
Status:   Installed
Version:  v20.10.1
```



Si l'état s'affiche `Updating`, assurez-vous de le résoudre avant de continuer. Pour obtenir la liste des valeurs d'état possibles, reportez-vous à la section ["ici"](#).

2. Créer le `TridentOrchestrator` CRD en utilisant le manifeste fourni avec le programme d'installation Trident.

```
# Download the release required [23.04.0]
mkdir 23.04.0
cd 23.04.0
wget
https://github.com/NetApp/trident/releases/download/v23.04.0/trident-
installer-23.04.0.tar.gz
tar -xf trident-installer-23.04.0.tar.gz
cd trident-installer
kubectl create -f
deploy/crds/trident.netapp.io_tridentorchestrators_crd_post1.16.yaml
```

3. Supprimez l'opérateur délimité par l'espace de noms à l'aide de son manifeste.

a. Vérifiez que vous êtes dans le bon répertoire.

```
pwd
/root/20.10.1/trident-installer
```

b. Supprimez l'opérateur namespace-scoped.

```
kubectl delete -f deploy/<BUNDLE.YAML> -n trident

serviceaccount "trident-operator" deleted
clusterrole.rbac.authorization.k8s.io "trident-operator" deleted
clusterrolebinding.rbac.authorization.k8s.io "trident-operator"
deleted
deployment.apps "trident-operator" deleted
podsecuritypolicy.policy "tridentoperatorpods" deleted
```

c. Vérifiez que l'opérateur Trident a été supprimé.



```
kubectl get all -n trident
```

NAME	READY	STATUS	RESTARTS	AGE
pod/trident-csi-68d979fb85-dsrmn	6/6	Running	12	99d
pod/trident-csi-8jfhf	2/2	Running	6	105d
pod/trident-csi-jtnjz	2/2	Running	6	105d
pod/trident-csi-lcxvh	2/2	Running	8	105d

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

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

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

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

4. (Facultatif) si les paramètres d'installation doivent être modifiés, mettez à jour le `TridentProvisioner` spécifications Cela peut inclure des modifications telles que la modification : les valeurs pour `tridentImage`, `autosupportImage`, un référentiel d'images privé, et la fourniture `imagePullSecrets`) après avoir supprimé l'opérateur du périmètre de l'espace de noms et avant d'installer l'opérateur du périmètre de cluster. Pour obtenir une liste complète des paramètres pouvant être mis à jour, reportez-vous au ["options de configuration"](#).

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

5. Installez l'opérateur Trident cluster-scoped.

- a. Assurez-vous que vous êtes dans le bon répertoire.

```
pwd
/root/23.04.0/trident-installer
```

- b. Installer l'opérateur cluster-scoped dans le même namespace.

Trident fournit un fichier bundle qui peut être utilisé 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"`.

```
kubectl create -f deploy/<BUNDLE.YAML>

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

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

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

- c. Examinez les pods Trident dans le namespace. Le `trident-controller` les noms de pods reflètent la convention de nommage introduite en 23.01.

```
kubectl get pods -n trident

NAME                                READY   STATUS    RESTARTS
AGE
trident-controller-79df798bdc-m79dc 6/6     Running  0
1m41s
trident-node-linux-xrst8            2/2     Running  0
1m41s
trident-operator-5574dbbc68-nthjv   1/1     Running  0
1m52s
```

- d. Vérifiez que Trident a été mis à jour avec la version prévue.

```
kubectl describe torc trident | grep Message -A 3
Message:                Trident installed
Namespace:              trident
Status:                 Installed
Version:                v23.04.0
```

## Mettre à niveau l'installation d'un opérateur basé sur Helm

Effectuer les étapes suivantes pour mettre à niveau l'installation d'un opérateur reposant sur 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. Téléchargez la dernière version d'Astra Trident.
2. Utilisez le `helm upgrade` commande où `trident-operator-23.04.0.tgz` reflète la version vers laquelle vous souhaitez effectuer la mise à niveau.

```
helm upgrade <name> trident-operator-23.04.0.tgz
```

Si vous définissez des options autres que celles par défaut lors de l'installation initiale (par exemple, spécifier des registres privés en miroir pour les images Trident et CSI), utilisez `--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.



Par exemple, pour modifier la valeur par défaut de `tridentDebug`, exécutez la commande suivante :

```
helm upgrade <name> trident-operator-23.04.0-custom.tgz --set
tridentDebug=true
```

3. 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.

### Résultats

L'opérateur de Trident identifiera une installation Astra Trident existante et la mettra à niveau vers la même version que l'opérateur.

### Mise à niveau à partir d'une installation autre que celle d'un opérateur

Vous pouvez effectuer la mise à niveau vers la dernière version de l'opérateur Trident à partir d'un `tridentctl` installation.

## Étapes

1. Téléchargez la dernière version d'Astra Trident.

```
# Download the release required [23.04.0]
mkdir 23.04.0
cd 23.04.0
wget
https://github.com/NetApp/trident/releases/download/v22.01.1/trident-
installer-23.04.0.tar.gz
tar -xf trident-installer-23.04.0.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.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:          v23.04.0

```

## Résultats

Les systèmes back-end et demandes de volume persistant sont automatiquement disponibles.

## Mise à niveau avec tridentctl

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

### Mise à niveau d'Astra Trident à l'aide de `tridentctl`

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.

## Mise à niveau des volumes avec `tridentctl`

Après la mise à niveau, vous pouvez exploiter le riche ensemble des fonctionnalités disponibles dans les nouvelles versions de Trident (telles que les copies Snapshot de volume à la demande) afin de mettre à niveau les volumes à l'aide du `tridentctl upgrade` commande.

Si des volumes sont hérités, vous devez les mettre à niveau d'un type NFS ou iSCSI vers un type CSI pour utiliser l'ensemble des nouvelles fonctionnalités d'Astra Trident. Un volume persistant existant provisionné par Trident prend en charge l'ensemble classique de fonctionnalités.

### Avant de commencer

Tenez compte des points suivants avant de décider de mettre à niveau les volumes vers le type CSI :

- Vous n'avez peut-être pas besoin de mettre à niveau tous les volumes. Les volumes déjà créés continuent à être accessibles et fonctionneront normalement.
- Un PV peut être monté dans le cadre d'un déploiement/StatefulSet lors de la mise à niveau. Il n'est pas nécessaire de faire descendre le déploiement/StatefulSet.
- Vous **ne pouvez pas** connecter un PV à un pod autonome lors de la mise à niveau. Vous devez arrêter le pod avant de mettre à niveau le volume.
- Vous pouvez mettre à niveau uniquement un volume lié à un volume persistant. Les volumes qui ne sont pas liés à des demandes de volume persistant doivent être supprimés et importés avant la mise à niveau.

## Étapes

1. Courez `kubectl get pv` Pour répertorier les volumes persistants.

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

Actuellement, quatre volumes persistants ont été créés par Trident 20.07, à l'aide de `netapp.io/trident` provisionnement.

## 2. Courez `kubectl describe pv` Pour en savoir plus sur le volume persistant.

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

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

Le volume persistant a été créé à l'aide du `netapp.io/trident` de type provisionnement et en mode NFS. Pour prendre en charge toutes les nouvelles fonctionnalités proposées par Astra Trident, ce PV doit être mis à niveau vers le type CSI.

3. Exécutez le `tridentctl upgrade volume <name-of-trident-volume>` Commande de mise à niveau d'un volume Astra Trident hérité vers les spécifications CSI.

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

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

4. Exécutez un `kubectl describe pv` Pour vérifier que le volume est un volume CSI.



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

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

## Désinstaller Astra Trident

Selon l'installation d'Astra Trident, il existe plusieurs options pour le désinstaller.

### Désinstaller en utilisant 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
```

## Désinstaller en utilisant l'opérateur Trident

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

- **Modifier `TridentOrchestrator` Pour définir l'indicateur de désinstallation :** vous pouvez modifier `TridentOrchestrator` et jeu `spec.uninstall=true`. Modifiez le `TridentOrchestrator` CR et régler le `uninstall` marquer comme indiqué ci-dessous :

```
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 le souhaitez  
Réinstallez Trident.

- **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. Pour supprimer entièrement Astra Trident (y compris les CRD qu'il crée) et nettoyer efficacement la ardoise, vous pouvez la modifier `TridentOrchestrator` pour passer `wipeout` option. Voir l'exemple suivant :

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

Cela désinstalle complètement Astra Trident et efface toutes les métadonnées relatives aux systèmes back-end et aux volumes gérés. Les installations ultérieures sont traitées comme des installations neuves.



Vous ne devez envisager de supprimer les CRD que lorsque vous effectuez une désinstallation complète. Cette opération ne peut pas être annulée. **Ne nettoyez pas les CRD à moins que vous ne cherchiez à recommencer et à créer une nouvelle installation Astra Trident.**

## Désinstaller à l'aide de `tridentctl`

Exécutez le `uninstall` commande dans `tridentctl` Comme suit, supprime toutes les ressources associées à Astra Trident, à l'exception des CRD et des objets associés, ce qui facilite l'exécution du programme d'installation pour la mise à jour vers une version plus récente.

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

Pour supprimer totalement Astra Trident, il est conseillé de supprimer les finaliseurs des CRD créés par Astra Trident et de supprimer les CRD.

## Revenir à la version antérieure d'Astra Trident

Découvrez les étapes de la restauration d'une version antérieure d'Astra Trident.

### Quand revenir à une version antérieure

Vous pouvez envisager de rétrograder pour diverses raisons, comme les suivantes :

- Planification des mesures d'urgence
- Résolution immédiate des bugs observés après une mise à niveau
- Problèmes de dépendance, mises à niveau infructueuses et incomplètes

Vous devez envisager une version antérieure si vous passez à une version d'Astra Trident qui utilise des CRD. Comme Astra Trident utilise des CRD pour la maintenance de l'état, toutes les entités de stockage créées (systèmes back-end, classes de stockage, volumes persistants et copies de volume) ont associé des objets CRD au lieu de données écrites dans le `trident` PV (utilisé par la version installée précédente d'Astra Trident). Les nouveaux volumes persistants, systèmes back-end et classes de stockage sont tous gérés en tant qu'objets CRD.

Ne tentez de revenir à une version antérieure d'Astra Trident qui s'exécute avec des CRD (19.07 et versions ultérieures). Cela permet de s'assurer que les opérations effectuées sur la version actuelle d'Astra Trident sont visibles après la mise à niveau vers le bas.

### Si vous ne devez pas revenir à une version antérieure

Vous ne devez pas revenir à une version antérieure de Trident utilisée `etcd` pour conserver l'état (19.04 et versions antérieures). Toutes les opérations effectuées avec la version actuelle d'Astra Trident ne sont pas prises en compte après la mise à niveau vers le bas. Les nouveaux volumes persistants ne sont pas utilisables lors du déplacement vers une version antérieure. Les modifications apportées aux objets tels que les systèmes back-end, les volumes persistants, les classes de stockage et les copies Snapshot de volume (créées/mises à jour/supprimées) ne sont pas visibles pour Astra Trident lors du retour dans une version antérieure. Revenir à une version antérieure ne perturbe pas l'accès aux volumes persistants créés à l'aide de cette version antérieure, sauf si ces derniers ont été mis à niveau.

## Procédure de rétrogradation lorsque Astra Trident est installé à l'aide de l'opérateur

Pour les installations effectuées à l'aide de l'opérateur Trident, le processus de mise à niveau vers une version antérieure est différent et ne nécessite pas l'utilisation de `tridentctl`.

Pour les installations utilisant l'opérateur Trident, il est possible de rétrograder avec l'un des éléments suivants :

- Une version installée à l'aide de l'opérateur à étendue de l'espace de noms (20.07 - 20.10).
- Version installée à l'aide de l'opérateur à étendue de cluster (21.01 et versions ultérieures).

### Revenir à l'opérateur du groupe d'instruments

Pour rétrograder d'Astra Trident à une version utilisant le groupe d'opérateurs du cluster, suivez les étapes indiquées ci-dessous.

#### Étapes

1. **"Désinstaller Astra Trident". Ne supprimez pas les CRD sauf si vous souhaitez supprimer complètement une installation existante.**
2. L'opérateur Trident peut être supprimé via le manifeste de l'opérateur associé à votre version de Trident. Par exemple : `https://github.com/NetApp/trident/tree/stable/vXX.XX/deploy/bundle.yaml` où `vXX.XX` est le numéro de version (par exemple `v22.10`) et `bundle.yaml` est le nom du fichier YAML du bundle.
3. Poursuivez la rétrogradation en installant la version souhaitée d'Astra Trident. Suivez la documentation de la version souhaitée.

### Revenir à l'opérateur de la portée de l'espace de noms

Cette section résume les étapes à suivre pour rétrograder à une version d'Astra Trident comprise entre 20.07 et 20.10, qui sera installée à l'aide de l'opérateur à espace de noms.

#### Étapes

1. **"Désinstaller Astra Trident". Ne pas utiliser les CRD sauf si vous souhaitez supprimer complètement une installation existante.**  
Assurez-vous que le `tridentorchestrator` est supprimé.

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

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

2. L'opérateur Trident peut être supprimé via le manifeste de l'opérateur associé à votre version de Trident. Par exemple : `https://github.com/NetApp/trident/tree/stable/vXX.XX/deploy/bundle.yaml` où `vXX.XX` est le numéro de version (par exemple `v22.10`) et `bundle.yaml` est le nom du fichier YAML du bundle.
3. Supprimez le `tridentorchestrator` CRD.

```
#Check to see if ``tridentorchestrators.trident.netapp.io`` CRD is present and delete it.
```

```
kubectl get crd tridentorchestrators.trident.netapp.io
```

```
NAME                                CREATED AT
tridentorchestrators.trident.netapp.io  2021-01-21T21:11:37Z
```

```
kubectl delete crd tridentorchestrators.trident.netapp.io
```

```
customresourcedefinition.apiextensions.k8s.io
"tridentorchestrators.trident.netapp.io" deleted
```

Astra Trident a été désinstallé.

4. Poursuivez la rétrogradation en installant la version souhaitée. Suivez la documentation de la version souhaitée.

### Rétrograder en utilisant Helm

Pour rétrograder, utilisez `helm rollback` commande. Voir l'exemple suivant :

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

### Processus de rétrogradation lors de l'installation d'Astra Trident à l'aide de `tridentctl`

Si vous avez installé Astra Trident à l'aide de `tridentctl`, le processus de rétrogradation implique les étapes suivantes. Cette séquence vous guide dans le processus de rétrogradation pour passer d'Astra Trident 21.07 à 20.07.



Avant de lancer la restauration vers une version antérieure, vous devez prendre une copie Snapshot de votre cluster Kubernetes `etcd`. Cela vous permet de sauvegarder l'état actuel des CRD d'Astra Trident.

### Étapes

1. Assurez-vous que Trident est installé à l'aide de `tridentctl`. Si vous ne savez pas comment Astra Trident est installé, exécutez ce test simple :
  - a. Lister les pods présents dans l'espace de noms Trident.
  - b. Identifier la version d'Astra Trident exécutée dans votre cluster. Vous pouvez utiliser `tridentctl` Vous pouvez également consulter l'image utilisée dans les pods Trident.
  - c. Si vous **ne voyez pas** `A. tridentOrchestrator`, (ou) `a tridentprovisioner`, (ou) un pod nommé `trident-operator-xxxxxxxxxxx-xxxxx`, Astra Trident \* est installé\* avec `tridentctl`.
2. Désinstallez Astra Trident avec l'ancien `tridentctl` binaire. Dans ce cas, vous désinstallerez avec le binaire 21.07.

```

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

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

```

3. Une fois le fichier terminé, procurez-vous le binaire Trident pour la version souhaitée (dans cet exemple, 20.07) et installez Astra Trident. Vous pouvez générer des YAML personnalisés pour un ["installation personnalisée"](#) si nécessaire.

```

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

```

Le processus de rétrogradation est terminé.

## Informations sur le copyright

Copyright © 2024 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.