



## **Utilisez Trident**

Trident

NetApp

January 14, 2026

This PDF was generated from <https://docs.netapp.com/fr-fr/trident-2502/trident-use/worker-node-prep.html> on January 14, 2026. Always check [docs.netapp.com](https://docs.netapp.com) for the latest.

# Sommaire

Utilisez Trident . . . . .	1
Préparez le nœud de travail . . . . .	1
Choisir les bons outils . . . . .	1
Détection des services de nœud . . . . .	1
Volumes NFS . . . . .	2
Volumes iSCSI . . . . .	2
Volumes NVMe/TCP . . . . .	6
Volumes SCSI sur FC . . . . .	7
Configuration et gestion des systèmes back-end . . . . .	10
Configuration des systèmes back-end . . . . .	10
Azure NetApp Files . . . . .	10
Google Cloud NetApp volumes . . . . .	29
Configurer un système Cloud Volumes Service pour Google Cloud backend . . . . .	46
Configurer un système NetApp HCI ou SolidFire backend . . . . .	58
Pilotes SAN de ONTAP . . . . .	63
Pilotes NAS de ONTAP . . . . .	92
Amazon FSX pour NetApp ONTAP . . . . .	123
Création de systèmes back-end avec kubectl . . . . .	157
Gestion des systèmes back-end . . . . .	164
Créer et gérer des classes de stockage . . . . .	174
Créer une classe de stockage . . . . .	174
Gérer les classes de stockage . . . . .	177
Provisionnement et gestion des volumes . . . . .	179
Provisionner un volume . . . . .	179
Développement des volumes . . . . .	183
Importer des volumes . . . . .	194
Personnaliser les noms et les étiquettes des volumes . . . . .	202
Partager un volume NFS entre les espaces de noms . . . . .	205
Cloner des volumes entre des espaces de noms . . . . .	209
RéPLICATION de volumes à l'aide de SnapMirror . . . . .	212
Utiliser la topologie CSI . . . . .	218
Travailler avec des instantanés . . . . .	225

# Utilisez Trident

## Préparez le nœud de travail

Tous les nœuds workers du cluster Kubernetes doivent pouvoir monter les volumes provisionnés pour vos pods. Pour préparer les nœuds workers, vous devez installer les outils NFS, iSCSI, NVMe/TCP ou FC en fonction de votre sélection de pilotes.

### Choisir les bons outils

Si vous utilisez une combinaison de pilotes, vous devez installer tous les outils requis pour vos pilotes. Les outils installés par défaut sont installés dans les versions récentes de Red Hat Enterprise Linux CoreOS (RHCOS).

#### Outils NFS

"[Installez les outils NFS](#)" si vous utilisez : ontap-nas, , ontap-nas-economy, ontap-nas-flexgroup, azure-netapp-files gcp-cvs .

#### Outils iSCSI

"[Installez les outils iSCSI](#)" si vous utilisez : ontap-san, , ontap-san-economy solidfire-san .

#### Outils NVMe

"[Installez les outils NVMe](#)" Si vous utilisez ontap-san pour le protocole NVMe (Nonvolatile Memory Express) over TCP (NVMe/TCP).



NetApp recommande ONTAP 9.12 ou version ultérieure pour NVMe/TCP.

#### Outils SCSI sur FC

Pour plus d'informations sur la configuration des hôtes SAN FC et FC-NVMe, reportez-vous à la section "[Manières de configurer FC ; hôtes SAN FC-NVMe](#)".

"[Installez les outils FC](#)" Si vous utilisez ontap-san avec sanType fcp (SCSI sur FC).

**Points à prendre en compte :** \* SCSI sur FC est pris en charge sur les environnements OpenShift et KubeVirt. \* SCSI sur FC n'est pas pris en charge sur Docker. \* L'auto-rétablissement iSCSI ne s'applique pas au SCSI sur FC.

## Détection des services de nœud

Trident tente de détecter automatiquement si le nœud peut exécuter des services iSCSI ou NFS.



La découverte de services de nœuds identifie les services détectés, mais ne garantit pas que les services sont correctement configurés. Inversement, l'absence d'un service découvert ne garantit pas l'échec du montage du volume.

## Révision des événements

Trident crée des événements pour le nœud afin d'identifier les services détectés. Pour passer en revue ces événements, exécutez :

```
kubectl get event -A --field-selector involvedObject.name=<Kubernetes node name>
```

## Examiner les services découverts

Trident identifie les services activés pour chaque nœud sur le nœud Trident CR. Pour afficher les services découverts, exécutez :

```
tridentctl get node -o wide -n <Trident namespace>
```

## Volumes NFS

Installez les outils NFS à l'aide des commandes de votre système d'exploitation. Assurez-vous que le service NFS est démarré pendant le démarrage.

### RHEL 8+

```
sudo yum install -y nfs-utils
```

### Ubuntu

```
sudo apt-get install -y nfs-common
```



Redémarrez les nœuds workers après l'installation des outils NFS afin d'éviter toute défaillance lors de la connexion des volumes aux conteneurs.

## Volumes iSCSI

Trident peut automatiquement établir une session iSCSI, analyser les LUN et détecter les périphériques à chemins d'accès multiples, les formater et les monter sur un pod.

### Fonctionnalités d'auto-rétablissement de l'iSCSI

Pour les systèmes ONTAP, Trident exécute l'autorétablissement iSCSI toutes les cinq minutes pour :

1. **Identifier** l'état de session iSCSI souhaité et l'état de session iSCSI en cours.
2. **Comparer** l'état souhaité à l'état actuel pour identifier les réparations nécessaires. Trident détermine les priorités de réparation et le moment où les réparations doivent être effectuées.
3. **Effectuez les réparations** requises pour rétablir l'état de session iSCSI actuel à l'état de session iSCSI souhaité.



Les journaux d'activité d'auto-rétablissement se trouvent dans `trident-main` le conteneur sur le pod Demeset respectif. Pour afficher les journaux, vous devez avoir défini `debug` la valeur « `true` » lors de l'installation de Trident.

Les fonctionnalités d'auto-rétablissement iSCSI de Trident permettent d'éviter :

- Sessions iSCSI obsolètes ou non saines pouvant survenir après un problème de connectivité réseau.  
Dans le cas d'une session obsolète, Trident attend sept minutes avant de se déconnecter pour rétablir la connexion avec un portail.



Par exemple, si les secrets CHAP ont tourné sur le contrôleur de stockage et que le réseau perd la connectivité, les anciens secrets CHAP (*obsolète*) pourraient persister. L'auto-rétablissement peut reconnaître ceci et rétablir automatiquement la session pour appliquer les secrets CHAP mis à jour.

- Sessions iSCSI manquantes
- LUN manquantes

## Points à prendre en compte avant de mettre à niveau Trident

- Si seuls les igroups par nœud (introduits dans 23.04+) sont utilisés, l'auto-rétablissement iSCSI lance des renumérisations SCSI pour tous les périphériques du bus SCSI.
- Si seuls les igroups dont le périmètre est back-end (*obsolète* à partir de la version 23.04) sont utilisés, l'auto-rétablissement iSCSI lance des renumérisations SCSI pour obtenir les ID de LUN exacts dans le bus SCSI.
- Si une combinaison d'igroups par nœud et d'igroups scoped back-end est utilisée, l'autorétablissement iSCSI lance des renumérisations SCSI pour obtenir des ID de LUN exacts sur le bus SCSI.

## Installez les outils iSCSI

Installez les outils iSCSI à l'aide des commandes de votre système d'exploitation.

### Avant de commencer

- Chaque nœud du cluster Kubernetes doit avoir un IQN unique. **C'est une condition préalable nécessaire.**
- Si vous utilisez RHCOS version 4.5 ou ultérieure, ou toute autre distribution Linux compatible avec RHEL, avec le `solidfire-san` pilote et Element OS 12.5 ou version antérieure, assurez-vous que l'algorithme d'authentification CHAP est défini sur MD5 dans les `/etc/iscsi/iscsid.conf` algorithmes CHAP sécurisés conformes à la norme FIPS SHA1, SHA-256 et SHA3-256 sont disponibles avec Element 12.7.

```
sudo sed -i 's/^\\(node.session.auth.chap_algs\\) .*\\1 = MD5/'  
/etc/iscsi/iscsid.conf
```

- Lors de l'utilisation de nœuds worker exécutant RHEL/Red Hat Enterprise Linux CoreOS (RHCOS) avec des PV iSCSI, spécifiez la `discard` mountOption dans la classe de stockage pour effectuer la récupération d'espace en ligne. Reportez-vous à la "[Documentation Red Hat](#)".

## RHEL 8+

1. Installez les packages système suivants :

```
sudo yum install -y lsscsi iscsi-initiator-utils device-mapper-multipath
```

2. Vérifiez que la version iscsi-initiator-utils est 6.2.0.874-2.el7 ou ultérieure :

```
rpm -q iscsi-initiator-utils
```

3. Activer les chemins d'accès multiples :

```
sudo mpathconf --enable --with_multipathd y --find_multipaths n
```



Assurez-vous que /etc/multipath.conf contient find\_multipaths no moins de defaults.

4. Assurez-vous que iscsid et multipathd sont en cours d'exécution :

```
sudo systemctl enable --now iscsid multipathd
```

5. Activer et démarrer iscsi:

```
sudo systemctl enable --now iscsi
```

## Ubuntu

1. Installez les packages système suivants :

```
sudo apt-get install -y open-iscsi lsscsi sg3-utils multipath-tools scsitools
```

2. Vérifiez que la version Open-iscsi est 2.0.874-5ubuntu2.10 ou ultérieure (pour bionic) ou 2.0.874-7.1ubuntu6.1 ou ultérieure (pour focal) :

```
dpkg -l open-iscsi
```

3. Définir la numérisation sur manuelle :

```
sudo sed -i 's/^\\(node.session.scan\\) .*\\1 = manual/'  
/etc/iscsi/iscsid.conf
```

#### 4. Activer les chemins d'accès multiples :

```
sudo tee /etc/multipath.conf <<-EOF  
defaults {  
    user_friendly_names yes  
    find_multipaths no  
}  
EOF  
sudo systemctl enable --now multipath-tools.service  
sudo service multipath-tools restart
```



Assurez-vous que `/etc/multipath.conf` contient `find_multipaths no` moins de `defaults`.

#### 5. Assurez-vous que `open-iscsi` et `multipath-tools` sont activés et en cours d'exécution :

```
sudo systemctl status multipath-tools  
sudo systemctl enable --now open-iscsi.service  
sudo systemctl status open-iscsi
```



Pour Ubuntu 18.04, vous devez détecter les ports cibles avec `iscsiadm` avant de démarrer `open-iscsi` pour que le démon iSCSI démarre. Vous pouvez également modifier le `iscsi` service pour qu'il démarre `iscsid` automatiquement.

### Configurez ou désactivez l'auto-rétablissement iSCSI

Vous pouvez configurer les paramètres d'auto-rétablissement iSCSI Trident suivants pour corriger les sessions obsolètes :

- **Intervalle d'auto-rétablissement iSCSI** : détermine la fréquence à laquelle l'auto-rétablissement iSCSI est appelé (par défaut : 5 minutes). Vous pouvez le configurer pour qu'il s'exécute plus fréquemment en définissant un nombre plus petit ou moins fréquemment en définissant un nombre plus grand.



La définition de l'intervalle d'auto-rétablissement iSCSI sur 0 arrête complètement l'auto-rétablissement iSCSI. Nous ne recommandons pas de désactiver l'auto-rétablissement iSCSI. Il ne doit être désactivé que dans certains cas lorsque l'auto-rétablissement iSCSI ne fonctionne pas comme prévu ou à des fins de débogage.

- **Délai d'attente d'auto-rétablissement iSCSI** : détermine la durée d'attente de l'auto-rétablissement iSCSI avant de se déconnecter d'une session défectueuse et de tenter de se reconnecter (par défaut : 7 minutes). Vous pouvez le configurer sur un nombre plus grand de sorte que les sessions identifiées

comme non saines doivent attendre plus longtemps avant d'être déconnectées, puis une tentative de connexion est faite, ou un nombre plus petit pour se déconnecter et se connecter plus tôt.

### Gouvernail

Pour configurer ou modifier les paramètres d'autorétablissement iSCSI, passez les `iscsiSelfHealingInterval` paramètres et `iscsiSelfHealingWaitTime` lors de l'installation de Helm ou de la mise à jour Helm.

L'exemple suivant définit l'intervalle d'auto-rétablissement iSCSI sur 3 minutes et le temps d'attente d'auto-rétablissement sur 6 minutes :

```
helm install trident trident-operator-100.2502.0.tgz --set  
iscsiSelfHealingInterval=3m0s --set iscsiSelfHealingWaitTime=6m0s -n  
trident
```

### tridentctl

Pour configurer ou modifier les paramètres d'autorétablissement iSCSI, passez les `iscsi-self-healing-interval` paramètres et `iscsi-self-healing-wait-time` lors de l'installation ou de la mise à jour de tridentctl.

L'exemple suivant définit l'intervalle d'auto-rétablissement iSCSI sur 3 minutes et le temps d'attente d'auto-rétablissement sur 6 minutes :

```
tridentctl install --iscsi-self-healing-interval=3m0s --iscsi-self  
-healing-wait-time=6m0s -n trident
```

## Volumes NVMe/TCP

Installez les outils NVMe à l'aide des commandes correspondant à votre système d'exploitation.

- NVMe requiert RHEL 9 ou version ultérieure.
- Si la version du noyau de votre nœud Kubernetes est trop ancienne ou si le package NVMe n'est pas disponible pour votre version du noyau, vous devrez peut-être mettre à jour la version du noyau de votre nœud avec le package NVMe.

## RHEL 9

```
sudo yum install nvme-cli  
sudo yum install linux-modules-extra-$(uname -r)  
sudo modprobe nvme-tcp
```

## Ubuntu

```
sudo apt install nvme-cli  
sudo apt -y install linux-modules-extra-$(uname -r)  
sudo modprobe nvme-tcp
```

## Vérifiez l'installation

Après l'installation, vérifiez que chaque nœud du cluster Kubernetes dispose d'un NQN unique via la commande :

```
cat /etc/nvme/hostnqn
```



Trident modifie la `ctrl_device_tmo` valeur pour s'assurer que NVMe ne renonce pas au chemin s'il tombe en panne. Ne modifiez pas ce paramètre.

## Volumes SCSI sur FC

Vous pouvez désormais utiliser le protocole Fibre Channel (FC) avec Trident pour provisionner et gérer les ressources de stockage sur un système ONTAP.

### Prérequis

Configurez les paramètres réseau et nœud requis pour FC.

#### Paramètres réseau

1. Obtenez le WWPN des interfaces cibles. Pour plus d'informations, reportez-vous à la section "[Interface réseau affiche](#)".
2. Procurez-vous le WWPN pour les interfaces sur l'initiateur (hôte).

Reportez-vous aux utilitaires correspondants du système d'exploitation hôte.

3. Configurer la segmentation sur le commutateur FC à l'aide des WWPN de l'hôte et de la cible.

Pour plus d'informations, reportez-vous à la documentation du fournisseur du commutateur Respecive.

Pour plus d'informations, reportez-vous à la documentation ONTAP suivante :

- "[Présentation de la segmentation Fibre Channel et FCoE](#)"

- "Manières de configurer FC ; hôtes SAN FC-NVMe"

## Installez les outils FC

Installez les outils FC à l'aide des commandes de votre système d'exploitation.

- Lors de l'utilisation de nœuds worker exécutant RHEL/Red Hat Enterprise Linux CoreOS (RHCOS) avec FC PVS, spécifiez la `discard mountOption` dans la classe de stockage pour effectuer la récupération d'espace en ligne. Reportez-vous à la ["Documentation Red Hat"](#).

## RHEL 8+

1. Installez les packages système suivants :

```
sudo yum install -y lsscsi device-mapper-multipath
```

2. Activer les chemins d'accès multiples :

```
sudo mpathconf --enable --with_multipathd y --find_multipaths n
```



Assurez-vous que /etc/multipath.conf contient find\_multipaths no moins de defaults.

3. Assurez-vous que multipathd est en cours d'exécution :

```
sudo systemctl enable --now multipathd
```

## Ubuntu

1. Installez les packages système suivants :

```
sudo apt-get install -y lsscsi sg3-utils multipath-tools scsitools
```

2. Activer les chemins d'accès multiples :

```
sudo tee /etc/multipath.conf <<-EOF
defaults {
    user_friendly_names yes
    find_multipaths no
}
EOF
sudo systemctl enable --now multipath-tools.service
sudo service multipath-tools restart
```



Assurez-vous que /etc/multipath.conf contient find\_multipaths no moins de defaults.

3. Assurez-vous que multipath-tools est activé et en cours d'exécution :

```
sudo systemctl status multipath-tools
```

# Configuration et gestion des systèmes back-end

## Configuration des systèmes back-end

Un back-end définit la relation entre Trident et un système de stockage. Il explique à Trident comment communiquer avec ce système de stockage et comment Trident doit provisionner les volumes à partir de celui-ci.

Trident propose automatiquement des pools de stockage back-end correspondant aux exigences définies par une classe de stockage. Découvrez comment configurer le système back-end pour votre système de stockage.

- "["Configurer un back-end Azure NetApp Files"](#)
- "["Configurez un système back-end Google Cloud NetApp volumes"](#)
- "["Configurer un système back-end Cloud Volumes Service pour Google Cloud Platform"](#)
- "["Configurer un système NetApp HCI ou SolidFire backend"](#)
- "["Configurer un système back-end avec des pilotes NAS ONTAP ou Cloud Volumes ONTAP"](#)
- "["Configurer un système back-end avec des pilotes SAN ONTAP ou Cloud Volumes ONTAP"](#)
- "["Utilisez Trident avec Amazon FSX pour NetApp ONTAP"](#)

## Azure NetApp Files

### Configurer un back-end Azure NetApp Files

Vous pouvez configurer Azure NetApp Files en tant que back-end pour Trident. Vous pouvez relier des volumes NFS et SMB à l'aide d'un back-end Azure NetApp Files. Trident prend également en charge la gestion des identifiants à l'aide d'identités gérées pour les clusters Azure Kubernetes Services (AKS).

#### Détails du pilote Azure NetApp Files

Trident fournit les pilotes de stockage Azure NetApp Files suivants pour communiquer avec le cluster. Les modes d'accès pris en charge sont : *ReadWriteOnce* (RWO), *ReadOnlyMey* (ROX), *ReadWriteMaly* (RWX), *ReadWriteOncePod* (RWOP).

Conducteur	Protocole	Mode Volume	Modes d'accès pris en charge	Systèmes de fichiers pris en charge
azure-netapp-files	NFS SMB	Système de fichiers	RWO, ROX, RWX, RWOP	nfs, smb

### Considérations

- Le service Azure NetApp Files ne prend pas en charge les volumes inférieurs à 50 Gio. Trident crée automatiquement des volumes de 50 Gio si un volume plus petit est demandé.
- Trident prend en charge les volumes SMB montés sur les pods s'exécutant sur les nœuds Windows uniquement.

## Identités gérées pour AKS

Trident prend en charge "identités gérées" les clusters Azure Kubernetes Services. Pour tirer parti de la gestion rationalisée des informations d'identification offerte par les identités gérées, vous devez disposer des éléments suivants :

- Cluster Kubernetes déployé à l'aide d'AKS
- Identités gérées configurées sur le cluster AKS kubernetes
- Trident installé qui inclut le `cloudProvider` à spécifier "Azure".

### Opérateur Trident

Pour installer Trident à l'aide de l'opérateur Trident, modifiez `tridentoperator_cr.yaml` pour définir sur `cloudProvider` "Azure". Par exemple :

```
apiVersion: trident.netapp.io/v1
kind: TridentOrchestrator
metadata:
  name: trident
spec:
  debug: true
  namespace: trident
  imagePullPolicy: IfNotPresent
  cloudProvider: "Azure"
```

### Gouvernail

L'exemple suivant installe les ensembles Trident `cloudProvider` sur Azure à l'aide de la variable d'environnement `$CP` :

```
helm install trident trident-operator-100.2502.0.tgz --create
--namespace --namespace <trident-namespace> --set cloudProvider=$CP
```

<code> </code>

L'exemple suivant installe Trident et définit l'`cloudProvider` `indicateur sur `Azure:

```
tridentctl install --cloud-provider="Azure" -n trident
```

## Identité cloud pour AKS

L'identité cloud permet aux pods Kubernetes d'accéder aux ressources Azure en s'authentifiant comme identité de workload au lieu de fournir des informations d'identification Azure explicites.

Pour tirer parti de l'identité cloud dans Azure, vous devez disposer des éléments suivants :

- Cluster Kubernetes déployé à l'aide d'AKS
- Identité de la charge de travail et émetteur oidc configurés sur le cluster AKS Kubernetes
- Trident installé, qui inclut le `cloudProvider` à spécifier "Azure" et `cloudIdentity` spécifier l'identité de la charge de travail

## Opérateur Trident

Pour installer Trident à l'aide de l'opérateur Trident, modifiez `tridentoperator_cr.yaml` pour définir `cloudProvider` "Azure" et définir `cloudIdentity` sur `azure.workload.identity/client-id: xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxx`.

Par exemple :

```
apiVersion: trident.netapp.io/v1
kind: TridentOrchestrator
metadata:
  name: trident
spec:
  debug: true
  namespace: trident
  imagePullPolicy: IfNotPresent
  cloudProvider: "Azure"
  cloudIdentity: 'azure.workload.identity/client-id: xxxxxxxx-xxxx-
xxxx-xxxx-xxxxxxxxxx' # Edit
```

## Gouvernail

Définissez les valeurs des indicateurs **cloud-Provider (CP)** et **cloud-Identity (ci)** à l'aide des variables d'environnement suivantes :

```
export CP="Azure"
export CI="'azure.workload.identity/client-id: xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxx'"
```

L'exemple suivant installe Trident et définit `cloudProvider` dans Azure à l'aide de la variable d'environnement `$CP` et définit `cloudIdentity` à l'aide de la variable d'environnement `$CI` :

```
helm install trident trident-operator-100.2502.0.tgz --set
cloudProvider=$CP --set cloudIdentity="$CI"
```

`<code> </code>`

Définissez les valeurs des indicateurs **cloud Provider** et **cloud Identity** à l'aide des variables d'environnement suivantes :

```
export CP="Azure"
export CI="azure.workload.identity/client-id: xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxx"
```

L'exemple suivant installe Trident et définit l' `cloud-provider`` indicateur sur ` `$CP`, et `cloud-identity` sur `$CI`:

```
tridentctl install --cloud-provider=$CP --cloud-identity="$CI" -n  
trident
```

## Préparez la configuration d'un back-end Azure NetApp Files

Avant de pouvoir configurer le système back-end Azure NetApp Files, vous devez vous assurer que les exigences suivantes sont respectées.

### Prérequis pour les volumes NFS et SMB

Si vous utilisez Azure NetApp Files pour la première fois ou dans un nouvel emplacement, une configuration initiale est requise pour configurer Azure NetApp Files et créer un volume NFS. Reportez-vous à la "["Azure : configurez Azure NetApp Files et créez un volume NFS"](#)".

Pour configurer et utiliser un "["Azure NetApp Files"](#)" back-end, vous avez besoin des éléments suivants :

- subscriptionID, tenantID clientID , , location Et clientSecret sont facultatifs lorsque vous utilisez des identités gérées sur un cluster AKS.
  - tenantID, clientID Et clientSecret sont facultatifs lors de l'utilisation d'une identité de cloud sur un cluster AKS.
- Un pool de capacité. Reportez-vous à la "["Microsoft : créez un pool de capacité pour Azure NetApp Files"](#)".
- Sous-réseau délégué à Azure NetApp Files. Reportez-vous à la "["Microsoft : déléguer un sous-réseau à Azure NetApp Files"](#)".
- subscriptionID Depuis un abonnement Azure avec Azure NetApp Files activé.
- tenantID, clientID Et clientSecret à partir d'un "["Enregistrement d'applications"](#)" dans Azure Active Directory avec les autorisations suffisantes pour le service Azure NetApp Files. L'enregistrement de l'application doit utiliser l'une des options suivantes :
  - Le rôle propriétaire ou contributeur "["Prédéfinie par Azure"](#)".
  - A "["Rôle de contributeur personnalisé"](#)" au niveau de ('assignableScopes`l'abonnement ) avec les autorisations suivantes qui sont limitées à ce que Trident exige. Après avoir créé le rôle personnalisé, "["Attribuez le rôle à l'aide du portail Azure"](#)".

## Rôle de contributeur personnalisé

```
{
  "id": "/subscriptions/<subscription-id>/providers/Microsoft.Authorization/roleDefinitions/<role-definition-id>",
  "properties": {
    "roleName": "custom-role-with-limited-perms",
    "description": "custom role providing limited permissions",
    "assignableScopes": [
      "/subscriptions/<subscription-id>"
    ],
    "permissions": [
      {
        "actions": [
          "Microsoft.NetApp/netAppAccounts/capacityPools/read",
          "Microsoft.NetApp/netAppAccounts/capacityPools/write",
          "Microsoft.NetApp/netAppAccounts/capacityPools/volumes/read",
          "Microsoft.NetApp/netAppAccounts/capacityPools/volumes/write",
          "Microsoft.NetApp/netAppAccounts/capacityPools/volumes/delete",
          "Microsoft.NetApp/netAppAccounts/capacityPools/volumes/snapshots/read",
          "Microsoft.NetApp/netAppAccounts/capacityPools/volumes/snapshots/write",
          "Microsoft.NetApp/netAppAccounts/capacityPools/volumes/snapshots/delete",
          "Microsoft.NetApp/netAppAccounts/capacityPools/volumes/MountTargets/read",
          "Microsoft.Network/virtualNetworks/read",
          "Microsoft.Network/virtualNetworks/subnets/read",
          "Microsoft.Features/featureProviders/subscriptionFeatureRegistrations/read",
          "Microsoft.Features/featureProviders/subscriptionFeatureRegistrations/write",
          "Microsoft.Features/featureProviders/subscriptionFeatureRegistrations/delete",
        ]
      }
    ]
  }
}
```

```

        "Microsoft.Features/features/read",
        "Microsoft.Features/operations/read",
        "Microsoft.Features/providers/features/read",

        "Microsoft.Features/providers/features/register/action",

        "Microsoft.Features/providers/features/unregister/action",

        "Microsoft.Features/subscriptionFeatureRegistrations/read"
    ],
    "notActions": [],
    "dataActions": [],
    "notDataActions": []
}
]
}
}

```

- L'Azure location qui contient au moins un "[sous-réseau délégué](#)". À partir de Trident 22.01, le location paramètre est un champ obligatoire au niveau supérieur du fichier de configuration back-end. Les valeurs d'emplacement spécifiées dans les pools virtuels sont ignorées.
- Pour utiliser Cloud Identity, obtenez le client ID à partir d'un "[identité gérée attribuée par l'utilisateur](#)" et spécifiez cet ID dans azure.workload.identity/client-id: xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx.

#### **Exigences supplémentaires pour les volumes SMB**

Pour créer un volume SMB, vous devez disposer des éléments suivants :

- Active Directory configuré et connecté à Azure NetApp Files. Reportez-vous à la "[Microsoft : création et gestion des connexions Active Directory pour Azure NetApp Files](#)".
- Cluster Kubernetes avec un nœud de contrôleur Linux et au moins un nœud worker Windows exécutant Windows Server 2022. Trident prend en charge les volumes SMB montés sur les pods s'exécutant sur les nœuds Windows uniquement.
- Au moins un secret Trident contenant vos informations d'identification Active Directory afin que Azure NetApp Files puisse s'authentifier auprès d'Active Directory. Pour générer un secret smbcreds:

```
kubectl create secret generic smbcreds --from-literal username=user
--from-literal password='password'
```

- Un proxy CSI configuré en tant que service Windows. Pour configurer un csi-proxy, reportez-vous "[GitHub : proxy CSI](#)" à la section ou "[GitHub : proxy CSI pour Windows](#)" pour les nœuds Kubernetes s'exécutant sous Windows.

## Exemples et options de configuration du back-end Azure NetApp Files

Découvrez les options de configuration NFS et SMB backend pour Azure NetApp Files et passez en revue les exemples de configuration.

### Options de configuration du back-end

Trident utilise votre configuration back-end (sous-réseau, réseau virtuel, niveau de service et emplacement) pour créer des volumes Azure NetApp Files sur des pools de capacité disponibles à l'emplacement souhaité et qui correspondent au niveau de service et au sous-réseau requis.



Trident ne prend pas en charge les pools de capacité QoS manuelle.

Les systèmes Azure NetApp Files back-end proposent ces options de configuration.

Paramètre	Description	Valeur par défaut
version		Toujours 1
storageDriverName	Nom du pilote de stockage	« azure-netapp-files »
backendName	Nom personnalisé ou système back-end de stockage	Nom du pilote + "_" + caractères aléatoires
subscriptionID	L'ID d'abonnement de votre abonnement Azure Facultatif lorsque les identités gérées sont activées sur un cluster AKS.	
tenantID	L'ID de locataire d'un enregistrement d'application facultatif lorsque des identités gérées ou une identité cloud sont utilisées sur un cluster AKS.	
clientID	L'ID client d'un enregistrement d'application facultatif lorsque des identités gérées ou une identité cloud sont utilisées sur un cluster AKS.	
clientSecret	Le secret client d'une application Registration Facultatif lorsque des identités gérées ou une identité cloud sont utilisées sur un cluster AKS.	
serviceLevel	Un de Standard, , Premium ou Ultra	« » (aléatoire)
location	Nom de l'emplacement Azure où les nouveaux volumes seront créés Facultatif lorsque les identités gérées sont activées sur un cluster AKS.	

Paramètre	Description	Valeur par défaut
resourceGroups	Liste des groupes de ressources pour le filtrage des ressources découvertes	«[] » (sans filtre)
netappAccounts	Liste des comptes NetApp permettant de filtrer les ressources découvertes	«[] » (sans filtre)
capacityPools	Liste des pools de capacité pour le filtrage des ressources découvertes	«[] » (sans filtre, aléatoire)
virtualNetwork	Nom d'un réseau virtuel avec un sous-réseau délégué	« »
subnet	Nom d'un sous-réseau délégué à Microsoft.Netapp/volumes	« »
networkFeatures	Ensemble de fonctions vNet pour un volume, peut être Basic ou Standard. Les fonctions réseau ne sont pas disponibles dans toutes les régions et peuvent être activées dans un abonnement. Si vous spécifiez networkFeatures lorsque la fonctionnalité n'est pas activée, le provisionnement des volumes échoue.	« »
nfsMountOptions	Contrôle précis des options de montage NFS. Ignoré pour les volumes SMB. Pour monter des volumes avec NFS version 4.1, incluez nfsvers=4 dans la liste des options de montage délimitées par des virgules pour choisir NFS v4.1. Les options de montage définies dans une définition de classe de stockage remplacent les options de montage définies dans la configuration backend.	« nfsvers=3 »
limitVolumeSize	Echec du provisionnement si la taille du volume demandé est supérieure à cette valeur	« » (non appliqué par défaut)
debugTraceFlags	Indicateurs de débogage à utiliser lors du dépannage. Exemple, \{"api": false, "method": true, "discovery": true}. Ne l'utilisez pas à moins que vous ne soyez en mesure de résoudre les problèmes et que vous ayez besoin d'un vidding détaillé des journaux.	nul

Paramètre	Description	Valeur par défaut
nasType	Configurez la création de volumes NFS ou SMB. Les options sont nfs, smb ou NULL. La valeur null par défaut sur les volumes NFS.	nfs
supportedTopologies	Représente une liste de régions et de zones prises en charge par ce back-end. Pour plus d'informations, reportez-vous " <a href="#">Utiliser la topologie CSI</a> " à .	



Pour plus d'informations sur les fonctionnalités réseau, reportez-vous "["Configurer les fonctions réseau d'un volume Azure NetApp Files"](#)" à la section .

## Autorisations et ressources requises

Si vous recevez une erreur « aucun pool de capacité trouvé » lors de la création d'une demande de volume persistant, il est probable que votre enregistrement d'application ne dispose pas des autorisations et des ressources requises (sous-réseau, réseau virtuel, pool de capacité). Si le débogage est activé, Trident consigne les ressources Azure découvertes lors de la création du back-end. Vérifiez que vous utilisez un rôle approprié.

Les valeurs de resourceGroups, netappAccounts, capacityPools, , virtualNetwork et subnet peuvent être spécifiées à l'aide de noms courts ou complets. Les noms complets sont recommandés dans la plupart des cas, car les noms abrégés peuvent faire correspondre plusieurs ressources avec le même nom.

Les resourceGroups valeurs , , netappAccounts et capacityPools sont des filtres qui limitent l'ensemble des ressources découvertes à celles disponibles pour ce back-end de stockage et peuvent être spécifiées dans n'importe quelle combinaison. Les noms complets suivent le format suivant :

Type	Format
Groupe de ressources	<groupe de ressources>
Compte NetApp	<groupe de ressources>/<compte netapp>
Pool de capacité	<groupe de ressources>/<compte netapp>/<pool de capacité>
Réseau virtuel	<groupe de ressources>/<région virtuelle>
Sous-réseau	<groupe de ressources>/<région virtuelle>/<sous-réseau>

## Provisionnement de volume

Vous pouvez contrôler le provisionnement de volume par défaut en spécifiant les options suivantes dans une section spéciale du fichier de configuration. Voir [Exemples de configurations](#) pour plus de détails.

Paramètre	Description	Valeur par défaut
exportRule	Règles d'exportation pour les nouveaux volumes. exportRule Doit être une liste séparée par des virgules de toute combinaison d'adresses IPv4 ou de sous-réseaux IPv4 en notation CIDR. Ignoré pour les volumes SMB.	« 0.0.0.0/0 »
snapshotDir	Contrôle la visibilité du répertoire .snapshot	« True » pour NFSv4 « false » pour NFSv3
size	Taille par défaut des nouveaux volumes	« 100G »
unixPermissions	Les autorisations unix des nouveaux volumes (4 chiffres octaux). Ignoré pour les volumes SMB.	« » (fonction d'aperçu, liste blanche requise dans l'abonnement)

#### Exemples de configurations

Les exemples suivants montrent des configurations de base qui laissent la plupart des paramètres par défaut. C'est la façon la plus simple de définir un back-end.

## Configuration minimale

Il s'agit de la configuration back-end minimale absolue. Avec cette configuration, Trident détecte tous vos comptes NetApp, pools de capacité et sous-réseaux délégués à Azure NetApp Files à l'emplacement configuré, et place de nouveaux volumes dans l'un de ces pools et sous-réseaux de manière aléatoire. Comme `nasType` est omis, la `nfs` valeur par défaut s'applique et le back-end provisionne les volumes NFS.

Cette configuration est idéale lorsque vous commencez à utiliser Azure NetApp Files et que vous essayez d'autres fonctionnalités, mais dans la pratique, vous voudrez ajouter de l'étendue aux volumes que vous provisionnez.

```
---
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-anf-1
  namespace: trident
spec:
  version: 1
  storageDriverName: azure-netapp-files
  subscriptionID: 9f87c765-4774-fake-ae98-a721add45451
  tenantID: 68e4f836-edc1-fake-bff9-b2d865ee56cf
  clientID: dd043f63-bf8e-fake-8076-8de91e5713aa
  clientSecret: SECRET
  location: eastus
```

## Identités gérées pour AKS

Cette configuration backend omet subscriptionID tenantID clientID , , et clientSecret, qui sont facultatifs lors de l'utilisation d'identités gérées.

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-anf-1
  namespace: trident
spec:
  version: 1
  storageDriverName: azure-netapp-files
  capacityPools:
    - ultra-pool
  resourceGroups:
    - aks-ami-eastus-rg
  netappAccounts:
    - smb-na
  virtualNetwork: eastus-prod-vnet
  subnet: eastus-anf-subnet
```

## Identité cloud pour AKS

Cette configuration backend omet tenantID clientID , et clientSecret , qui sont facultatifs lors de l'utilisation d'une identité de nuage.

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-anf-1
  namespace: trident
spec:
  version: 1
  storageDriverName: azure-netapp-files
  capacityPools:
    - ultra-pool
  resourceGroups:
    - aks-ami-eastus-rg
  netappAccounts:
    - smb-na
  virtualNetwork: eastus-prod-vnet
  subnet: eastus-anf-subnet
  location: eastus
  subscriptionID: 9f87c765-4774-fake-ae98-a721add45451
```

## Configuration de niveau de service spécifique avec filtres de pool de capacité

Cette configuration back-end place les volumes dans l'emplacement d'Azure eastus dans un Ultra pool de capacité. Trident découvre automatiquement tous les sous-réseaux délégués à Azure NetApp Files à cet emplacement et place un nouveau volume sur l'un d'entre eux de manière aléatoire.

```
---
version: 1
storageDriverName: azure-netapp-files
subscriptionID: 9f87c765-4774-fake-ae98-a721add45451
tenantID: 68e4f836-edc1-fake-bff9-b2d865ee56cf
clientID: dd043f63-bf8e-fake-8076-8de91e5713aa
clientSecret: SECRET
location: eastus
serviceLevel: Ultra
capacityPools:
  - application-group-1/account-1/ultra-1
  - application-group-1/account-1/ultra-2
```

## Configuration avancée

Cette configuration back-end réduit davantage l'étendue du placement des volumes sur un seul sous-réseau et modifie également certains paramètres par défaut du provisionnement des volumes.

```
---  
version: 1  
storageDriverName: azure-netapp-files  
subscriptionID: 9f87c765-4774-fake-ae98-a721add45451  
tenantID: 68e4f836-edc1-fake-bff9-b2d865ee56cf  
clientID: dd043f63-bf8e-fake-8076-8de91e5713aa  
clientSecret: SECRET  
location: eastus  
serviceLevel: Ultra  
capacityPools:  
  - application-group-1/account-1/ultra-1  
  - application-group-1/account-1/ultra-2  
virtualNetwork: my-virtual-network  
subnet: my-subnet  
networkFeatures: Standard  
nfsMountOptions: vers=3,proto=tcp,timeo=600  
limitVolumeSize: 500Gi  
defaults:  
  exportRule: 10.0.0.0/24,10.0.1.0/24,10.0.2.100  
  snapshotDir: "true"  
  size: 200Gi  
  unixPermissions: "0777"
```

## Configuration de pool virtuel

Cette configuration back-end définit plusieurs pools de stockage dans un seul fichier. Cette fonction est utile lorsque plusieurs pools de capacité prennent en charge différents niveaux de service, et que vous souhaitez créer des classes de stockage dans Kubernetes qui les représentent. Des étiquettes de pool virtuel ont été utilisées pour différencier les pools en fonction de performance .

```
---
version: 1
storageDriverName: azure-netapp-files
subscriptionID: 9f87c765-4774-fake-ae98-a721add45451
tenantID: 68e4f836-edc1-fake-bff9-b2d865ee56cf
clientID: dd043f63-bf8e-fake-8076-8de91e5713aa
clientSecret: SECRET
location: eastus
resourceGroups:
  - application-group-1
networkFeatures: Basic
nfsMountOptions: vers=3,proto=tcp,timeo=600
labels:
  cloud: azure
storage:
  - labels:
      performance: gold
      serviceLevel: Ultra
      capacityPools:
        - ultra-1
        - ultra-2
      networkFeatures: Standard
    - labels:
        performance: silver
        serviceLevel: Premium
        capacityPools:
          - premium-1
    - labels:
        performance: bronze
        serviceLevel: Standard
        capacityPools:
          - standard-1
          - standard-2
```

## Configuration des topologies prises en charge

Trident facilite le provisionnement des volumes pour les workloads en fonction des régions et des zones de disponibilité. Le `supportedTopologies` bloc de cette configuration back-end est utilisé pour fournir une liste de régions et de zones par back-end. Les valeurs de région et de zone spécifiées ici doivent correspondre aux valeurs de région et de zone indiquées sur les étiquettes de chaque nœud de cluster Kubernetes. Ces régions et zones représentent la liste des valeurs autorisées pouvant être fournies dans une classe de stockage. Pour les classes de stockage qui contiennent un sous-ensemble des régions et zones fournies dans un back-end, Trident crée des volumes dans la région et la zone mentionnées. Pour plus d'informations, reportez-vous "[Utiliser la topologie CSI](#)" à .

```
---  
version: 1  
storageDriverName: azure-netapp-files  
subscriptionID: 9f87c765-4774-fake-ae98-a721add45451  
tenantID: 68e4f836-edc1-fake-bff9-b2d865ee56cf  
clientID: dd043f63-bf8e-fake-8076-8de91e5713aa  
clientSecret: SECRET  
location: eastus  
serviceLevel: Ultra  
capacityPools:  
  - application-group-1/account-1/ultra-1  
  - application-group-1/account-1/ultra-2  
supportedTopologies:  
  - topology.kubernetes.io/region: eastus  
    topology.kubernetes.io/zone: eastus-1  
  - topology.kubernetes.io/region: eastus  
    topology.kubernetes.io/zone: eastus-2
```

## Définitions des classes de stockage

Les définitions suivantes `StorageClass` font référence aux pools de stockage ci-dessus.

### Exemples de définitions utilisant le parameter.selector champ

A l'aide de `parameter.selector`, vous pouvez spécifier pour chaque `StorageClass` pool virtuel utilisé pour héberger un volume. Les aspects définis dans le pool sélectionné seront définis pour le volume.

```

---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: gold
provisioner: csi.trident.netapp.io
parameters:
  selector: performance=gold
allowVolumeExpansion: true

---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: silver
provisioner: csi.trident.netapp.io
parameters:
  selector: performance=silver
allowVolumeExpansion: true

---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: bronze
provisioner: csi.trident.netapp.io
parameters:
  selector: performance=bronze
allowVolumeExpansion: true

```

## Exemples de définitions pour les volumes SMB

A l'aide de `nasType`, `node-stage-secret-name`, et `node-stage-secret-namespace`, vous pouvez spécifier un volume SMB et fournir les informations d'identification Active Directory requises.

## Configuration de base sur l'espace de noms par défaut

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: anf-sc-smb
provisioner: csi.trident.netapp.io
parameters:
  backendType: "azure-netapp-files"
  trident.netapp.io/nasType: "smb"
  csi.storage.k8s.io/node-stage-secret-name: "smbcreds"
  csi.storage.k8s.io/node-stage-secret-namespace: "default"
```

## Utilisation de secrets différents par espace de noms

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: anf-sc-smb
provisioner: csi.trident.netapp.io
parameters:
  backendType: "azure-netapp-files"
  trident.netapp.io/nasType: "smb"
  csi.storage.k8s.io/node-stage-secret-name: "smbcreds"
  csi.storage.k8s.io/node-stage-secret-namespace: ${pvc.namespace}
```

## Utilisation de secrets différents par volume

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: anf-sc-smb
provisioner: csi.trident.netapp.io
parameters:
  backendType: "azure-netapp-files"
  trident.netapp.io/nasType: "smb"
  csi.storage.k8s.io/node-stage-secret-name: ${pvc.name}
  csi.storage.k8s.io/node-stage-secret-namespace: ${pvc.namespace}
```



nasType: smb Filtres pour les pools qui prennent en charge les volumes SMB. nasType : nfs Ou nasType: null des filtres pour les pools NFS.

## Créer le backend

Après avoir créé le fichier de configuration backend, exécutez la commande suivante :

```
tridentctl create backend -f <backend-file>
```

Si la création du back-end échoue, la configuration du back-end est erronée. Vous pouvez afficher les journaux pour déterminer la cause en exécutant la commande suivante :

```
tridentctl logs
```

Après avoir identifié et corrigé le problème avec le fichier de configuration, vous pouvez exécuter de nouveau la commande create.

## Google Cloud NetApp volumes

### Configurez un système back-end Google Cloud NetApp volumes

Vous pouvez désormais configurer Google Cloud NetApp volumes en tant que back-end pour Trident. Vous pouvez connecter des volumes NFS et SMB à l'aide d'un back-end Google Cloud NetApp volumes.

#### Détails du pilote Google Cloud NetApp volumes

Trident fournit le `google-cloud-netapp-volumes` pilote pour communiquer avec le cluster. Les modes d'accès pris en charge sont : `ReadWriteOnce` (RWO), `ReadOnlyMany` (ROX), `ReadWriteMany` (RWX), `ReadWriteOncePod` (RWOP).

Conducteur	Protocole	Mode Volume	Modes d'accès pris en charge	Systèmes de fichiers pris en charge
<code>google-cloud-netapp-volumes</code>	NFS SMB	Système de fichiers	RWO, ROX, RWX, RWOP	<code>nfs</code> , <code>smb</code>

#### Identité cloud pour GKE

L'identité cloud permet aux pods Kubernetes d'accéder aux ressources Google Cloud en s'authentifiant comme identité de workload au lieu de fournir des informations d'identification Google Cloud explicites.

Pour tirer parti de l'identité cloud dans Google Cloud, vous devez disposer des éléments suivants :

- Cluster Kubernetes déployé à l'aide de GKE.
- Identité de la charge de travail configurée sur le cluster GKE et le serveur de métadonnées GKE configuré sur les pools de nœuds.

- Compte de service GCP avec le rôle d'administrateur Google Cloud NetApp volumes (rôles/NetApp.admin) ou un rôle personnalisé.
- Trident a installé, qui inclut le fournisseur cloud, afin de spécifier « GCP » et « cloudIdentity » en spécifiant le nouveau compte de service GCP. Un exemple est donné ci-dessous.

## Opérateur Trident

Pour installer Trident à l'aide de l'opérateur Trident, modifiez `tridentoperator_cr.yaml` pour définir sur `cloudProvider` "GCP" et définir `cloudIdentity` sur `iam.gke.io/gcp-service-account: cloudvolumes-admin-sa@mygcpproject.iam.gserviceaccount.com`.

Par exemple :

```
apiVersion: trident.netapp.io/v1
kind: TridentOrchestrator
metadata:
  name: trident
spec:
  debug: true
  namespace: trident
  imagePullPolicy: IfNotPresent
  cloudProvider: "GCP"
  cloudIdentity: 'iam.gke.io/gcp-service-account: cloudvolumes-admin-sa@mygcpproject.iam.gserviceaccount.com'
```

## Gouvernail

Définissez les valeurs des indicateurs **cloud-Provider (CP)** et **cloud-Identity (ci)** à l'aide des variables d'environnement suivantes :

```
export CP="GCP"
export ANNOTATION="'iam.gke.io/gcp-service-account: cloudvolumes-admin-sa@mygcpproject.iam.gserviceaccount.com'"
```

L'exemple suivant installe Trident et définit `cloudProvider GCP` à l'aide de la variable d'environnement `$CP` et définit le `cloudIdentity` à l'aide de la variable d'environnement `$ANNOTATION` :

```
helm install trident trident-operator-100.2502.0.tgz --set
cloudProvider=$CP --set cloudIdentity="$ANNOTATION"
```

<code> </code>

Définissez les valeurs des indicateurs **cloud Provider** et **cloud Identity** à l'aide des variables d'environnement suivantes :

```
export CP="GCP"
export ANNOTATION="'iam.gke.io/gcp-service-account: cloudvolumes-admin-sa@mygcpproject.iam.gserviceaccount.com'"
```

L'exemple suivant installe Trident et définit l' `cloud-provider``indicateur sur ``$CP`, et `cloud-identity` sur `$ANNOTATION`:

```
tridentctl install --cloud-provider=$CP --cloud  
-identity="$ANNOTATION" -n trident
```

## Préparez la configuration d'un système back-end Google Cloud NetApp volumes

Avant de pouvoir configurer votre système back-end Google Cloud NetApp volumes, vous devez vous assurer que les exigences suivantes sont respectées.

### Prérequis pour les volumes NFS

Si vous utilisez Google Cloud NetApp volumes pour la première fois ou dans un nouvel emplacement, une configuration initiale est requise pour configurer Google Cloud NetApp volumes et créer un volume NFS. Reportez-vous à la "["Avant de commencer"](#)".

Vérifiez les points suivants avant de configurer le système back-end Google Cloud NetApp volumes :

- Compte Google Cloud configuré avec le service Google Cloud NetApp volumes. Reportez-vous à la "["Google Cloud NetApp volumes"](#)".
- Numéro de projet de votre compte Google Cloud. Reportez-vous à la "["Identification des projets"](#)".
- Un compte de service Google Cloud avec le rôle d'administrateur NetApp volumes (roles/netapp.admin). Reportez-vous à la "["Rôles et autorisations de gestion des identités et des accès"](#)".
- Fichier de clé API pour votre compte GCNV. Reportez-vous à "["Créez une clé de compte de service"](#)".
- Un pool de stockage. Reportez-vous à la "["Présentation des pools de stockage"](#)".

Pour plus d'informations sur la configuration de l'accès à Google Cloud NetApp volumes, consultez "["Configurez l'accès à Google Cloud NetApp volumes"](#)".

## Options et exemples de configuration back-end de Google Cloud NetApp volumes

Découvrez les options de configuration back-end pour Google Cloud NetApp volumes et examinez des exemples de configuration.

### Options de configuration du back-end

Chaque back-end provisionne les volumes dans une seule région Google Cloud. Pour créer des volumes dans d'autres régions, vous pouvez définir des systèmes back-end supplémentaires.

Paramètre	Description	Valeur par défaut
version		Toujours 1
storageDriverName	Nom du pilote de stockage	La valeur de storageDriverName doit être indiquée comme « google-cloud-netapp-volumes ».

Paramètre	Description	Valeur par défaut
backendName	(Facultatif) Nom personnalisé du système back-end de stockage	Nom du pilote + " " + partie de la clé API
storagePools	Paramètre facultatif utilisé pour spécifier les pools de stockage pour la création du volume.	
projectNumber	Numéro de projet de compte Google Cloud. La valeur est disponible sur la page d'accueil du portail Google Cloud.	
location	Emplacement Google Cloud dans lequel Trident crée des volumes GCNV. Lors de la création de clusters Kubernetes répartis entre régions, les volumes créés dans un location peuvent être utilisés dans des workloads planifiés sur des nœuds répartis sur plusieurs régions Google Cloud. Le trafic entre les régions coûte plus cher.	
apiKey	Clé d'API pour le compte de service Google Cloud avec le netapp.admin rôle. Il inclut le contenu au format JSON du fichier de clé privée d'un compte de service Google Cloud (copié en compte dans le fichier de configuration back-end). Le apiKey doit inclure des paires clé-valeur pour les clés suivantes : type project_id, client_email,, client_id,, auth_uri token_uri auth_provider_x509_cert_url, et client_x509_cert_url.	
nfsMountOptions	Contrôle précis des options de montage NFS.	« nfsvers=3 »
limitVolumeSize	Echec du provisionnement si la taille du volume demandé est supérieure à cette valeur.	« » (non appliqué par défaut)
serviceLevel	Niveau de service d'un pool de stockage et de ses volumes. Les valeurs sont flex, standard, , premium`ou `extreme.	
network	Réseau Google Cloud utilisé pour les volumes GCNV.	
debugTraceFlags	Indicateurs de débogage à utiliser lors du dépannage. Exemple, { "api":false, "method":true}. Ne l'utilisez pas à moins que vous ne soyez en mesure de résoudre les problèmes et que vous ayez besoin d'un vidage détaillé des journaux.	nul
nasType	Configurez la création de volumes NFS ou SMB. Les options sont nfs, smb ou NULL. La valeur null par défaut sur les volumes NFS.	nfs

Paramètre	Description	Valeur par défaut
supportedTopologies	Représente une liste de régions et de zones prises en charge par ce back-end. Pour plus d'informations, reportez-vous " <a href="#">Utiliser la topologie CSI</a> " à . Par exemple : supportedTopologies: - topology.kubernetes.io/region: asia-east1 topology.kubernetes.io/zone: asia-east1-a	

### Options de provisionnement de volumes

Vous pouvez contrôler le provisionnement du volume par défaut dans la `defaults` section du fichier de configuration.

Paramètre	Description	Valeur par défaut
exportRule	Règles d'exportation pour les nouveaux volumes. Doit être une liste séparée par des virgules de toute combinaison d'adresses IPv4.	« 0.0.0.0/0 »
snapshotDir	Accès au <code>.snapshot</code> répertoire	« True » pour NFSv4 « false » pour NFSv3
snapshotReserve	Pourcentage de volume réservé pour les snapshots	« » (accepter la valeur par défaut de 0)
unixPermissions	Les autorisations unix des nouveaux volumes (4 chiffres octaux).	« »

### Exemples de configurations

Les exemples suivants montrent des configurations de base qui laissent la plupart des paramètres par défaut. C'est la façon la plus simple de définir un back-end.

## Configuration minimale

Il s'agit de la configuration back-end minimale absolue. Avec cette configuration, Trident détecte tous vos pools de stockage délégués aux volumes Google Cloud NetApp dans l'emplacement configuré et place de nouveaux volumes dans l'un de ces pools de manière aléatoire. Comme `nasType` est omis, la `nfs` valeur par défaut s'applique et le back-end provisionne les volumes NFS.

Cette configuration est idéale lorsque vous n'utilisez que Google Cloud NetApp volumes et que vous essayez d'effectuer des opérations. Dans la pratique, vous devrez probablement fournir une étendue supplémentaire pour les volumes que vous provisionnez.

```

---
apiVersion: v1
kind: Secret
metadata:
  name: backend-tbc-gcnv-secret
type: Opaque
stringData:
  private_key_id: f2cb6ed6d7cc10c453f7d3406fc700c5df0ab9ec
  private_key: |
    -----BEGIN PRIVATE KEY-----\n
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzzE4jK3b1/qp8B4Kws8zX5ojY9m\n
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzzE4jK3b1/qp8B4Kws8zX5ojY9m\n
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzzE4jK3b1/qp8B4Kws8zX5ojY9m\n
    XsYg6gyxy4zq70lwWgLwGa==\n
    -----END PRIVATE KEY-----\n

---
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-gcnv
spec:
  version: 1
  storageDriverName: google-cloud-netapp-volumes
  projectNumber: "123455380079"
  location: europe-west6
  serviceLevel: premium
  apiKey:
    type: service_account
    project_id: my-gcnv-project
    client_email: myproject-prod@my-gcnv-
    project.iam.gserviceaccount.com
    client_id: "103346282737811234567"
    auth_uri: https://accounts.google.com/o/oauth2/auth
    token_uri: https://oauth2.googleapis.com/token
    auth_provider_x509_cert_url:
      https://www.googleapis.com/oauth2/v1/certs
    client_x509_cert_url:
      https://www.googleapis.com/robot/v1/metadata/x509/myproject-prod%40my-
      gcnv-project.iam.gserviceaccount.com
    credentials:
      name: backend-tbc-gcnv-secret

```

## Configuration pour les volumes SMB

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-gcnv1
  namespace: trident
spec:
  version: 1
  storageDriverName: google-cloud-netapp-volumes
  projectNumber: "123456789"
  location: asia-east1
  serviceLevel: flex
  nasType: smb
  apiKey:
    type: service_account
    project_id: cloud-native-data
    client_email: trident-sample@cloud-native-
data.iam.gserviceaccount.com
    client_id: "123456789737813416734"
    auth_uri: https://accounts.google.com/o/oauth2/auth
    token_uri: https://oauth2.googleapis.com/token
    auth_provider_x509_cert_url:
      https://www.googleapis.com/oauth2/v1/certs
    client_x509_cert_url:
      https://www.googleapis.com/robot/v1/metadata/x509/trident-
sample%40cloud-native-data.iam.gserviceaccount.com
  credentials:
    name: backend-tbc-gcnv-secret
```

## Configuration avec filtre StoragePools

```

---
apiVersion: v1
kind: Secret
metadata:
  name: backend-tbc-gcnv-secret
type: Opaque
stringData:
  private_key_id: f2cb6ed6d7cc10c453f7d3406fc700c5df0ab9ec
  private_key: |
    -----BEGIN PRIVATE KEY-----
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzzE4jK3b1/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzzE4jK3b1/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzzE4jK3b1/qp8B4Kws8zX5ojY9m
    XsYg6gyxy4zq70lwWgLwGa==
    -----END PRIVATE KEY-----

---
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-gcnv
spec:
  version: 1
  storageDriverName: google-cloud-netapp-volumes
  projectNumber: "123455380079"
  location: europe-west6
  serviceLevel: premium
  storagePools:
    - premium-pool1-europe-west6
    - premium-pool2-europe-west6
  apiKey:
    type: service_account
    project_id: my-gcnv-project
    client_email: myproject-prod@my-gcnv-
project.iam.gserviceaccount.com
    client_id: "103346282737811234567"
    auth_uri: https://accounts.google.com/o/oauth2/auth
    token_uri: https://oauth2.googleapis.com/token
    auth_provider_x509_cert_url:
      https://www.googleapis.com/oauth2/v1/certs
      client_x509_cert_url:
        https://www.googleapis.com/robot/v1/metadata/x509/myproject-prod%40my-
gcnv-project.iam.gserviceaccount.com
    credentials:
      name: backend-tbc-gcnv-secret

```

## Configuration de pool virtuel

Cette configuration back-end définit plusieurs pools virtuels dans un seul fichier. Les pools virtuels sont définis dans la storage section. Ces fonctionnalités sont utiles lorsque plusieurs pools de stockage prennent en charge différents niveaux de services et que vous souhaitez créer dans Kubernetes des classes de stockage qui les représentent. Les étiquettes de pool virtuel sont utilisées pour différencier les pools. Par exemple, dans l'exemple ci-dessous performance , le libellé et serviceLevel le type sont utilisés pour différencier les pools virtuels.

Vous pouvez également définir des valeurs par défaut applicables à tous les pools virtuels et remplacer les valeurs par défaut des pools virtuels individuels. Dans l'exemple suivant, snapshotReserve et exportRule servent de valeurs par défaut pour tous les pools virtuels.

Pour plus d'informations, reportez-vous "[Pools virtuels](#)" à .

```
---  
apiVersion: v1  
kind: Secret  
metadata:  
  name: backend-tbc-gcnv-secret  
type: Opaque  
stringData:  
  private_key_id: f2cb6ed6d7cc10c453f7d3406fc700c5df0ab9ec  
  private_key: |  
    -----BEGIN PRIVATE KEY-----  
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzzE4jK3b1/qp8B4Kws8zX5ojY9m  
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzzE4jK3b1/qp8B4Kws8zX5ojY9m  
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzzE4jK3b1/qp8B4Kws8zX5ojY9m  
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGzlzzE4jK3b1/qp8B4Kws8zX5ojY9m  
    XsYg6gyxy4zq70lwWgLwGa==  
    -----END PRIVATE KEY-----  
  
---  
apiVersion: trident.netapp.io/v1  
kind: TridentBackendConfig  
metadata:  
  name: backend-tbc-gcnv  
spec:  
  version: 1  
  storageDriverName: google-cloud-netapp-volumes  
  projectNumber: "123455380079"  
  location: europe-west6  
  apiKey:  
    type: service_account  
    project_id: my-gcnv-project  
    client_email: myproject-prod@my-gcnv-  
    project.iam.gserviceaccount.com  
    client_id: "103346282737811234567"
```

```

auth_uri: https://accounts.google.com/o/oauth2/auth
token_uri: https://oauth2.googleapis.com/token
auth_provider_x509_cert_url:
https://www.googleapis.com/oauth2/v1/certs
client_x509_cert_url:
https://www.googleapis.com/robot/v1/metadata/x509/myproject-prod%40my-
gcnv-project.iam.gserviceaccount.com
credentials:
  name: backend-tbc-gcnv-secret
defaults:
  snapshotReserve: "10"
  exportRule: 10.0.0.0/24
storage:
  - labels:
      performance: extreme
      serviceLevel: extreme
      defaults:
        snapshotReserve: "5"
        exportRule: 0.0.0.0/0
    - labels:
      performance: premium
      serviceLevel: premium
    - labels:
      performance: standard
      serviceLevel: standard

```

## Identité cloud pour GKE

```

apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-gcp-gcnv
spec:
  version: 1
  storageDriverName: google-cloud-netapp-volumes
  projectNumber: '012345678901'
  network: gcnv-network
  location: us-west2
  serviceLevel: Premium
  storagePool: pool-premium1

```

## Configuration des topologies prises en charge

Trident facilite le provisionnement des volumes pour les workloads en fonction des régions et des zones de disponibilité. Le `supportedTopologies` bloc de cette configuration back-end est utilisé pour fournir une liste de régions et de zones par back-end. Les valeurs de région et de zone spécifiées ici doivent correspondre aux valeurs de région et de zone indiquées sur les étiquettes de chaque nœud de cluster Kubernetes. Ces régions et zones représentent la liste des valeurs autorisées pouvant être fournies dans une classe de stockage. Pour les classes de stockage qui contiennent un sous-ensemble des régions et zones fournies dans un back-end, Trident crée des volumes dans la région et la zone mentionnées. Pour plus d'informations, reportez-vous "[Utiliser la topologie CSI](#)" à .

```
---  
version: 1  
storageDriverName: google-cloud-netapp-volumes  
subscriptionID: 9f87c765-4774-fake-ae98-a721add45451  
tenantID: 68e4f836-edc1-fake-bff9-b2d865ee56cf  
clientID: dd043f63-bf8e-fake-8076-8de91e5713aa  
clientSecret: SECRET  
location: asia-east1  
serviceLevel: flex  
supportedTopologies:  
  - topology.kubernetes.io/region: asia-east1  
    topology.kubernetes.io/zone: asia-east1-a  
  - topology.kubernetes.io/region: asia-east1  
    topology.kubernetes.io/zone: asia-east1-b
```

### Et la suite ?

Après avoir créé le fichier de configuration backend, exécutez la commande suivante :

```
kubectl create -f <backend-file>
```

Pour vérifier que le back-end est correctement créé, exécutez la commande suivante :

```
kubectl get tridentbackendconfig
```

NAME	BACKEND NAME	BACKEND UUID
PHASE	STATUS	
backend-tbc-gcnv	backend-tbc-gcnv	b2fd1ff9-b234-477e-88fd-713913294f65
Bound	Success	

Si la création du back-end échoue, la configuration du back-end est erronée. Vous pouvez décrire le back-end à l'aide de la `kubectl get tridentbackendconfig <backend-name>` commande ou afficher les journaux pour déterminer la cause en exécutant la commande suivante :

```
tridentctl logs
```

Après avoir identifié et corrigé le problème avec le fichier de configuration, vous pouvez supprimer le back-end et exécuter à nouveau la commande `create`.

#### Définitions des classes de stockage

Voici une définition de base `StorageClass` qui fait référence au back-end ci-dessus.

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: gcnv-nfs-sc
provisioner: csi.trident.netapp.io
parameters:
  backendType: "google-cloud-netapp-volumes"
```

#### Exemples de définitions utilisant le `parameter.selector` champ :

A l'aide de `parameter.selector`, vous pouvez spécifier pour chaque `StorageClass` système "[pool virtuel](#)" utilisé pour héberger un volume. Les aspects définis dans le pool sélectionné seront définis pour le volume.

```

apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: extreme-sc
provisioner: csi.trident.netapp.io
parameters:
  selector: performance=extreme
  backendType: google-cloud-netapp-volumes

---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: premium-sc
provisioner: csi.trident.netapp.io
parameters:
  selector: performance=premium
  backendType: google-cloud-netapp-volumes

---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: standard-sc
provisioner: csi.trident.netapp.io
parameters:
  selector: performance=standard
  backendType: google-cloud-netapp-volumes

```

Pour plus de détails sur les classes de stockage, reportez-vous "[Créer une classe de stockage](#)" à la section .

### **Exemples de définitions pour les volumes SMB**

A l'aide de `nasType`, `node-stage-secret-name`, et `node-stage-secret-namespace`, vous pouvez spécifier un volume SMB et fournir les informations d'identification Active Directory requises. N'importe quel utilisateur/mot de passe Active Directory avec des autorisations n'importe quel/aucune peut être utilisé pour le secret d'étape du noeud.

## Configuration de base sur l'espace de noms par défaut

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: gcnv-sc-smb
provisioner: csi.trident.netapp.io
parameters:
  backendType: "google-cloud-netapp-volumes"
  trident.netapp.io/nasType: "smb"
  csi.storage.k8s.io/node-stage-secret-name: "smbcreds"
  csi.storage.k8s.io/node-stage-secret-namespace: "default"
```

## Utilisation de secrets différents par espace de noms

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: gcnv-sc-smb
provisioner: csi.trident.netapp.io
parameters:
  backendType: "google-cloud-netapp-volumes"
  trident.netapp.io/nasType: "smb"
  csi.storage.k8s.io/node-stage-secret-name: "smbcreds"
  csi.storage.k8s.io/node-stage-secret-namespace: ${pvc.namespace}
```

## Utilisation de secrets différents par volume

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: gcnv-sc-smb
provisioner: csi.trident.netapp.io
parameters:
  backendType: "google-cloud-netapp-volumes"
  trident.netapp.io/nasType: "smb"
  csi.storage.k8s.io/node-stage-secret-name: ${pvc.name}
  csi.storage.k8s.io/node-stage-secret-namespace: ${pvc.namespace}
```



nasType: smb Filtres pour les pools qui prennent en charge les volumes SMB. nasType: nfs Ou nasType: null des filtres pour les pools NFS.

## Exemple de définition de PVC

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: gcnv-nfs-pvc
spec:
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 100Gi
  storageClassName: gcnv-nfs-sc
```

Pour vérifier si la demande de volume persistant est liée, exécutez la commande suivante :

```
kubectl get pvc gcnv-nfs-pvc
```

NAME	STATUS	VOLUME	CAPACITY
ACCESS MODES	STORAGECLASS	AGE	
gcnv-nfs-pvc	Bound	pvc-b00f2414-e229-40e6-9b16-ee03eb79a213	100Gi
RWX	gcnv-nfs-sc	1m	

## Configurer un système Cloud Volumes Service pour Google Cloud backend

Découvrez comment configurer NetApp Cloud Volumes Service pour Google Cloud en tant que back-end pour votre installation Trident à l'aide des exemples de configurations fournis.

### Détails du pilote Google Cloud

Trident fournit le gcp-cvs pilote pour communiquer avec le cluster. Les modes d'accès pris en charge sont : *ReadWriteOnce* (RWO), *ReadOnlyMey* (ROX), *ReadWriteMaly* (RWX), *ReadWriteOncePod* (RWOP).

Conducteur	Protocole	Mode Volume	Modes d'accès pris en charge	Systèmes de fichiers pris en charge
gcp-cvs	NFS	Système de fichiers	RWO, ROX, RWX, RWOP	nfs

## En savoir plus sur la prise en charge de Trident pour Cloud Volumes Service pour Google Cloud

Trident peut créer des volumes Cloud Volumes Service en deux "[types de service](#)" :

- **CVS-Performance** : type de service Trident par défaut. Ce type de service aux performances optimisées est parfaitement adapté aux charges de travail de production qui exigent des performances élevées. Le type de service CVS-Performance est une option matérielle prenant en charge les volumes d'une taille minimale de 100 Gio. Vous pouvez choisir l'une des "[trois niveaux de service](#)" options suivantes :
  - standard
  - premium
  - extreme
- **CVS**: Le type de service CVS fournit une haute disponibilité zonale avec des niveaux de performance limités à modérés. Le type de service CVS est une option logicielle utilisant des pools de stockage pour prendre en charge des volumes de 1 Gio. Le pool de stockage peut contenir jusqu'à 50 volumes dans lesquels tous les volumes partagent la capacité et les performances du pool. Vous pouvez choisir l'une des "[deux niveaux de service](#)" options suivantes :
  - standardsw
  - zonedredundantstandardsw

### Ce dont vous avez besoin

Pour configurer et utiliser le "[Cloud Volumes Service pour Google Cloud](#)" back-end, vous avez besoin des éléments suivants :

- Un compte Google Cloud configuré avec NetApp Cloud Volumes Service
- Numéro de projet de votre compte Google Cloud
- Compte de service Google Cloud avec le `netappcloudvolumes.admin` rôle
- Fichier de clé API pour votre compte Cloud Volumes Service

### Options de configuration du back-end

Chaque back-end provisionne les volumes dans une seule région Google Cloud. Pour créer des volumes dans d'autres régions, vous pouvez définir des systèmes back-end supplémentaires.

Paramètre	Description	Valeur par défaut
version		Toujours 1
storageDriverName	Nom du pilote de stockage	« gcp-cvs »
backendName	Nom personnalisé ou système back-end de stockage	Nom du pilote + " _ " + partie de la clé API
storageClass	Paramètre facultatif utilisé pour spécifier le type de service CVS. Utilisez <code>software</code> pour sélectionner le type de service CVS. Sinon, Trident suppose le type de service CVS-Performance ( <code>hardware</code> ).	
storagePools	Type de service CVS uniquement. Paramètre facultatif utilisé pour spécifier les pools de stockage pour la création du volume.	

Paramètre	Description	Valeur par défaut
projectNumber	Numéro de projet de compte Google Cloud. La valeur est disponible sur la page d'accueil du portail Google Cloud.	
hostProjectNumber	Requis si l'utilisation d'un réseau VPC partagé. Dans ce scénario, <code>projectNumber</code> est le projet de service, et <code>hostProjectNumber</code> est le projet hôte.	
apiRegion	Région Google Cloud dans laquelle Trident crée des volumes Cloud Volumes Service. Lors de la création de clusters Kubernetes répartis entre régions, les volumes créés dans un <code>apiRegion</code> peuvent être utilisés dans des workloads planifiés sur des nœuds répartis sur plusieurs régions Google Cloud. Le trafic entre les régions coûte plus cher.	
apiKey	Clé d'API pour le compte de service Google Cloud avec le <code>netappcloudvolumes.admin</code> rôle. Il inclut le contenu au format JSON du fichier de clé privée d'un compte de service Google Cloud (copié en compte dans le fichier de configuration back-end).	
proxyURL	URL proxy si le serveur proxy doit se connecter au compte CVS. Le serveur proxy peut être un proxy HTTP ou HTTPS. Pour un proxy HTTPS, la validation du certificat est ignorée pour permettre l'utilisation de certificats auto-signés dans le serveur proxy. Les serveurs proxy avec authentification activée ne sont pas pris en charge.	
nfsMountOptions	Contrôle précis des options de montage NFS.	« nfsvers=3 »
limitVolumeSize	Echec du provisionnement si la taille du volume demandé est supérieure à cette valeur.	« » (non appliqué par défaut)
serviceLevel	Niveau de service CVS-Performance ou CVS pour les nouveaux volumes. Les valeurs de performances CVS sont standard, , premium ou extreme. Les valeurs CVS sont standardsw ou zoneredundantstandardsw.	CVS-Performance par défaut est « standard ». CVS default est "standardsw".
network	Réseau Google Cloud utilisé pour les volumes Cloud Volumes Service.	« par défaut »
debugTraceFlags	Indicateurs de débogage à utiliser lors du dépannage. Exemple, <code>\{"api":false, "method":true\}</code> . Ne l'utilisez pas à moins que vous ne soyez en mesure de résoudre les problèmes et que vous ayez besoin d'un viddage détaillé des journaux.	nul

Paramètre	Description	Valeur par défaut
allowedTopologies	Pour activer l'accès interrégional, votre définition de classe de stockage pour allowedTopologies doit inclure toutes les régions. Par exemple : <ul style="list-style-type: none"> <li>- key: topology.kubernetes.io/region</li> <li>values:</li> <li>- us-east1</li> <li>- europe-west1</li> </ul>	

## Options de provisionnement de volumes

Vous pouvez contrôler le provisionnement du volume par défaut dans la defaults section du fichier de configuration.

Paramètre	Description	Valeur par défaut
exportRule	Règles d'exportation pour les nouveaux volumes. Doit être une liste séparée par des virgules d'une combinaison d'adresses IPv4 ou de sous-réseaux IPv4 en notation CIDR.	« 0.0.0.0/0 »
snapshotDir	Accès au .snapshot répertoire	« faux »
snapshotReserve	Pourcentage de volume réservé pour les snapshots	« » (Accepter CVS par défaut de 0)
size	La taille des nouveaux volumes. CVS-Performance minimum est de 100 Gio. CVS est au minimum de 1 Gio.	Le type de service CVS-Performance utilise par défaut « 100 Gio ». Le type de service CVS n'est pas défini par défaut mais nécessite au moins 1 Gio.

## Exemples de type de service CVS-Performance

Les exemples suivants fournissent des exemples de configuration pour le type de service CVS-Performance.

## Exemple 1 : configuration minimale

Il s'agit de la configuration back-end minimale avec le type de service CVS-Performance par défaut et le niveau de service « standard » par défaut.

```
---  
version: 1  
storageDriverName: gcp-cvs  
projectNumber: "012345678901"  
apiRegion: us-west2  
apiKey:  
  type: service_account  
  project_id: my-gcp-project  
  private_key_id: <id_value>  
  private_key: |  
    -----BEGIN PRIVATE KEY-----  
    <key_value>  
    -----END PRIVATE KEY-----  
  client_email: cloudvolumes-admin-sa@my-gcp-  
project.iam.gserviceaccount.com  
  client_id: "123456789012345678901"  
  auth_uri: https://accounts.google.com/o/oauth2/auth  
  token_uri: https://oauth2.googleapis.com/token  
  auth_provider_x509_cert_url:  
    https://www.googleapis.com/oauth2/v1/certs  
  client_x509_cert_url:  
    https://www.googleapis.com/robot/v1/metadata/x509/cloudvolumes-admin-  
sa%40my-gcp-project.iam.gserviceaccount.com
```

## Exemple 2 : configuration du niveau de service

Dans cet exemple, nous présentons les options de configuration du back-end, y compris les niveaux de service et les valeurs par défaut des volumes.

```
---  
version: 1  
storageDriverName: gcp-cvs  
projectNumber: '012345678901'  
apiRegion: us-west2  
apiKey:  
  type: service_account  
  project_id: my-gcp-project  
  private_key_id: "<id_value>"  
  private_key: |  
    -----BEGIN PRIVATE KEY-----  
    <key_value>  
    -----END PRIVATE KEY-----  
  client_email: cloudvolumes-admin-sa@my-gcp-  
project.iam.gserviceaccount.com  
  client_id: '123456789012345678901'  
  auth_uri: https://accounts.google.com/o/oauth2/auth  
  token_uri: https://oauth2.googleapis.com/token  
  auth_provider_x509_cert_url:  
    https://www.googleapis.com/oauth2/v1/certs  
  client_x509_cert_url:  
    https://www.googleapis.com/robot/v1/metadata/x509/cloudvolumes-admin-  
sa%40my-gcp-project.iam.gserviceaccount.com  
proxyURL: http://proxy-server-hostname/  
nfsMountOptions: vers=3,proto=tcp,timeo=600  
limitVolumeSize: 10Ti  
serviceLevel: premium  
defaults:  
  snapshotDir: 'true'  
  snapshotReserve: '5'  
  exportRule: 10.0.0.0/24,10.0.1.0/24,10.0.2.100  
  size: 5Ti
```

### Exemple 3 : configuration de pool virtuel

Cet exemple utilise `storage` pour configurer les pools virtuels et le `StorageClasses` qui les renvoie. Reportez-vous [Définitions des classes de stockage](#) à la pour savoir comment les classes de stockage ont été définies.

Ici, des valeurs par défaut spécifiques sont définies pour tous les pools virtuels, qui définissent le `snapshotReserve` à 5 % et le `exportRule` à 0.0.0.0/0. Les pools virtuels sont définis dans la `storage` section. Chaque pool virtuel individuel définit son propre pool `serviceLevel` et certains pools remplacent les valeurs par défaut. Des étiquettes de pool virtuel ont été utilisées pour différencier les pools en fonction de `performance` et `'protection'` de .

```
---
version: 1
storageDriverName: gcp-cvs
projectNumber: '012345678901'
apiRegion: us-west2
apiKey:
  type: service_account
  project_id: my-gcp-project
  private_key_id: "<id_value>"
  private_key: |
    -----BEGIN PRIVATE KEY-----
    <key_value>
    -----END PRIVATE KEY-----
client_email: cloudvolumes-admin-sa@my-gcp-
project.iam.gserviceaccount.com
client_id: '123456789012345678901'
auth_uri: https://accounts.google.com/o/oauth2/auth
token_uri: https://oauth2.googleapis.com/token
auth_provider_x509_cert_url:
  https://www.googleapis.com/oauth2/v1/certs
  client_x509_cert_url:
    https://www.googleapis.com/robot/v1/metadata/x509/cloudvolumes-admin-
sa%40my-gcp-project.iam.gserviceaccount.com
nfsMountOptions: vers=3,proto=tcp,timeo=600
defaults:
  snapshotReserve: '5'
  exportRule: 0.0.0.0/0
labels:
  cloud: gcp
region: us-west2
storage:
- labels:
    performance: extreme
    protection: extra
    serviceLevel: extreme
```

```

defaults:
  snapshotDir: 'true'
  snapshotReserve: '10'
  exportRule: 10.0.0.0/24
- labels:
    performance: extreme
    protection: standard
    serviceLevel: extreme
- labels:
    performance: premium
    protection: extra
    serviceLevel: premium
defaults:
  snapshotDir: 'true'
  snapshotReserve: '10'
- labels:
    performance: premium
    protection: standard
    serviceLevel: premium
- labels:
    performance: standard
    serviceLevel: standard

```

## Définitions des classes de stockage

Les définitions de classe de stockage suivantes s'appliquent à l'exemple de configuration de pool virtuel. A l'aide de `parameters.selector`, vous pouvez spécifier pour chaque classe de stockage le pool virtuel utilisé pour héberger un volume. Les aspects définis dans le pool sélectionné seront définis pour le volume.

## Exemple de classe de stockage

```
---  
apiVersion: storage.k8s.io/v1  
kind: StorageClass  
metadata:  
  name: cvs-extreme-extra-protection  
provisioner: csi.trident.netapp.io  
parameters:  
  selector: performance=extreme; protection=extra  
allowVolumeExpansion: true  
---  
apiVersion: storage.k8s.io/v1  
kind: StorageClass  
metadata:  
  name: cvs-extreme-standard-protection  
provisioner: csi.trident.netapp.io  
parameters:  
  selector: performance=premium; protection=standard  
allowVolumeExpansion: true  
---  
apiVersion: storage.k8s.io/v1  
kind: StorageClass  
metadata:  
  name: cvs-premium-extra-protection  
provisioner: csi.trident.netapp.io  
parameters:  
  selector: performance=premium; protection=extra  
allowVolumeExpansion: true  
---  
apiVersion: storage.k8s.io/v1  
kind: StorageClass  
metadata:  
  name: cvs-premium  
provisioner: csi.trident.netapp.io  
parameters:  
  selector: performance=premium; protection=standard  
allowVolumeExpansion: true  
---  
apiVersion: storage.k8s.io/v1  
kind: StorageClass  
metadata:  
  name: cvs-standard  
provisioner: csi.trident.netapp.io  
parameters:  
  selector: performance=standard
```

```
allowVolumeExpansion: true
---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: cvs-extra-protection
provisioner: csi.trident.netapp.io
parameters:
  selector: protection=extra
allowVolumeExpansion: true
```

- La première classe de stockage (`cvs-extreme-extra-protection`) est mappée sur le premier pool virtuel. Il s'agit du seul pool offrant des performances extrêmes avec une réserve Snapshot de 10 %.
- La dernière classe de stockage (`cvs-extra-protection`) fait appel à n'importe quel pool de stockage qui fournit une réserve de snapshots de 10 %. Trident détermine quel pool virtuel est sélectionné et veille à ce que les exigences de réserve d'instantanés soient respectées.

## Exemples de type de service CVS

Les exemples suivants fournissent des exemples de configuration pour le type de service CVS.

## Exemple 1 : configuration minimale

Il s'agit de la configuration dorsale minimale qui utilise `storageClass` pour spécifier le type de service CVS et le niveau de service par défaut `standardsw`.

```
---
version: 1
storageDriverName: gcp-cvs
projectNumber: '012345678901'
storageClass: software
apiRegion: us-east4
apiKey:
  type: service_account
  project_id: my-gcp-project
  private_key_id: "<id_value>"
  private_key: |
    -----BEGIN PRIVATE KEY-----
    <key_value>
    -----END PRIVATE KEY-----
client_email: cloudvolumes-admin-sa@my-gcp-
project.iam.gserviceaccount.com
client_id: '123456789012345678901'
auth_uri: https://accounts.google.com/o/oauth2/auth
token_uri: https://oauth2.googleapis.com/token
auth_provider_x509_cert_url:
https://www.googleapis.com/oauth2/v1/certs
client_x509_cert_url:
https://www.googleapis.com/robot/v1/metadata/x509/cloudvolumes-admin-
sa%40my-gcp-project.iam.gserviceaccount.com
serviceLevel: standardsw
```

## Exemple 2 : configuration du pool de stockage

Cet exemple de configuration back-end utilise `storagePools` pour configurer un pool de stockage.

```
---
version: 1
storageDriverName: gcp-cvs
backendName: gcp-std-so-with-pool
projectNumber: '531265380079'
apiRegion: europe-west1
apiKey:
  type: service_account
  project_id: cloud-native-data
  private_key_id: "<id_value>"
  private_key: |-
    -----BEGIN PRIVATE KEY-----
    <key_value>
    -----END PRIVATE KEY-----
client_email: cloudvolumes-admin-sa@cloud-native-
data.iam.gserviceaccount.com
client_id: '107071413297115343396'
auth_uri: https://accounts.google.com/o/oauth2/auth
token_uri: https://oauth2.googleapis.com/token
auth_provider_x509_cert_url:
https://www.googleapis.com/oauth2/v1/certs
  client_x509_cert_url:
https://www.googleapis.com/robot/v1/metadata/x509/cloudvolumes-admin-
sa%40cloud-native-data.iam.gserviceaccount.com
storageClass: software
zone: europe-west1-b
network: default
storagePools:
- 1bc7f380-3314-6005-45e9-c7dc8c2d7509
serviceLevel: Standardsw
```

## Et la suite ?

Après avoir créé le fichier de configuration backend, exécutez la commande suivante :

```
tridentctl create backend -f <backend-file>
```

Si la création du back-end échoue, la configuration du back-end est erronée. Vous pouvez afficher les journaux pour déterminer la cause en exécutant la commande suivante :

```
tridentctl logs
```

Après avoir identifié et corrigé le problème avec le fichier de configuration, vous pouvez exécuter de nouveau la commande `create`.

## Configurer un système NetApp HCI ou SolidFire backend

Découvrez comment créer et utiliser un back-end Element avec votre installation Trident.

### Détails du pilote d'élément

Trident fournit le `solidfire-san` pilote de stockage pour communiquer avec le cluster. Les modes d'accès pris en charge sont : `ReadWriteOnce` (RWO), `ReadOnlyMey` (ROX), `ReadWriteMaly` (RWX), `ReadWriteOncePod` (RWOP).

Le `solidfire-san` pilote de stockage prend en charge les modes `file` et `block` volume. Pour le `Filesystem` volumeMode, Trident crée un volume et un système de fichiers. Le type de système de fichiers est spécifié par la classe de stockage.

Conducteur	Protocole	Mode Volume	Modes d'accès pris en charge	Systèmes de fichiers pris en charge
solidfire-san	iSCSI	Bloc	RWO, ROX, RWX, RWOP	Aucun système de fichiers. Périphérique de bloc brut.
solidfire-san	iSCSI	Système de fichiers	RWO, RWOP	xfs, ext3 ext4

### Avant de commencer

Vous aurez besoin des éléments suivants avant de créer un back-end d'élément.

- Système de stockage pris en charge exécutant le logiciel Element.
- Identifiants de locataire ou administrateur de cluster NetApp HCI/SolidFire pouvant gérer les volumes
- Tous vos nœuds workers Kubernetes doivent avoir installé les outils iSCSI appropriés. Reportez-vous à la "[informations de préparation du nœud de travail](#)".

### Options de configuration du back-end

Voir le tableau suivant pour les options de configuration du back-end :

Paramètre	Description	Valeur par défaut
version		Toujours 1
storageDriverName	Nom du pilote de stockage	Toujours « SolidFire-san »

Paramètre	Description	Valeur par défaut
backendName	Nom personnalisé ou système back-end de stockage	Adresse IP « SolidFire_ » + stockage (iSCSI)
Endpoint	MVIP pour le cluster SolidFire avec les identifiants de locataire	
SVIP	Port et adresse IP de stockage (iSCSI)	
labels	Ensemble d'étiquettes arbitraires au format JSON à appliquer aux volumes.	« »
TenantName	Nom du locataire à utiliser (créé si introuvable)	
InitiatorIFace	Limitez le trafic iSCSI à une interface hôte spécifique	« par défaut »
UseCHAP	Utilisez CHAP pour authentifier iSCSI. Trident utilise CHAP.	vrai
AccessGroups	Liste des ID de groupes d'accès à utiliser	Recherche l'ID d'un groupe d'accès nommé « Trident »
Types	Spécifications de QoS	
limitVolumeSize	Echec du provisionnement si la taille du volume demandé est supérieure à cette valeur	« » (non appliqué par défaut)
debugTraceFlags	Indicateurs de débogage à utiliser lors du dépannage. Exemple, {"api":false, "method":true}	nul



Ne pas utiliser debugTraceFlags sauf si vous effectuez un dépannage et que vous avez besoin d'un vidage de journal détaillé.

### Exemple 1 : configuration back-end pour un solidfire-san pilote avec trois types de volume

Cet exemple montre un fichier back-end utilisant l'authentification CHAP et la modélisation de trois types de volumes avec des garanties de QoS spécifiques. Il est probable que vous définissiez ensuite les classes de stockage pour les consommer à l'aide du IOPS paramètre de classe de stockage.

```

---
version: 1
storageDriverName: solidfire-san
Endpoint: https://<user>:<password>@<mvip>/json-rpc/8.0
SVIP: <svip>:3260
TenantName: <tenant>
labels:
  k8scluster: dev1
  backend: dev1-element-cluster
UseCHAP: true
Types:
  - Type: Bronze
    Qos:
      minIOPS: 1000
      maxIOPS: 2000
      burstIOPS: 4000
  - Type: Silver
    Qos:
      minIOPS: 4000
      maxIOPS: 6000
      burstIOPS: 8000
  - Type: Gold
    Qos:
      minIOPS: 6000
      maxIOPS: 8000
      burstIOPS: 10000

```

## Exemple 2 : configuration du back-end et des classes de stockage pour le solidfire-san pilote avec pools virtuels

Cet exemple représente le fichier de définition du back-end configuré avec des pools virtuels ainsi que des classes de stockage qui les renvoient.

Trident copie les étiquettes présentes sur un pool de stockage vers la LUN de stockage back-end au moment du provisionnement. Pour plus de commodité, les administrateurs du stockage peuvent définir des étiquettes par pool virtuel et les volumes de groupe par étiquette.

Dans l'exemple de fichier de définition back-end illustré ci-dessous, des valeurs par défaut spécifiques sont définies pour tous les pools de stockage, qui définissent le type sur Silver. Les pools virtuels sont définis dans la storage section. Dans cet exemple, certains pools de stockage définissent leur propre type et certains d'entre eux remplacent les valeurs par défaut définies ci-dessus.

```

---
version: 1
storageDriverName: solidfire-san
Endpoint: https://<user>:<password>@<mvip>/json-rpc/8.0

```

```

SVIP: <svip>:3260
TenantName: <tenant>
UseCHAP: true
Types:
  - Type: Bronze
    Qos:
      minIOPS: 1000
      maxIOPS: 2000
      burstIOPS: 4000
  - Type: Silver
    Qos:
      minIOPS: 4000
      maxIOPS: 6000
      burstIOPS: 8000
  - Type: Gold
    Qos:
      minIOPS: 6000
      maxIOPS: 8000
      burstIOPS: 10000
type: Silver
labels:
  store: solidfire
  k8scluster: dev-1-cluster
region: us-east-1
storage:
  - labels:
      performance: gold
      cost: "4"
      zone: us-east-1a
      type: Gold
  - labels:
      performance: silver
      cost: "3"
      zone: us-east-1b
      type: Silver
  - labels:
      performance: bronze
      cost: "2"
      zone: us-east-1c
      type: Bronze
  - labels:
      performance: silver
      cost: "1"
      zone: us-east-1d

```

Les définitions de classe de stockage suivantes font référence aux pools virtuels ci-dessus. En utilisant ce

parameters.selector champ, chaque classe de stockage indique quel(s) pool(s) virtuel(s) peut(NT) être utilisé(s) pour héberger un volume. Les aspects définis dans le pool virtuel sélectionné seront définis pour le volume.

La première classe de stockage (`solidfire-gold-four`) est mappée sur le premier pool virtuel. Il s'agit de la seule piscine offrant des performances or avec un Volume Type QoS de Gold. La dernière classe de stockage (`solidfire-silver`) fait référence à n'importe quel pool de stockage offrant des performances Silver. Trident décide du pool virtuel sélectionné et s'assure que les besoins en stockage sont satisfait.

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: solidfire-gold-four
provisioner: csi.trident.netapp.io
parameters:
  selector: performance=gold; cost=4
  fsType: ext4

---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: solidfire-silver-three
provisioner: csi.trident.netapp.io
parameters:
  selector: performance=silver; cost=3
  fsType: ext4

---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: solidfire-bronze-two
provisioner: csi.trident.netapp.io
parameters:
  selector: performance=bronze; cost=2
  fsType: ext4

---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: solidfire-silver-one
provisioner: csi.trident.netapp.io
parameters:
  selector: performance=silver; cost=1
  fsType: ext4
```

```

---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: solidfire-silver
provisioner: csi.trident.netapp.io
parameters:
  selector: performance=silver
  fsType: ext4

```

## Trouvez plus d'informations

- ["Groupes d'accès de volume"](#)

## Pilotes SAN de ONTAP

### Présentation du pilote SAN ONTAP

Découvrez comment configurer un back-end ONTAP avec les pilotes ONTAP et SAN Cloud Volumes ONTAP.

### Détails du pilote SAN ONTAP

Trident fournit les pilotes de stockage SAN suivants pour communiquer avec le cluster ONTAP. Les modes d'accès pris en charge sont : *ReadWriteOnce* (RWO), *ReadOnlyMey* (ROX), *ReadWriteMaly* (RWX), *ReadWriteOncePod* (RWOP).

Conducteur	Protocole	Mode Volume	Modes d'accès pris en charge	Systèmes de fichiers pris en charge
ontap-san	ISCSI SCSI sur FC	Bloc	RWO, ROX, RWX, RWOP	Pas de système de fichiers, périphérique de bloc brut
ontap-san	ISCSI SCSI sur FC	Système de fichiers	RWO, RWOP ROX et RWX ne sont pas disponibles en mode de volume du système de fichiers.	xfs, ext3 ext4

Conducteur	Protocole	Mode Volume	Modes d'accès pris en charge	Systèmes de fichiers pris en charge
ontap-san	NVMe/TCP  Reportez-vous à la <a href="#">Autres considérations relatives au NVMe/TCP.</a>	Bloc	RWO, ROX, RWX, RWOP	Pas de système de fichiers, périphérique de bloc brut
ontap-san	NVMe/TCP  Reportez-vous à la <a href="#">Autres considérations relatives au NVMe/TCP.</a>	Système de fichiers	RWO, RWOP  ROX et RWX ne sont pas disponibles en mode de volume du système de fichiers.	xfs, ext3 ext4
ontap-san-economy	ISCSI	Bloc	RWO, ROX, RWX, RWOP	Pas de système de fichiers, périphérique de bloc brut
ontap-san-economy	ISCSI	Système de fichiers	RWO, RWOP  ROX et RWX ne sont pas disponibles en mode de volume du système de fichiers.	xfs, ext3 ext4

- ! • À utiliser ontap-san-economy uniquement si le nombre d'utilisations de volume persistant doit être supérieur à "[Limites de volume ONTAP prises en charge](#)".
- À utiliser ontap-nas-economy uniquement si le nombre d'utilisations de volume persistant doit être supérieur à et si "[Limites de volume ONTAP prises en charge](#)" le ontap-san-economy pilote ne peut pas être utilisé.
- N'utilisez pas cette option ontap-nas-economy si vous prévoyez d'avoir besoin de protection des données, de reprise après incident ou de mobilité.
- NetApp ne recommande pas l'utilisation de l'autogrow FlexVol dans tous les pilotes ONTAP, sauf ONTAP-san. Pour contourner ce problème, Trident prend en charge l'utilisation de la réserve Snapshot et adapte les volumes FlexVol en conséquence.

#### Autorisations utilisateur

Trident s'attend à être exécuté en tant qu'administrateur ONTAP ou SVM, en général avec l'utilisateur du cluster ou un vsadmin utilisateur SVM, ou en tant qu' admin `utilisateur avec un nom différent et le même rôle. Pour les déploiements Amazon FSX pour NetApp ONTAP, Trident

prévoit d'être exécuté en tant qu'administrateur ONTAP ou SVM, en utilisant l'utilisateur du cluster `fsxadmin` ou un `vsadmin` utilisateur SVM, ou un utilisateur avec un nom différent ayant le même rôle. `fsxadmin` L'utilisateur est un remplaçant limité pour l'utilisateur admin du cluster.

 Si vous utilisez le `limitAggregateUsage` paramètre, les autorisations d'administration du cluster sont requises. Lors de l'utilisation d'Amazon FSX for NetApp ONTAP avec Trident, le `limitAggregateUsage` paramètre ne fonctionnera pas avec les `vsadmin` comptes d'utilisateur et `fsxadmin`. L'opération de configuration échoue si vous spécifiez ce paramètre.

S'il est possible de créer au sein de ONTAP un rôle plus restrictif qu'un pilote Trident peut utiliser, nous ne le recommandons pas. La plupart des nouvelles versions de Trident appellent des API supplémentaires qui devront être prises en compte, ce qui complique les mises à niveau et risque d'erreurs.

#### Autres considérations relatives au NVMe/TCP

Trident prend en charge le protocole NVMe (non-volatile Memory Express) avec le `ontap-san` pilote, notamment :

- IPv6
- Copies Snapshot et clones de volumes NVMe
- Redimensionnement d'un volume NVMe
- Importation d'un volume NVMe créé en dehors de Trident afin que son cycle de vie puisse être géré par Trident
- Chemins d'accès multiples natifs NVMe
- Arrêt normal ou sans gracieuse des nœuds K8s (24.06)

Trident ne prend pas en charge :

- DH-HMAC-CHAP pris en charge par NVMe de manière native
- Chemins d'accès multiples du mappeur de périphériques (DM)
- Cryptage LUKS

#### Préparez la configuration du système back-end avec les pilotes SAN ONTAP

Découvrez les exigences et les options d'authentification pour la configuration d'un back-end ONTAP avec des pilotes SAN ONTAP.

##### De formation

Pour tous les backends ONTAP, Trident exige qu'au moins un agrégat soit attribué au SVM.

Reportez-vous à cet article de la base de connaissances pour savoir comment attribuer des agrégats à SVM dans les systèmes ASA r2 : "La création d'une unité de stockage par l'administrateur SVM à l'aide de l'interface de ligne de commande échoue avec l'erreur « Aucun agrégat candidat n'est disponible pour les services de stockage »".

N'oubliez pas que vous pouvez également exécuter plusieurs pilotes et créer des classes de stockage qui pointent vers l'un ou l'autre. Par exemple, vous pouvez configurer une `san-dev` classe qui utilise le `ontap-san` pilote et une `san-default` classe qui utilise celui-ci `ontap-san-economy`.

Tous vos nœuds workers Kubernetes doivent avoir installé les outils iSCSI appropriés. Voir "["Préparez le nœud de travail"](#)" pour plus de détails.

### Authentifiez le back-end ONTAP

Trident propose deux modes d'authentification d'un back-end ONTAP.

- Basé sur les informations d'identification : nom d'utilisateur et mot de passe pour un utilisateur ONTAP disposant des autorisations requises. Il est recommandé d'utiliser un rôle de connexion de sécurité prédéfini, tel que `admin` ou `vsadmin` pour garantir une compatibilité maximale avec les versions de ONTAP.
- Basé sur un certificat : Trident peut également communiquer avec un cluster ONTAP à l'aide d'un certificat installé sur le back-end. Dans ce cas, la définition `backend` doit contenir des valeurs encodées Base64 du certificat client, de la clé et du certificat d'autorité de certification de confiance, le cas échéant (recommandé).

Vous pouvez mettre à jour les systèmes back-end existants pour passer d'une méthode basée sur les identifiants à une méthode basée sur les certificats. Toutefois, une seule méthode d'authentification est prise en charge à la fois. Pour passer à une méthode d'authentification différente, vous devez supprimer la méthode existante de la configuration `backend`.



Si vous tentez de fournir **les deux identifiants et les certificats**, la création du back-end échoue avec une erreur indiquant que plus d'une méthode d'authentification a été fournie dans le fichier de configuration.

### Activer l'authentification basée sur les informations d'identification

Trident exige que les identifiants d'un administrateur SVM-scoped/cluster-scoped communiquent avec le back-end ONTAP. Il est recommandé d'utiliser des rôles standard prédéfinis tels que `admin` ou `vsadmin`. La compatibilité avec les futures versions d'ONTAP qui exposent les API de fonctionnalités à utiliser dans les futures versions d'Trident est ainsi garantie. Un rôle de connexion de sécurité personnalisé peut être créé et utilisé avec Trident, mais il n'est pas recommandé.

Voici un exemple de définition du back-end :

## YAML

```
---
```

```
version: 1
backendName: ExampleBackend
storageDriverName: ontap-san
managementLIF: 10.0.0.1
svm: svm_nfs
username: vsadmin
password: password
```

## JSON

```
{
  "version": 1,
  "backendName": "ExampleBackend",
  "storageDriverName": "ontap-san",
  "managementLIF": "10.0.0.1",
  "svm": "svm_nfs",
  "username": "vsadmin",
  "password": "password"
}
```

Gardez à l'esprit que la définition du back-end est le seul endroit où les informations d'identification sont stockées en texte brut. Une fois le système backend créé, les noms d'utilisateur/mots de passe sont codés avec Base64 et stockés sous forme de secrets Kubernetes. La création ou la mise à jour d'un back-end est la seule étape qui nécessite la connaissance des informations d'identification. Il s'agit donc d'une opération uniquement administrative, qui doit être effectuée par l'administrateur Kubernetes/du stockage.

### Activer l'authentification basée sur certificat

Les systèmes back-end, nouveaux et existants, peuvent utiliser un certificat et communiquer avec le système back-end ONTAP. Trois paramètres sont requis dans la définition du back-end.

- ClientCertificate : valeur encodée en Base64 du certificat client.
- ClientPrivateKey : valeur encodée en Base64 de la clé privée associée.
- TrustedCACertificate : valeur encodée Base64 du certificat CA de confiance. Si vous utilisez une autorité de certification approuvée, ce paramètre doit être fourni. Ceci peut être ignoré si aucune autorité de certification approuvée n'est utilisée.

Un flux de travail type comprend les étapes suivantes.

### Étapes

1. Générez un certificat client et une clé. Lors de la génération, définissez le nom commun (CN) sur l'utilisateur ONTAP pour qu'il s'authentifie.

```
openssl req -x509 -nodes -days 1095 -newkey rsa:2048 -keyout k8senv.key  
-out k8senv.pem -subj "/C=US/ST=NC/L=RTP/O=NetApp/CN=admin"
```

- Ajoutez un certificat d'autorité de certification de confiance au cluster ONTAP. Il se peut déjà que l'administrateur de stockage gère cet espace. Ignorer si aucune autorité de certification approuvée n'est utilisée.

```
security certificate install -type server -cert-name <trusted-ca-cert-name> -vserver <vserver-name>  
ssl modify -vserver <vserver-name> -server-enabled true -client-enabled true -common-name <common-name> -serial <SN-from-trusted-CA-cert> -ca <cert-authority>
```

- Installez le certificat client et la clé (à partir de l'étape 1) sur le cluster ONTAP.

```
security certificate install -type client-ca -cert-name <certificate-name> -vserver <vserver-name>  
security ssl modify -vserver <vserver-name> -client-enabled true
```

- Vérifiez que le rôle de connexion de sécurité ONTAP prend en charge cert la méthode d'authentification.

```
security login create -user-or-group-name admin -application ontapi  
-authentication-method cert  
security login create -user-or-group-name admin -application http  
-authentication-method cert
```

- Testez l'authentification à l'aide d'un certificat généré. Remplacer <ONTAP Management LIF> et <vserver name> par Management LIF IP et SVM name.

```
curl -X POST -Lk https://<ONTAP-Management-LIF>/servlets/netapp.servlets.admin.XMLrequest_filer --key k8senv.key  
--cert ~/k8senv.pem -d '<?xml version="1.0" encoding="UTF-8"?><netapp  
xmlns="http://www.netapp.com/filer/admin" version="1.21"  
vfiler="<vserver-name>"><vserver-get></vserver-get></netapp>'
```

- Encodez le certificat, la clé et le certificat CA de confiance avec Base64.

```
base64 -w 0 k8senv.pem >> cert_base64  
base64 -w 0 k8senv.key >> key_base64  
base64 -w 0 trustedca.pem >> trustedca_base64
```

7. Créez le back-end à l'aide des valeurs obtenues à partir de l'étape précédente.

```
cat cert-backend.json
{
  "version": 1,
  "storageDriverName": "ontap-san",
  "backendName": "SanBackend",
  "managementLIF": "1.2.3.4",
  "svm": "vserver_test",
  "clientCertificate": "Faaaakkkeeee...Vaaallluuuueeee",
  "clientPrivateKey": "LS0tFaKE...0VaLuES0tLS0K",
  "trustedCACertificate": "QNFinfO...SiqOyN",
  "storagePrefix": "myPrefix_"
}

tridentctl create backend -f cert-backend.json -n trident
+-----+-----+
+-----+-----+
|      NAME      | STORAGE DRIVER |          UUID          |
STATE | VOLUMES |
+-----+-----+
+-----+-----+
| SanBackend | ontap-san     | 586b1cd5-8cf8-428d-a76c-2872713612c1 |
online |           0 |
+-----+-----+
+-----+-----+
```

### Mettre à jour les méthodes d'authentification ou faire pivoter les informations d'identification

Vous pouvez mettre à jour un back-end existant pour utiliser une méthode d'authentification différente ou pour faire pivoter leurs informations d'identification. Cela fonctionne de deux manières : les systèmes back-end qui utilisent le nom d'utilisateur/mot de passe peuvent être mis à jour pour utiliser des certificats ; les systèmes back-end qui utilisent des certificats peuvent être mis à jour en fonction du nom d'utilisateur/mot de passe. Pour ce faire, vous devez supprimer la méthode d'authentification existante et ajouter la nouvelle méthode d'authentification. Utilisez ensuite le fichier backend.json mis à jour contenant les paramètres requis pour exécuter `tridentctl backend update`.

```

cat cert-backend-updated.json
{
  "version": 1,
  "storageDriverName": "ontap-san",
  "backendName": "SanBackend",
  "managementLIF": "1.2.3.4",
  "svm": "vserver_test",
  "username": "vsadmin",
  "password": "password",
  "storagePrefix": "myPrefix_"
}

#Update backend with tridentctl
tridentctl update backend SanBackend -f cert-backend-updated.json -n
trident
+-----+-----+-----+
+-----+-----+
|     NAME      | STORAGE DRIVER |                         UUID          |
STATE | VOLUMES |
+-----+-----+-----+
+-----+-----+
| SanBackend | ontap-san       | 586b1cd5-8cf8-428d-a76c-2872713612c1 |
online |         9 |           |
+-----+-----+-----+
+-----+-----+

```

i Lors de la rotation des mots de passe, l'administrateur du stockage doit d'abord mettre à jour le mot de passe de l'utilisateur sur ONTAP. Cette opération est suivie d'une mise à jour du back-end. Lors de la rotation de certificats, plusieurs certificats peuvent être ajoutés à l'utilisateur. Le back-end est ensuite mis à jour pour utiliser le nouveau certificat, en suivant lequel l'ancien certificat peut être supprimé du cluster ONTAP.

La mise à jour d'un back-end n'interrompt pas l'accès aux volumes qui ont déjà été créés, et n'a aucun impact sur les connexions de volume effectuées après. Une mise à jour back-end réussie indique que Trident peut communiquer avec le back-end ONTAP et gérer les futures opérations de volume.

## Créez un rôle ONTAP personnalisé pour Trident

Vous pouvez créer un rôle de cluster ONTAP avec une Privileges minimale afin de ne pas avoir à utiliser le rôle ONTAP admin pour effectuer des opérations dans Trident. Lorsque vous incluez le nom d'utilisateur dans une configuration Trident backend, Trident utilise le rôle de cluster ONTAP que vous avez créé pour effectuer les opérations.

Pour plus d'informations sur la création de rôles personnalisés Trident, reportez-vous à la section "[Générateur de rôle personnalisé Trident](#)".

## Utilisation de l'interface de ligne de commandes ONTAP

1. Créez un rôle à l'aide de la commande suivante :

```
security login role create <role_name> -cmddirname "command" -access all  
-vserver <svm_name>
```

2. Créez un nom d'utilisateur pour l'utilisateur Trident :

```
security login create -username <user_name> -application ontapi  
-authmethod <password> -role <name_of_role_in_step_1> -vserver  
<svm_name> -comment "user_description"
```

3. Mapper le rôle à l'utilisateur :

```
security login modify username <user_name> -vserver <svm_name> -role  
<role_name> -application ontapi -application console -authmethod  
<password>
```

## À l'aide de System Manager

Dans ONTAP System Manager, effectuez les opérations suivantes :

1. **Créer un rôle personnalisé :**

- Pour créer un rôle personnalisé au niveau du cluster, sélectionnez **Cluster > Paramètres**.  
(Ou) pour créer un rôle personnalisé au niveau du SVM, sélectionner **stockage > Storage VM > >> Paramètres > required SVM utilisateurs et rôles**.
- Sélectionnez l'icône de flèche (→) en regard de **utilisateurs et rôles**.
- Sélectionnez **+Ajouter sous rôles**.
- Définissez les règles du rôle et cliquez sur **Enregistrer**.

2. **Mapper le rôle à l'utilisateur Trident:** + effectuez les étapes suivantes sur la page **utilisateurs et rôles** :

- Sélectionnez Ajouter l'icône **+** sous **utilisateurs**.
- Sélectionnez le nom d'utilisateur requis et sélectionnez un rôle dans le menu déroulant pour **role**.
- Cliquez sur **Enregistrer**.

Pour plus d'informations, reportez-vous aux pages suivantes :

- "Rôles personnalisés pour l'administration de ONTAP" ou "Définissez des rôles personnalisés"
- "Travaillez avec les rôles et les utilisateurs"

## Authentifier les connexions avec le protocole CHAP bidirectionnel

Trident peut authentifier les sessions iSCSI avec le protocole CHAP bidirectionnel pour les `ontap-san` pilotes et `ontap-san-economy`. Pour ce faire, vous devez activer `useCHAP` l'option dans votre définition de back-end. Lorsque ce paramètre est défini sur `true`, Trident configure la sécurité initiateur par défaut du SVM sur CHAP bidirectionnel et définit le nom d'utilisateur et les secrets à partir du fichier back-end. NetApp recommande d'utiliser le protocole CHAP bidirectionnel pour l'authentification des connexions. Voir l'exemple

de configuration suivant :

```
---  
version: 1  
storageDriverName: ontap-san  
backendName: ontap_san_chap  
managementLIF: 192.168.0.135  
svm: ontap_iscsi_svm  
useCHAP: true  
username: vsadmin  
password: password  
chapInitiatorSecret: cl9qxIm36DKyawxy  
chapTargetInitiatorSecret: rxqigXgkesIpwxyz  
chapTargetUsername: iJF4heBRT0TCwxyz  
chapUsername: uh2aNCLSd6cNwxyz
```

 Le `useCHAP` paramètre est une option booléenne qui ne peut être configurée qu'une seule fois. Elle est définie sur `FALSE` par défaut. Une fois la valeur `true` définie, vous ne pouvez pas la définir sur `false`.

En plus `useCHAP=true` de , les `chapInitiatorSecret` champs , `chapTargetInitiatorSecret`, `chapTargetUsername` et `chapUsername` doivent être inclus dans la définition du back-end. Les secrets peuvent être modifiés après la création d'un backend en exécutant `tridentctl update`.

## Comment cela fonctionne

En définissant la `useCHAP` valeur sur `true`, l'administrateur du stockage demande à Trident de configurer CHAP sur le back-end de stockage. Ceci inclut les éléments suivants :

- Configuration du protocole CHAP sur le SVM :
  - Si le type de sécurité initiateur par défaut du SVM est `none` (défini par défaut) **et** il n'y a pas de LUN préexistantes déjà présentes dans le volume, Trident définit le type de sécurité par défaut sur `CHAP` et passe à la configuration de l'initiateur CHAP et du nom d'utilisateur et des secrets cible.
  - Si le SVM contient des LUN, Trident n'activera pas CHAP sur le SVM. Cela garantit que l'accès aux LUNs déjà présentes sur le SVM n'est pas restreint.
- Configuration de l'initiateur CHAP et du nom d'utilisateur cible et des secrets ; ces options doivent être spécifiées dans la configuration `backend` (comme indiqué ci-dessus).

Une fois le back-end créé, Trident crée un code CRD correspondant `tridentbackend` et stocke les secrets CHAP et les noms d'utilisateur comme secrets Kubernetes. Tous les volumes persistants créés par Trident sur ce back-end seront montés et rattachés via CHAP.

## Rotation des identifiants et mise à jour des systèmes back-end

Vous pouvez mettre à jour les informations d'identification CHAP en mettant à jour les paramètres CHAP dans `backend.json` le fichier. Ceci nécessitera la mise à jour des secrets CHAP et l'utilisation de la `tridentctl update` commande pour refléter ces modifications.



Lors de la mise à jour des secrets CHAP pour un backend, vous devez utiliser `tridentctl` pour mettre à jour le backend. Ne mettez pas à jour les informations d'identification sur le cluster de stockage via l'interface de ligne de commande ONTAP ou ONTAP System Manager, car Trident ne pourra pas récupérer ces modifications.

```
cat backend-san.json
{
    "version": 1,
    "storageDriverName": "ontap-san",
    "backendName": "ontap_san_chap",
    "managementLIF": "192.168.0.135",
    "svm": "ontap_iscsi_svm",
    "useCHAP": true,
    "username": "vsadmin",
    "password": "password",
    "chapInitiatorSecret": "c19qxUpDaTeD",
    "chapTargetInitiatorSecret": "rqxigXgkeUpDaTeD",
    "chapTargetUsername": "iJF4heBRT0TCwxyz",
    "chapUsername": "uh2aNCLSd6cNwxyz",
}

./tridentctl update backend ontap_san_chap -f backend-san.json -n trident
+-----+-----+
+-----+-----+
|     NAME          | STORAGE DRIVER |           UUID           |
STATE | VOLUMES |
+-----+-----+
+-----+-----+
| ontap_san_chap | ontap-san      | aa458f3b-ad2d-4378-8a33-1a472ffbeb5c |
online |       7 |
+-----+-----+
+-----+-----+
```

Les connexions existantes ne seront pas affectées ; elles continueront à rester actives si les informations d'identification sont mises à jour par Trident sur le SVM. Les nouvelles connexions utilisent les informations d'identification mises à jour et les connexions existantes restent actives. La déconnexion et la reconnexion des anciens volumes persistants se traduiront par l'utilisation des identifiants mis à jour.

### Options et exemples de configuration du SAN ONTAP

Découvrez comment créer et utiliser les pilotes SAN ONTAP avec votre installation Trident. Cette section fournit des exemples de configuration back-end et des détails sur le mappage des systèmes back-end aux classes de stockage.

## Options de configuration du back-end

Voir le tableau suivant pour les options de configuration du back-end :

Paramètre	Description	Valeur par défaut
version		Toujours 1
storageDriveName	Nom du pilote de stockage	ontap-san ou ontap-san-economy
backendName	Nom personnalisé ou système back-end de stockage	Nom du pilote + " " + dataLIF
managementLIF	<p>Adresse IP d'un cluster ou d'une LIF de management du SVM.</p> <p>Un nom de domaine complet (FQDN) peut être spécifié.</p> <p>Peut être configuré pour utiliser des adresses IPv6 si Trident a été installé à l'aide de l'indicateur IPv6. Les adresses IPv6 doivent être définies entre crochets, par exemple [28e8:d9fb:a825:b7bf:69a8:d02f:9e7b:3555].</p> <p>Pour un basculement MetroCluster transparent, consultez le <a href="#">Exemple MetroCluster</a>.</p> <p> Si vous utilisez des identifiants « vsadmin », doit être celui du SVM ; si vous managementLIF utilisez des identifiants « admin », managementLIF doit être celui du cluster.</p>	
dataLIF	<p>Adresse IP de la LIF de protocole. Peut être configuré pour utiliser des adresses IPv6 si Trident a été installé à l'aide de l'indicateur IPv6. Les adresses IPv6 doivent être définies entre crochets, par exemple [28e8:d9fb:a825:b7bf:69a8:d02f:9e7b:3555]. <b>Ne spécifiez pas pour iSCSI.</b> Trident utilise "<a href="#">Mappage de LUN sélectif ONTAP</a>" pour détecter les LIFs iSCSI nécessaires pour établir une session multi-chemins. Un avertissement est généré si dataLIF est explicitement défini. <b>Omettre pour MetroCluster.</b> Voir la <a href="#">Exemple MetroCluster</a>.</p>	Dérivé par la SVM
svm	Machine virtuelle de stockage à utiliser <b>omet pour MetroCluster.</b> Voir la <a href="#">Exemple MetroCluster</a> .	Dérivé si un SVM managementLIF est spécifié

Paramètre	Description	Valeur par défaut
useCHAP	Utilisez CHAP pour authentifier iSCSI pour les pilotes SAN ONTAP [Boolean]. Set to <code>true</code> for Trident to configure and use bidirectionnelle CHAP as the default Authentication for the SVM donné au backend. Voir " <a href="#">"Préparez la configuration du système backend avec les pilotes SAN ONTAP"</a> " pour plus de détails.	false
chapInitiatorSecret	Secret de l'initiateur CHAP. Requis si <code>useCHAP=true</code>	« »
labels	Ensemble d'étiquettes arbitraires au format JSON à appliquer aux volumes	« »
chapTargetInitiatorSecret	Secret de l'initiateur cible CHAP. Requis si <code>useCHAP=true</code>	« »
chapUsername	Nom d'utilisateur entrant. Requis si <code>useCHAP=true</code>	« »
chapTargetUsername	Nom d'utilisateur cible. Requis si <code>useCHAP=true</code>	« »
clientCertificate	Valeur encodée en Base64 du certificat client. Utilisé pour l'authentification par certificat	« »
clientPrivateKey	Valeur encodée en Base64 de la clé privée du client. Utilisé pour l'authentification par certificat	« »
trustedCACertificate	Valeur encodée en Base64 du certificat CA de confiance. Facultatif. Utilisé pour l'authentification basée sur des certificats.	« »
username	Le nom d'utilisateur devait communiquer avec le cluster ONTAP. Utilisé pour l'authentification basée sur les identifiants.	« »
password	Mot de passe requis pour communiquer avec le cluster ONTAP. Utilisé pour l'authentification basée sur les identifiants.	« »
svm	Serveur virtuel de stockage à utiliser	Dérivé si un SVM managementLIF est spécifié
storagePrefix	Préfixe utilisé pour le provisionnement des nouveaux volumes dans la SVM. Ne peut pas être modifié ultérieurement. Pour mettre à jour ce paramètre, vous devez créer un nouveau backend.	trident

Paramètre	Description	Valeur par défaut
aggregate	<p>Agrégat pour le provisionnement (facultatif ; si défini, doit être attribué au SVM) Pour le ontap-nas-flexgroup pilote, cette option est ignorée. S'ils ne sont pas affectés, les agrégats disponibles peuvent être utilisés pour provisionner un volume FlexGroup.</p> <p> Lorsque l'agrégat est mis à jour au SVM, il est mis à jour automatiquement dans Trident par SVM d'interrogation sans avoir à redémarrer le contrôleur Trident. Lorsque vous avez configuré un agrégat spécifique dans Trident pour provisionner des volumes, si l'agrégat est renommé ou déplacé hors du SVM, le back-end passe à l'état Failed dans Trident lors de l'interrogation de l'agrégat du SVM. Il faut remplacer l'agrégat par un agrégat présent sur la SVM ou le retirer complètement pour remettre le back-end en ligne.</p> <p><b>Ne spécifiez pas pour ASA r2.</b></p>	« »
limitAggregateUsage	Echec du provisionnement si l'utilisation est supérieure à ce pourcentage. Si vous utilisez un backend Amazon FSX for NetApp ONTAP, ne spécifiez pas limitAggregateUsage. Les fournies fsxadmin et vsadmin ne contiennent pas les autorisations requises pour récupérer l'utilisation des agrégats et la limiter à l'aide de Trident. <b>Ne spécifiez pas pour ASA r2.</b>	« » (non appliqué par défaut)
limitVolumeSize	Echec du provisionnement si la taille du volume demandé est supérieure à cette valeur. Limite également la taille maximale des volumes qu'il gère pour les LUN.	« » (non appliqué par défaut)
lunsPerFlexvol	Nombre maximal de LUN par FlexVol, doit être compris dans la plage [50, 200]	100
debugTraceFlags	Indicateurs de débogage à utiliser lors du dépannage. Exemple, {"api":false, "method":true} n'utilisez que si vous effectuez un dépannage et que vous avez besoin d'un vidage de journal détaillé.	null

Paramètre	Description	Valeur par défaut
useREST	<p>Paramètre booléen pour utiliser les API REST de ONTAP.</p> <p>useREST Lorsqu'il est défini sur <code>true</code>, Trident utilise les API REST ONTAP pour communiquer avec le back-end ; lorsqu'il est défini sur <code>false</code>, Trident utilise les appels ONTAPI (ZAPI) pour communiquer avec le back-end. Cette fonctionnalité requiert ONTAP 9.11.1 et versions ultérieures. En outre, le rôle de connexion ONTAP utilisé doit avoir accès à l' <code>ontapi</code> application. Ceci est satisfait par les rôles et prédefinis <code>vsadmin</code> <code>cluster-admin</code>. À partir de la version Trident 24.06 et de ONTAP 9.15.1 ou version ultérieure,</p> <p>useREST est défini sur <code>true</code> par défaut ; passez useREST à <code>false</code> pour utiliser les appels ONTAPI (ZAPI).</p> <p>useREST Est pleinement qualifié pour NVMe/TCP. <b>Si spécifié, toujours défini sur true pour ASA r2.</b></p>	true Pour ONTAP 9.15.1 ou version ultérieure, sinon <code>false</code> .
sanType	Utilisez pour sélectionner <code>iscsi</code> pour iSCSI, <code>nvme</code> pour NVMe/TCP ou <code>fcp</code> pour SCSI over Fibre Channel (FC).	<code>iscsi</code> si vide
formatOptions	<p>Utilisez <code>formatOptions</code> pour spécifier des arguments de ligne de commande pour la <code>mkfs</code> commande, qui seront appliqués chaque fois qu'un volume est formaté. Vous pouvez ainsi formater le volume en fonction de vos préférences. Assurez-vous de spécifier les options de formatage similaires à celles des options de commande <code>mkfs</code>, à l'exception du chemin du périphérique. Exemple : « <code>-E nojeter</code> »</p> <p><b>Pris en charge pour ontap-san les ontap-san-economy pilotes et uniquement.</b></p>	
limitVolumePoolSize	Taille maximale des FlexVol pouvant être demandées lors de l'utilisation de LUN dans le back-end ONTAP-san Economy.	« » (non appliqué par défaut)
denyNewVolumePools	Limite les <code>ontap-san-economy</code> systèmes back-end à la création de nouveaux volumes FlexVol afin qu'ils contiennent leurs LUN. Seuls les volumes FlexVol préexistants sont utilisés pour provisionner les nouveaux volumes persistants.	

### Recommandations pour l'utilisation des options de format

Trident recommande l'option suivante pour accélérer le processus de formatage :

#### **-E nojeter:**

- Conservez, n'essayez pas de supprimer des blocs au moment `mkfs` (la suppression initiale des blocs est

utile sur les périphériques SSD et le stockage fragmenté/à provisionnement fin). Ceci remplace l'option obsolète "-K" et il est applicable à tous les systèmes de fichiers (xfs, ext3 et ext4).

#### Options de configuration back-end pour les volumes de provisionnement

Vous pouvez contrôler le provisionnement par défaut à l'aide de ces options dans la `defaults` section de la configuration. Pour un exemple, voir les exemples de configuration ci-dessous.

Paramètre	Description	Valeur par défaut
spaceAllocation	Allocation d'espace pour les LUN	"Vrai" si spécifié, définissez sur <b>true</b> pour ASA r2.
spaceReserve	Mode de réservation d'espace ; « aucun » (fin) ou « volume » (épais). <b>Défini sur none pour ASA r2.</b>	« aucun »
snapshotPolicy	Règle Snapshot à utiliser. <b>Défini sur none pour ASA r2.</b>	« aucun »
qosPolicy	QoS policy group à affecter pour les volumes créés. Choisissez une de qosPolicy ou adaptiveQosPolicy par pool de stockage/back-end. L'utilisation de groupes de règles de qualité de service avec Trident nécessite ONTAP 9.8 ou une version ultérieure. Vous devez utiliser un groupe de règles QoS non partagé et vous assurer que le groupe de règles est appliqué à chaque composant individuellement. Un groupe de règles de QoS partagées applique le débit total de toutes les charges de travail.	« »
adaptiveQosPolicy	Groupe de règles de QoS adaptative à attribuer aux volumes créés. Choisissez une de qosPolicy ou adaptiveQosPolicy par pool de stockage/back-end	« »
snapshotReserve	Pourcentage du volume réservé pour les snapshots. <b>Ne spécifiez pas pour ASA r2.</b>	« 0 » si <code>snapshotPolicy</code> est « aucun », sinon « »
splitOnClone	Séparer un clone de son parent lors de sa création	« faux »
encryption	Activez le chiffrement de volume NetApp (NVE) sur le nouveau volume. La valeur par défaut est <code>false</code> . Pour utiliser cette option, NVE doit être sous licence et activé sur le cluster. Si NAE est activé sur le back-end, tout volume provisionné dans Trident est activé. Pour plus d'informations, reportez-vous à la section : " <a href="#">Fonctionnement de Trident avec NVE et NAE</a> ".	"FALSE" si spécifié, définissez sur <b>true</b> pour ASA r2.
luksEncryption	Activez le cryptage LUKS. Reportez-vous à la " <a href="#">Utiliser la configuration de clé unifiée Linux (LUKS)</a> ".	« » Défini sur <code>false</code> pour ASA r2.
tieringPolicy	Stratégie de hiérarchisation pour utiliser "none" <b>ne spécifiez pas pour ASA r2.</b>	
nameTemplate	Modèle pour créer des noms de volume personnalisés.	« »

## Exemples de provisionnement de volumes

Voici un exemple avec des valeurs par défaut définies :

```
---
version: 1
storageDriverName: ontap-san
managementLIF: 10.0.0.1
svm: trident_svm
username: admin
password: <password>
labels:
  k8scluster: dev2
  backend: dev2-sanbackend
storagePrefix: alternate-trident
debugTraceFlags:
  api: false
  method: true
defaults:
  spaceReserve: volume
  qosPolicy: standard
  spaceAllocation: 'false'
  snapshotPolicy: default
  snapshotReserve: '10'
```

 Pour tous les volumes créés à l'aide du `ontap-san` pilote, Trident ajoute 10 % de capacité supplémentaire au FlexVol pour prendre en charge les métadonnées des LUN. La LUN sera provisionnée avec la taille exacte que l'utilisateur demande dans la demande de volume persistant. Trident ajoute 10 % au FlexVol (s'affiche en tant que taille disponible dans ONTAP). Les utilisateurs obtiennent à présent la capacité utilisable requise. Cette modification empêche également que les LUN ne soient en lecture seule, à moins que l'espace disponible soit pleinement utilisé. Cela ne s'applique pas à l'économie d'`ontap-san`.

Pour les systèmes back-end définis par `snapshotReserve`, Trident calcule la taille des volumes comme suit :

```
Total volume size = [(PVC requested size) / (1 - (snapshotReserve percentage) / 100)] * 1.1
```

Le modèle 1.1 représente les 10 % supplémentaires de Trident ajoutés au FlexVol pour prendre en charge les métadonnées de LUN. Pour `snapshotReserve = 5 %` et `demande de volume persistant = 5 Gio`, la taille totale du volume est de 5,79 Gio et la taille disponible est de 5,5 Gio. `volume show` La commande doit afficher des résultats similaires à cet exemple :

Vserver	Volume	Aggregate	State	Type	Size	Available	Used%
	_pvc_89f1c156_3801_4de4_9f9d_034d54c395f4		online	RW	10GB	5.00GB	0%
	_pvc_e42ec6fe_3baa_4af6_996d_134adb8e6d		online	RW	5.79GB	5.50GB	0%
	_pvc_e8372153_9ad9_474a_951a_08ae15e1c0ba		online	RW	1GB	511.8MB	0%
3 entries were displayed.							

Actuellement, le redimensionnement est le seul moyen d'utiliser le nouveau calcul pour un volume existant.

#### Exemples de configuration minimaux

Les exemples suivants montrent des configurations de base qui laissent la plupart des paramètres par défaut. C'est la façon la plus simple de définir un back-end.



Si vous utilisez Amazon FSX on NetApp ONTAP avec Trident, NetApp vous recommande de spécifier des noms DNS pour les LIF au lieu d'adresses IP.

#### Exemple de SAN ONTAP

Il s'agit d'une configuration de base utilisant le `ontap-san` pilote.

```
---
version: 1
storageDriverName: ontap-san
managementLIF: 10.0.0.1
svm: svm_iscsi
labels:
  k8scluster: test-cluster-1
  backend: testcluster1-sanbackend
username: vsadmin
password: <password>
```

## Exemple MetroCluster

Vous pouvez configurer le back-end pour éviter de devoir mettre à jour manuellement la définition du back-end après le basculement et le rétablissement pendant "[RéPLICATION ET RESTAURATION DES SVM](#)".

Pour un basculement et un retour en arrière transparents, préciser le SVM en utilisant `managementLIF` et omettre les `svm` paramètres. Par exemple :

```
version: 1
storageDriverName: ontap-san
managementLIF: 192.168.1.66
username: vsadmin
password: password
```

## Exemple d'économie SAN ONTAP

```
version: 1
storageDriverName: ontap-san-economy
managementLIF: 10.0.0.1
svm: svm_iscsi_eco
username: vsadmin
password: <password>
```

## Exemple d'authentification basée sur un certificat

Dans cet exemple de configuration de base `clientCertificate`, `clientPrivateKey`, et `trustedCACertificate` (facultatif, si vous utilisez une autorité de certification approuvée) sont renseignés `backend.json` et prennent respectivement les valeurs codées en base64 du certificat client, de la clé privée et du certificat de l'autorité de certification approuvée.

```
---  
version: 1  
storageDriverName: ontap-san  
backendName: DefaultSANBackend  
managementLIF: 10.0.0.1  
svm: svm_iscsi  
useCHAP: true  
chapInitiatorSecret: c19qxIm36DKyawxy  
chapTargetInitiatorSecret: rqxigXgkesIpwxyz  
chapTargetUsername: iJF4heBRT0TCwxyz  
chapUsername: uh2aNCLSd6cNwxyz  
clientCertificate: ZXROZXJwYXB...ICMgJ3BhcGVyc2  
clientPrivateKey: vciwKIyAgZG...0cnksIGRlc2NyaX  
trustedCACertificate: zcyBbaG...b3Igb3duIGNsYXNz
```

## Exemples CHAP bidirectionnels

Ces exemples créent un back-end avec `useCHAP` défini sur `true`.

### Exemple CHAP de SAN ONTAP

```
---  
version: 1  
storageDriverName: ontap-san  
managementLIF: 10.0.0.1  
svm: svm_iscsi  
labels:  
  k8scluster: test-cluster-1  
  backend: testcluster1-sanbackend  
useCHAP: true  
chapInitiatorSecret: cl9qxIm36DKyawxy  
chapTargetInitiatorSecret: rqxigXgkesIpwxyz  
chapTargetUsername: iJF4heBRT0TCwxyz  
chapUsername: uh2aNCLSd6cNwxyz  
username: vsadmin  
password: <password>
```

### Exemple CHAP d'économie SAN ONTAP

```
---  
version: 1  
storageDriverName: ontap-san-economy  
managementLIF: 10.0.0.1  
svm: svm_iscsi_eco  
useCHAP: true  
chapInitiatorSecret: cl9qxIm36DKyawxy  
chapTargetInitiatorSecret: rqxigXgkesIpwxyz  
chapTargetUsername: iJF4heBRT0TCwxyz  
chapUsername: uh2aNCLSd6cNwxyz  
username: vsadmin  
password: <password>
```

## Exemple NVMe/TCP

Un SVM doit être configuré avec NVMe sur votre back-end ONTAP. Il s'agit d'une configuration back-end de base pour NVMe/TCP.

```
---  
version: 1  
backendName: NVMeBackend  
storageDriverName: ontap-san  
managementLIF: 10.0.0.1  
svm: svm_nvme  
username: vsadmin  
password: password  
sanType: nvme  
useREST: true
```

## Exemple de SCSI sur FC (FCP)

Vous devez avoir un SVM configuré avec FC sur votre back-end ONTAP. Il s'agit d'une configuration back-end de base pour FC.

```
---  
version: 1  
backendName: fcp-backend  
storageDriverName: ontap-san  
managementLIF: 10.0.0.1  
svm: svm_fc  
username: vsadmin  
password: password  
sanType: fcp  
useREST: true
```

## Exemple de configuration back-end avec nomTemplate

```
---  
version: 1  
storageDriverName: ontap-san  
backendName: ontap-san-backend  
managementLIF: <ip address>  
svm: svm0  
username: <admin>  
password: <password>  
defaults:  
  nameTemplate:  
    "{{.volume.Name}}_{{.labels.cluster}}_{{.volume.Namespace}}_{{.vo\\}}  
      lume.RequestName}"  
  labels:  
    cluster: ClusterA  
    PVC: "{{.volume.Namespace}}_{{.volume.RequestName}}"
```

## Exemple de formatoptions pour le pilote ONTAP-san-Economy

```
---  
version: 1  
storageDriverName: ontap-san-economy  
managementLIF: ""  
svm: svml  
username: ""  
password: "!"  
storagePrefix: whelk_  
debugTraceFlags:  
  method: true  
  api: true  
defaults:  
  formatOptions: -E nodiscard
```

## Exemples de systèmes back-end avec pools virtuels

Dans ces exemples de fichiers de définition back-end, des valeurs par défaut spécifiques sont définies pour tous les pools de stockage, comme spaceReserve aucun, spaceAllocation faux et encryption faux. Les pools virtuels sont définis dans la section stockage.

Trident définit les étiquettes de provisionnement dans le champ « Commentaires ». Les commentaires sont définis sur les copies FlexVol volume Trident. Toutes les étiquettes présentes sur un pool virtuel sont apposées sur le volume de stockage au moment du provisionnement. Pour plus de commodité, les administrateurs du stockage peuvent définir des étiquettes par pool virtuel et les volumes de groupe par étiquette.

Dans ces exemples, certains pools de stockage définissent leurs propres `spaceReserve` valeurs , `spaceAllocation` et, `et encryption` certains pools remplacent les valeurs par défaut.

## **Exemple de SAN ONTAP**

```
---  
version: 1  
storageDriverName: ontap-san  
managementLIF: 10.0.0.1  
svm: svm_iscsi  
useCHAP: true  
chapInitiatorSecret: cl9qxIm36DKyawxy  
chapTargetInitiatorSecret: rqxigXgkesIpwxyz  
chapTargetUsername: iJF4heBRT0TCwxyz  
chapUsername: uh2aNCLSd6cNwxyz  
username: vsadmin  
password: <password>  
defaults:  
    spaceAllocation: "false"  
    encryption: "false"  
    qosPolicy: standard  
labels:  
    store: san_store  
    kubernetes-cluster: prod-cluster-1  
region: us_east_1  
storage:  
    - labels:  
        protection: gold  
        creditpoints: "40000"  
        zone: us_east_1a  
        defaults:  
            spaceAllocation: "true"  
            encryption: "true"  
            adaptiveQosPolicy: adaptive-extreme  
    - labels:  
        protection: silver  
        creditpoints: "20000"  
        zone: us_east_1b  
        defaults:  
            spaceAllocation: "false"  
            encryption: "true"  
            qosPolicy: premium  
    - labels:  
        protection: bronze  
        creditpoints: "5000"  
        zone: us_east_1c  
        defaults:  
            spaceAllocation: "true"  
            encryption: "false"
```

## Exemple d'économie SAN ONTAP

```
---  
version: 1  
storageDriverName: ontap-san-economy  
managementLIF: 10.0.0.1  
svm: svm_iscsi_eco  
useCHAP: true  
chapInitiatorSecret: cl9qxIm36DKyawxy  
chapTargetInitiatorSecret: rqxigXgkesIpwxyz  
chapTargetUsername: iJF4heBRT0TCwxyz  
chapUsername: uh2aNCLSd6cNwxyz  
username: vsadmin  
password: <password>  
defaults:  
  spaceAllocation: "false"  
  encryption: "false"  
labels:  
  store: san_economy_store  
region: us_east_1  
storage:  
  - labels:  
    app: oracledb  
    cost: "30"  
    zone: us_east_1a  
    defaults:  
      spaceAllocation: "true"  
      encryption: "true"  
  - labels:  
    app: postgresdb  
    cost: "20"  
    zone: us_east_1b  
    defaults:  
      spaceAllocation: "false"  
      encryption: "true"  
  - labels:  
    app: mysql ldb  
    cost: "10"  
    zone: us_east_1c  
    defaults:  
      spaceAllocation: "true"  
      encryption: "false"  
  - labels:  
    department: legal  
    creditpoints: "5000"  
    zone: us_east_1c
```

```
defaults:  
  spaceAllocation: "true"  
  encryption: "false"
```

## Exemple NVMe/TCP

```
---  
version: 1  
storageDriverName: ontap-san  
sanType: nvme  
managementLIF: 10.0.0.1  
svm: nvme_svm  
username: vsadmin  
password: <password>  
useREST: true  
defaults:  
  spaceAllocation: "false"  
  encryption: "true"  
storage:  
  - labels:  
    app: testApp  
    cost: "20"  
  defaults:  
    spaceAllocation: "false"  
    encryption: "false"
```

## Mappage des systèmes back-end aux classes de stockage

Les définitions de classe de stockage suivantes font référence à la [Exemples de systèmes back-end avec pools virtuels](#). En utilisant ce parameters.selector champ, chaque classe de stockage indique quels pools virtuels peuvent être utilisés pour héberger un volume. Les aspects définis dans le pool virtuel sélectionné seront définis pour le volume.

- La protection-gold classe de stockage est mappée sur le premier pool virtuel du ontap-san back-end. Il s'agit du seul pool offrant une protection de niveau Gold.

```

apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: protection-gold
provisioner: csi.trident.netapp.io
parameters:
  selector: "protection=gold"
  fsType: "ext4"

```

- La protection-not-gold classe de stockage sera mappée sur le deuxième et le troisième pool virtuel du ontap-san back-end. Ce sont les seuls pools offrant un niveau de protection autre que Gold.

```

apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: protection-not-gold
provisioner: csi.trident.netapp.io
parameters:
  selector: "protection!=gold"
  fsType: "ext4"

```

- La app-mysqldb classe de stockage sera mappée sur le troisième pool virtuel du ontap-san-economy back-end. Il s'agit du seul pool offrant la configuration du pool de stockage pour l'application de type mysqldb.

```

apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: app-mysqldb
provisioner: csi.trident.netapp.io
parameters:
  selector: "app=mysql"
  fsType: "ext4"

```

- La protection-silver-creditpoints-20k classe de stockage sera mappée sur le second pool virtuel du ontap-san back-end. Il s'agit de la seule piscine offrant une protection de niveau argent et 20000 points de crédit.

```

apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: protection-silver-creditpoints-20k
provisioner: csi.trident.netapp.io
parameters:
  selector: "protection=silver; creditpoints=20000"
  fsType: "ext4"

```

- La creditpoints-5k classe de stockage est mappée sur le troisième pool virtuel du ontap-san back-end et le quatrième pool virtuel du ontap-san-economy back-end. Il s'agit des seules offres de pool avec 5000 points de crédit.

```

apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: creditpoints-5k
provisioner: csi.trident.netapp.io
parameters:
  selector: "creditpoints=5000"
  fsType: "ext4"

```

- La my-test-app-sc classe de stockage est mappée sur le testAPP pool virtuel du ontap-san pilote avec sanType: nvme. C'est la seule offre de piscine testApp .

```

---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: my-test-app-sc
provisioner: csi.trident.netapp.io
parameters:
  selector: "app=testApp"
  fsType: "ext4"

```

Trident décide du pool virtuel sélectionné et s'assure que les besoins en stockage sont satisfait.

## Pilotes NAS de ONTAP

### Présentation du pilote NAS ONTAP

Découvrez comment configurer un back-end ONTAP avec les pilotes ONTAP et NAS Cloud Volumes ONTAP.

## Détails du pilote NAS ONTAP

Trident fournit les pilotes de stockage NAS suivants pour communiquer avec le cluster ONTAP. Les modes d'accès pris en charge sont : *ReadWriteOnce* (RWO), *ReadOnlyMey* (ROX), *ReadWriteMaly* (RWX), *ReadWriteOncePod* (RWOP).

Conducteur	Protocole	Mode Volume	Modes d'accès pris en charge	Systèmes de fichiers pris en charge
ontap-nas	NFS SMB	Système de fichiers	RWO, ROX, RWX, RWOP	"", nfs, smb
ontap-nas-economy	NFS SMB	Système de fichiers	RWO, ROX, RWX, RWOP	"", nfs, smb
ontap-nas-flexgroup	NFS SMB	Système de fichiers	RWO, ROX, RWX, RWOP	"", nfs, smb

- À utiliser `ontap-san-economy` uniquement si le nombre d'utilisations de volume persistant doit être supérieur à "[Limites de volume ONTAP prises en charge](#)".
- À utiliser `ontap-nas-economy` uniquement si le nombre d'utilisations de volume persistant doit être supérieur à et si "[Limites de volume ONTAP prises en charge](#)" le `ontap-san-economy` pilote ne peut pas être utilisé.
- N'utilisez pas cette option `ontap-nas-economy` si vous prévoyez d'avoir besoin de protection des données, de reprise après incident ou de mobilité.
- NetApp ne recommande pas l'utilisation de l'autogrow FlexVol dans tous les pilotes ONTAP, sauf ONTAP-san. Pour contourner ce problème, Trident prend en charge l'utilisation de la réserve Snapshot et adapte les volumes FlexVol en conséquence.

## Autorisations utilisateur

Trident s'attend à être exécuté en tant qu'administrateur ONTAP ou SVM, en général avec l'utilisateur du cluster ou un `vsadmin` utilisateur SVM, ou en tant qu'`admin` utilisateur avec un nom différent et le même rôle.

Pour les déploiements Amazon FSX pour NetApp ONTAP, Trident prévoit d'être exécuté en tant qu'administrateur ONTAP ou SVM, en utilisant l'utilisateur du cluster `fsxadmin` ou un `vsadmin` utilisateur SVM, ou un utilisateur avec un nom différent ayant le même rôle. `fsxadmin` L'utilisateur est un remplaçant limité pour l'utilisateur `admin` du cluster.

 Si vous utilisez le `limitAggregateUsage` paramètre, les autorisations d'administration du cluster sont requises. Lors de l'utilisation d'Amazon FSX for NetApp ONTAP avec Trident, le `limitAggregateUsage` paramètre ne fonctionnera pas avec les `vsadmin` comptes d'utilisateur et `fsxadmin`. L'opération de configuration échoue si vous spécifiez ce paramètre.

S'il est possible de créer au sein de ONTAP un rôle plus restrictif qu'un pilote Trident peut utiliser, nous ne le recommandons pas. La plupart des nouvelles versions de Trident appellent des API supplémentaires qui devront être prises en compte, ce qui complique les mises à niveau et risque d'erreurs.

## Préparez la configuration d'un système back-end avec les pilotes NAS ONTAP

Découvrez les exigences, les options d'authentification et les règles d'exportation pour la configuration d'un back-end ONTAP avec des pilotes NAS ONTAP.

### De formation

- Pour tous les backends ONTAP, Trident exige qu'au moins un agrégat soit attribué au SVM.
- Vous pouvez exécuter plusieurs pilotes et créer des classes de stockage qui pointent vers l'un ou l'autre. Par exemple, vous pouvez configurer une classe Gold qui utilise le `ontap-nas` pilote et une classe Bronze qui utilise celui-ci `ontap-nas-economy`.
- Tous vos nœuds workers Kubernetes doivent avoir installé les outils NFS appropriés. Voir "[ici](#)" pour plus de détails.
- Trident prend en charge les volumes SMB montés sur les pods s'exécutant sur les nœuds Windows uniquement. Voir [Préparez-vous au provisionnement des volumes SMB](#) pour plus de détails.

### Authentifiez le back-end ONTAP

Trident propose deux modes d'authentification d'un back-end ONTAP.

- Basé sur les informations d'identification : ce mode requiert des autorisations suffisantes pour le backend ONTAP. Il est recommandé d'utiliser un compte associé à un rôle de connexion de sécurité prédéfini, tel que `admin` ou `vsadmin` pour assurer une compatibilité maximale avec les versions de ONTAP.
- Basé sur un certificat : ce mode nécessite l'installation d'un certificat sur le back-end pour que Trident puisse communiquer avec un cluster ONTAP. Dans ce cas, la définition `backend` doit contenir des valeurs encodées Base64 du certificat client, de la clé et du certificat d'autorité de certification de confiance, le cas échéant (recommandé).

Vous pouvez mettre à jour les systèmes back-end existants pour passer d'une méthode basée sur les identifiants à une méthode basée sur les certificats. Toutefois, une seule méthode d'authentification est prise en charge à la fois. Pour passer à une méthode d'authentification différente, vous devez supprimer la méthode existante de la configuration `backend`.

 Si vous tentez de fournir **les deux identifiants et les certificats**, la création du back-end échoue avec une erreur indiquant que plus d'une méthode d'authentification a été fournie dans le fichier de configuration.

### Activer l'authentification basée sur les informations d'identification

Trident exige que les identifiants d'un administrateur SVM-scoped/cluster-scoped communiquent avec le back-end ONTAP. Il est recommandé d'utiliser des rôles standard prédéfinis tels que `admin` ou `vsadmin`. La compatibilité avec les futures versions d'ONTAP qui exposent les API de fonctionnalités à utiliser dans les futures versions d'Trident est ainsi garantie. Un rôle de connexion de sécurité personnalisé peut être créé et utilisé avec Trident, mais il n'est pas recommandé.

Voici un exemple de définition du back-end :

## YAML

```
---
```

```
version: 1
backendName: ExampleBackend
storageDriverName: ontap-nas
managementLIF: 10.0.0.1
dataLIF: 10.0.0.2
svm: svm_nfs
username: vsadmin
password: password
```

## JSON

```
{
  "version": 1,
  "backendName": "ExampleBackend",
  "storageDriverName": "ontap-nas",
  "managementLIF": "10.0.0.1",
  "dataLIF": "10.0.0.2",
  "svm": "svm_nfs",
  "username": "vsadmin",
  "password": "password"
}
```

Gardez à l'esprit que la définition du back-end est le seul endroit où les informations d'identification sont stockées en texte brut. Une fois le système backend créé, les noms d'utilisateur/mots de passe sont codés avec Base64 et stockés sous forme de secrets Kubernetes. La création/la conversion d'un back-end est la seule étape qui nécessite la connaissance des informations d'identification. Il s'agit donc d'une opération uniquement administrative, qui doit être effectuée par l'administrateur Kubernetes/du stockage.

### Activez l'authentification basée sur les certificats

Les systèmes back-end, nouveaux et existants, peuvent utiliser un certificat et communiquer avec le système back-end ONTAP. Trois paramètres sont requis dans la définition du back-end.

- ClientCertificate : valeur encodée en Base64 du certificat client.
- ClientPrivateKey : valeur encodée en Base64 de la clé privée associée.
- TrustedCACertificate : valeur encodée Base64 du certificat CA de confiance. Si vous utilisez une autorité de certification approuvée, ce paramètre doit être fourni. Ceci peut être ignoré si aucune autorité de certification approuvée n'est utilisée.

Un flux de travail type comprend les étapes suivantes.

### Étapes

1. Générez un certificat client et une clé. Lors de la génération, définissez le nom commun (CN) sur

l'utilisateur ONTAP pour qu'il s'authentifie.

```
openssl req -x509 -nodes -days 1095 -newkey rsa:2048 -keyout k8senv.key  
-out k8senv.pem -subj "/C=US/ST=NC/L=RTP/O=NetApp/CN=vsadmin"
```

2. Ajoutez un certificat d'autorité de certification de confiance au cluster ONTAP. Il se peut déjà que l'administrateur de stockage gère cet espace. Ignorer si aucune autorité de certification approuvée n'est utilisée.

```
security certificate install -type server -cert-name <trusted-ca-cert-name> -vserver <vserver-name>  
ssl modify -vserver <vserver-name> -server-enabled true -client-enabled true -common-name <common-name> -serial <SN-from-trusted-CA-cert> -ca <cert-authority>
```

3. Installez le certificat client et la clé (à partir de l'étape 1) sur le cluster ONTAP.

```
security certificate install -type client-ca -cert-name <certificate-name> -vserver <vserver-name>  
security ssl modify -vserver <vserver-name> -client-enabled true
```

4. Vérifiez que le rôle de connexion de sécurité ONTAP prend en charge cert la méthode d'authentification.

```
security login create -user-or-group-name vsadmin -application ontapi  
-authentication-method cert -vserver <vserver-name>  
security login create -user-or-group-name vsadmin -application http  
-authentication-method cert -vserver <vserver-name>
```

5. Testez l'authentification à l'aide d'un certificat généré. Remplacer <ONTAP Management LIF> et <vserver name> par Management LIF IP et SVM name. Vous devez vous assurer que la stratégie de service de la LIF est définie sur default-data-management.

```
curl -X POST -Lk https://<ONTAP-Management-LIF>/servlets/netapp.servlets.admin.XMLrequest_filer --key k8senv.key  
--cert ~/k8senv.pem -d '<?xml version="1.0" encoding="UTF-8"?><netapp  
xmlns="http://www.netapp.com/filer/admin" version="1.21"  
vfiler="<vserver-name>"><vserver-get></vserver-get></netapp>'
```

6. Encodez le certificat, la clé et le certificat CA de confiance avec Base64.

```
base64 -w 0 k8senv.pem >> cert_base64  
base64 -w 0 k8senv.key >> key_base64  
base64 -w 0 trustedca.pem >> trustedca_base64
```

## 7. Créez le back-end à l'aide des valeurs obtenues à partir de l'étape précédente.

```
cat cert-backend-updated.json  
{  
  "version": 1,  
  "storageDriverName": "ontap-nas",  
  "backendName": "NasBackend",  
  "managementLIF": "1.2.3.4",  
  "dataLIF": "1.2.3.8",  
  "svm": "vserver_test",  
  "clientCertificate": "Faaaakkkeeee...Vaaallluuuueeee",  
  "clientPrivateKey": "LS0tFAKE...0VaLuES0tLS0K",  
  "storagePrefix": "myPrefix_"  
}  
  
#Update backend with tridentctl  
tridentctl update backend NasBackend -f cert-backend-updated.json -n  
trident  
+-----+-----+-----+  
+-----+-----+  
|      NAME      | STORAGE DRIVER |          UUID          |  
STATE | VOLUMES |  
+-----+-----+-----+  
+-----+-----+  
| NasBackend | ontap-nas     | 98e19b74-aec7-4a3d-8dcf-128e5033b214 |  
online |         9 |  
+-----+-----+-----+  
+-----+-----+
```

## Mettre à jour les méthodes d'authentification ou faire pivoter les informations d'identification

Vous pouvez mettre à jour un back-end existant pour utiliser une méthode d'authentification différente ou pour faire pivoter leurs informations d'identification. Cela fonctionne de deux manières : les systèmes back-end qui utilisent le nom d'utilisateur/mot de passe peuvent être mis à jour pour utiliser des certificats ; les systèmes back-end qui utilisent des certificats peuvent être mis à jour en fonction du nom d'utilisateur/mot de passe. Pour ce faire, vous devez supprimer la méthode d'authentification existante et ajouter la nouvelle méthode d'authentification. Utilisez ensuite le fichier backend.json mis à jour contenant les paramètres requis pour exécuter `tridentctl update backend`.

```
cat cert-backend-updated.json
```

```
{  
  "version": 1,  
  "storageDriverName": "ontap-nas",  
  "backendName": "NasBackend",  
  "managementLIF": "1.2.3.4",  
  "dataLIF": "1.2.3.8",  
  "svm": "vserver_test",  
  "username": "vsadmin",  
  "password": "password",  
  "storagePrefix": "myPrefix_"  
}
```

```
#Update backend with tridentctl  
tridentctl update backend NasBackend -f cert-backend-updated.json -n  
trident  
+-----+-----+-----+  
+-----+-----+  
|      NAME      | STORAGE DRIVER |          UUID          |  
STATE | VOLUMES |  
+-----+-----+-----+  
+-----+-----+  
| NasBackend | ontap-nas       | 98e19b74-aec7-4a3d-8dcf-128e5033b214 |  
online |         9 |  
+-----+-----+-----+  
+-----+-----+
```

 Lors de la rotation des mots de passe, l'administrateur du stockage doit d'abord mettre à jour le mot de passe de l'utilisateur sur ONTAP. Cette opération est suivie d'une mise à jour du back-end. Lors de la rotation de certificats, plusieurs certificats peuvent être ajoutés à l'utilisateur. Le back-end est ensuite mis à jour pour utiliser le nouveau certificat, en suivant lequel l'ancien certificat peut être supprimé du cluster ONTAP.

La mise à jour d'un back-end n'interrompt pas l'accès aux volumes qui ont déjà été créés, et n'a aucun impact sur les connexions de volume effectuées après. Une mise à jour back-end réussie indique que Trident peut communiquer avec le back-end ONTAP et gérer les futures opérations de volume.

## Créez un rôle ONTAP personnalisé pour Trident

Vous pouvez créer un rôle de cluster ONTAP avec une Privileges minimale afin de ne pas avoir à utiliser le rôle ONTAP admin pour effectuer des opérations dans Trident. Lorsque vous incluez le nom d'utilisateur dans une configuration Trident backend, Trident utilise le rôle de cluster ONTAP que vous avez créé pour effectuer les opérations.

Pour plus d'informations sur la création de rôles personnalisés Trident, reportez-vous à la section "[Générateur de rôle personnalisé Trident](#)".

### Utilisation de l'interface de ligne de commandes ONTAP

- Créez un rôle à l'aide de la commande suivante :

```
security login role create <role_name> -cmddirname "command" -access all  
-vserver <svm_name>
```

- Créez un nom d'utilisateur pour l'utilisateur Trident :

```
security login create -username <user_name> -application ontapi  
-authmethod <password> -role <name_of_role_in_step_1> -vserver  
<svm_name> -comment "user_description"
```

- Mapper le rôle à l'utilisateur :

```
security login modify username <user_name> -vserver <svm_name> -role  
<role_name> -application ontapi -application console -authmethod  
<password>
```

### À l'aide de System Manager

Dans ONTAP System Manager, effectuez les opérations suivantes :

- Créer un rôle personnalisé :

- Pour créer un rôle personnalisé au niveau du cluster, sélectionnez **Cluster > Paramètres**.  
(Ou) pour créer un rôle personnalisé au niveau du SVM, sélectionner **stockage > Storage VM > >> Paramètres > required SVM utilisateurs et rôles**.

- Sélectionnez l'icône de flèche (→) en regard de **utilisateurs et rôles**.
  - Sélectionnez **+Ajouter** sous **rôles**.
  - Définissez les règles du rôle et cliquez sur **Enregistrer**.
- Mapper le rôle à l'utilisateur Trident: + effectuez les étapes suivantes sur la page **utilisateurs et rôles** :
    - Sélectionnez Ajouter l'icône + sous **utilisateurs**.
    - Sélectionnez le nom d'utilisateur requis et sélectionnez un rôle dans le menu déroulant pour **role**.
    - Cliquez sur **Enregistrer**.

Pour plus d'informations, reportez-vous aux pages suivantes :

- "[Rôles personnalisés pour l'administration de ONTAP](#)" ou "[Définissez des rôles personnalisés](#)"
- "[Travaillez avec les rôles et les utilisateurs](#)"

### Gestion des règles d'exportation NFS

Trident utilise des export policy NFS pour contrôler l'accès aux volumes qu'il provisionne.

Trident propose deux options pour les règles d'export :

- Trident peut gérer la politique d'export de manière dynamique. Dans ce mode de fonctionnement, l'administrateur du stockage spécifie une liste de blocs CIDR qui représentent des adresses IP recevables. Trident ajoute automatiquement aux règles d'export les adresses IP de nœud applicables comprises dans ces plages au moment de la publication. Sinon, lorsqu'aucun CIDR n'est spécifié, toutes les adresses IP de monodiffusion à périmètre global trouvées sur le nœud auquel le volume est publié seront ajoutées à la export policy.
- Les administrateurs du stockage peuvent créer une export-policy et ajouter des règles manuellement. Trident utilise la export policy par défaut sauf si un autre nom de export policy est spécifié dans la configuration.

## Gérez les règles d'exportation de manière dynamique

Trident permet de gérer de manière dynamique les politiques d'exportation des systèmes back-end ONTAP. Cela permet à l'administrateur du stockage de spécifier un espace d'adresse autorisé pour les adresses IP du nœud de travail, au lieu de définir manuellement des règles explicites. Il simplifie considérablement la gestion des export policy ; les modifications apportées à l'export policy ne nécessitent plus d'intervention manuelle sur le cluster de stockage. De plus, cela permet de restreindre l'accès au cluster de stockage uniquement aux nœuds workers qui montez des volumes et dont les adresses IP se situent dans la plage spécifiée, et de prendre en charge une gestion automatisée et précise.

 N'utilisez pas NAT (Network Address Translation) lorsque vous utilisez des stratégies d'exportation dynamiques. Avec NAT, le contrôleur de stockage voit l'adresse NAT front-end et non l'adresse IP réelle de l'hôte. L'accès sera donc refusé lorsqu'aucune correspondance n'est trouvée dans les règles d'exportation.

## Exemple

Deux options de configuration doivent être utilisées. Voici un exemple de définition de back-end :

```
---
version: 1
storageDriverName: ontap-nas-economy
backendName: ontap_nas_auto_export
managementLIF: 192.168.0.135
svm: svml
username: vsadmin
password: password
autoExportCIDRs:
  - 192.168.0.0/24
autoExportPolicy: true
```

 Lorsque vous utilisez cette fonctionnalité, vous devez vous assurer que la jonction root dans votre SVM possède une export policy précédemment créée avec une règle d'exportation qui autorise le bloc CIDR (comme la export policy par défaut) du nœud. Respectez toujours les bonnes pratiques recommandées par NetApp pour dédier une SVM à Trident.

Voici une explication du fonctionnement de cette fonction à l'aide de l'exemple ci-dessus :

- `autoExportPolicy` est défini sur `true`. Cela signifie que Trident crée une export policy pour chaque volume provisionné avec ce back-end pour la `svm1` SVM et gère l'ajout et la suppression de règles à l'aide de `autoexportCIDRs` blocs d'adresse. Tant qu'un volume n'est pas rattaché à un nœud, le volume utilise une export policy vide sans règle pour empêcher tout accès indésirable à ce volume. Lorsqu'un volume est publié sur un nœud, Trident crée une export policy portant le même nom que le qtree sous-jacent contenant l'IP de nœud dans le bloc CIDR spécifié. Ces adresses IP seront également ajoutées à la export policy utilisée par le FlexVol volume parent
  - Par exemple :
    - Back-end UUID `403b5326-8482-40db-96d0-d83fb3f4daec`
    - `autoExportPolicy` réglé sur `true`
    - préfixe de stockage `trident`
    - UUID de PVC `a79bcf5f-7b6d-4a40-9876-e2551f159c1c`
    - Qtree nommée `Trident_pvc_a79bcf5f_7b6d_4a40_9876_e2551f159c1c` crée une export policy pour la FlexVol nommée, une export policy pour le qtree `trident_pvc_a79bcf5f_7b6d_4a40_9876_e2551f159c1c` nommé `trident-403b5326-8482-40db96d0-d83fb3f4daec` et une export policy vide nommée `trident_empty` sur le SVM. Les règles de la FlexVol export policy seront un superset de toutes les règles contenues dans les qtree export policies. Les règles d'export vides seront réutilisées par tous les volumes qui ne sont pas attachés.
- `autoExportCIDRs` contient une liste de blocs d'adresses. Ce champ est facultatif et il prend par défaut la valeur `["0.0.0.0/0", ":/0"]`. S'il n'est pas défini, Trident ajoute toutes les adresses de monodiffusion à portée globale trouvées sur les nœuds de travail avec des publications.

Dans cet exemple, l'`192.168.0.0/24` espace d'adresse est fourni. Cela signifie que les adresses IP des nœuds Kubernetes comprises dans cette plage d'adresses avec les publications seront ajoutées à la règle d'export créée par Trident. Lorsque Trident enregistre un nœud sur lequel il s'exécute, il récupère les adresses IP du nœud et les compare aux blocs d'adresse fournis dans `autoExportCIDRs`. Au moment de la publication, après le filtrage des adresses IP, Trident crée les règles d'export policy pour les adresses IP du client pour le nœud sur lequel il publie.

Vous pouvez mettre à jour `autoExportPolicy` et `autoExportCIDRs` pour les systèmes back-end une fois que vous les avez créés. Vous pouvez ajouter de nouveaux rapports CIDR pour un back-end qui est géré automatiquement ou supprimé des rapports CIDR existants. Faites preuve de prudence lors de la suppression des CIDR pour vous assurer que les connexions existantes ne sont pas tombées. Vous pouvez également choisir de désactiver `autoExportPolicy` pour un backend et de revenir à une export policy créée manuellement. Pour cela, vous devrez définir le `exportPolicy` paramètre dans votre configuration back-end.

Une fois que Trident a créé ou mis à jour un back-end, vous pouvez vérifier le back-end à l'aide de `tridentctl` ou du CRD correspondant `tridentbackend`:

```
./tridentctl get backends ontap_nas_auto_export -n trident -o yaml
items:
- backendUUID: 403b5326-8482-40db-96d0-d83fb3f4daec
  config:
    aggregate: ""
    autoExportCIDRs:
    - 192.168.0.0/24
    autoExportPolicy: true
    backendName: ontap_nas_auto_export
    chapInitiatorSecret: ""
    chapTargetInitiatorSecret: ""
    chapTargetUsername: ""
    chapUsername: ""
    dataLIF: 192.168.0.135
    debug: false
    debugTraceFlags: null
    defaults:
      encryption: "false"
      exportPolicy: <automatic>
      fileSystemType: ext4
```

Lorsqu'un nœud est supprimé, Trident vérifie toutes les export policies pour supprimer les règles d'accès correspondant au nœud. En supprimant cette adresse IP de nœud des politiques d'exportation des systèmes back-end gérés, Trident empêche les montages indésirables, sauf si cette adresse IP est réutilisée par un nouveau nœud du cluster.

Pour les systèmes back-end existants, la mise à jour du back-end `tridentctl update backend` permet à Trident de gérer automatiquement les règles d'exportation. Deux nouvelles règles d'exportation nommées en fonction du nom UUID et du nom de qtree du système back-end sont alors créées, le cas échéant. Les volumes présents sur le back-end utiliseront les nouvelles règles d'export créées une fois qu'elles auront été démontées et remontées.

 La suppression d'un back-end avec des règles d'exportation gérées automatiquement supprimera l'export policy créée de manière dynamique. Si le back-end est recréé, il est traité comme un nouveau backend et entraîne la création d'une nouvelle export policy.

Si l'adresse IP d'un nœud actif est mise à jour, vous devez redémarrer le pod Trident sur le nœud. Trident mettra ensuite à jour la politique d'exportation des systèmes back-end qu'elle gère pour refléter cette modification de propriété intellectuelle.

#### Préparez-vous au provisionnement des volumes SMB

Avec un peu de préparation supplémentaire, vous pouvez provisionner des volumes SMB à l'aide de `ontap-nas` pilotes.

 Vous devez configurer les protocoles NFS et SMB/CIFS au SVM pour créer un `ontap-nas-economy` volume SMB pour les clusters ONTAP sur site. La configuration de l'un de ces protocoles entraîne l'échec de la création du volume SMB.



autoExportPolicy N'est pas pris en charge pour les volumes SMB.

## Avant de commencer

Avant de pouvoir provisionner des volumes SMB, vous devez disposer des éléments suivants :

- Cluster Kubernetes avec un nœud de contrôleur Linux et au moins un nœud worker Windows exécutant Windows Server 2022. Trident prend en charge les volumes SMB montés sur les pods s'exécutant sur les nœuds Windows uniquement.
- Au moins un secret Trident contenant vos informations d'identification Active Directory. Pour générer un secret `smbcreds`:

```
kubectl create secret generic smbcreds --from-literal username=user  
--from-literal password='password'
```

- Un proxy CSI configuré en tant que service Windows. Pour configurer un `csi-proxy`, reportez-vous "[GitHub : proxy CSI](#)" à la section ou "[GitHub : proxy CSI pour Windows](#)" pour les nœuds Kubernetes s'exécutant sous Windows.

## Étapes

1. Pour les ONTAP sur site, vous pouvez créer un partage SMB ou Trident en créer un pour vous.



Les partages SMB sont requis pour Amazon FSX pour ONTAP.

Vous pouvez créer les partages d'administration SMB de deux manières, soit à l'aide du "[Console de gestion Microsoft](#)" composant logiciel enfichable dossiers partagés, soit à l'aide de l'interface de ligne de commande ONTAP. Pour créer les partages SMB à l'aide de l'interface de ligne de commandes ONTAP :

- a. Si nécessaire, créez la structure du chemin d'accès au répertoire pour le partage.

La `vserver cifs share create` commande vérifie le chemin spécifié dans l'option `-path` lors de la création du partage. Si le chemin spécifié n'existe pas, la commande échoue.

- b. Créer un partage SMB associé au SVM spécifié :

```
vserver cifs share create -vserver vserver_name -share-name  
share_name -path path [-share-properties share_properties,...]  
[other_attributes] [-comment text]
```

- c. Vérifiez que le partage a été créé :

```
vserver cifs share show -share-name share_name
```



Reportez-vous "[Créez un partage SMB](#)" à pour plus de détails.

2. Lors de la création du back-end, vous devez configurer le suivant pour spécifier les volumes SMB. Pour toutes les options de configuration du back-end FSX for ONTAP, reportez-vous à "[Exemples et options de](#)

configuration de FSX pour ONTAP" la section .

Paramètre	Description	Exemple
smbShare	Vous pouvez spécifier l'une des options suivantes : le nom d'un partage SMB créé à l'aide de la console de gestion Microsoft ou de l'interface de ligne de commande ONTAP ; un nom permettant à Trident de créer le partage SMB ; ou bien laisser le paramètre vide pour empêcher l'accès au partage commun aux volumes. Ce paramètre est facultatif pour les ONTAP sur site. Ce paramètre est requis pour Amazon FSX pour les systèmes back-end ONTAP et ne peut pas être vide.	smb-share
nasType	<b>Doit être réglé sur smb.</b> Si nul, la valeur par défaut est nfs .	smb
securityStyle	Style de sécurité pour les nouveaux volumes. <b>Doit être défini sur ntfs ou mixed pour les volumes SMB.</b>	ntfs Ou mixed pour les volumes SMB
unixPermissions	Mode pour les nouveaux volumes. <b>Doit rester vide pour les volumes SMB.</b>	« »

## Options et exemples de configuration du NAS ONTAP

Apprenez à créer et à utiliser des pilotes NAS ONTAP avec votre installation Trident. Cette section fournit des exemples de configuration back-end et des détails sur le mappage des systèmes back-end aux classes de stockage.

### Options de configuration du back-end

Voir le tableau suivant pour les options de configuration du back-end :

Paramètre	Description	Valeur par défaut
version		Toujours 1
storageDrive rName	Nom du pilote de stockage	ontap-nas, ontap-nas-economy OU ontap-nas-flexgroup
backendName	Nom personnalisé ou système back-end de stockage	Nom du pilote + " _ " + dataLIF
managementLIF	Adresse IP d'un cluster ou LIF de gestion De SVM Un nom de domaine complet (FQDN) peut être spécifié. Peut être configuré pour utiliser des adresses IPv6 si Trident a été installé à l'aide de l'indicateur IPv6. Les adresses IPv6 doivent être définies entre crochets, par exemple [28e8:d9fb:a825:b7bf:69a8:d02f:9e7b:3555]. Pour un basculement MetroCluster transparent, consultez le <a href="#">Exemple MetroCluster</a> .	« 10.0.0.1 », « [2001:1234:abcd::fefe] »

Paramètre	Description	Valeur par défaut
dataLIF	Adresse IP de la LIF de protocole. NetApp recommande de spécifier dataLIF. Si non fourni, Trident récupère les LIFs de données du SVM. Vous pouvez spécifier un nom de domaine complet (FQDN) à utiliser pour les opérations de montage NFS, ce qui vous permet de créer un DNS circulaire pour équilibrer la charge sur plusieurs dataLIFs. Peut être modifié après le réglage initial. Reportez-vous à la . Peut être configuré pour utiliser des adresses IPv6 si Trident a été installé à l'aide de l'indicateur IPv6. Les adresses IPv6 doivent être définies entre crochets, par exemple [28e8:d9fb:a825:b7bf:69a8:d02f:9e7b:3555]. <b>Omettre pour MetroCluster.</b> Voir la <a href="#">Exemple MetroCluster</a> .	Adresse spécifiée ou dérivée d'un SVM, si non spécifiée (non recommandé)
svm	Machine virtuelle de stockage à utiliser <b>omet pour MetroCluster.</b> Voir la <a href="#">Exemple MetroCluster</a> .	Dérivé si un SVM managementLIF est spécifié
autoExportPolicy	Activer la création et la mise à jour automatiques des règles d'exportation [booléennes]. Grâce aux autoExportPolicy options et autoExportCIDRs, Trident peut gérer automatiquement les règles d'export.	faux
autoExportCIDRs	Liste des CIDR permettant de filtrer les adresses IP des nœuds Kubernetes par rapport à lorsque autoExportPolicy est activé. Grâce aux autoExportPolicy options et autoExportCIDRs, Trident peut gérer automatiquement les règles d'export.	["0.0.0.0/0", ":/0"]
labels	Ensemble d'étiquettes arbitraires au format JSON à appliquer aux volumes	« »
clientCertificate	Valeur encodée en Base64 du certificat client. Utilisé pour l'authentification par certificat	« »
clientPrivateKey	Valeur encodée en Base64 de la clé privée du client. Utilisé pour l'authentification par certificat	« »
trustedCACertificate	Valeur encodée en Base64 du certificat CA de confiance. Facultatif. Utilisé pour l'authentification par certificat	« »
username	Nom d'utilisateur pour la connexion au cluster/SVM. Utilisé pour l'authentification basée sur les identifiants	
password	Mot de passe pour la connexion au cluster/SVM. Utilisé pour l'authentification basée sur les identifiants	

Paramètre	Description	Valeur par défaut
storagePrefix	<p>Préfixe utilisé pour le provisionnement des nouveaux volumes dans la SVM. Ne peut pas être mis à jour une fois que vous l'avez défini</p> <p> Si vous utilisez ONTAP-nas-Economy et un préfixe de stockage de 24 caractères ou plus, le préfixe de stockage n'est pas intégré dans les qtrees, même s'il figure dans le nom du volume.</p>	« trident »
aggregate	<p>Agrégat pour le provisionnement (facultatif ; si défini, doit être attribué au SVM) Pour le <code>ontap-nas-flexgroup</code> pilote, cette option est ignorée. S'ils ne sont pas affectés, les agrégats disponibles peuvent être utilisés pour provisionner un volume FlexGroup.</p> <p> Lorsque l'agrégat est mis à jour au SVM, il est mis à jour automatiquement dans Trident par SVM d'interrogation sans avoir à redémarrer le contrôleur Trident. Lorsque vous avez configuré un agrégat spécifique dans Trident pour provisionner des volumes, si l'agrégat est renommé ou déplacé hors du SVM, le back-end passe à l'état Failed dans Trident lors de l'interrogation de l'agrégat du SVM. Il faut remplacer l'agrégat par un agrégat présent sur la SVM ou le retirer complètement pour remettre le back-end en ligne.</p>	« »
limitAggregateUsage	Echec du provisionnement si l'utilisation est supérieure à ce pourcentage. <b>Ne s'applique pas à Amazon FSX pour ONTAP</b>	« » (non appliqué par défaut)

Paramètre	Description	Valeur par défaut
FlexgroupAggregateList	<p>Liste des agrégats pour le provisionnement (facultatif ; si défini, doit être affecté au SVM) Tous les agrégats affectés au SVM sont utilisés pour provisionner un volume FlexGroup. Pris en charge pour le pilote de stockage <b>ONTAP-nas-FlexGroup</b>.</p> <p> Lorsque la liste des agrégats est mise à jour au SVM, elle est mise à jour automatiquement dans Trident par SVM d'interrogation sans devoir redémarrer le contrôleur Trident. Lorsque vous avez configuré une liste d'agrégats spécifique dans Trident pour provisionner des volumes, si la liste d'agrégats est renommée ou déplacée hors du SVM, le back-end passe à l'état Failed dans Trident lors de l'interrogation de l'agrégat du SVM. Il faut remplacer la liste des agrégats par une liste présente sur la SVM ou la supprimer définitivement pour remettre le système back-end en ligne.</p>	« »
limitVolumeSize	Echec du provisionnement si la taille du volume demandé est supérieure à cette valeur. Limite également la taille maximale des volumes gérés pour les qtrees et qtreesPerFlexvol permet de personnaliser le nombre maximal de qtrees par FlexVol volume	« » (non appliqué par défaut)
debugTraceFlags	Indicateurs de débogage à utiliser lors du dépannage. Exemple, {"api":false, "method":true} n'utilisez debugTraceFlags que si vous effectuez un dépannage et que vous avez besoin d'un vidage de journal détaillé.	nul
nasType	Configurez la création de volumes NFS ou SMB. Les options sont nfs, smb ou NULL. La valeur null par défaut sur les volumes NFS.	nfs
nfsMountOptions	Liste des options de montage NFS séparée par des virgules. Les options de montage des volumes persistants Kubernetes sont normalement spécifiées dans les classes de stockage, mais si aucune option de montage n'est spécifiée dans une classe de stockage, Trident revient à utiliser les options de montage spécifiées dans le fichier de configuration du back-end de stockage. Si aucune option de montage n'est spécifiée dans la classe de stockage ou le fichier de configuration, Trident ne définit aucune option de montage sur un volume persistant associé.	« »

Paramètre	Description	Valeur par défaut
qtreesPerFlexVol	Nombre maximal de qtrees par FlexVol, qui doit être compris dans la plage [50, 300]	« 200 »
smbShare	Vous pouvez spécifier l'une des options suivantes : le nom d'un partage SMB créé à l'aide de la console de gestion Microsoft ou de l'interface de ligne de commande ONTAP ; un nom permettant à Trident de créer le partage SMB ; ou bien laisser le paramètre vide pour empêcher l'accès au partage commun aux volumes. Ce paramètre est facultatif pour les ONTAP sur site. Ce paramètre est requis pour Amazon FSX pour les systèmes back-end ONTAP et ne peut pas être vide.	smb-share
useREST	Paramètre booléen pour utiliser les API REST de ONTAP. <code>useREST</code> Lorsqu'il est défini sur <code>true</code> , Trident utilise les API REST ONTAP pour communiquer avec le back-end ; lorsqu'il est défini sur <code>false</code> , Trident utilise les appels ONTAPI (ZAPI) pour communiquer avec le back-end. Cette fonctionnalité requiert ONTAP 9.11.1 et versions ultérieures. En outre, le rôle de connexion ONTAP utilisé doit avoir accès à l' <code>ontapi</code> application. Ceci est satisfait par les rôles et prédéfinis <code>vsadmin</code> <code>cluster-admin</code> . À partir de la version Trident 24.06 et de ONTAP 9.15.1 ou version ultérieure, <code>useREST</code> est défini sur <code>true</code> par défaut ; passez <code>useREST</code> à <code>false</code> pour utiliser les appels ONTAPI (ZAPI).	<code>true</code> Pour ONTAP 9.15.1 ou version ultérieure, sinon <code>false</code> .
limitVolumePoolSize	Taille de FlexVol maximale requise lors de l'utilisation de qtrees dans le back-end ONTAP-nas-Economy.	« » (non appliqué par défaut)
denyNewVolumePools	Empêche les <code>ontap-nas-economy</code> systèmes back-end de créer de nouveaux volumes FlexVol pour contenir leurs qtrees. Seuls les volumes FlexVol préexistants sont utilisés pour provisionner les nouveaux volumes persistants.	

#### Options de configuration back-end pour les volumes de provisionnement

Vous pouvez contrôler le provisionnement par défaut à l'aide de ces options dans la `defaults` section de la configuration. Pour un exemple, voir les exemples de configuration ci-dessous.

Paramètre	Description	Valeur par défaut
spaceAllocation	Allocation d'espace pour les qtrees	« vrai »
spaceReserve	Mode de réservation d'espace ; « aucun » (fin) ou « volume » (épais)	« aucun »
snapshotPolicy	Règle Snapshot à utiliser	« aucun »

Paramètre	Description	Valeur par défaut
qosPolicy	QoS policy group à affecter pour les volumes créés. Choisissez une de qosPolicy ou adaptiveQosPolicy par pool de stockage/back-end	« »
adaptiveQosPolicy	Groupe de règles de QoS adaptative à attribuer aux volumes créés. Choisissez une de qosPolicy ou adaptiveQosPolicy par pool de stockage/back-end. Non pris en charge par l'économie ontap-nas.	« »
snapshotReserve	Pourcentage de volume réservé pour les snapshots	« 0 » si snapshotPolicy est « aucun », sinon « »
splitOnClone	Séparer un clone de son parent lors de sa création	« faux »
encryption	Activez le chiffrement de volume NetApp (NVE) sur le nouveau volume. La valeur par défaut est <code>false</code> . Pour utiliser cette option, NVE doit être sous licence et activé sur le cluster. Si NAE est activé sur le back-end, tout volume provisionné dans Trident est activé. Pour plus d'informations, reportez-vous à la section : " <a href="#">Fonctionnement de Trident avec NVE et NAE</a> ".	« faux »
tieringPolicy	Règle de hiérarchisation à utiliser « aucun »	
unixPermissions	Mode pour les nouveaux volumes	« 777 » pour les volumes NFS ; vide (non applicable) pour les volumes SMB
snapshotDir	Contrôle l'accès au <code>.snapshot</code> répertoire	« True » pour NFSv4 « false » pour NFSv3
exportPolicy	Export policy à utiliser	« par défaut »
securityStyle	Style de sécurité pour les nouveaux volumes. Prise en charge NFS <code>mixed</code> et unix styles de sécurité. Prise en charge SMB <code>mixed</code> et ntfs styles de sécurité.	NFS la valeur par défaut est <code>unix</code> . SMB la valeur par défaut est <code>ntfs</code> .
nameTemplate	Modèle pour créer des noms de volume personnalisés.	« »

 L'utilisation de groupes de règles de qualité de service avec Trident nécessite ONTAP 9.8 ou une version ultérieure. Vous devez utiliser un groupe de règles QoS non partagé et vous assurer que le groupe de règles est appliqué à chaque composant individuellement. Un groupe de règles de QoS partagées applique le débit total de toutes les charges de travail.

## Exemples de provisionnement de volumes

Voici un exemple avec des valeurs par défaut définies :

```

---
version: 1
storageDriverName: ontap-nas
backendName: customBackendName
managementLIF: 10.0.0.1
dataLIF: 10.0.0.2
labels:
  k8scluster: dev1
  backend: dev1-nasbackend
svm: trident_svm
username: cluster-admin
password: <password>
limitAggregateUsage: 80%
limitVolumeSize: 50Gi
nfsMountOptions: nfsvers=4
debugTraceFlags:
  api: false
  method: true
defaults:
  spaceReserve: volume
  qosPolicy: premium
  exportPolicy: myk8scluster
  snapshotPolicy: default
  snapshotReserve: "10"

```

Pour `ontap-nas` et `ontap-nas-flexgroups`, Trident utilise maintenant un nouveau calcul pour s'assurer que le FlexVol est correctement dimensionné avec le pourcentage `snapshotReserve` et le PVC. Lorsque l'utilisateur demande une demande de volume persistant, Trident crée la FlexVol d'origine avec plus d'espace en utilisant le nouveau calcul. Ce calcul garantit que l'utilisateur reçoit l'espace inscriptible demandé dans la demande de volume persistant et qu'il ne dispose pas d'un espace minimal par rapport à ce qu'il a demandé. Avant le 21.07, lorsque l'utilisateur demande une demande de volume persistant (par exemple, 5 Gio), et le `snapshotReserve` à 50 %, ils ne bénéficient que d'un espace inscriptible de 2,5 Gio. En effet, ce que l'utilisateur a demandé est le volume entier et est un pourcentage de ce volume `snapshotReserve`. Avec Trident 21.07, l'utilisateur demande l'espace inscriptible, et Trident définit le `snapshotReserve` nombre comme le pourcentage du volume dans son ensemble. Cela ne s'applique pas à `ontap-nas-economy`. Voir l'exemple suivant pour voir comment cela fonctionne :

Le calcul est le suivant :

```
Total volume size = (PVC requested size) / (1 - (snapshotReserve percentage) / 100)
```

Pour les snapshots Reserve = 50 %, et demande en volume PVC = 5 Gio, la taille totale du volume est  $5/0,5 = 10$  Gio et la taille disponible est de 5 Gio, ce que l'utilisateur a demandé dans la demande de volume persistant. `volume show` La commande doit afficher des résultats similaires à cet exemple :

Vserver	Volume	Aggregate	State	Type	Size	Available	Used%
	_pvc_89f1c156_3801_4de4_9f9d_034d54c395f4		online	RW	10GB	5.00GB	0%
	_pvc_e8372153_9ad9_474a_951a_08ae15e1c0ba		online	RW	1GB	511.8MB	0%
2 entries were displayed.							

Les systèmes back-end des installations précédentes provisionnent les volumes comme expliqué ci-dessus lors de la mise à niveau de Trident. Pour les volumes que vous avez créés avant la mise à niveau, vous devez redimensionner leurs volumes afin que la modification puisse être observée. Par exemple, une demande de volume persistant de 2 Gio associée à `snapshotReserve=50` la précédente a donné lieu à un volume qui fournit 1 Gio d'espace inscriptible. Le redimensionnement du volume à 3 Gio, par exemple, fournit l'application avec 3 Gio d'espace inscriptible sur un volume de 6 Gio.

#### Exemples de configuration minimaux

Les exemples suivants montrent des configurations de base qui laissent la plupart des paramètres par défaut. C'est la façon la plus simple de définir un back-end.



Si vous utilisez Amazon FSX sur NetApp ONTAP avec Trident, nous vous recommandons de spécifier des noms DNS pour les LIF au lieu d'adresses IP.

#### Exemple d'économie NAS ONTAP

```
---
version: 1
storageDriverName: ontap-nas-economy
managementLIF: 10.0.0.1
dataLIF: 10.0.0.2
svm: svm_nfs
username: vsadmin
password: password
```

#### Exemple de FlexGroup NAS ONTAP

```
---
version: 1
storageDriverName: ontap-nas-flexgroup
managementLIF: 10.0.0.1
dataLIF: 10.0.0.2
svm: svm_nfs
username: vsadmin
password: password
```

## Exemple MetroCluster

Vous pouvez configurer le back-end pour éviter de devoir mettre à jour manuellement la définition du back-end après le basculement et le rétablissement pendant "[RéPLICATION ET RESTAURATION DES SVM](#)".

Pour un basculement et un rétablissement transparents, préciser le SVM en utilisant `managementLIF` et omettre les `dataLIF` paramètres et `svm`. Par exemple :

```
---  
version: 1  
storageDriverName: ontap-nas  
managementLIF: 192.168.1.66  
username: vsadmin  
password: password
```

## Exemple de volumes SMB

```
---  
version: 1  
backendName: ExampleBackend  
storageDriverName: ontap-nas  
managementLIF: 10.0.0.1  
nasType: smb  
securityStyle: ntfs  
unixPermissions: ""  
dataLIF: 10.0.0.2  
svm: svm_nfs  
username: vsadmin  
password: password
```

## Exemple d'authentification basée sur un certificat

Il s'agit d'un exemple de configuration back-end minimale. `clientCertificate`, `clientPrivateKey` Et `trustedCACertificate` (facultatif, si vous utilisez une autorité de certification approuvée) sont renseignés `backend.json` et prennent respectivement les valeurs codées en base64 du certificat client, de la clé privée et du certificat de l'autorité de certification approuvée.

```
---  
version: 1  
backendName: DefaultNASBackend  
storageDriverName: ontap-nas  
managementLIF: 10.0.0.1  
dataLIF: 10.0.0.15  
svm: nfs_svm  
clientCertificate: ZXROZXJwYXB...ICMgJ3BhcGVyc2  
clientPrivateKey: vciwKIyAgZG...0cnksIGRlc2NyaX  
trustedCACertificate: zcyBbaG...b3Igb3duIGNsYXNz  
storagePrefix: myPrefix_
```

## Exemple de règle d'export automatique

Cet exemple montre comment vous pouvez demander à Trident d'utiliser des règles d'export dynamiques pour créer et gérer automatiquement les règles d'export. Cela fonctionne de la même manière pour les `ontap-nas-economy` pilotes et `ontap-nas-flexgroup`.

```
---  
version: 1  
storageDriverName: ontap-nas  
managementLIF: 10.0.0.1  
dataLIF: 10.0.0.2  
svm: svm_nfs  
labels:  
  k8scluster: test-cluster-east-1a  
  backend: test1-nasbackend  
autoExportPolicy: true  
autoExportCIDRs:  
- 10.0.0.0/24  
username: admin  
password: password  
nfsMountOptions: nfsvers=4
```

## Exemple d'adresses IPv6

Cet exemple illustre managementLIF l'utilisation d'une adresse IPv6.

```
---  
version: 1  
storageDriverName: ontap-nas  
backendName: nas_ipv6_backend  
managementLIF: "[5c5d:5edf:8f:7657:bef8:109b:1b41:d491]"  
labels:  
  k8scluster: test-cluster-east-1a  
  backend: test1-ontap-ipv6  
svm: nas_ipv6_svm  
username: vsadmin  
password: password
```

## Exemple d'Amazon FSX pour ONTAP avec des volumes SMB

Le smbShare paramètre est requis pour FSX for ONTAP utilisant des volumes SMB.

```
---  
version: 1  
backendName: SMBBackend  
storageDriverName: ontap-nas  
managementLIF: example.mgmt.fqdn.aws.com  
nasType: smb  
dataLIF: 10.0.0.15  
svm: nfs_svm  
smbShare: smb-share  
clientCertificate: ZXROZXJwYXB...ICMgJ3BhcGVyc2  
clientPrivateKey: vciwKIyAgZG...0cnksIGRlc2NyaX  
trustedCACertificate: zcyBbaG...b3Igb3duIGNsYXNz  
storagePrefix: myPrefix_
```

## Exemple de configuration back-end avec nomTemplate

```
---  
version: 1  
storageDriverName: ontap-nas  
backendName: ontap-nas-backend  
managementLIF: <ip address>  
svm: svm0  
username: <admin>  
password: <password>  
defaults:  
  nameTemplate:  
    "{{.volume.Name}}_{{.labels.cluster}}_{{.volume.Namespace}}_{{.vo\\}}  
      lume.RequestName}"  
  labels:  
    cluster: ClusterA  
    PVC: "{{.volume.Namespace}}_{{.volume.RequestName}}"
```

## Exemples de systèmes back-end avec pools virtuels

Dans les exemples de fichiers de définition back-end présentés ci-dessous, des valeurs par défaut spécifiques sont définies pour tous les pools de stockage, comme spaceReserve aucun, spaceAllocation faux et encryption faux. Les pools virtuels sont définis dans la section stockage.

Trident définit les étiquettes de provisionnement dans le champ « Commentaires ». Les commentaires sont définis sur FlexVol pour ontap-nas ou FlexGroup pour ontap-nas-flexgroup. Trident copie toutes les étiquettes présentes sur un pool virtuel vers le volume de stockage lors du provisionnement. Pour plus de commodité, les administrateurs du stockage peuvent définir des étiquettes par pool virtuel et les volumes de groupe par étiquette.

Dans ces exemples, certains pools de stockage définissent leurs propres spaceReserve valeurs , spaceAllocation et, et encryption certains pools remplacent les valeurs par défaut.

## Exemple de NAS ONTAP

```
---  
version: 1  
storageDriverName: ontap-nas  
managementLIF: 10.0.0.1  
svm: svm_nfs  
username: admin  
password: <password>  
nfsMountOptions: nfsvers=4  
defaults:  
    spaceReserve: none  
    encryption: "false"  
    qosPolicy: standard  
labels:  
    store: nas_store  
    k8scluster: prod-cluster-1  
region: us_east_1  
storage:  
    - labels:  
        app: msoffice  
        cost: "100"  
        zone: us_east_1a  
        defaults:  
            spaceReserve: volume  
            encryption: "true"  
            unixPermissions: "0755"  
            adaptiveQosPolicy: adaptive-premium  
    - labels:  
        app: slack  
        cost: "75"  
        zone: us_east_1b  
        defaults:  
            spaceReserve: none  
            encryption: "true"  
            unixPermissions: "0755"  
    - labels:  
        department: legal  
        creditpoints: "5000"  
        zone: us_east_1b  
        defaults:  
            spaceReserve: none  
            encryption: "true"  
            unixPermissions: "0755"  
    - labels:  
        app: wordpress
```

```
cost: "50"
zone: us_east_1c
defaults:
  spaceReserve: none
  encryption: "true"
  unixPermissions: "0775"
- labels:
  app: mysqlDb
  cost: "25"
  zone: us_east_1d
  defaults:
    spaceReserve: volume
    encryption: "false"
    unixPermissions: "0775"
```

## Exemple de FlexGroup NAS ONTAP

```
---  
version: 1  
storageDriverName: ontap-nas-flexgroup  
managementLIF: 10.0.0.1  
svm: svm_nfs  
username: vsadmin  
password: <password>  
defaults:  
    spaceReserve: none  
    encryption: "false"  
labels:  
    store: flexgroup_store  
    k8scluster: prod-cluster-1  
region: us_east_1  
storage:  
    - labels:  
        protection: gold  
        creditpoints: "50000"  
        zone: us_east_1a  
        defaults:  
            spaceReserve: volume  
            encryption: "true"  
            unixPermissions: "0755"  
    - labels:  
        protection: gold  
        creditpoints: "30000"  
        zone: us_east_1b  
        defaults:  
            spaceReserve: none  
            encryption: "true"  
            unixPermissions: "0755"  
    - labels:  
        protection: silver  
        creditpoints: "20000"  
        zone: us_east_1c  
        defaults:  
            spaceReserve: none  
            encryption: "true"  
            unixPermissions: "0775"  
    - labels:  
        protection: bronze  
        creditpoints: "10000"  
        zone: us_east_1d  
        defaults:
```

```
spaceReserve: volume
encryption: "false"
unixPermissions: "0775"
```

## Exemple d'économie NAS ONTAP

```
---  
version: 1  
storageDriverName: ontap-nas-economy  
managementLIF: 10.0.0.1  
svm: svm_nfs  
username: vsadmin  
password: <password>  
defaults:  
    spaceReserve: none  
    encryption: "false"  
labels:  
    store: nas_economy_store  
region: us_east_1  
storage:  
    - labels:  
        department: finance  
        creditpoints: "6000"  
        zone: us_east_1a  
        defaults:  
            spaceReserve: volume  
            encryption: "true"  
            unixPermissions: "0755"  
    - labels:  
        protection: bronze  
        creditpoints: "5000"  
        zone: us_east_1b  
        defaults:  
            spaceReserve: none  
            encryption: "true"  
            unixPermissions: "0755"  
    - labels:  
        department: engineering  
        creditpoints: "3000"  
        zone: us_east_1c  
        defaults:  
            spaceReserve: none  
            encryption: "true"  
            unixPermissions: "0775"  
    - labels:  
        department: humanresource  
        creditpoints: "2000"  
        zone: us_east_1d  
        defaults:  
            spaceReserve: volume
```

```
  encryption: "false"
  unixPermissions: "0775"
```

## Mappage des systèmes back-end aux classes de stockage

Les définitions de classe de stockage suivantes font référence [Exemples de systèmes back-end avec pools virtuels](#). En utilisant ce parameters.selector champ, chaque classe de stockage indique quels pools virtuels peuvent être utilisés pour héberger un volume. Les aspects définis dans le pool virtuel sélectionné seront définis pour le volume.

- La protection-gold classe de stockage est mappée sur le premier et le second pool virtuel du ontap-nas-flexgroup back-end. Il s'agit des seuls pools offrant une protection de niveau Gold.

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: protection-gold
provisioner: csi.trident.netapp.io
parameters:
  selector: "protection=gold"
  fsType: "ext4"
```

- La protection-not-gold classe de stockage est mappée sur le troisième et le quatrième pool virtuel du ontap-nas-flexgroup back-end. Ce sont les seuls pools offrant un niveau de protection autre que l'or.

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: protection-not-gold
provisioner: csi.trident.netapp.io
parameters:
  selector: "protection!=gold"
  fsType: "ext4"
```

- La app-mysqldb classe de stockage est mappée sur le quatrième pool virtuel du ontap-nas back-end. Il s'agit du seul pool offrant la configuration du pool de stockage pour l'application de type mysqldb.

```

apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: app-mysqldb
provisioner: csi.trident.netapp.io
parameters:
  selector: "app=mysqldb"
  fsType: "ext4"

```

- La protection-silver-creditpoints-20k classe de stockage sera mappée sur le troisième pool virtuel du ontap-nas-flexgroup back-end. Il s'agit de la seule piscine offrant une protection de niveau argent et 20000 points de crédit.

```

apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: protection-silver-creditpoints-20k
provisioner: csi.trident.netapp.io
parameters:
  selector: "protection=silver; creditpoints=20000"
  fsType: "ext4"

```

- La creditpoints-5k classe de stockage est mappée sur le troisième pool virtuel du ontap-nas back-end et sur le second pool virtuel du ontap-nas-economy back-end. Il s'agit des seules offres de pool avec 5000 points de crédit.

```

apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: creditpoints-5k
provisioner: csi.trident.netapp.io
parameters:
  selector: "creditpoints=5000"
  fsType: "ext4"

```

Trident décide du pool virtuel sélectionné et s'assure que les besoins en stockage sont satisfaits.

#### **Mise à jour dataLIF après la configuration initiale**

Vous pouvez modifier la dataLIF après la configuration initiale en exécutant la commande suivante pour fournir le nouveau fichier JSON back-end avec une dataLIF mise à jour.

```
tridentctl update backend <backend-name> -f <path-to-backend-json-file-with-updated-dataLIF>
```



Si des ESV sont associées à un ou plusieurs pods, vous devez arrêter tous les pods correspondants, puis les remonter pour que la nouvelle dataLIF prenne effet.

## Amazon FSX pour NetApp ONTAP

### Utilisez Trident avec Amazon FSX pour NetApp ONTAP

"[Amazon FSX pour NetApp ONTAP](#)" Est un service AWS entièrement géré qui permet aux clients de lancer et d'exécuter des systèmes de fichiers optimisés par le système d'exploitation du stockage NetApp ONTAP. La solution FSX pour ONTAP vous permet d'exploiter les fonctionnalités, les performances et les capacités d'administration de NetApp que vous connaissez bien, tout en profitant de la simplicité, de l'agilité, de la sécurité et de l'évolutivité du stockage de données sur AWS. FSX pour ONTAP prend en charge les fonctionnalités du système de fichiers ONTAP et les API d'administration.

Vous pouvez intégrer votre système de fichiers Amazon FSX pour NetApp ONTAP avec Trident pour vous assurer que les clusters Kubernetes s'exécutant dans Amazon Elastic Kubernetes Service (EKS) peuvent provisionner des volumes persistants de bloc et de fichier soutenus par ONTAP.

Un système de fichiers est la ressource principale d'Amazon FSX, similaire à un cluster ONTAP sur site. Au sein de chaque SVM, vous pouvez créer un ou plusieurs volumes, qui sont des conteneurs de données qui stockent les fichiers et les dossiers dans votre système de fichiers. Avec Amazon FSX pour NetApp ONTAP sera fourni en tant que système de fichiers géré dans le cloud. Le nouveau type de système de fichiers est appelé **NetApp ONTAP**.

Grâce à Trident avec Amazon FSX pour NetApp ONTAP, vous pouvez vous assurer que les clusters Kubernetes s'exécutant dans Amazon Elastic Kubernetes Service (EKS) peuvent provisionner des volumes persistants de bloc et de fichier soutenus par ONTAP.

### De formation

En plus de "[Configuration requise pour Trident](#)", pour intégrer FSX for ONTAP avec Trident, vous avez besoin de :

- Un cluster Amazon EKS ou un cluster Kubernetes autogéré avec `kubectl` installé.
- Système de fichiers Amazon FSX for NetApp ONTAP et machine virtuelle de stockage (SVM) accessibles depuis les nœuds workers de votre cluster.
- Nœuds de travail préparés pour "[NFS ou iSCSI](#)".



Assurez-vous de suivre les étapes de préparation de nœud requises pour Amazon Linux et Ubuntu "[Images de machine Amazon](#)" (amis) en fonction de votre type d'ami EKS.

### Considérations

- Volumes SMB :

- Les volumes SMB sont pris en charge à l'aide du `ontap-nas` pilote uniquement.
- Les volumes SMB ne sont pas pris en charge par le module d'extension Trident EKS.
- Trident prend en charge les volumes SMB montés sur les pods s'exécutant sur les nœuds Windows uniquement. Voir "["Préparez-vous au provisionnement des volumes SMB"](#)" pour plus de détails.
- Avant Trident 24.02, les volumes créés sur les systèmes de fichiers Amazon FSX pour lesquels les sauvegardes automatiques sont activées ne pouvaient pas être supprimés par Trident. Pour éviter ce problème dans Trident 24.02 ou version ultérieure, spécifiez `fsxFilesystemID`, `AWS`, `AWS apiRegion` `apikey` et `AWS secretKey` dans le fichier de configuration back-end pour AWS FSX pour ONTAP.



Si vous spécifiez un rôle IAM dans Trident, vous pouvez omettre de spécifier explicitement les `apiRegion` champs, `apiKey` et `secretKey` dans Trident. Pour plus d'informations, reportez-vous "["Exemples et options de configuration de FSX pour ONTAP"](#)" à .

## Authentification

Trident propose deux modes d'authentification.

- Basé sur les informations d'identification (recommandé) : stocke les informations d'identification de manière sécurisée dans AWS secrets Manager. Vous pouvez utiliser `fsxadmin` l'utilisateur pour votre système de fichiers ou l' `vsadmin` utilisateur configuré pour votre SVM.



Trident s'attend à être exécuté en tant qu' `vsadmin`` utilisateur SVM ou en tant qu' utilisateur avec un nom différent qui a le même rôle. Amazon FSX pour NetApp ONTAP a un ``fsxadmin` utilisateur qui remplace de façon limitée l'utilisateur du cluster ONTAP `admin`. Nous vous recommandons vivement d'utiliser `vsadmin` Trident.

- Basé sur des certificats : Trident communiquera avec le SVM sur votre système de fichiers FSX à l'aide d'un certificat installé sur votre SVM.

Pour plus d'informations sur l'activation de l'authentification, reportez-vous à la section authentification de votre type de pilote :

- "["Authentification NAS de ONTAP"](#)"
- "["Authentification SAN de ONTAP"](#)"

## Ami (Amazon machine Images) testé

Le cluster EKS prend en charge plusieurs systèmes d'exploitation, mais AWS a optimisé certains ami (Amazon machine image) pour les conteneurs et EKS. Les amis suivants ont été testés avec Trident 24.10.

AMI	NAS	NAS-Economie	SAN	SAN-Economie
AL2023_x86_64_ST ANDARD	Oui	Oui	Oui	Oui
AL2_x86_64	Oui	Oui	Oui**	Oui**
BOTTLEROCKET_x86_64	Oui*	Oui	S/O	S/O

AL2023_ARM_64_STANDARD	Oui	Oui	Oui	Oui
AL2_ARM_64	Oui	Oui	Oui**	Oui**
BOTTLEROCKET_A_RM_64	Oui*	Oui	S/O	S/O

- \*Doit utiliser "nolock" dans les options de montage.
- \*\* Impossible de supprimer le PV sans redémarrer le nœud



Si votre ami souhaité n'est pas répertorié ici, cela ne signifie pas qu'il n'est pas pris en charge ; cela signifie simplement qu'il n'a pas été testé. Cette liste sert de guide pour les amis connus pour fonctionner.

#### Tests effectués avec :

- Version EKS : 1.30
- Méthode d'installation : Helm et en tant qu'add-on AWS
- Pour NAS, NFS v3 et NFS v4.1 ont été testés.
- Pour le SAN, iSCSI uniquement a été testé, pas NVMe-of.

#### Tests effectués :

- Créer : classe de stockage, pvc, pod
- Suppression : pod, pvc (normal, qtree/lun – économique, NAS avec sauvegarde AWS)

#### Trouvez plus d'informations

- "[Documentation Amazon FSX pour NetApp ONTAP](#)"
- "[Billet de blog sur Amazon FSX pour NetApp ONTAP](#)"

#### Créez un rôle IAM et un code secret AWS

Vous pouvez configurer les pods Kubernetes pour accéder aux ressources AWS en vous authentifiant en tant que rôle IAM AWS au lieu de fournir des informations d'identification AWS explicites.



Pour vous authentifier à l'aide d'un rôle IAM AWS, un cluster Kubernetes doit être déployé à l'aide d'EKS.

#### Créez un secret AWS secrets Manager

Comme Trident émettra des API pour un vServer FSX afin de gérer le stockage pour vous, il aura besoin d'informations d'identification pour le faire. La façon sécurisée de transmettre ces informations d'identification est de passer par un secret AWS secrets Manager. Par conséquent, si vous n'en avez pas déjà un, vous devrez créer un secret AWS secrets Manager contenant les informations d'identification du compte vsadmin.

Cet exemple crée un secret AWS secrets Manager pour stocker les informations d'identification Trident CSI :

```
aws secretsmanager create-secret --name trident-secret --description "Trident CSI credentials"\n    --secret-string\n    "{\"username\":\"vsadmin\",\"password\":\"<svmpassword>\"}\"\n
```

### Créer une politique IAM

Trident a également besoin des autorisations AWS pour s'exécuter correctement. Par conséquent, vous devez créer une stratégie qui donne à Trident les autorisations dont elle a besoin.

Les exemples suivants créent une politique IAM à l'aide de l'interface de ligne de commande AWS :

```
aws iam create-policy --policy-name AmazonFSxNCSIReaderPolicy --policy\n    -document file://policy.json\n        --description "This policy grants access to Trident CSI to FSxN and\n        Secrets manager"
```

### Exemple JSON de règles :

```

{
  "Statement": [
    {
      "Action": [
        "fsx:DescribeFileSystems",
        "fsx:DescribeVolumes",
        "fsx>CreateVolume",
        "fsx:RestoreVolumeFromSnapshot",
        "fsx:DescribeStorageVirtualMachines",
        "fsx:UntagResource",
        "fsx:UpdateVolume",
        "fsx:TagResource",
        "fsx:DeleteVolume"
      ],
      "Effect": "Allow",
      "Resource": "*"
    },
    {
      "Action": "secretsmanager:GetSecretValue",
      "Effect": "Allow",
      "Resource": "arn:aws:secretsmanager:<aws-region>:<aws-account-
id>:secret:<aws-secret-manager-name>*"
    }
  ],
  "Version": "2012-10-17"
}

```

## Créer un rôle IAM pour le compte de service

Une fois la stratégie créée, utilisez-la lors de la création du rôle qui sera affecté au compte de service sous lequel Trident s'exécutera :

## CLI AWS

```
aws iam create-role --role-name AmazonEKS_FSxN_CSI_DriverRole \
--assume-role-policy-document file://trust-relationship.json
```

### fichier trust-relationship.json :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Federated": "arn:aws:iam::<account_id>:oidc-provider/<oidc_provider>"
      },
      "Action": "sts:AssumeRoleWithWebIdentity",
      "Condition": {
        "StringEquals": {
          "<oidc_provider>:aud": "sts.amazonaws.com",
          "<oidc_provider>:sub": "system:serviceaccount:trident:trident-controller"
        }
      }
    }
  ]
}
```

Mettez à jour les valeurs suivantes dans le trust-relationship.json fichier :

- <account\_id> - votre ID de compte AWS
- <oidc\_provider> - l'OIDC de votre cluster EKS. Vous pouvez obtenir le fournisseur oidc\_Provider en exécutant :

```
aws eks describe-cluster --name my-cluster --query
"cluster.identity.oidc.issuer"\n
--output text | sed -e "s/^https:\/\/\///"
```

### Joindre le rôle IAM à la politique IAM :

Une fois le rôle créé, reliez la stratégie (créeée à l'étape ci-dessus) au rôle à l'aide de la commande suivante :

```
aws iam attach-role-policy --role-name my-role --policy-arn <IAM policy ARN>
```

#### Vérifier que le fournisseur OIDC est associé :

Vérifiez que votre fournisseur OIDC est associé à votre cluster. Vous pouvez le vérifier à l'aide de la commande suivante :

```
aws iam list-open-id-connect-providers | grep $oidc_id | cut -d "/" -f4
```

Si la sortie est vide, utiliser la commande suivante pour associer IAM OIDC à votre cluster :

```
eksctl utils associate-iam-oidc-provider --cluster $cluster_name  
--approve
```

#### eksctl

L'exemple suivant crée un rôle IAM pour le compte de service dans EKS :

```
eksctl create iamserviceaccount --name trident-controller --namespace  
trident \  
--cluster <my-cluster> --role-name AmazonEKS_FSxN_CSI_DriverRole  
--role-only \  
--attach-policy-arn <IAM-Policy ARN> --approve
```

#### Installation de Trident

Trident rationalise la gestion du stockage Amazon FSX for NetApp ONTAP dans Kubernetes pour que vos développeurs et administrateurs puissent donner la priorité au déploiement d'applications.

Vous pouvez installer Trident à l'aide de l'une des méthodes suivantes :

- Gouvernail
- Module complémentaire EKS

Si vous souhaitez utiliser la fonctionnalité snapshot, installez le module complémentaire CSI snapshot Controller. Pour plus d'informations, reportez-vous à la section "[Activer la fonctionnalité snapshot pour les volumes CSI](#)" .

#### Installez Trident via Helm

1. Téléchargez le programme d'installation de Trident

Le programme d'installation de Trident contient tout ce dont vous avez besoin pour déployer l'opérateur

Trident et installer Trident. Téléchargez et extrayez la dernière version du programme d'installation de Trident à partir de la section Ressources sur GitHub.

```
wget  
https://github.com/NetApp/trident/releases/download/v25.02.0/trident-  
installer-25.02.0.tar.gz  
tar -xf trident-installer-25.02.0.tar.gz  
cd trident-installer
```

2. Définissez les valeurs des indicateurs **cloud Provider** et **cloud Identity** à l'aide des variables d'environnement suivantes :

L'exemple suivant installe Trident et définit l'`cloud-provider` `indicateur sur `$CP`, et `cloud-identity` sur `$CI`:

```
helm install trident trident-operator-100.2502.0.tgz \  
--set cloudProvider="AWS" \  
--set cloudIdentity="'eks.amazonaws.com/role-arn:  
arn:aws:iam::<accountID>:role/<AmazonEKS_FSxN_CSI_DriverRole>'" \  
--namespace trident \  
--create-namespace
```

Vous pouvez utiliser `helm list` la commande pour consulter les détails de l'installation tels que le nom, l'espace de noms, le graphique, l'état, la version de l'application et le numéro de révision.

```
helm list -n trident
```

NAME	NAMESPACE	REVISION	UPDATED
STATUS	CHART		APP VERSION
trident-operator	trident	1	2024-10-14 14:31:22.463122
+0300 IDT	deployed	trident-operator-100.2502.0	25.02.0

### Installez Trident via le module complémentaire EKS

Le module complémentaire Trident EKS inclut les derniers correctifs de sécurité et de bogues, et est validé par AWS pour une utilisation avec Amazon EKS. Le module complémentaire EKS vous permet de vous assurer de manière cohérente que vos clusters Amazon EKS sont sécurisés et stables et de réduire la quantité de travail à effectuer pour installer, configurer et mettre à jour des modules complémentaires.

### Prérequis

Vérifiez les points suivants avant de configurer le module complémentaire Trident pour AWS EKS :

- Un compte de cluster Amazon EKS avec abonnement complémentaire
- Autorisations AWS sur AWS Marketplace :  
  "aws-marketplace:ViewSubscriptions",  
  "aws-marketplace:Subscribe",  
  "aws-marketplace:Unsubscribe"
- Type ami : Amazon Linux 2 (AL2\_x86\_64) ou Amazon Linux 2 ARM (AL2\_ARM\_64)
- Type de nœud : AMD ou ARM
- Un système de fichiers Amazon FSX pour NetApp ONTAP

#### **Activez le module complémentaire Trident pour AWS**

## eksctl

L'exemple de commande suivant installe le module complémentaire Trident EKS :

```
eksctl create addon --name netapp_trident-operator --cluster  
<cluster_name> \  
--service-account-role-arn arn:aws:iam::<account_id>:role/<role_name>  
--force
```

## Console de gestion

1. Ouvrez la console Amazon EKS à l'adresse <https://console.aws.amazon.com/eks/home#/clusters>.
2. Dans le volet de navigation de gauche, sélectionnez **clusters**.
3. Sélectionnez le nom du cluster pour lequel vous souhaitez configurer le module complémentaire NetApp Trident CSI.
4. Sélectionnez **Compléments**, puis **obtenir plus de modules complémentaires**.
5. Sur la page **Select add-ons**, procédez comme suit :
  - a. Dans la section EKS-addons du Marketplace AWS, cochez la case **Trident by NetApp**.
  - b. Sélectionnez **Suivant**.
6. Sur la page **configurer les compléments sélectionnés**, procédez comme suit :
  - a. Sélectionnez la **version** que vous souhaitez utiliser.
  - b. Pour **Sélectionner le rôle IAM**, laisser à **non défini**.
  - c. Suivez le schéma de configuration **Add-on** et définissez le paramètre **configurationValues** dans la section **Configuration Values** sur le fil de rôle que vous avez créé à l'étape précédente (la valeur doit être au format suivant :

```
{  
  
    "cloudIdentity": "'eks.amazonaws.com/role-arn: <role ARN>'"  
  
}
```

Si vous sélectionnez remplacer pour la méthode de résolution des conflits, un ou plusieurs des paramètres du module complémentaire existant peuvent être remplacés par les paramètres du module complémentaire Amazon EKS. Si vous n'activez pas cette option et qu'il y a un conflit avec vos paramètres existants, l'opération échoue. Vous pouvez utiliser le message d'erreur qui en résulte pour résoudre le conflit. Avant de sélectionner cette option, assurez-vous que le module complémentaire Amazon EKS ne gère pas les paramètres que vous devez gérer vous-même.

7. Choisissez **Suivant**.
8. Sur la page **consulter et ajouter**, choisissez **Créer**.

Une fois l'installation du module complémentaire terminée, le module complémentaire installé s'affiche.

## CLI AWS

- Créez le add-on.json fichier :

```
{  
    "clusterName": "<eks-cluster>",  
    "addonName": "netapp_trident-operator",  
    "addonVersion": "v25.02.1-eksbuild.1",  
    "serviceAccountRoleArn": "<role ARN>",  
    "configurationValues": {  
        "cloudIdentity": "'eks.amazonaws.com/role-arn: <role ARN>'",  
        "cloudProvider": "AWS"  
    }  
}
```



Remplacer <role ARN> par l'ARN du rôle créé à l'étape précédente.

- Installez le module complémentaire Trident EKS.

```
aws eks create-addon --cli-input-json file://add-on.json
```

## Mettez à jour le module complémentaire Trident EKS

## eksctl

- Vérifiez la version actuelle de votre module complémentaire FSxN Trident CSI. Remplacez `my-cluster` par le nom de votre cluster.

```
eksctl get addon --name netapp_trident-operator --cluster my-cluster
```

### Exemple de sortie :

NAME	VERSION	STATUS	ISSUES
IAMROLE	UPDATE AVAILABLE	CONFIGURATION VALUES	
netapp_trident-operator	v25.02.1-eksbuild.1	ACTIVE	0
{ "cloudIdentity": "'eks.amazonaws.com/role-arn:arn:aws:iam::139763910815:role/AmazonEKS_FSXN_CSI_DriverRole'" }			

- Mettez à jour le complément à la version renvoyée sous **MISE À JOUR DISPONIBLE** dans la sortie de l'étape précédente.

```
eksctl update addon --name netapp_trident-operator --version v25.02.1-eksbuild.1 --cluster my-cluster --force
```

Si vous supprimez l' `--force` option et que l'un des paramètres du module complémentaire Amazon EKS entre en conflit avec vos paramètres existants, la mise à jour du module complémentaire Amazon EKS échoue ; un message d'erreur s'affiche pour vous aider à résoudre le conflit. Avant de spécifier cette option, assurez-vous que le module complémentaire Amazon EKS ne gère pas les paramètres que vous devez gérer, car ces paramètres sont remplacés par cette option. Pour plus d'informations sur les autres options de ce paramètre, reportez-vous à la section "[Addons](#)". Pour plus d'informations sur la gestion de terrain Amazon EKS Kubernetes, reportez-vous à la section "[Gestion de terrain Kubernetes](#)".

## Console de gestion

1. Ouvrez la console Amazon EKS <https://console.aws.amazon.com/eks/home#/clusters>.
2. Dans le volet de navigation de gauche, sélectionnez **clusters**.
3. Sélectionnez le nom du cluster pour lequel vous souhaitez mettre à jour le module complémentaire NetApp Trident CSI.
4. Sélectionnez l'onglet **Compléments**.
5. Sélectionnez **Trident by NetApp**, puis **Edit**.
6. Sur la page **configurer Trident par NetApp**, procédez comme suit :
  - a. Sélectionnez la **version** que vous souhaitez utiliser.
  - b. Développez les **Paramètres de configuration facultatifs** et modifiez-les si nécessaire.
  - c. Sélectionnez **Enregistrer les modifications**.

## CLI AWS

L'exemple suivant met à jour le module complémentaire EKS :

```
aws eks update-addon --cluster-name my-cluster netapp_trident-operator
vpc-cni --addon-version v25.02.1-eksbuild.1 \
--service-account-role-arn <role-ARN> --configuration-values '{}'
--resolve-conflicts --preserve
```

## Désinstallez/supprimez le module complémentaire Trident EKS

Vous avez deux options pour supprimer un module complémentaire Amazon EKS :

- **Préserver le logiciel complémentaire sur votre cluster** – cette option supprime la gestion Amazon EKS de tous les paramètres. Il supprime également la possibilité pour Amazon EKS de vous informer des mises à jour et de mettre à jour automatiquement le module complémentaire Amazon EKS après avoir lancé une mise à jour. Cependant, il conserve le logiciel complémentaire sur votre cluster. Cette option fait du complément une installation auto-gérée, plutôt qu'un module complémentaire Amazon EKS. Avec cette option, vous n'avez plus à subir de temps d'indisponibilité. Conservez `--preserve` l'option dans la commande pour conserver le complément.
- **Supprimer entièrement le logiciel complémentaire de votre cluster** – NetApp vous recommande de supprimer le module complémentaire Amazon EKS de votre cluster uniquement s'il n'y a pas de ressources qui en dépendent sur votre cluster. Supprimez l'`--preserve` option de la `delete` commande pour supprimer le complément.



Si le complément est associé à un compte IAM, le compte IAM n'est pas supprimé.

## eksctl

La commande suivante désinstalle le module complémentaire Trident EKS :

```
eksctl delete addon --cluster K8s-arm --name netapp_trident-operator
```

## Console de gestion

1. Ouvrez la console Amazon EKS à l'adresse <https://console.aws.amazon.com/eks/home#/clusters>.
2. Dans le volet de navigation de gauche, sélectionnez **clusters**.
3. Sélectionnez le nom du cluster pour lequel vous souhaitez supprimer le module complémentaire NetApp Trident CSI.
4. Sélectionnez l'onglet **Compléments**, puis sélectionnez **Trident by NetApp**.\*
5. Sélectionnez **Supprimer**.
6. Dans la boîte de dialogue **Remove netapp\_trident-operator confirmation**, procédez comme suit :
  - a. Si vous souhaitez qu'Amazon EKS cesse de gérer les paramètres du module complémentaire, sélectionnez **préserver sur le cluster**. Procédez ainsi si vous souhaitez conserver l'extension logicielle sur votre cluster afin de pouvoir gérer tous les paramètres du module complémentaire vous-même.
  - b. Entrez **netapp\_trident-operator**.
  - c. Sélectionnez **Supprimer**.

## CLI AWS

Remplacez `my-cluster` par le nom de votre cluster, puis exécutez la commande suivante.

```
aws eks delete-addon --cluster-name my-cluster --addon-name  
netapp_trident-operator --preserve
```

## Configurez le back-end de stockage

### Intégration des pilotes SAN et NAS de ONTAP

Pour créer un back-end de stockage, vous devez créer un fichier de configuration au format JSON ou YAML. Le fichier doit spécifier le type de stockage souhaité (NAS ou SAN), le système de fichiers et le SVM pour le récupérer et comment s'authentifier auprès de lui. L'exemple suivant montre comment définir un stockage NAS et utiliser un secret AWS pour stocker les identifiants de la SVM que vous souhaitez utiliser :

## YAML

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-ontap-nas
  namespace: trident
spec:
  version: 1
  storageDriverName: ontap-nas
  backendName: tbc-ontap-nas
  svm: svm-name
  aws:
    fsxFilesystemID: fs-xxxxxxxxxx
  credentials:
    name: "arn:aws:secretsmanager:us-west-2:xxxxxxxx:secret:secret-
name"
    type: awsarn
```

## JSON

```
{
  "apiVersion": "trident.netapp.io/v1",
  "kind": "TridentBackendConfig",
  "metadata": {
    "name": "backend-tbc-ontap-nas",
    "namespace": "trident"
  },
  "spec": {
    "version": 1,
    "storageDriverName": "ontap-nas",
    "backendName": "tbc-ontap-nas",
    "svm": "svm-name",
    "aws": {
      "fsxFilesystemID": "fs-xxxxxxxxxx"
    },
    "managementLIF": null,
    "credentials": {
      "name": "arn:aws:secretsmanager:us-west-2:xxxxxxxx:secret:secret-
name",
      "type": "awsarn"
    }
  }
}
```

Exécutez les commandes suivantes pour créer et valider la configuration back-end Trident (TBC) :

- Créez la configuration Trident backend (TBC) à partir du fichier yaml et exécutez la commande suivante :

```
kubectl create -f backendconfig.yaml -n trident
```

```
tridentbackendconfig.trident.netapp.io/backend-tbc-ontap-nas created
```

- Vérifiez que la configuration du back-end Trident (TBC) a été créée avec succès :

```
Kubectl get tbc -n trident
```

NAME	PHASE	STATUS	BACKEND NAME	BACKEND UUID
backend-tbc-ontap-nas	b9ff-f96d916ac5e9	Bound	tbc-ontap-nas	933e0071-66ce-4324-

#### Détails du pilote FSX pour ONTAP

Vous pouvez intégrer Trident avec Amazon FSX for NetApp ONTAP à l'aide des pilotes suivants :

- **ontap-san**: Chaque volume persistant provisionné est un LUN au sein de son propre volume Amazon FSX pour NetApp ONTAP. Recommandé pour le stockage en mode bloc.
- **ontap-nas**: Chaque volume persistant provisionné est un volume Amazon FSX pour NetApp ONTAP complet. Recommandé pour les protocoles NFS et SMB.
- **ontap-san-economy**: Chaque volume persistant provisionné est une LUN avec un nombre configurable de LUN par volume Amazon FSX pour NetApp ONTAP.
- **ontap-nas-economy**: Chaque volume persistant provisionné est un qtree, avec un nombre configurable de qtree par volume Amazon FSX pour NetApp ONTAP.
- **ontap-nas-flexgroup**: Chaque volume persistant provisionné est un volume Amazon FSX pour NetApp ONTAP FlexGroup complet.

Pour plus de détails sur le pilote, reportez-vous à "[Pilotes NAS](#)" et "[Pilotes SAN](#)".

Une fois le fichier de configuration créé, exécutez cette commande pour le créer dans votre EKS :

```
kubectl create -f configuration_file
```

Pour vérifier le statut, lancer la commande suivante :

```
kubectl get tbc -n trident
```

NAME	BACKEND NAME	BACKEND UUID
PHASE	STATUS	
backend-fsx-ontap-nas	backend-fsx-ontap-nas	7a551921-997c-4c37-a1d1-
f2f4c87fa629	Bound	Success

### Configuration avancée back-end et exemples

Voir le tableau suivant pour les options de configuration du back-end :

Paramètre	Description	Exemple
version		Toujours 1
storageDriverName	Nom du pilote de stockage	ontap-nas ontap-nas-economy, ontap-nas-flexgroup, , ontap-san ontap-san-economy
backendName	Nom personnalisé ou système back-end de stockage	Nom du pilote + "_" + dataLIF
managementLIF	Adresse IP d'un cluster ou LIF de gestion De SVM Un nom de domaine complet (FQDN) peut être spécifié. Peut être configuré pour utiliser des adresses IPv6 si Trident a été installé à l'aide de l'indicateur IPv6. Les adresses IPv6 doivent être définies entre crochets, telles que [28e8:d9fb:a825:b7bf:69a8:d02f:9e7b:3555]. Si vous fournissez fsxFilesystemID sous le aws champ, il n'est pas nécessaire de fournir le managementLIF car Trident récupère les informations du SVM managementLIF auprès d'AWS. Donc, vous devez fournir des informations d'identification pour un utilisateur sous la SVM (par exemple : vsadmin) et l'utilisateur doit avoir le vsadmin rôle.	« 10.0.0.1 », « [2001:1234:abcd::fefe] »

Paramètre	Description	Exemple
dataLIF	<p>Adresse IP de la LIF de protocole.</p> <p><b>Pilotes NAS ONTAP:</b> NetApp recommande de spécifier dataLIF. Si non fourni, Trident récupère les LIFs de données du SVM. Vous pouvez spécifier un nom de domaine complet (FQDN) à utiliser pour les opérations de montage NFS, ce qui vous permet de créer un DNS circulaire pour équilibrer la charge sur plusieurs dataLIFs. Peut être modifié après le réglage initial. Reportez-vous à la . <b>Pilotes SAN ONTAP :</b> ne pas spécifier pour iSCSI. Trident utilise ONTAP Selective LUN Map pour découvrir les LIF iSCI nécessaires à l'établissement d'une session à chemins multiples. Un avertissement est généré si dataLIF est explicitement défini. Peut être configuré pour utiliser des adresses IPv6 si Trident a été installé à l'aide de l'indicateur IPv6. Les adresses IPv6 doivent être définies entre crochets, telles que [28e8:d9fb:a825:b7bf:69a8:d02f:9e7b:3555].</p>	
autoExportPolicy	Activer la création et la mise à jour automatiques des règles d'exportation [booléennes]. Grâce aux autoExportPolicy options et autoExportCIDRs, Trident peut gérer automatiquement les règles d'export.	false
autoExportCIDRs	Liste des CIDR permettant de filtrer les adresses IP des nœuds Kubernetes par rapport à lorsque autoExportPolicy est activé. Grâce aux autoExportPolicy options et autoExportCIDRs, Trident peut gérer automatiquement les règles d'export.	« [« 0.0.0.0/0 », «:/0 »] »
labels	Ensemble d'étiquettes arbitraires au format JSON à appliquer aux volumes	« »
clientCertificate	Valeur encodée en Base64 du certificat client. Utilisé pour l'authentification par certificat	« »

Paramètre	Description	Exemple
clientPrivateKey	Valeur encodée en Base64 de la clé privée du client. Utilisé pour l'authentification par certificat	« »
trustedCACertificate	Valeur encodée en Base64 du certificat CA de confiance. Facultatif. Utilisé pour l'authentification basée sur des certificats.	« »
username	Nom d'utilisateur pour la connexion au cluster ou au SVM. Utilisé pour l'authentification basée sur les identifiants. Par exemple, vsadmin.	
password	Mot de passe pour se connecter au cluster ou au SVM. Utilisé pour l'authentification basée sur les identifiants.	
svm	Serveur virtuel de stockage à utiliser	Dérivé si une LIF de gestion SVM est spécifiée.
storagePrefix	Préfixe utilisé pour le provisionnement des nouveaux volumes dans la SVM. Ne peut pas être modifié après sa création. Pour mettre à jour ce paramètre, vous devez créer un nouveau backend.	trident
limitAggregateUsage	<b>Ne spécifiez pas pour Amazon FSX pour NetApp ONTAP.</b> Les fournies fsxadmin et vsadmin ne contiennent pas les autorisations requises pour récupérer l'utilisation des agrégats et la limiter à l'aide de Trident.	Ne pas utiliser.
limitVolumeSize	Echec du provisionnement si la taille du volume demandé est supérieure à cette valeur. Limite également la taille maximale des volumes gérés pour les qtrees et les LUN, et qtreesPerFlexvol permet de personnaliser le nombre maximal de qtrees par FlexVol volume	« » (non appliqué par défaut)
lunsPerFlexvol	Le nombre maximal de LUN par FlexVol volume doit être compris dans la plage [50, 200]. SAN uniquement.	« 100 »

Paramètre	Description	Exemple
debugTraceFlags	Indicateurs de débogage à utiliser lors du dépannage. Exemple, {"api":false, "method":true} n'utilisez debugTraceFlags que si vous effectuez un dépannage et que vous avez besoin d'un vidage de journal détaillé.	nul
nfsMountOptions	Liste des options de montage NFS séparée par des virgules. Les options de montage des volumes persistants Kubernetes sont normalement spécifiées dans les classes de stockage, mais si aucune option de montage n'est spécifiée dans une classe de stockage, Trident revient à utiliser les options de montage spécifiées dans le fichier de configuration du back-end de stockage. Si aucune option de montage n'est spécifiée dans la classe de stockage ou le fichier de configuration, Trident ne définit aucune option de montage sur un volume persistant associé.	« »
nasType	Configurez la création de volumes NFS ou SMB. Les options sont nfs, , smb ou null. <b>Doit être défini sur smb pour les volumes SMB.</b> La valeur null par défaut sur les volumes NFS.	nfs
qtreesPerFlexvol	Nombre maximal de qtree par FlexVol volume, doit être compris dans la plage [50, 300]	"200"
smbShare	Vous pouvez spécifier l'une des options suivantes : le nom d'un partage SMB créé à l'aide de la console de gestion Microsoft ou de l'interface de ligne de commande ONTAP, ou un nom permettant à Trident de créer le partage SMB. Ce paramètre est requis pour Amazon FSX pour les systèmes back-end ONTAP.	smb-share

Paramètre	Description	Exemple
useREST	Paramètre booléen pour utiliser les API REST de ONTAP. Lorsqu'il est défini sur true, Trident utilise les API REST ONTAP pour communiquer avec le back-end. Cette fonctionnalité requiert ONTAP 9.11.1 et versions ultérieures. En outre, le rôle de connexion ONTAP utilisé doit avoir accès à l'ontap application. Ceci est satisfait par les rôles et prédéfinis vsadmin cluster-admin .	false
aws	<p>Vous pouvez spécifier ce qui suit dans le fichier de configuration d'AWS FSX pour ONTAP : -</p> <ul style="list-style-type: none"> <li>- fsxFilesystemID: spécifiez l'ID du système de fichiers AWS FSX.</li> <li>- apiRegion: Nom de la région de l'API AWS.</li> <li>- apikey: Clé d'API AWS.</li> <li>- secretKey: Clé secrète AWS.</li> </ul>	"""         """         """
credentials	Spécifiez les informations d'identification de SVM FSX à stocker dans AWS secrets Manager. - name: Amazon Resource Name (ARN) du secret, qui contient les références de SVM. - type: Réglé sur awsarn. Pour plus d'informations, reportez-vous à la section " <a href="#">Créez un secret AWS secrets Manager</a> " .	

#### Options de configuration back-end pour les volumes de provisionnement

Vous pouvez contrôler le provisionnement par défaut à l'aide de ces options dans la defaults section de la configuration. Pour un exemple, voir les exemples de configuration ci-dessous.

Paramètre	Description	Valeur par défaut
spaceAllocation	Allocation d'espace pour les LUN	true
spaceReserve	Mode de réservation d'espace ; « aucun » (fin) ou « volume » (épais)	none
snapshotPolicy	Règle Snapshot à utiliser	none

Paramètre	Description	Valeur par défaut
qosPolicy	QoS policy group à affecter pour les volumes créés. Choisissez une de qosPolicy ou adaptiveQosPolicy par pool de stockage ou back-end. L'utilisation de groupes de règles de qualité de service avec Trident nécessite ONTAP 9.8 ou une version ultérieure. Vous devez utiliser un groupe de règles QoS non partagé et vous assurer que le groupe de règles est appliqué à chaque composant individuellement. Un groupe de règles de QoS partagées applique le débit total de toutes les charges de travail.	« »
adaptiveQosPolicy	Groupe de règles de QoS adaptative à attribuer aux volumes créés. Choisissez une de qosPolicy ou adaptiveQosPolicy par pool de stockage ou back-end. Non pris en charge par l'économie ontap-nas.	« »
snapshotReserve	Pourcentage du volume réservé pour les snapshots « 0 »	Si snapshotPolicy est none, else ""
splitOnClone	Séparer un clone de son parent lors de sa création	false
encryption	Activez le chiffrement de volume NetApp (NVE) sur le nouveau volume. La valeur par défaut est false. Pour utiliser cette option, NVE doit être sous licence et activé sur le cluster. Si NAE est activé sur le back-end, tout volume provisionné dans Trident est activé. Pour plus d'informations, reportez-vous à la section : " <a href="#">Fonctionnement de Trident avec NVE et NAE</a> ".	false
luksEncryption	Activez le cryptage LUKS. Reportez-vous à la <a href="#">"Utiliser la configuration de clé unifiée Linux (LUKS)"</a> . SAN uniquement.	« »
tieringPolicy	Règle de hiérarchisation à utiliser none	
unixPermissions	Mode pour les nouveaux volumes. <b>Laisser vide pour les volumes SMB.</b>	« »

Paramètre	Description	Valeur par défaut
securityStyle	Style de sécurité pour les nouveaux volumes. Prise en charge NFS mixed et unix styles de sécurité. Prise en charge SMB mixed et ntfs styles de sécurité.	NFS la valeur par défaut est unix. SMB la valeur par défaut est ntfs.

### Préparez-vous au provisionnement des volumes SMB

Vous pouvez provisionner des volumes SMB à l'aide du `ontap-nas` pilote. Avant de terminer [l'intégration des pilotes SAN et NAS de ONTAP](#), procédez comme suit.

#### Avant de commencer

Avant de pouvoir provisionner des volumes SMB à l'aide du `ontap-nas` pilote, vous devez disposer des éléments suivants.

- Cluster Kubernetes avec un nœud de contrôleur Linux et au moins un nœud worker Windows exécutant Windows Server 2019. Trident prend en charge les volumes SMB montés sur les pods s'exécutant sur les nœuds Windows uniquement.
- Au moins un secret Trident contenant vos informations d'identification Active Directory. Pour générer un secret `smbcreds`:

```
kubectl create secret generic smbcreds --from-literal username=user
--from-literal password='password'
```

- Un proxy CSI configuré en tant que service Windows. Pour configurer un `csi-proxy`, reportez-vous ["GitHub : proxy CSI"](#) à la section ou ["GitHub : proxy CSI pour Windows"](#) pour les nœuds Kubernetes s'exécutant sous Windows.

#### Étapes

1. Création de partages SMB. Vous pouvez créer les partages d'administration SMB de deux manières, soit à l'aide du ["Console de gestion Microsoft"](#) composant logiciel enfichable dossiers partagés, soit à l'aide de l'interface de ligne de commande ONTAP. Pour créer les partages SMB à l'aide de l'interface de ligne de commandes ONTAP :

- a. Si nécessaire, créez la structure du chemin d'accès au répertoire pour le partage.

La `vserver cifs share create` commande vérifie le chemin spécifié dans l'option `-path` lors de la création du partage. Si le chemin spécifié n'existe pas, la commande échoue.

- b. Créez un partage SMB associé au SVM spécifié :

```
vserver cifs share create -vserver vserver_name -share-name
share_name -path path [-share-properties share_properties,...]
[other_attributes] [-comment text]
```

- c. Vérifiez que le partage a été créé :

```
vserver cifs share show -share-name share_name
```



Reportez-vous "[Créez un partage SMB](#)" à pour plus de détails.

2. Lors de la création du back-end, vous devez configurer le suivant pour spécifier les volumes SMB. Pour toutes les options de configuration du back-end FSX for ONTAP, reportez-vous à "[Exemples et options de configuration de FSX pour ONTAP](#)" la section .

Paramètre	Description	Exemple
smbShare	Vous pouvez spécifier l'une des options suivantes : le nom d'un partage SMB créé à l'aide de la console de gestion Microsoft ou de l'interface de ligne de commande ONTAP, ou un nom permettant à Trident de créer le partage SMB. Ce paramètre est requis pour Amazon FSX pour les systèmes back-end ONTAP.	smb-share
nasType	<b>Doit être réglé sur smb.</b> Si nul, la valeur par défaut est nfs .	smb
securityStyle	Style de sécurité pour les nouveaux volumes. <b>Doit être défini sur ntfs ou mixed pour les volumes SMB.</b>	ntfs Ou mixed pour les volumes SMB
unixPermissions	Mode pour les nouveaux volumes. <b>Doit rester vide pour les volumes SMB.</b>	« »

## Configurez une classe de stockage et un PVC

Configurez un objet StorageClass Kubernetes et créez la classe de stockage pour indiquer à Trident comment provisionner les volumes. Créez une demande de volume persistant qui utilise la classe de stockage Kubernetes configurée pour demander l'accès au volume persistant. Vous pouvez ensuite monter le volume persistant sur un pod.

### Créer une classe de stockage

### Configuration d'un objet StorageClass Kubernetes

Le système "[Objet classe de stockage Kubernetes](#)" identifie Trident en tant que mécanisme de provisionnement utilisé pour cette classe et indique à Trident comment provisionner un volume. Par exemple :

```

apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontap-gold
provisioner: csi.trident.netapp.io
parameters:
  backendType: "ontap-nas"
  provisioningType: "thin"
  snapshots: "true"

```

Pour provisionner des volumes NFSv3 sur AWS Bottlerocket, ajoutez les éléments requis `mountOptions` à la classe de stockage :

```

apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontap-gold
provisioner: csi.trident.netapp.io
parameters:
  backendType: "ontap-nas"
  media: "ssd"
  provisioningType: "thin"
  snapshots: "true"
mountOptions:
  - nfsvers=3
  - nolock

```

Reportez-vous "[Kubernetes et objets Trident](#)" à pour plus de détails sur l'interaction des classes de stockage avec les `PersistentVolumeClaim` paramètres et pour le contrôle de la manière dont Trident provisionne les volumes.

## Créer une classe de stockage

### Étapes

- Il s'agit d'un objet Kubernetes. Utilisez donc `kubectl` pour le créer dans Kubernetes.

```
kubectl create -f storage-class-ontapnas.yaml
```

- Vous devriez maintenant voir une classe de stockage **Basic-csi** dans Kubernetes et Trident, et Trident aurait dû détecter les pools sur le back-end.

```
kubectl get sc basic-csi
```

NAME	PROVISIONER	AGE
basic-csi	csi.trident.netapp.io	15h

## Créer la PVC

Une "[PersistentVolumeClaim](#)" demande de volume persistant est une demande d'accès au volume persistant sur le cluster.

Le PVC peut être configuré pour demander un stockage d'une certaine taille ou d'un certain mode d'accès. À l'aide de la classe de stockage associée, l'administrateur du cluster peut contrôler plus que la taille du volume persistant et le mode d'accès, tels que les performances ou le niveau de service.

Une fois la demande de volume créée, vous pouvez la monter dans un pod.

## Exemples de manifestes

### Exemple de manifeste de volume persistant

Cet exemple de manifeste montre un volume persistant de base de 10Gi associé à StorageClass basic-csi.

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: pv-storage
  labels:
    type: local
spec:
  storageClassName: ontap-gold
  capacity:
    storage: 10Gi
  accessModes:
    - ReadWriteMany
  hostPath:
    path: "/my/host/path"
```

## Exemples de manifestes de demande de volume persistant

Ces exemples présentent les options de configuration de base de la PVC.

### PVC avec accès RWX

Cet exemple montre une demande de volume persistant de base avec accès RWX associée à une classe de stockage nommée basic-csi.

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: pvc-storage
spec:
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 1Gi
  storageClassName: ontap-gold
```

### PVC avec NVMe/TCP

Cet exemple montre une demande de volume persistant de base pour NVMe/TCP avec accès RWX associée à une classe de stockage nommée protection-gold.

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: pvc-san-nvme
spec:
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 300Mi
  storageClassName: protection-gold
```

## Créer le volume persistant et la demande de volume persistant

### Étapes

1. Créer la PVC.

```
kubectl create -f pvc.yaml
```

## 2. Vérifiez l'état du PVC.

```
kubectl get pvc
```

NAME	STATUS	VOLUME	CAPACITY	ACCESS MODES	STORAGECLASS	AGE
pvc-storage	Bound	pv-name	2Gi	RWO		5m

Reportez-vous "[Kubernetes et objets Trident](#)" à pour plus de détails sur l'interaction des classes de stockage avec les PersistentVolumeClaim paramètres et pour le contrôle de la manière dont Trident provisionne les volumes.

### Attributs Trident

Ces paramètres déterminent quels pools de stockage gérés par Trident doivent être utilisés pour provisionner les volumes d'un type donné.

Attribut	Type	Valeurs	Offre	Demande	Pris en charge par
support <sup>1</sup>	chaîne	hdd, hybride, ssd	Le pool contient des supports de ce type ; hybride signifie les deux	Type de support spécifié	ontap-nas, ontap-nas-économie, ontap-nas-flexgroup, ontap-san, solidfire-san
Type de provisionnement	chaîne	fin, épais	Le pool prend en charge cette méthode de provisionnement	Méthode de provisionnement spécifiée	thick : tous les systèmes ONTAP ; thin : tous les systèmes ONTAP et solidfire-san
Type de dos	chaîne	ontap-nas, économie ontap-nas, ontap-nas-flexgroup, ontap-san, solidfire-san, gcp-cvs, azure-netapp-files, ontap-san-economy	Le pool appartient à ce type de système back-end	Backend spécifié	Tous les conducteurs
snapshots	bool	vrai, faux	Le pool prend en charge les volumes dotés de snapshots	Volume sur lequel les snapshots sont activés	ontap-nas, ontap-san, solidfire-san, gcp-cvs

Attribut	Type	Valeurs	Offre	Demande	Pris en charge par
clones	bool	vrai, faux	Le pool prend en charge les volumes de clonage	Volume sur lequel les clones sont activés	ontap-nas, ontap-san, solidfire-san, gcp-cvs
le cryptage	bool	vrai, faux	Le pool prend en charge les volumes chiffrés	Volume avec chiffrement activé	ontap-nas, économie ontap-nas, ontap-nas-flexgroups, ontap-san
LES IOPS	int	entier positif	Le pool est en mesure de garantir l'IOPS dans cette plage	Volume garanti ces IOPS	solidfire-san

<sup>1</sup> : non pris en charge par les systèmes ONTAP Select

### Déploiement de l'application exemple

Une fois la classe de stockage et la demande de volume persistant créées, vous pouvez monter le volume persistant sur un pod. Cette section répertorie l'exemple de commande et de configuration permettant d'attacher le volume persistant à un pod.

#### Étapes

1. Montez le volume dans un pod.

```
kubectl create -f pv-pod.yaml
```

Ces exemples montrent les configurations de base pour attacher le PVC à un pod : **Configuration de base** :

```

kind: Pod
apiVersion: v1
metadata:
  name: pv-pod
spec:
  volumes:
    - name: pv-storage
      persistentVolumeClaim:
        claimName: basic
  containers:
    - name: pv-container
      image: nginx
      ports:
        - containerPort: 80
          name: "http-server"
      volumeMounts:
        - mountPath: "/my/mount/path"
          name: pv-storage

```



Vous pouvez surveiller la progression à l'aide de `kubectl get pod --watch`.

- Vérifiez que le volume est monté sur `/my/mount/path`.

```
kubectl exec -it pv-pod -- df -h /my/mount/path
```

Filesystem		Size			
Used	Avail	Use%	Mounted on		
192.168.188.78:/trident_pvc_ae45ed05_3ace_4e7c_9080_d2a83ae03d06	320K	1.0G	1%	/my/mount/path	1.1G

Vous pouvez maintenant supprimer le Pod. L'application Pod n'existera plus, mais le volume restera.

```
kubectl delete pod pv-pod
```

### Configurer le module complémentaire Trident EKS sur un cluster EKS

NetApp Trident rationalise la gestion du stockage Amazon FSX for NetApp ONTAP dans Kubernetes pour que vos développeurs et administrateurs puissent donner la priorité au déploiement d'applications. Le module complémentaire NetApp Trident EKS inclut les derniers correctifs de sécurité et de bogues, et est validé par AWS pour une utilisation avec Amazon EKS. Le module complémentaire EKS vous permet de vous assurer de

manière cohérente que vos clusters Amazon EKS sont sécurisés et stables et de réduire la quantité de travail à effectuer pour installer, configurer et mettre à jour des modules complémentaires.

## Prérequis

Vérifiez les points suivants avant de configurer le module complémentaire Trident pour AWS EKS :

- Un compte de cluster Amazon EKS avec des autorisations d'utilisation de modules complémentaires. Reportez-vous à la ["Add-ons Amazon EKS"](#).
- Autorisations AWS sur AWS Marketplace :  
"aws-marketplace:ViewSubscriptions",  
"aws-marketplace:Subscribe",  
"aws-marketplace:Unsubscribe"
- Type ami : Amazon Linux 2 (AL2\_x86\_64) ou Amazon Linux 2 ARM (AL2\_ARM\_64)
- Type de nœud : AMD ou ARM
- Un système de fichiers Amazon FSX pour NetApp ONTAP

## Étapes

1. Veillez à créer un rôle IAM et un code AWS secret pour permettre aux pods d'EKS d'accéder aux ressources AWS. Pour obtenir des instructions, reportez-vous à la section ["Créez un rôle IAM et un code secret AWS"](#).
2. Sur votre cluster EKS Kubernetes, accédez à l'onglet **Add-ons**.

The screenshot shows the AWS EKS Cluster Details page for a cluster named "tri-env-eks". At the top, there are buttons for "Delete cluster", "Upgrade version", and "View dashboard". Below the cluster name, a message indicates that standard support for Kubernetes version 1.30 ends on July 28, 2025, and provides a link to the pricing page. The "Cluster info" section displays the status as "Active", Kubernetes version as "1.30", support period as "Standard support until July 28, 2025", and provider as "EKS". The "Add-ons" tab is selected, showing 3 available add-ons. A notification at the bottom of the add-ons list says "New versions are available for 1 add-on." The "Add-ons" section includes a search bar, filters for category and status, and buttons for "View details", "Edit", "Remove", and "Get more add-ons".

3. Accédez à **add-ons** AWS Marketplace et choisissez la catégorie *Storage*.

## AWS Marketplace add-ons (1)



Discover, subscribe to and configure EKS add-ons to enhance your EKS clusters.

Find add-on

Filtering options

Any category ▾

NetApp, Inc. ▾

Any pricing model ▾

Clear filters

NetApp, Inc.

< 1 >



### NetApp Trident



NetApp Trident streamlines Amazon FSx for NetApp ONTAP storage management in Kubernetes to let your developers and administrators focus on application deployment. FSx for ONTAP flexibility, scalability, and integration capabilities make it the ideal choice for organizations seeking efficient containerized storage workflows. [Product details](#)

Standard Contract

**Category**  
storage

**Listed by**  
[NetApp, Inc.](#)

**Supported versions**  
1.31, 1.30, 1.29, 1.28,  
1.27, 1.26, 1.25, 1.24,  
1.23

**Pricing starting at**  
[View pricing details](#)

[Cancel](#)

[Next](#)

4. Localisez **NetApp Trident** et cochez la case du module complémentaire Trident, puis cliquez sur **Suivant**.
5. Choisissez la version souhaitée du module complémentaire.

### NetApp Trident

Remove add-on

Listed by



Category

storage

Status

Ready to install



#### You're subscribed to this software

You can view the terms and pricing details for this product or choose another offer if one is available.

View subscription



#### Version

Select the version for this add-on.

v24.10.0-eksbuild.1



#### Select IAM role

Select an IAM role to use with this add-on. To create a new custom role, follow the instructions in the [Amazon EKS User Guide](#)

Not set



#### ► Optional configuration settings

[Cancel](#)

[Previous](#)

[Next](#)

6. Sélectionnez l'option rôle IAM à hériter du nœud.

## Review and add

### Step 1: Select add-ons

[Edit](#)

#### Selected add-ons (1)

 Find add-on

&lt; 1 &gt;

Add-on name	Type	Status
netapp_trident-operator	storage	<input checked="" type="checkbox"/> Ready to install

### Step 2: Configure selected add-ons settings

[Edit](#)

#### Selected add-ons version (1)

&lt; 1 &gt;

Add-on name	Version	IAM role for service account (IRSA)
netapp_trident-operator	v24.10.0-eksbuild.1	Not set

#### EKS Pod Identity (0)

&lt; 1 &gt;

Add-on name	IAM role	Service account
No Pod Identity associations		
None of the selected add-on(s) have Pod Identity associations.		

[Cancel](#)[Previous](#)[Create](#)

7. Suivez le schéma de configuration **Add-on** et définissez le paramètre valeurs de configuration de la section **valeurs de configuration** sur le fil de rôle que vous avez créé à l'étape précédente (étape 1). La valeur doit être au format suivant :

```
{  
  "cloudIdentity": "'eks.amazonaws.com/role-arn: <role ARN>'"  
}
```

**i** Si vous sélectionnez remplacer pour la méthode de résolution des conflits, un ou plusieurs des paramètres du module complémentaire existant peuvent être remplacés par les paramètres du module complémentaire Amazon EKS. Si vous n'activez pas cette option et qu'il y a un conflit avec vos paramètres existants, l'opération échoue. Vous pouvez utiliser le message d'erreur qui en résulte pour résoudre le conflit. Avant de sélectionner cette option, assurez-vous que le module complémentaire Amazon EKS ne gère pas les paramètres que vous devez gérer vous-même.

#### ▼ Optional configuration settings

##### Add-on configuration schema

Refer to the JSON schema below. The configuration values entered in the code editor will be validated against this schema.

```
version: 1.0,
"examples": [
  {
    "cloudIdentity": ""
  }
],
"properties": {
  "cloudIdentity": {
    "default": "",
    "examples": [
      ""
    ],
    "title": "The cloudIdentity Schema",
    "type": "string"
  }
}
```

##### Configuration values | Info

Specify any additional JSON or YAML configurations that should be applied to the add-on.

```
1 ▾ {
2   "cloudIdentity": "eks.amazonaws.com/role-arn: arn:aws:iam
3     ::186785786363:role/tri-env-eks-trident-controller-role"
 }
```

## 8. Sélectionnez **Créer**.

## 9. Vérifiez que l'état du complément est *Active*.

The screenshot shows the AWS Lambda console interface. At the top, there's a search bar with 'netapp' and a dropdown menu. Below it, a table displays the 'NetApp Trident' add-on details:

Category	Status	Version	EKS Pod Identity	IAM role for service account (IRSA)
storage	Active	v24.10.0-eksbuild.1	-	Not set

Below the table, it says 'Listed by NetApp, Inc.' and has a 'View subscription' button.

## 10. Exécutez la commande suivante pour vérifier que Trident est correctement installé sur le cluster :

```
kubectl get pods -n trident
```

## 11. Poursuivez l'installation et la configuration du système back-end de stockage. Pour plus d'informations, voir "[Configurez le back-end de stockage](#)".

**Installez/désinstallez le module complémentaire Trident EKS à l'aide de l'interface de ligne de commande**

**Installez le module complémentaire NetApp Trident EKS à l'aide de l'interface de ligne de commande :**

L'exemple de commande suivant installe le module complémentaire Trident EKS :

```
eksctl create addon --cluster clusterName --name netapp_trident-operator  
--version v25.02.1-eksbuild.1 (avec une version dédiée)
```

**Désinstallez le module complémentaire NetApp Trident EKS à l'aide de l'interface de ligne de commande :**

La commande suivante désinstalle le module complémentaire Trident EKS :

```
eksctl delete addon --cluster K8s-arm --name netapp_trident-operator
```

## Création de systèmes back-end avec kubectl

Un back-end définit la relation entre Trident et un système de stockage. Il explique à Trident comment communiquer avec ce système de stockage et comment Trident doit provisionner les volumes à partir de celui-ci. Une fois Trident installé, l'étape suivante consiste à créer un back-end. La `TridentBackendConfig` définition personnalisée des ressources (CRD) vous permet de créer et de gérer des systèmes back-end Trident directement via l'interface Kubernetes. Pour cela, vous pouvez utiliser `kubectl` ou l'outil CLI équivalent pour votre distribution Kubernetes.

`TridentBackendConfig`

`TridentBackendConfig` (`tbc`, `tbconfig`, `tbackendconfig`) Est un système CRD front-end, qui vous permet de gérer des systèmes Trident back-end à l'aide de `kubectl`. Kubernetes et les administrateurs du stockage peuvent désormais créer et gérer des systèmes back-end directement via l'interface de ligne de commande Kubernetes sans avoir besoin d'un utilitaire de ligne de commande dédié (`tridentctl`).

Lors de la création d'un `TridentBackendConfig` objet, les événements suivants se produisent :

- Un back-end est créé automatiquement par Trident en fonction de la configuration que vous fournissez. Ceci est représenté en interne sous la forme d'une `TridentBackend` CR (`tbe`, `tridentbackend`).
- Le `TridentBackendConfig` est lié de manière unique à un `TridentBackend` qui a été créé par Trident.

Chacun `TridentBackendConfig` gère un mappage un-à-un avec un `TridentBackend`. Le premier est l'interface fournie à l'utilisateur pour concevoir et configurer des systèmes back-end ; le second est la façon dont Trident représente l'objet back-end réel.

 `TridentBackend` Les CRS sont créés automatiquement par Trident. Vous ne devez pas les modifier. Si vous souhaitez effectuer des mises à jour vers des systèmes back-end, modifiez l'objet pour procéder `TridentBackendConfig` à cette opération.

Reportez-vous à l'exemple suivant pour connaître le format de `TridentBackendConfig` la CR :

```

apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-ontap-san
spec:
  version: 1
  backendName: ontap-san-backend
  storageDriverName: ontap-san
  managementLIF: 10.0.0.1
  dataLIF: 10.0.0.2
  svm: trident_svm
  credentials:
    name: backend-tbc-ontap-san-secret

```

Vous pouvez également consulter les exemples de configuration de la plate-forme/du service de stockage souhaité dans le "[programme d'installation trident](#)" répertoire.

``spec`` Prend en charge les paramètres de configuration spécifiques au back-end. Dans cet exemple, le back-end utilise le `ontap-san` pilote de stockage et les paramètres de configuration qui sont tabulés ici. Pour obtenir la liste des options de configuration du pilote de stockage souhaité, reportez-vous au [xref:{relative\\_path}backends.html\["informations de configuration backend pour votre pilote de stockage"\]](#).

La `spec` section inclut également `credentials` les champs et `deletionPolicy`, qui sont récemment introduits dans la `TridentBackendConfig` CR :

- `credentials`: Ce paramètre est un champ obligatoire qui contient les informations d'identification utilisées pour s'authentifier auprès du système/service de stockage. Cette configuration est définie sur un code secret Kubernetes créé par l'utilisateur. Les informations d'identification ne peuvent pas être transmises en texte brut et entraînent une erreur.
- `deletionPolicy`: Ce champ définit ce qui doit se produire lorsque le `TridentBackendConfig` est supprimé. Il peut prendre l'une des deux valeurs possibles :
  - `delete`: Cela entraîne la suppression de la `TridentBackendConfig` CR et du back-end associé. Il s'agit de la valeur par défaut.
  - `retain`: Lorsqu'une `TridentBackendConfig` demande de modification est supprimée, la définition du back-end est toujours présente et peut être gérée avec `tridentctl`. La définition de la stratégie de suppression `retain` permet aux utilisateurs de revenir à une version antérieure (antérieure à 21.04) et de conserver les systèmes back-end créés. La valeur de ce champ peut être mise à jour après la création d'un `TridentBackendConfig`.

 Le nom d'un backend est défini à l'aide de `spec.backendName`. S'il n'est pas spécifié, le nom du backend est défini sur le nom de `TridentBackendConfig` l'objet (metadata.name). Il est recommandé de définir explicitement les noms de back-end à l'aide de `spec.backendName`.



Les systèmes back-end créés avec `tridentctl` n'ont pas d'objet associé `TridentBackendConfig`. Vous pouvez choisir de gérer ces systèmes back-end avec `kubectl` en créant une `TridentBackendConfig` demande de modification. Veillez à spécifier des paramètres de configuration identiques (tels que `spec.backendName`, , , `spec.storagePrefix` `spec.storageDriverName` etc.). Trident lie automatiquement le nouveau système créé `TridentBackendConfig` avec le système back-end existant.

## Présentation des étapes

Pour créer un nouveau back-end à l'aide de `kubectl`, procédez comme suit :

1. Créez un "[Le secret de Kubernetes](#)". Le secret contient les informations d'identification dont Trident a besoin pour communiquer avec le cluster/service de stockage.
2. Créez un `TridentBackendConfig` objet. Elle contient des informations spécifiques sur le cluster/service de stockage et fait référence au secret créé à l'étape précédente.

Après avoir créé un backend, vous pouvez observer son état en utilisant et en `kubectl get tbc <tbc-name> -n <trident-namespace>` rassemblant des détails supplémentaires.

### Étape 1 : créez un code secret Kubernetes

Créez un secret qui contient les informations d'identification d'accès pour le back-end. Ce point est unique à chaque service/plateforme de stockage. Voici un exemple :

```
kubectl -n trident create -f backend-tbc-ontap-san-secret.yaml
```

```
apiVersion: v1
kind: Secret
metadata:
  name: backend-tbc-ontap-san-secret
type: Opaque
stringData:
  username: cluster-admin
  password: password
```

Ce tableau récapitule les champs à inclure dans le Secret pour chaque plate-forme de stockage :

Description des champs secrets de la plate-forme de stockage	Secret	Description des champs
Azure NetApp Files	ID client	ID client d'un enregistrement d'application
Cloud Volumes Service pour GCP	id_clé_privée	ID de la clé privée. Partie de la clé API pour le compte de service GCP avec le rôle d'administrateur CVS

Description des champs secrets de la plate-forme de stockage	Secret	Description des champs
Cloud Volumes Service pour GCP	clé_privée	Clé privée. Partie de la clé API pour le compte de service GCP avec le rôle d'administrateur CVS
Element (NetApp HCI/SolidFire)	Point final	MVIP pour le cluster SolidFire avec les identifiants de locataire
ONTAP	nom d'utilisateur	Nom d'utilisateur pour la connexion au cluster/SVM. Utilisé pour l'authentification basée sur les identifiants
ONTAP	mot de passe	Mot de passe pour la connexion au cluster/SVM. Utilisé pour l'authentification basée sur les identifiants
ONTAP	ClientPrivateKey	Valeur encodée en Base64 de la clé privée du client. Utilisé pour l'authentification basée sur des certificats
ONTAP	ChapUsername	Nom d'utilisateur entrant. Requis si useCHAP=vrai. Pour ontap-san et ontap-san-economy
ONTAP	Chapeau InitiatorSecret	Secret de l'initiateur CHAP. Requis si useCHAP=vrai. Pour ontap-san et ontap-san-economy
ONTAP	ChapTargetUsername	Nom d'utilisateur cible. Requis si useCHAP=vrai. Pour ontap-san et ontap-san-economy
ONTAP	ChapTargetInitiatorSecret	Secret de l'initiateur cible CHAP. Requis si useCHAP=vrai. Pour ontap-san et ontap-san-economy

Le secret créé à cette étape sera référencé dans le `spec.credentials` champ de l'`'TridentBackendConfig'` objet créé à l'étape suivante.

## Étape 2 : créer la `TridentBackendConfig CR`

Vous êtes maintenant prêt à créer votre `TridentBackendConfig CR`. Dans cet exemple, un back-end qui utilise le `ontap-san` pilote est créé à l'aide de l'`'TridentBackendConfig'` objet illustré ci-dessous :

```
kubectl -n trident create -f backend-tbc-ontap-san.yaml
```

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-ontap-san
spec:
  version: 1
  backendName: ontap-san-backend
  storageDriverName: ontap-san
  managementLIF: 10.0.0.1
  dataLIF: 10.0.0.2
  svm: trident_svm
  credentials:
    name: backend-tbc-ontap-san-secret
```

### Étape 3 : vérifier l'état du TridentBackendConfig CR

Maintenant que vous avez créé la TridentBackendConfig demande de modification, vous pouvez vérifier son état. Voir l'exemple suivant :

```
kubectl -n trident get tbc backend-tbc-ontap-san
NAME          BACKEND NAME      BACKEND UUID
PHASE   STATUS
backend-tbc-ontap-san  ontap-san-backend  8d24fce7-6f60-4d4a-8ef6-
bab2699e6ab8    Bound    Success
```

Un back-end a été créé avec succès et lié à la TridentBackendConfig demande de modification.

La phase peut prendre l'une des valeurs suivantes :

- **Bound:** La TridentBackendConfig CR est associée à un back-end, et ce back-end contient configRef défini sur l'uid de la TridentBackendConfig CR.
- **Unbound:** Représenté à l'aide de "". L' TridentBackendConfig` objet n'est pas lié à un backend. Par défaut, toutes les demandes de modification nouvellement créées `TridentBackendConfig sont dans cette phase. Une fois la phase modifiée, elle ne peut plus revenir à Unbound.
- **Deleting:** La TridentBackendConfig demande de modification deletionPolicy a été définie sur supprimer. Lorsque la TridentBackendConfig CR est supprimée, elle passe à l'état Suppression.
  - Si aucune demande de volume persistant n'existe sur le back-end, la suppression du entraîne la suppression de Trident, TridentBackendConfig ainsi que de la TridentBackendConfig demande de modification.
  - Si un ou plusieurs ESV sont présents sur le back-end, il passe à l'état de suppression. La

TridentBackendConfig CR passe ensuite également en phase de suppression. Le back-end et TridentBackendConfig sont supprimés uniquement après la suppression de toutes les ESV.

- Lost: Le back-end associé à la TridentBackendConfig CR a été accidentellement ou délibérément supprimé et la TridentBackendConfig CR a toujours une référence au back-end supprimé. La TridentBackendConfig demande de modification peut toujours être supprimée quelle que soit la deletionPolicy valeur.
- Unknown: Trident n'est pas en mesure de déterminer l'état ou l'existence du back-end associé à la TridentBackendConfig CR. Par exemple, si le serveur d'API ne répond pas ou si le tridentbackends.trident.netapp.io CRD est manquant. Cela peut nécessiter une intervention.

À ce stade, un système back-end est créé avec succès ! Plusieurs opérations peuvent également être gérées, telles que "[mises à jour du système back-end et suppressions](#)".

#### (Facultatif) étape 4 : pour plus de détails

Vous pouvez exécuter la commande suivante pour obtenir plus d'informations sur votre système back-end :

```
kubectl -n trident get tbc backend-tbc-ontap-san -o wide
```

NAME	BACKEND NAME	BACKEND UUID	
PHASE	STATUS	STORAGE DRIVER	DELETION POLICY
backend-tbc-ontap-san	Bound	ontap-san-backend	8d24fce7-6f60-4d4a-8ef6-bab2699e6ab8
		Success	ontap-san
			delete

En outre, vous pouvez également obtenir un vimage YAML/JSON de TridentBackendConfig.

```
kubectl -n trident get tbc backend-tbc-ontap-san -o yaml
```

```

apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  creationTimestamp: 2021-04-21T20:45:11Z
  finalizers:
    - trident.netapp.io
  generation: 1
  name: backend-tbc-ontap-san
  namespace: trident
  resourceVersion: "947143"
  uid: 35b9d777-109f-43d5-8077-c74a4559d09c
spec:
  backendName: ontap-san-backend
  credentials:
    name: backend-tbc-ontap-san-secret
  managementLIF: 10.0.0.1
  dataLIF: 10.0.0.2
  storageDriverName: ontap-san
  svm: trident_svm
  version: 1
status:
  backendInfo:
    backendName: ontap-san-backend
    backendUUID: 8d24fce7-6f60-4d4a-8ef6-bab2699e6ab8
  deletionPolicy: delete
  lastOperationStatus: Success
  message: Backend 'ontap-san-backend' created
  phase: Bound

```

**backendInfo** Contient le `backendName` et le `backendUUID` du back-end créé en réponse à la `TridentBackendConfig` demande de modification. Le `lastOperationStatus` champ représente l'état de la dernière opération de la `TridentBackendConfig` CR, qui peut être déclenchée par l'utilisateur (par exemple, l'utilisateur a modifié quelque chose dans `spec`) ou déclenchée par Trident (par exemple, lors d'un redémarrage de Trident). Il peut s'agir d'un succès ou d'un échec. `phase` Représente l'état de la relation entre la `TridentBackendConfig` CR et le back-end. Dans l'exemple ci-dessus, `phase` a la valeur liée, ce qui signifie que la `TridentBackendConfig` CR est associée au back-end.

Vous pouvez exécuter `kubectl -n trident describe tbc <tbc-cr-name>` la commande pour obtenir des détails sur les journaux d'événements.

 Vous ne pouvez pas mettre à jour ou supprimer un back-end contenant un objet associé à `TridentBackendConfig` l'aide de `tridentctl`. Pour comprendre les étapes impliquées dans le passage entre `tridentctl` et `TridentBackendConfig`, ["voir ici"](#).

## Gestion des systèmes back-end

### Effectuer la gestion back-end avec kubectl

Découvrez comment effectuer des opérations de gestion back-end à l'aide de kubectl.

#### Supprimer un back-end

En supprimant un `TridentBackendConfig`, vous demandez à Trident de supprimer/conserver les systèmes back-end (sur la base de `deletionPolicy` la ). Pour supprimer un back-end, assurez-vous que `deletionPolicy` est défini sur supprimer. Pour supprimer uniquement le `TridentBackendConfig`, assurez-vous que `deletionPolicy` est défini sur conserver. Cela permet de s'assurer que le back-end est toujours présent et peut être géré à l'aide de `tridentctl`.

Exécutez la commande suivante :

```
kubectl delete tbc <tbc-name> -n trident
```

Trident ne supprime pas les secrets Kubernetes utilisés par `TridentBackendConfig`. L'utilisateur Kubernetes est chargé de nettoyer les secrets. Il faut faire attention lors de la suppression des secrets. Vous devez supprimer les secrets uniquement s'ils ne sont pas utilisés par les systèmes back-end.

#### Affichez les systèmes back-end existants

Exécutez la commande suivante :

```
kubectl get tbc -n trident
```

Vous pouvez également exécuter `tridentctl get backend -n trident` ou `tridentctl get backend -o yaml -n trident` pour obtenir la liste de tous les systèmes back-end existants. Cette liste inclura également les systèmes back-end créés avec `tridentctl`.

#### Mettre à jour un back-end

Il peut y avoir plusieurs raisons de mettre à jour un backend :

- Les informations d'identification du système de stockage ont été modifiées. Pour mettre à jour les informations d'identification, le secret Kubernetes utilisé dans l' `'TridentBackendConfig'` objet doit être mis à jour. Trident met automatiquement à jour le back-end avec les informations d'identification les plus récentes fournies. Exécutez la commande suivante pour mettre à jour le code secret Kubernetes :

```
kubectl apply -f <updated-secret-file.yaml> -n trident
```

- Les paramètres (tels que le nom du SVM ONTAP utilisé) doivent être mis à jour.

- Vous pouvez mettre à jour les `TridentBackendConfig` objets directement via Kubernetes à l'aide de la commande suivante :

```
kubectl apply -f <updated-backend-file.yaml>
```

- Vous pouvez également modifier la CR existante à l' `TridentBackendConfig` aide de la commande suivante :

```
kubectl edit tbc <tbc-name> -n trident
```

-  i
- En cas d'échec d'une mise à jour du back-end, le système back-end continue de rester dans sa dernière configuration connue. Vous pouvez afficher les journaux pour déterminer la cause en exécutant `kubectl get tbc <tbc-name> -o yaml -n trident` ou `kubectl describe tbc <tbc-name> -n trident`.
  - Après avoir identifié et corrigé le problème avec le fichier de configuration, vous pouvez relancer la commande update.

## Gestion back-end avec tridentctl

Découvrez comment effectuer des opérations de gestion back-end à l'aide de `tridentctl`.

### Créer un back-end

Après avoir créé un "[fichier de configuration back-end](#)", exécutez la commande suivante :

```
tridentctl create backend -f <backend-file> -n trident
```

Si la création du système back-end échoue, la configuration du système back-end était erronée. Vous pouvez afficher les journaux pour déterminer la cause en exécutant la commande suivante :

```
tridentctl logs -n trident
```

Une fois que vous avez identifié et corrigé le problème avec le fichier de configuration, il vous suffit d'exécuter à nouveau la `create` commande.

### Supprimer un back-end

Pour supprimer un back-end de Trident, procédez comme suit :

1. Récupérer le nom du système back-end :

```
tridentctl get backend -n trident
```

2. Supprimer le backend :

```
tridentctl delete backend <backend-name> -n trident
```



Si Trident a provisionné des volumes et des snapshots à partir de ce back-end, la suppression du back-end empêche le provisionnement de nouveaux volumes. Le back-end continuera à exister dans un état « Suppression ».

### Affichez les systèmes back-end existants

Pour afficher les systèmes back-end dont Trident a conscience, procédez comme suit :

- Pour obtenir un récapitulatif, exécutez la commande suivante :

```
tridentctl get backend -n trident
```

- Pour obtenir tous les détails, exécutez la commande suivante :

```
tridentctl get backend -o json -n trident
```

### Mettre à jour un back-end

Après avoir créé un nouveau fichier de configuration back-end, exécutez la commande suivante :

```
tridentctl update backend <backend-name> -f <backend-file> -n trident
```

En cas d'échec de la mise à jour back-end, quelque chose était incorrect avec la configuration back-end ou vous avez tenté une mise à jour non valide. Vous pouvez afficher les journaux pour déterminer la cause en exécutant la commande suivante :

```
tridentctl logs -n trident
```

Une fois que vous avez identifié et corrigé le problème avec le fichier de configuration, il vous suffit d'exécuter à nouveau la `update` commande.

### Identifier les classes de stockage qui utilisent un système back-end

Voici un exemple du type de questions que vous pouvez répondre avec le JSON qui `tridentctl` sort pour les objets `backend`. Cela utilise l' `jq` utilitaire que vous devez installer.

```
tridentctl get backend -o json | jq '[.items[] | {backend: .name, storageClasses: [.storage[].storageClasses]|unique}]'
```

Cela s'applique également aux systèmes back-end créés par l'utilisation de `TridentBackendConfig`.

## Passez d'une option de gestion back-end à une autre

Découvrez les différentes méthodes de gestion des systèmes back-end dans Trident.

### Options de gestion des systèmes back-end

Avec l'introduction de `TridentBackendConfig`, les administrateurs disposent désormais de deux méthodes uniques de gestion des systèmes back-end. Ceci pose les questions suivantes :

- Les systèmes back-end créés à l'aide de peuvent-ils `tridentctl` être gérés avec `TridentBackendConfig`?
- Les systèmes back-end créés à l'aide de peuvent-ils `TridentBackendConfig` être gérés à `tridentctl` ?

#### Gestion des `tridentctl` systèmes back-end avec `TridentBackendConfig`

Cette section décrit les étapes requises pour gérer les systèmes back-end créés `tridentctl` directement via l'interface Kubernetes en créant `TridentBackendConfig` des objets.

Cela s'applique aux scénarios suivants :

- Systèmes back-end existants, qui n'ont pas de système `TridentBackendConfig` parce qu'ils ont été créés avec `tridentctl`.
- Nouveaux systèmes back-end créés avec `tridentctl`, alors que d'autres `TridentBackendConfig` objets existent.

Dans les deux scénarios, les systèmes back-end continueront d'être présents, avec Trident qui planifie les volumes et les exécute. Les administrateurs peuvent choisir l'une des deux options suivantes :

- Continuez à utiliser `tridentctl` pour gérer les systèmes back-end créés à l'aide de celui-ci.
- Lier les systèmes back-end créés à l'aide d'`tridentctl`` un nouvel ``TridentBackendConfig` objet. Cela signifie que les systèmes back-end seront gérés sans utilisation `kubectl tridentctl`.

Pour gérer un back-end pré-existant à l'aide de `kubectl`, vous devez créer un `TridentBackendConfig` qui se lie au back-end existant. Voici un aperçu du fonctionnement de ces éléments :

1. Créez un code secret Kubernetes. La clé secrète contient les informations d'identification dont Trident a besoin pour communiquer avec le cluster/service de stockage.
2. Créer un `TridentBackendConfig` objet. Elle contient des informations spécifiques sur le cluster/service de stockage et fait référence au secret créé à l'étape précédente. Veillez à spécifier des paramètres de configuration identiques (tels que `spec.backendName`, , , `spec.storagePrefix` `spec.storageDriverName` etc.). `spec.backendName` doit être défini sur le nom du back-end existant.

### Étape 0 : identifier le back-end

Pour créer un `TridentBackendConfig` qui se lie à un back-end existant, vous devez obtenir la configuration back-end. Dans cet exemple, supposons qu'un back-end a été créé à l'aide de la définition JSON suivante :

```
tridentctl get backend ontap-nas-backend -n trident
+-----+-----+
+-----+-----+-----+
|       NAME      | STORAGE DRIVER |          UUID
| STATE   | VOLUMES  |
+-----+-----+
+-----+-----+
| ontap-nas-backend | ontap-nas      | 52f2eb10-e4c6-4160-99fc-
96b3be5ab5d7 | online | 25 |
+-----+-----+
+-----+-----+
```

```
cat ontap-nas-backend.json
```

```
{
  "version": 1,
  "storageDriverName": "ontap-nas",
  "managementLIF": "10.10.10.1",
  "dataLIF": "10.10.10.2",
  "backendName": "ontap-nas-backend",
  "svm": "trident_svm",
  "username": "cluster-admin",
  "password": "admin-password",
  "defaults": {
    "spaceReserve": "none",
    "encryption": "false"
  },
  "labels": {
    "store": "nas_store"
  },
  "region": "us_east_1",
  "storage": [
    {
      "labels": {
        "app": "msoffice",
        "cost": "100"
      },
      "zone": "us_east_1a",
      "defaults": {
        "spaceReserve": "volume",
        "encryption": "true",
        "unixPermissions": "0755"
      }
    },
    {
      "labels": {
        "app": "mysqldb",
        "cost": "25"
      },
      "zone": "us_east_1d",
      "defaults": {
        "spaceReserve": "volume",
        "encryption": "false",
        "unixPermissions": "0775"
      }
    }
  ]
}
```

## Étape 1 : créez un code secret Kubernetes

Créez un secret qui contient les informations d'identification du back-end, comme indiqué dans cet exemple :

```
cat tbc-ontap-nas-backend-secret.yaml
```

```
apiVersion: v1
kind: Secret
metadata:
  name: ontap-nas-backend-secret
type: Opaque
stringData:
  username: cluster-admin
  password: admin-password
```

```
kubectl create -f tbc-ontap-nas-backend-secret.yaml -n trident
secret/backend-tbc-ontap-san-secret created
```

## Étape 2 : créer une TridentBackendConfig demande de modification

L'étape suivante consiste à créer une TridentBackendConfig demande de modification qui sera automatiquement lié au pré-existant ontap-nas-backend (comme dans cet exemple). Assurez-vous que les exigences suivantes sont respectées :

- Le même nom de back-end est défini dans spec.backendName.
- Les paramètres de configuration sont identiques au back-end d'origine.
- Les pools virtuels (le cas échéant) doivent conserver le même ordre que dans le back-end d'origine.
- Les identifiants sont fournis via un code secret Kubernetes et non en texte brut.

Dans ce cas, le se TridentBackendConfig présente comme suit :

```
cat backend-tbc-ontap-nas.yaml
```

```

apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: tbc-ontap-nas-backend
spec:
  version: 1
  storageDriverName: ontap-nas
  managementLIF: 10.10.10.1
  dataLIF: 10.10.10.2
  backendName: ontap-nas-backend
  svm: trident_svm
  credentials:
    name: mysecret
  defaults:
    spaceReserve: none
    encryption: 'false'
  labels:
    store: nas_store
  region: us_east_1
  storage:
    - labels:
        app: msoffice
        cost: '100'
        zone: us_east_1a
      defaults:
        spaceReserve: volume
        encryption: 'true'
        unixPermissions: '0755'
    - labels:
        app: mysqldb
        cost: '25'
        zone: us_east_1d
      defaults:
        spaceReserve: volume
        encryption: 'false'
        unixPermissions: '0775'

```

```

kubectl create -f backend-tbc-ontap-nas.yaml -n trident
tridentbackendconfig.trident.netapp.io/tbc-ontap-nas-backend created

```

### Étape 3 : vérifier l'état du TridentBackendConfig CR

Une fois le TridentBackendConfig créé, sa phase doit être Bound. Il devrait également refléter le même nom de back-end et UUID que celui du back-end existant.

```

kubectl get tbc tbc-ontap-nas-backend -n trident
NAME                  BACKEND NAME      BACKEND UUID
PHASE    STATUS
tbc-ontap-nas-backend  ontap-nas-backend  52f2eb10-e4c6-4160-99fc-
96b3be5ab5d7    Bound     Success

#confirm that no new backends were created (i.e., TridentBackendConfig did
not end up creating a new backend)
tridentctl get backend -n trident
+-----+-----+
+-----+-----+-----+
|       NAME          | STORAGE DRIVER |           UUID
| STATE   | VOLUMES | 
+-----+-----+
+-----+-----+-----+
| ontap-nas-backend | ontap-nas        | 52f2eb10-e4c6-4160-99fc-
96b3be5ab5d7 | online |      25 |
+-----+-----+
+-----+-----+-----+

```

Le back-end sera désormais entièrement géré à l'aide de l' `tbc-ontap-nas-backend` `TridentBackendConfig` objet.

#### Gestion des TridentBackendConfig systèmes back-end avec tridentctl

`tridentctl` elle peut être utilisée pour lister les systèmes back-end créés à l'aide de `TridentBackendConfig`. En outre, les administrateurs peuvent choisir de gérer intégralement ces systèmes back-end `tridentctl` en supprimant et en `TridentBackendConfig` veillant à ce que `spec.deletionPolicy` soit défini sur `retain`.

#### Étape 0 : identifier le back-end

Supposons, par exemple, que le backend suivant a été créé à l'aide de TridentBackendConfig:

```

kubectl get tbc backend-tbc-ontap-san -n trident -o wide
NAME                  BACKEND NAME      BACKEND UUID
PHASE    STATUS     STORAGE DRIVER   DELETION POLICY
backend-tbc-ontap-san  ontap-san-backend  81abcb27-ea63-49bb-b606-
0a5315ac5f82    Bound    Success    ontap-san        delete

tridentctl get backend ontap-san-backend -n trident
+-----+
+-----+-----+
|       NAME      | STORAGE DRIVER |           UUID
| STATE | VOLUMES |           |
+-----+-----+
+-----+-----+
| ontap-san-backend | ontap-san      | 81abcb27-ea63-49bb-b606-
0a5315ac5f82 | online | 33 |
+-----+-----+
+-----+-----+

```

A partir de la sortie, il est vu que `TridentBackendConfig` a été créé avec succès et est lié à un backend [observer l'UUID du backend].

#### Étape 1 : confirmer `deletionPolicy` est défini sur `retain`

Examinons la valeur de `deletionPolicy`. Ce paramètre doit être défini sur `retain`. Cela garantit que lorsqu'une `TridentBackendConfig` demande de modification est supprimée, la définition du back-end est toujours présente et peut être gérée avec `tridentctl`.

```

kubectl get tbc backend-tbc-ontap-san -n trident -o wide
NAME                  BACKEND NAME      BACKEND UUID
PHASE    STATUS     STORAGE DRIVER   DELETION POLICY
backend-tbc-ontap-san  ontap-san-backend  81abcb27-ea63-49bb-b606-
0a5315ac5f82    Bound    Success    ontap-san        delete

# Patch value of deletionPolicy to retain
kubectl patch tbc backend-tbc-ontap-san --type=merge -p
'{"spec":{"deletionPolicy":"retain"}}' -n trident
tridentbackendconfig.trident.netapp.io/backend-tbc-ontap-san patched

#Confirm the value of deletionPolicy
kubectl get tbc backend-tbc-ontap-san -n trident -o wide
NAME                  BACKEND NAME      BACKEND UUID
PHASE    STATUS     STORAGE DRIVER   DELETION POLICY
backend-tbc-ontap-san  ontap-san-backend  81abcb27-ea63-49bb-b606-
0a5315ac5f82    Bound    Success    ontap-san        retain

```



Ne passez pas à l'étape suivante, sauf si `deletionPolicy` est défini sur `retain`.

## Étape 2 : supprimez la TridentBackendConfig CR

La dernière étape consiste à supprimer la `TridentBackendConfig` demande de modification. Après avoir confirmé que `deletionPolicy` est défini sur `retain`, vous pouvez procéder à la suppression :

```
kubectl delete tbc backend-tbc-ontap-san -n trident
tridentbackendconfig.trident.netapp.io "backend-tbc-ontap-san" deleted

tridentctl get backend ontap-san-backend -n trident
+-----+-----+
+-----+-----+-----+
|       NAME      | STORAGE DRIVER |          UUID
| STATE   | VOLUMES | 
+-----+-----+
+-----+-----+-----+
| ontap-san-backend | ontap-san     | 81abcb27-ea63-49bb-b606-
0a5315ac5f82 | online |      33 |
+-----+-----+
+-----+-----+
```

Lors de la suppression de `TridentBackendConfig` l'objet, Trident le supprime simplement sans réellement supprimer le back-end lui-même.

# Créer et gérer des classes de stockage

## Créer une classe de stockage

Configurez un objet `StorageClass` Kubernetes et créez la classe de stockage pour indiquer à Trident comment provisionner les volumes.

### Configuration d'un objet `StorageClass` Kubernetes

Le système "[Objet classe de stockage Kubernetes](#)" identifie Trident en tant que mécanisme de provisionnement utilisé pour cette classe et indique à Trident comment provisionner un volume. Par exemple :

```

apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: <Name>
provisioner: csi.trident.netapp.io
mountOptions: <Mount Options>
parameters:
  <Trident Parameters>
allowVolumeExpansion: true
volumeBindingMode: Immediate

```

Reportez-vous "[Kubernetes et objets Trident](#)" à pour plus de détails sur l'interaction des classes de stockage avec les PersistentVolumeClaim paramètres et pour le contrôle de la manière dont Trident provisionne les volumes.

## Créer une classe de stockage

Après avoir créé l'objet StorageClass, vous pouvez créer la classe de stockage. [Échantillons de classe de stockage](#) fournit des exemples de base que vous pouvez utiliser ou modifier.

### Étapes

- Il s'agit d'un objet Kubernetes. Utilisez donc kubectl pour le créer dans Kubernetes.

```
kubectl create -f sample-input/storage-class-basic-csi.yaml
```

- Vous devriez maintenant voir une classe de stockage **Basic-csi** dans Kubernetes et Trident, et Trident aurait dû détecter les pools sur le back-end.

```
kubectl get sc basic-csi
```

NAME	PROVISIONER	AGE
basic-csi	csi.trident.netapp.io	15h

```
./tridentctl -n trident get storageclass basic-csi -o json
```

```
{
  "items": [
    {
      "Config": {
        "version": "1",
        "name": "basic-csi",
        "attributes": {
          "backendType": "ontap-nas"
        },
        "storagePools": null,
        "additionalStoragePools": null
      },
      "storage": {
        "ontapnas_10.0.0.1": [
          "aggr1",
          "aggr2",
          "aggr3",
          "aggr4"
        ]
      }
    }
  ]
}
```

### Échantillons de classe de stockage

Trident fournit "[définition simple de classes de stockage pour des systèmes back-end spécifiques](#)".

Vous pouvez également modifier `sample-input/storage-class-csi.yaml.template` le fichier fourni avec le programme d'installation et le remplacer `BACKEND_TYPE` par le nom du pilote de stockage.

```

./tridentctl -n trident get backend
+-----+-----+
+-----+-----+
|     NAME      | STORAGE DRIVER |                         UUID          |
STATE   | VOLUMES   |
+-----+-----+
+-----+-----+
| nas-backend | ontap-nas       | 98e19b74-aec7-4a3d-8dcf-128e5033b214 |
online  |           0  |
+-----+-----+
+-----+-----+
cp sample-input/storage-class-csi.yaml.template sample-input/storage-class-
basic-csi.yaml

# Modify __BACKEND_TYPE__ with the storage driver field above (e.g.,
ontap-nas)
vi sample-input/storage-class-basic-csi.yaml

```

## Gérer les classes de stockage

Vous pouvez afficher les classes de stockage existantes, définir une classe de stockage par défaut, identifier le back-end de la classe de stockage et supprimer les classes de stockage.

### Afficher les classes de stockage existantes

- Pour afficher les classes de stockage Kubernetes existantes, exécutez la commande suivante :

```
kubectl get storageclass
```

- Pour afficher les détails de la classe de stockage Kubernetes, exécutez la commande suivante :

```
kubectl get storageclass <storage-class> -o json
```

- Pour afficher les classes de stockage synchronisées de Trident, exécutez la commande suivante :

```
tridentctl get storageclass
```

- Pour afficher le détail de la classe de stockage synchronisée de Trident, exécutez la commande suivante :

```
tridentctl get storageclass <storage-class> -o json
```

## Définir une classe de stockage par défaut

Kubernetes 1.6 a ajouté la possibilité de définir une classe de stockage par défaut. Cette classe de stockage sera utilisée pour provisionner un volume persistant si un utilisateur ne en spécifie pas une dans une demande de volume persistant.

- Définissez une classe de stockage par défaut en définissant l'annotation `storageclass.kubernetes.io/is-default-class` sur `true` dans la définition de la classe de stockage. Selon la spécification, toute autre valeur ou absence de l'annotation est interprétée comme fausse.
- Vous pouvez configurer une classe de stockage existante comme classe de stockage par défaut à l'aide de la commande suivante :

```
kubectl patch storageclass <storage-class-name> -p '{"metadata": {"annotations":{"storageclass.kubernetes.io/is-default-class":"true"}}}'
```

- De même, vous pouvez supprimer l'annotation de classe de stockage par défaut à l'aide de la commande suivante :

```
kubectl patch storageclass <storage-class-name> -p '{"metadata": {"annotations":{"storageclass.kubernetes.io/is-default-class":"false"}}}'
```

Il existe également des exemples dans le bundle du programme d'installation de Trident qui incluent cette annotation.



Votre cluster ne doit contenir qu'une seule classe de stockage par défaut à la fois. Kubernetes n'empêche pas techniquement d'en avoir plusieurs, mais il se comporte comme s'il n'existe aucune classe de stockage par défaut.

## Identifier le système back-end pour une classe de stockage

Voici un exemple du type de questions que vous pouvez répondre avec le fichier JSON qui `tridentctl` sort pour les objets backend Trident. Cela utilise l' `jq` utilitaire, que vous devrez peut-être installer en premier.

```
tridentctl get storageclass -o json | jq '[.items[] | {storageClass:.Config.name, backends: [.storage] | unique}]'
```

## Supprimer une classe de stockage

Pour supprimer une classe de stockage de Kubernetes, exécutez la commande suivante :

```
kubectl delete storageclass <storage-class>
```

<storage-class> doit être remplacé par votre classe de stockage.

Tous les volumes persistants créés via cette classe de stockage ne sont pas concernés et Trident continuera de les gérer.



Trident applique un espace vide `fsType` pour les volumes qu'il crée. Pour les systèmes backend iSCSI, il est recommandé d'appliquer la `parameters.fsType` classe de stockage. Vous devez supprimer les classes de stockage existantes et les recréer avec les classes `parameters.fsType` spécifiées.

## Provisionnement et gestion des volumes

### Provisionner un volume

Créez une demande de volume persistant qui utilise la classe de stockage Kubernetes configurée pour demander l'accès au volume persistant. Vous pouvez ensuite monter le volume persistant sur un pod.

#### Présentation

Une "["PersistentVolumeClaim"](#)" demande de volume persistant est une demande d'accès au volume persistant sur le cluster.

Le PVC peut être configuré pour demander un stockage d'une certaine taille ou d'un certain mode d'accès. À l'aide de la classe de stockage associée, l'administrateur du cluster peut contrôler plus que la taille du volume persistant et le mode d'accès, tels que les performances ou le niveau de service.

Une fois la demande de volume créée, vous pouvez la monter dans un pod.

#### Créer la PVC

##### Étapes

1. Créer la PVC.

```
kubectl create -f pvc.yaml
```

2. Vérifiez l'état du PVC.

```
kubectl get pvc
```

NAME	STATUS	VOLUME	CAPACITY	ACCESS MODES	STORAGECLASS	AGE
pvc-storage	Bound	pv-name	1Gi	RWO		5m

1. Montez le volume dans un pod.

```
kubectl create -f pv-pod.yaml
```



Vous pouvez surveiller la progression à l'aide de `kubectl get pod --watch`.

2. Vérifiez que le volume est monté sur `/my/mount/path`.

```
kubectl exec -it task-pv-pod -- df -h /my/mount/path
```

3. Vous pouvez maintenant supprimer le Pod. L'application Pod n'existera plus, mais le volume restera.

```
kubectl delete pod pv-pod
```

### Exemples de manifestes

## Exemples de manifestes de demande de volume persistant

Ces exemples présentent les options de configuration de base de la PVC.

### PVC avec accès RWO

Cet exemple montre une demande de volume persistant de base avec accès RWO associée à une classe de stockage nommée basic-csi.

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: pvc-storage
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: basic-csi
```

### PVC avec NVMe/TCP

Cet exemple montre une demande de volume persistant de base pour NVMe/TCP avec accès RWO associée à une classe de stockage nommée protection-gold.

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: pvc-san-nvme
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 300Mi
  storageClassName: protection-gold
```

## Échantillons de manifeste de pod

Ces exemples présentent les configurations de base pour fixer la demande de volume persistant à un pod.

### Configuration de base

```
kind: Pod
apiVersion: v1
metadata:
  name: pv-pod
spec:
  volumes:
    - name: storage
      persistentVolumeClaim:
        claimName: pvc-storage
  containers:
    - name: pv-container
      image: nginx
      ports:
        - containerPort: 80
          name: "http-server"
      volumeMounts:
        - mountPath: "/my/mount/path"
          name: storage
```

### Configuration NVMe/TCP de base

```
apiVersion: v1
kind: Pod
metadata:
  name: pod-nginx
spec:
  volumes:
    - name: basic-pvc
      persistentVolumeClaim:
        claimName: pvc-san-nvme
  containers:
    - name: task-pv-container
      image: nginx
      volumeMounts:
        - mountPath: "/my/mount/path"
          name: basic-pvc
```

Reportez-vous "[Kubernetes et objets Trident](#)" à pour plus de détails sur l'interaction des classes de stockage avec les PersistentVolumeClaim paramètres et pour le contrôle de la manière dont Trident provisionne les

volumes.

## Développement des volumes

Trident permet aux utilisateurs de Kubernetes d'étendre leurs volumes après leur création. Recherchez des informations sur les configurations requises pour étendre les volumes iSCSI, NFS et FC.

### Développez un volume iSCSI

Vous pouvez développer un volume persistant iSCSI à l'aide du mécanisme de provisionnement CSI.



L'extension de volume iSCSI est prise en charge par les `ontap-san` `solidfire-san` pilotes, `ontap-san-economy` et requiert Kubernetes 1.16 et versions ultérieures.

#### Étape 1 : configurer la classe de stockage pour prendre en charge l'extension de volume

Modifiez la définition de classe de stockage pour définir le `allowVolumeExpansion` champ sur `true`.

```
cat storageclass-ontapsan.yaml
```

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontap-san
provisioner: csi.trident.netapp.io
parameters:
  backendType: "ontap-san"
allowVolumeExpansion: True
```

Pour une classe de stockage existante, modifiez-la pour inclure le `allowVolumeExpansion` paramètre.

#### Étape 2 : créez une demande de volume persistant avec la classe de stockage que vous avez créée

Modifiez la définition de la PVC et mettez à jour le `spec.resources.requests.storage` pour refléter la nouvelle taille souhaitée, qui doit être supérieure à la taille d'origine.

```
cat pvc-ontapsan.yaml
```

```

kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: san-pvc
spec:
  accessModes:
  - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: ontap-san

```

Trident crée un volume persistant et l'associe à cette demande de volume persistant.

```

kubectl get pvc
NAME      STATUS      VOLUME                                     CAPACITY
ACCESS MODES      STORAGECLASS      AGE
san-pvc   Bound      pvc-8a814d62-bd58-4253-b0d1-82f2885db671   1Gi
RWO          ontap-san           8s

kubectl get pv
NAME                           CAPACITY      ACCESS MODES
RECLAIM POLICY      STATUS      CLAIM      STORAGECLASS      REASON      AGE
pvc-8a814d62-bd58-4253-b0d1-82f2885db671   1Gi          RWO
Delete            Bound      default/san-pvc      ontap-san           10s

```

### Étape 3 : définissez un pod qui fixe la demande de volume persistant

Reliez le volume persistant à un pod pour qu'il soit redimensionné. Lors du redimensionnement d'un volume persistant iSCSI, deux scénarios sont possibles :

- Si le volume persistant est connecté à un pod, Trident étend le volume sur le back-end de stockage, analyse à nouveau le périphérique et redimensionne le système de fichiers.
- Lors d'une tentative de redimensionnement d'un volume persistant non attaché, Trident étend le volume sur le back-end de stockage. Une fois le volume de volume persistant lié à un pod, Trident analyse de nouveau le périphérique et redimensionne le système de fichiers. Kubernetes met ensuite à jour la taille de la demande de volume persistant une fois l'opération d'extension terminée.

Dans cet exemple, un pod est créé et utilise le san-pvc.

```
kubectl get pod
NAME          READY   STATUS    RESTARTS   AGE
ubuntu-pod    1/1     Running   0          65s

kubectl describe pvc san-pvc
Name:           san-pvc
Namespace:      default
StorageClass:   ontap-san
Status:         Bound
Volume:         pvc-8a814d62-bd58-4253-b0d1-82f2885db671
Labels:         <none>
Annotations:   pv.kubernetes.io/bind-completed: yes
                pv.kubernetes.io/bound-by-controller: yes
                volume.beta.kubernetes.io/storage-provisioner:
                csi.trident.netapp.io
Finalizers:    [kubernetes.io/pvc-protection]
Capacity:      1Gi
Access Modes:  RWO
VolumeMode:    Filesystem
Mounted By:    ubuntu-pod
```

#### Étape 4 : développez le volume persistant

Pour redimensionner la PV créée de 1Gi à 2Gi, modifiez la définition de la PVC et mettez à jour le spec.resources.requests.storage à 2Gi.

```
kubectl edit pvc san-pvc
```

```

# Please edit the object below. Lines beginning with a '#' will be
ignored,
# and an empty file will abort the edit. If an error occurs while saving
this file will be
# reopened with the relevant failures.
#
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  annotations:
    pv.kubernetes.io/bind-completed: "yes"
    pv.kubernetes.io/bound-by-controller: "yes"
    volume.beta.kubernetes.io/storage-provisioner: csi.trident.netapp.io
  creationTimestamp: "2019-10-10T17:32:29Z"
  finalizers:
  - kubernetes.io/pvc-protection
  name: san-pvc
  namespace: default
  resourceVersion: "16609"
  selfLink: /api/v1/namespaces/default/persistentvolumeclaims/san-pvc
  uid: 8a814d62-bd58-4253-b0d1-82f2885db671
spec:
  accessModes:
  - ReadWriteOnce
  resources:
    requests:
      storage: 2Gi
# ...

```

#### Étape 5 : validation de l'extension

Vous pouvez valider le fonctionnement correct de l'extension en vérifiant la taille de la demande de volume persistant, du volume PV et du volume Trident :

```

kubectl get pvc san-pvc
NAME      STATUS      VOLUME                                     CAPACITY
ACCESS MODES   STORAGECLASS   AGE
san-pvc    Bound      pvc-8a814d62-bd58-4253-b0d1-82f2885db671   2Gi
RWO          ontap-san     11m

kubectl get pv
NAME                                         CAPACITY   ACCESS MODES
RECLAIM POLICY   STATUS      CLAIM           STORAGECLASS   REASON   AGE
pvc-8a814d62-bd58-4253-b0d1-82f2885db671   2Gi        RWO
Delete          Bound      default/san-pvc   ontap-san
tridentctl get volumes -n trident
+-----+-----+-----+
+-----+-----+-----+
|           NAME           | SIZE   | STORAGE CLASS |
PROTOCOL |           BACKEND UUID           | STATE  | MANAGED  |
+-----+-----+-----+
+-----+-----+-----+
| pvc-8a814d62-bd58-4253-b0d1-82f2885db671 | 2.0 GiB | ontap-san    |
block    | a9b7bfff-0505-4e31-b6c5-59f492e02d33 | online | true      |
+-----+-----+-----+
+-----+-----+-----+

```

## Développez un volume FC

Vous pouvez étendre un volume persistant FC à l'aide du mécanisme de provisionnement CSI.



L'extension de volume FC est prise en charge par le `ontap-san` pilote et requiert Kubernetes 1.16 et versions ultérieures.

### Étape 1 : configurer la classe de stockage pour prendre en charge l'extension de volume

Modifiez la définition de classe de stockage pour définir le `allowVolumeExpansion` champ sur `true`.

```
cat storageclass-ontapsan.yaml
```

```

apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontap-san
provisioner: csi.trident.netapp.io
parameters:
  backendType: "ontap-san"
allowVolumeExpansion: True

```

Pour une classe de stockage existante, modifiez-la pour inclure le `allowVolumeExpansion` paramètre.

#### Étape 2 : créez une demande de volume persistant avec la classe de stockage que vous avez créée

Modifiez la définition de la PVC et mettez à jour le `spec.resources.storage` pour refléter la nouvelle taille souhaitée, qui doit être supérieure à la taille d'origine.

```
cat pvc-ontapsan.yaml
```

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: san-pvc
spec:
  accessModes:
  - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: ontap-san
```

Trident crée un volume persistant et l'associe à cette demande de volume persistant.

```
kubectl get pvc
NAME      STATUS      VOLUME                                     CAPACITY
ACCESS MODES      STORAGECLASS      AGE
san-pvc    Bound      pvc-8a814d62-bd58-4253-b0d1-82f2885db671   1Gi
RWO          ontap-san        8s

kubectl get pv
NAME                           CAPACITY      ACCESS MODES
RECLAIM POLICY      STATUS      CLAIM      STORAGECLASS      REASON      AGE
pvc-8a814d62-bd58-4253-b0d1-82f2885db671   1Gi           RWO
Delete            Bound      default/san-pvc      ontap-san        10s
```

#### Étape 3 : définissez un pod qui fixe la demande de volume persistant

Reliez le volume persistant à un pod pour qu'il soit redimensionné. Lors du redimensionnement d'un volume persistant FC, il existe deux scénarios :

- Si le volume persistant est connecté à un pod, Trident étend le volume sur le back-end de stockage, analyse à nouveau le périphérique et redimensionne le système de fichiers.
- Lors d'une tentative de redimensionnement d'un volume persistant non attaché, Trident étend le volume sur le back-end de stockage. Une fois le volume de volume persistant lié à un pod, Trident analyse de nouveau le périphérique et redimensionne le système de fichiers. Kubernetes met ensuite à jour la taille de

la demande de volume persistant une fois l'opération d'extension terminée.

Dans cet exemple, un pod est créé et utilise le san-pvc.

```
kubectl get pod
NAME          READY   STATUS    RESTARTS   AGE
ubuntu-pod    1/1     Running   0          65s

kubectl describe pvc san-pvc
Name:           san-pvc
Namespace:      default
StorageClass:   ontap-san
Status:         Bound
Volume:         pvc-8a814d62-bd58-4253-b0d1-82f2885db671
Labels:         <none>
Annotations:   pv.kubernetes.io/bind-completed: yes
                pv.kubernetes.io/bound-by-controller: yes
                volume.beta.kubernetes.io/storage-provisioner:
                csi.trident.netapp.io
Finalizers:    [kubernetes.io/pvc-protection]
Capacity:      1Gi
Access Modes:  RWO
VolumeMode:    Filesystem
Mounted By:    ubuntu-pod
```

#### Étape 4 : développez le volume persistant

Pour redimensionner la PV créée de 1Gi à 2Gi, modifiez la définition de la PVC et mettez à jour le spec.resources.requests.storage à 2Gi.

```
kubectl edit pvc san-pvc
```

```

# Please edit the object below. Lines beginning with a '#' will be
ignored,
# and an empty file will abort the edit. If an error occurs while saving
this file will be
# reopened with the relevant failures.
#
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  annotations:
    pv.kubernetes.io/bind-completed: "yes"
    pv.kubernetes.io/bound-by-controller: "yes"
    volume.beta.kubernetes.io/storage-provisioner: csi.trident.netapp.io
  creationTimestamp: "2019-10-10T17:32:29Z"
  finalizers:
  - kubernetes.io/pvc-protection
  name: san-pvc
  namespace: default
  resourceVersion: "16609"
  selfLink: /api/v1/namespaces/default/persistentvolumeclaims/san-pvc
  uid: 8a814d62-bd58-4253-b0d1-82f2885db671
spec:
  accessModes:
  - ReadWriteOnce
  resources:
    requests:
      storage: 2Gi
# ...

```

#### Étape 5 : validation de l'extension

Vous pouvez valider le fonctionnement correct de l'extension en vérifiant la taille de la demande de volume persistant, du volume PV et du volume Trident :

```

kubectl get pvc san-pvc
NAME      STATUS      VOLUME                                     CAPACITY
ACCESS MODES   STORAGECLASS   AGE
san-pvc    Bound      pvc-8a814d62-bd58-4253-b0d1-82f2885db671   2Gi
RWO          ontap-san     11m

kubectl get pv
NAME                                         CAPACITY   ACCESS MODES
RECLAIM POLICY   STATUS      CLAIM           STORAGECLASS   REASON   AGE
pvc-8a814d62-bd58-4253-b0d1-82f2885db671   2Gi        RWO
Delete          Bound      default/san-pvc   ontap-san
tridentctl get volumes -n trident
+-----+-----+-----+
+-----+-----+-----+
|           NAME           | SIZE   | STORAGE CLASS |
PROTOCOL |           BACKEND UUID           | STATE  | MANAGED  |
+-----+-----+-----+
+-----+-----+-----+
| pvc-8a814d62-bd58-4253-b0d1-82f2885db671 | 2.0 GiB | ontap-san    |
block     | a9b7bfff-0505-4e31-b6c5-59f492e02d33 | online | true      |
+-----+-----+-----+
+-----+-----+-----+

```

## Développez un volume NFS

Trident prend en charge l'extension de volume des volumes NFS PVS provisionnés sur `ontap-nas`, `ontap-nas-economy`, `ontap-nas-flexgroup` gcp-cvs et les `azure-netapp-files` systèmes back-end.

### Étape 1 : configurer la classe de stockage pour prendre en charge l'extension de volume

Pour redimensionner un PV NFS, l'administrateur doit d'abord configurer la classe de stockage pour permettre l'extension du volume en définissant le `allowVolumeExpansion` champ sur `true`:

```
cat storageclass-ontapnas.yaml
```

```

apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontapnas
provisioner: csi.trident.netapp.io
parameters:
  backendType: ontap-nas
allowVolumeExpansion: true

```

Si vous avez déjà créé une classe de stockage sans cette option, vous pouvez simplement modifier la classe

de stockage existante en utilisant `kubectl edit storageclass` pour autoriser l'extension de volume.

#### Étape 2 : créez une demande de volume persistant avec la classe de stockage que vous avez créée

```
cat pvc-ontapnas.yaml
```

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: ontapnas20mb
spec:
  accessModes:
  - ReadWriteOnce
  resources:
    requests:
      storage: 20Mi
  storageClassName: ontapnas
```

Trident devrait créer un volume persistant NFS de 20 Mio pour cette demande de volume persistant :

```
kubectl get pvc
NAME           STATUS    VOLUME
CAPACITY      ACCESS MODES   STORAGECLASS     AGE
ontapnas20mb  Bound     pvc-08f3d561-b199-11e9-8d9f-5254004dfdb7  20Mi
RWO           ontapnas        9s

kubectl get pv pvc-08f3d561-b199-11e9-8d9f-5254004dfdb7
NAME                           CAPACITY      ACCESS MODES
RECLAIM POLICY    STATUS    CLAIM           STORAGECLASS     REASON
AGE
pvc-08f3d561-b199-11e9-8d9f-5254004dfdb7  20Mi          RWO
Delete           Bound     default/ontapnas20mb  ontapnas
2m42s
```

#### Étape 3 : développez le volume persistant

Pour redimensionner le nouveau volume persistant de 20 Mio à 1 Gio, modifiez la demande de volume persistant et définissez `spec.resources.requests.storage` cette valeur sur 1 Gio :

```
kubectl edit pvc ontapnas20mb
```

```

# Please edit the object below. Lines beginning with a '#' will be
ignored,
# and an empty file will abort the edit. If an error occurs while saving
this file will be
# reopened with the relevant failures.
#
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  annotations:
    pv.kubernetes.io/bind-completed: "yes"
    pv.kubernetes.io/bound-by-controller: "yes"
    volume.beta.kubernetes.io/storage-provisioner: csi.trident.netapp.io
  creationTimestamp: 2018-08-21T18:26:44Z
  finalizers:
    - kubernetes.io/pvc-protection
  name: ontapnas20mb
  namespace: default
  resourceVersion: "1958015"
  selfLink: /api/v1/namespaces/default/persistentvolumeclaims/ontapnas20mb
  uid: c1bd7fa5-a56f-11e8-b8d7-fa163e59eaab
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
# ...

```

#### Étape 4 : validation de l'extension

Vous pouvez valider le redimensionnement travaillé correctement en vérifiant la taille de la demande de volume persistant, de la valeur PV et du volume Trident :

```

kubectl get pvc ontapnas20mb
NAME           STATUS    VOLUME
CAPACITY      ACCESS MODES   STORAGECLASS     AGE
ontapnas20mb   Bound      pvc-08f3d561-b199-11e9-8d9f-5254004dfdb7   1Gi
RWO            ontapnas   4m44s

kubectl get pv pvc-08f3d561-b199-11e9-8d9f-5254004dfdb7
NAME                           CAPACITY   ACCESS MODES
RECLAIM POLICY   STATUS    CLAIM          STORAGECLASS   REASON
AGE
pvc-08f3d561-b199-11e9-8d9f-5254004dfdb7   1Gi           RWO
Delete           Bound    default/ontapnas20mb   ontapnas
5m35s

tridentctl get volume pvc-08f3d561-b199-11e9-8d9f-5254004dfdb7 -n trident
+-----+-----+-----+
+-----+-----+-----+
|           NAME           | SIZE   | STORAGE CLASS |
PROTOCOL |           BACKEND UUID          | STATE  | MANAGED  |
+-----+-----+-----+
+-----+-----+-----+
| pvc-08f3d561-b199-11e9-8d9f-5254004dfdb7 | 1.0 GiB | ontapnas      |
file     | c5a6f6a4-b052-423b-80d4-8fb491a14a22 | online | true       |
+-----+-----+-----+
+-----+-----+-----+

```

## Importer des volumes

Vous pouvez importer des volumes de stockage existants en tant que volume persistant Kubernetes à l'aide de `tridentctl import`.

### Présentation et considérations

Vous pouvez importer un volume dans Trident vers :

- Conteneurisation d'une application et réutilisation de son jeu de données existant
- Utilisez un clone d'un jeu de données pour une application éphémère
- Reconstruction d'un cluster Kubernetes en panne
- Migration des données applicatives pendant la reprise après incident

### Considérations

Avant d'importer un volume, consultez les considérations suivantes.

- Trident peut importer des volumes ONTAP de type RW (lecture-écriture) uniquement. Les volumes de type DP (protection des données) sont des volumes de destination SnapMirror. Vous devez rompre la relation de miroir avant d'importer le volume dans Trident.

- Nous vous suggérons d'importer des volumes sans connexions actives. Pour importer un volume activement utilisé, clonez-le, puis effectuez l'importation.



C'est particulièrement important pour les volumes en mode bloc, car Kubernetes ignorerait la connexion précédente et pourrait facilement relier un volume actif à un pod. Cela peut entraîner une corruption des données.

- Bien que `StorageClass` doit être spécifié sur une demande de volume persistant, Trident n'utilise pas ce paramètre lors de l'importation. Les classes de stockage sont utilisées lors de la création du volume pour sélectionner un pool disponible en fonction des caractéristiques de stockage. Comme le volume existe déjà, aucune sélection de pool n'est requise pendant l'importation. Par conséquent, l'importation n'échouera pas même si le volume existe sur un back-end ou un pool qui ne correspond pas à la classe de stockage spécifiée dans le PVC.
- La taille du volume existant est déterminée et définie dans la PVC. Une fois le volume importé par le pilote de stockage, le volume persistant est créé avec un `SécurRef` dans la demande de volume persistant.
  - La règle de récupération est initialement définie sur `retain` dans le volume persistant. Une fois que Kubernetes a réussi à relier la demande de volume persistant et le volume persistant, la règle de récupération est mise à jour pour correspondre à la règle de récupération de la classe de stockage.
  - Si la règle de récupération de la classe de stockage est `delete`, le volume de stockage sera supprimé lors de la suppression du volume persistant.
- Par défaut, Trident gère la demande de volume persistant et renomme la FlexVol volume et la LUN sur le back-end. Vous pouvez passer `--no-manage` l'indicateur pour importer un volume non géré. Si vous utilisez `--no-manage`, Trident n'effectue aucune opération supplémentaire sur la PVC ou la PV pour le cycle de vie des objets. Le volume de stockage n'est pas supprimé lorsque le volume persistant est supprimé et d'autres opérations telles que le clone de volume et le redimensionnement de volume sont également ignorées.



Cette option est utile si vous souhaitez utiliser Kubernetes pour des workloads conteneurisés, mais que vous souhaitez gérer le cycle de vie du volume de stockage en dehors de Kubernetes.

- Une annotation est ajoutée pour la demande de volume persistant et la volume persistant, qui servent un double objectif : indiquer l'importation du volume et gérer la demande de volume persistant. Cette annotation ne doit pas être modifiée ni supprimée.

## Importer un volume

Vous pouvez utiliser `tridentctl import` pour importer un volume.

### Étapes

1. Créez le fichier de demande de volume persistant (PVC) (par exemple ) qui sera utilisé pour créer la demande de volume persistant `pvc.yaml`. Le fichier PVC doit inclure `name`, `namespace`, `accessModes` et `storageClassName`. Vous pouvez également spécifier `unixPermissions` dans votre définition de PVC.

Voici un exemple de spécification minimale :

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: my_claim
  namespace: my_namespace
spec:
  accessModes:
    - ReadWriteOnce
  storageClassName: my_storage_class
```



N'incluez pas de paramètres supplémentaires tels que le nom du volume persistant ou la taille du volume. Cela peut entraîner l'échec de la commande d'importation.

2. Utilisez `tridentctl import` la commande pour spécifier le nom du back-end Trident contenant le volume et le nom qui identifie de manière unique le volume sur le stockage (par exemple : ONTAP FlexVol, Element Volume, Cloud Volumes Service path). L'`-f` argument est requis pour spécifier le chemin d'accès au fichier PVC.

```
tridentctl import volume <backendName> <volumeName> -f <path-to-pvc-file>
```

## Exemples

Consultez les exemples d'importation de volume suivants pour les pilotes pris en charge.

### NAS ONTAP et FlexGroup NAS ONTAP

Trident prend en charge l'importation de volumes à l'aide des `ontap-nas` pilotes et `ontap-nas-flexgroup`.



- Le `ontap-nas-economy` pilote ne peut pas importer et gérer des qtrees.
- Les `ontap-nas` pilotes et `ontap-nas-flexgroup` n'autorisent pas les noms de volumes dupliqués.

Chaque volume créé avec le `ontap-nas` pilote est un FlexVol volume sur le cluster ONTAP. L'importation de volumes FlexVol avec le `ontap-nas` pilote fonctionne de la même manière. Les volumes FlexVol qui existent déjà sur un cluster ONTAP peuvent être importés en tant que `ontap-nas` demande de volume persistant. De même, les volumes FlexGroup peuvent être importés en tant que `ontap-nas-flexgroup` ESV.

### Exemples de NAS ONTAP

Voici un exemple d'importation de volume géré et de volume non géré.

## Volume géré

L'exemple suivant importe un volume nommé `managed_volume` sur un back-end nommé `ontap_nas`:

```
tridentctl import volume ontap_nas managed_volume -f <path-to-pvc-file>

+-----+-----+
+-----+-----+
|           NAME          |  SIZE   | STORAGE CLASS |
PROTOCOL |          BACKEND UUID          | STATE  | MANAGED  |
+-----+-----+
+-----+-----+
| pvc-bf5ad463-afbb-11e9-8d9f-5254004dfdb7 | 1.0 GiB | standard    |
file      | c5a6f6a4-b052-423b-80d4-8fb491a14a22 | online  | true     |
+-----+-----+
+-----+-----+
```

## Volume non géré

Lors de l'utilisation de l'``--no-manage`` argument, Trident ne renomme pas le volume.

L'exemple suivant importe `unmanaged_volume` sur le `ontap_nas` back-end :

```
tridentctl import volume nas_blog unmanaged_volume -f <path-to-pvc-
file> --no-manage

+-----+-----+
+-----+-----+
|           NAME          |  SIZE   | STORAGE CLASS |
PROTOCOL |          BACKEND UUID          | STATE  | MANAGED  |
+-----+-----+
+-----+-----+
| pvc-df07d542-afbc-11e9-8d9f-5254004dfdb7 | 1.0 GiB | standard    |
file      | c5a6f6a4-b052-423b-80d4-8fb491a14a22 | online  | false    |
+-----+-----+
+-----+-----+
```

## SAN ONTAP

Trident prend en charge l'importation de volumes à l'aide des `ontap-san` pilotes et `ontap-san-economy`.

Trident peut importer des volumes FlexVol SAN ONTAP contenant une seule LUN. Cela est cohérent avec le `ontap-san` pilote, qui crée une FlexVol volume pour chaque demande de volume persistant et une LUN au sein de la FlexVol volume. Trident importe le FlexVol volume et l'associe à la définition du PVC.

## Exemples de SAN ONTAP

Voici un exemple d'importation de volume géré et de volume non géré.

### Volume géré

Pour les volumes gérés, Trident renomme le FlexVol volume au pvc-<uuid> format et le LUN dans le FlexVol volume à lun0.

L'exemple suivant importe le `ontap-san-managed` FlexVol volume présent sur le `ontap_san_default` back-end :

```
tridentctl import volume ontapsan_san_default ontap-san-managed -f pvc-basic-import.yaml -n trident -d

+-----+-----+
+-----+-----+
|           NAME          |  SIZE   | STORAGE CLASS |
PROTOCOL |           BACKEND UUID        | STATE   | MANAGED    |
+-----+-----+
+-----+-----+
| pvc-d6ee4f54-4e40-4454-92fd-d00fc228d74a | 20 MiB | basic      |
block    | cd394786-ddd5-4470-adc3-10c5ce4ca757 | online   | true       |
+-----+-----+
+-----+-----+
```

### Volume non géré

L'exemple suivant importe `unmanaged_example_volume` sur le `ontap_san` back-end :

```
tridentctl import volume -n trident san_blog unmanaged_example_volume
-f pvc-import.yaml --no-manage

+-----+-----+
+-----+-----+
|           NAME          |  SIZE   | STORAGE CLASS |
PROTOCOL |           BACKEND UUID        | STATE   | MANAGED    |
+-----+-----+
+-----+-----+
| pvc-1fc999c9-ce8c-459c-82e4-ed4380a4b228 | 1.0 GiB | san-blog      |
block    | e3275890-7d80-4af6-90cc-c7a0759f555a | online   | false       |
+-----+-----+
+-----+-----+
```

Si des LUN sont mappées à des igroups qui partagent un IQN avec un IQN de nœud Kubernetes, comme illustré dans l'exemple suivant, vous recevrez l'erreur : `LUN already mapped to initiator(s) in this group`. Vous devez supprimer l'initiateur ou annuler le mappage de la LUN pour importer le volume.

Vserver	Igroup	Protocol	OS	Type	Initiators
svm0	k8s-nodename.example.com-fe5d36f2-cded-4f38-9eb0-c7719fc2f9f3	iscsi	linux	iqn.1994-05.com.redhat:4c2e1cf35e0	
svm0	unmanaged-example-igroup	mixed	linux	iqn.1994-05.com.redhat:4c2e1cf35e0	

## Elément

Trident prend en charge le logiciel NetApp Element et l'importation de volumes NetApp HCI à l'aide du solidfire-san pilote.



Le pilote d'élément prend en charge les noms de volume dupliqués. Cependant, Trident renvoie une erreur s'il existe des noms de volumes dupliqués. Pour contourner ce problème, clonez le volume, indiquez un nom de volume unique et importez le volume cloné.

## Exemple d'élément

L'exemple suivant importe un element-managed volume sur le back-end element\_default.

```
tridentctl import volume element_default element-managed -f pvc-basic-import.yaml -n trident -d

+-----+-----+
+-----+-----+-----+
|           NAME          |  SIZE   | STORAGE CLASS |
PROTOCOL |           BACKEND UUID        | STATE   | MANAGED    |
+-----+-----+-----+
+-----+-----+-----+
| pvc-970ce1ca-2096-4ecd-8545-ac7edc24a8fe | 10 GiB | basic-element |
block     | d3ba047a-ea0b-43f9-9c42-e38e58301c49 | online  | true       |
+-----+-----+-----+
+-----+-----+-----+
```

## Google Cloud Platform

Trident prend en charge l'importation de volumes à l'aide du gcp-cvs pilote.



Pour importer un volume soutenu par NetApp Cloud Volumes Service dans Google Cloud Platform, identifiez le volume par son chemin d'accès au volume. Le chemin du volume est la partie du chemin d'exportation du volume après :/. Par exemple, si le chemin d'exportation est 10.0.0.1:/adroit-jolly-swift, le chemin du volume est adroit-jolly-swift.

## Exemple de Google Cloud Platform

L'exemple suivant importe un `gcp-cvs` volume sur le back-end `gcpcvs_YEppr` avec le chemin du volume de `adroit-jolly-swift`.

```
tridentctl import volume gcpcvs_YEppr adroit-jolly-swift -f <path-to-pvc-file> -n trident

+-----+-----+
+-----+-----+-----+
|           NAME          | SIZE   | STORAGE CLASS |
PROTOCOL |           BACKEND UUID      | STATE  | MANAGED    |
+-----+-----+-----+
+-----+-----+-----+
| pvc-a46ccab7-44aa-4433-94b1-e47fc8c0fa55 | 93 GiB | gcp-storage | file
| e1a6e65b-299e-4568-ad05-4f0a105c888f | online | true       |
+-----+-----+-----+
+-----+-----+-----+
```

## Azure NetApp Files

Trident prend en charge l'importation de volumes à l'aide du `azure-netapp-files` pilote.



Pour importer un volume Azure NetApp Files, identifiez-le par son chemin d'accès au volume. Le chemin du volume est la partie du chemin d'exportation du volume après `:`. Par exemple, si le chemin de montage est `10.0.0.2:/importvol1`, le chemin du volume est `importvol1`.

## Exemple Azure NetApp Files

L'exemple suivant importe un `azure-netapp-files` volume sur le back-end `azurenappfiles_40517` avec le chemin du volume `importvol1`.

```
tridentctl import volume azurenappfiles_40517 importvol1 -f <path-to-pvc-file> -n trident

+-----+-----+
+-----+-----+-----+
|           NAME          | SIZE   | STORAGE CLASS |
PROTOCOL |           BACKEND UUID      | STATE  | MANAGED    |
+-----+-----+-----+
+-----+-----+-----+
| pvc-0ee95d60-fd5c-448d-b505-b72901b3a4ab | 100 GiB | anf-storage |
file     | 1c01274f-d94b-44a3-98a3-04c953c9a51e | online | true       |
+-----+-----+-----+
+-----+-----+-----+
```

## Google Cloud NetApp volumes

Trident prend en charge l'importation de volumes à l'aide du google-cloud-netapp-volumes pilote.

### Exemple de Google Cloud NetApp volumes

L'exemple suivant importe un google-cloud-netapp-volumes volume sur le back-end backend-tbc-gcnv1 avec le volume testvoleasiaeast1.

```
tridentctl import volume backend-tbc-gcnv1 "testvoleasiaeast1" -f < path-to-pvc> -n trident

+-----+-----+
+-----+-----+
+-----+-----+
|           NAME          |   SIZE   | STORAGE CLASS
| PROTOCOL |           BACKEND UUID          | STATE   | MANAGED |
+-----+-----+
+-----+-----+
+-----+-----+
| pvc-a69cda19-218c-4ca9-a941-aea05dd13dc0 | 10 GiB | gcnv-nfs-sc-
identity | file      | 8c18cdf1-0770-4bc0-bcc5-c6295fe6d837 | online | true
|
+-----+-----+
+-----+-----+
+-----+-----+
```

L'exemple suivant importe un google-cloud-netapp-volumes volume lorsque deux volumes sont présents dans la même région :

```

tridentctl import volume backend-tbc-gcnv1
"projects/123456789100/locations/asia-east1-a/volumes/testvoleasiaeast1"
-f <path-to-pvc> -n trident

+-----+
+-----+-----+
+-----+-----+
|           NAME          |   SIZE   | STORAGE CLASS
| PROTOCOL |           BACKEND UUID           | STATE   | MANAGED |
+-----+-----+
+-----+-----+
+-----+-----+
| pvc-a69cda19-218c-4ca9-a941-aea05dd13dc0 | 10 GiB | gcnv-nfs-sc-
identity | file      | 8c18cdf1-0770-4bc0-bcc5-c6295fe6d837 | online | true
|
+-----+-----+
+-----+-----+
+-----+-----+

```

## Personnaliser les noms et les étiquettes des volumes

Avec Trident, vous pouvez attribuer des noms et des libellés significatifs aux volumes que vous créez. Vous pouvez ainsi identifier et mapper facilement les volumes vers leurs ressources Kubernetes respectives. Vous pouvez également définir des modèles au niveau du back-end pour créer des noms de volume personnalisés et des étiquettes personnalisées. Tous les volumes que vous créez, importez ou clonez seront conformes aux modèles.

### Avant de commencer

Prise en charge des noms et des libellés de volumes personnalisables :

- Opérations de création, d'importation et de clonage de volumes.
- Dans le cas du pilote ontap-nas-Economy, seul le nom du volume Qtree est conforme au modèle de nom.
- Dans le cas d'un pilote ontap-san-Economy, seul le nom de LUN est conforme au modèle de nom.

### Limites

- Les noms de volume personnalisables sont uniquement compatibles avec les pilotes ONTAP sur site.
- Les noms de volume personnalisables ne s'appliquent pas aux volumes existants.

### Comportements clés des noms de volume personnalisables

- Si une défaillance survient en raison d'une syntaxe incorrecte dans un modèle de nom, la création du back-end échoue. Toutefois, si l'application modèle échoue, le volume sera nommé conformément à la convention de nommage existante.

2. Le préfixe de stockage n'est pas applicable lorsqu'un volume est nommé à l'aide d'un modèle de nom de la configuration back-end. Toute valeur de préfixe souhaitée peut être directement ajoutée au modèle.

## Exemples de configuration back-end avec modèle de nom et étiquettes

Les modèles de noms personnalisés peuvent être définis au niveau de la racine et/ou du pool.

### Exemple de niveau racine

```
{  
    "version": 1,  
    "storageDriverName": "ontap-nas",  
    "backendName": "ontap-nfs-backend",  
    "managementLIF": "<ip address>",  
    "svm": "svm0",  
    "username": "<admin>",  
    "password": "<password>",  
    "defaults": {  
        "nameTemplate":  
            "{{volume.Name}}_{{labels.cluster}}_{{volume.Namespace}}_{{volume.RequestName}}"  
    },  
    "labels": {  
        "cluster": "ClusterA",  
        "PVC": "{{volume.Namespace}}_{{volume.RequestName}}"  
    }  
}
```

## Exemple au niveau du pool

```
{  
    "version": 1,  
    "storageDriverName": "ontap-nas",  
    "backendName": "ontap-nfs-backend",  
    "managementLIF": "<ip address>",  
    "svm": "svm0",  
    "username": "<admin>",  
    "password": "<password>",  
    "useREST": true,  
    "storage": [  
        {  
            "labels": {  
                "labelname": "label1",  
                "name": "{{ .volume.Name }}"  
            },  
            "defaults": {  
                "nameTemplate": "pool01_{{ .volume.Name }}_{{ .labels.cluster }}_{{ .volume.Namespace }}_{{ .volume.RequestName }}"  
            }  
        },  
        {  
            "labels": {  
                "cluster": "label2",  
                "name": "{{ .volume.Name }}"  
            },  
            "defaults": {  
                "nameTemplate": "pool02_{{ .volume.Name }}_{{ .labels.cluster }}_{{ .volume.Namespace }}_{{ .volume.RequestName }}"  
            }  
        }  
    ]  
}
```

## Exemples de modèles de noms

### Exemple 1 :

```
"nameTemplate": "{{ .config.StoragePrefix }}_{{ .volume.Name }}_{{ .config.BackendName }}"
```

### Exemple 2 :

```
"nameTemplate": "pool_{{ .config.StoragePrefix }}_{{ .volume.Name }}_{{ slice .volume.RequestName 1 5 }}""
```

## Points à prendre en compte

1. Dans le cas des importations en volume, les étiquettes ne sont mises à jour que si le volume existant comporte des étiquettes dans un format spécifique. Par exemple :  
{"provisioning": {"Cluster": "ClusterA", "PVC": "pvcname"} }.
2. Dans le cas des importations de volume gérées, le nom du volume suit le modèle de nom défini au niveau racine dans la définition du back-end.
3. Trident ne prend pas en charge l'utilisation d'un opérateur de tranche avec le préfixe de stockage.
4. Si les modèles ne donnent pas de noms de volumes uniques, Trident ajoute quelques caractères aléatoires pour créer des noms de volumes uniques.
5. Si le nom personnalisé d'un volume économique NAS dépasse 64 caractères, Trident nommera les volumes conformément à la convention de nommage existante. Pour tous les autres pilotes ONTAP, si le nom du volume dépasse la limite de nom, le processus de création du volume échoue.

## Partager un volume NFS entre les espaces de noms

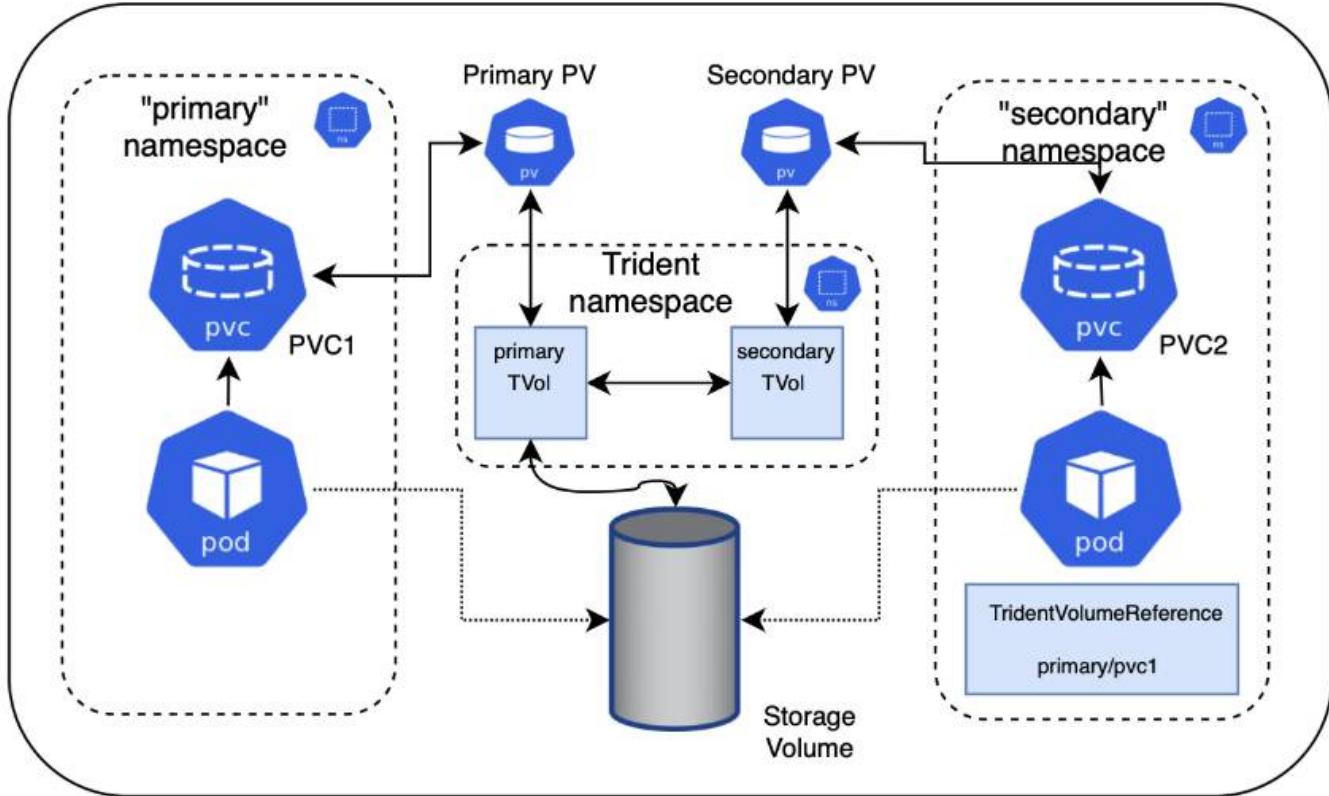
Avec Trident, vous pouvez créer un volume dans un espace de noms principal et le partager dans un ou plusieurs espaces de noms secondaires.

### Caractéristiques

Le système TridentVolumeReference CR vous permet de partager en toute sécurité des volumes ReadWriteMany (RWX) NFS sur un ou plusieurs espaces de noms Kubernetes. Cette solution Kubernetes-native présente plusieurs avantages :

- Plusieurs niveaux de contrôle d'accès pour assurer la sécurité
- Fonctionne avec tous les pilotes de volume NFS Trident
- Pas de dépendance à tridentctl ou à toute autre fonctionnalité Kubernetes non native

Ce schéma illustre le partage de volumes NFS entre deux espaces de noms Kubernetes.



## Démarrage rapide

Vous pouvez configurer le partage de volumes NFS en quelques étapes seulement.

**1**

### Configurez la PVC source pour partager le volume

Le propriétaire de l'espace de noms source autorise l'accès aux données dans la demande de volume persistant source.

**2**

### Accorder l'autorisation de créer une demande de modification dans l'espace de noms de destination

L'administrateur de cluster accorde l'autorisation au propriétaire de l'espace de noms de destination pour créer le CR TridentVolumeReference.

**3**

### Créer TridentVolumeReference dans l'espace de noms de destination

Le propriétaire de l'espace de noms de destination crée le CR TridentVolumeReference pour faire référence au PVC source.

**4**

### Créez la demande de volume persistant subordonnée dans l'espace de noms de destination

Le propriétaire de l'espace de noms de destination crée le PVC subalterne pour utiliser la source de données à partir du PVC source.

## Configurer les espaces de noms source et de destination

Pour garantir la sécurité, le partage de l'espace de noms croisé nécessite une collaboration et une action du propriétaire de l'espace de noms source, de l'administrateur de cluster et du propriétaire de l'espace de noms de destination. Le rôle utilisateur est désigné dans chaque étape.

### Étapes

1. **Propriétaire de l'espace de noms source :** Créez le PVC (pvc1) dans l'espace de noms source qui accorde l'autorisation de partager avec l'espace de noms de destination (namespace2) à l'aide de shareToNamespace l'annotation.

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: pvc1
  namespace: namespace1
  annotations:
    trident.netapp.io/shareToNamespace: namespace2
spec:
  accessModes:
    - ReadWriteMany
  storageClassName: trident-csi
  resources:
    requests:
      storage: 100Gi
```

Trident crée le volume de stockage PV et son volume de stockage NFS back-end.



- Vous pouvez partager le PVC sur plusieurs espaces de noms à l'aide d'une liste délimitée par des virgules. Par exemple trident.netapp.io/shareToNamespace: namespace2, namespace3, namespace4, .
- Vous pouvez partager sur tous les espaces de noms à l'aide de \*. Par exemple, trident.netapp.io/shareToNamespace: \*
- Vous pouvez mettre à jour la demande de volume persistant pour inclure l' `shareToNamespace` annotation à tout moment.

2. **Cluster admin:** Créez le rôle personnalisé et kubeconfig pour accorder l'autorisation au propriétaire de l'espace de noms de destination de créer le CR TridentVolumeReference dans l'espace de noms de destination.
3. **Nom de l'espace de noms de destination propriétaire :** Créez un CR TridentVolumeReference dans l'espace de noms de destination qui fait référence à l'espace de noms source pvc1 .

```

apiVersion: trident.netapp.io/v1
kind: TridentVolumeReference
metadata:
  name: my-first-tvr
  namespace: namespace2
spec:
  pvcName: pvc1
  pvcNamespace: namespacel

```

4. **Propriétaire de l'espace de noms de destination :** Créez un PVC (namespace2)(pvc2 dans l'espace de noms de destination ) en utilisant shareFromPVC l'annotation pour désigner le PVC source.

```

kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  annotations:
    trident.netapp.io/shareFromPVC: namespacel/pvc1
  name: pvc2
  namespace: namespace2
spec:
  accessModes:
    - ReadWriteMany
  storageClassName: trident-csi
  resources:
    requests:
      storage: 100Gi

```



La taille du PVC de destination doit être inférieure ou égale à la PVC source.

## Résultats

Trident lit l' `shareFromPVC` annotation sur la demande de volume de destination et crée le volume persistant de destination en tant que volume subordonné sans ressource de stockage propre qui pointe vers le volume persistant source et partage la ressource de stockage du volume persistant source. La demande de volume persistant et la demande de volume persistant de destination apparaissent comme normales.

## Supprimer un volume partagé

Vous pouvez supprimer un volume partagé entre plusieurs namespaces. Trident supprime l'accès au volume sur l'espace de noms source et maintient l'accès aux autres espaces de noms qui partagent le volume. Lorsque tous les espaces de noms qui référencent le volume sont supprimés, Trident supprime le volume.

### `tridentctl get` Permet d'interroger les volumes subordonnés

A l'aide de l'[tridentctl` utilitaire, vous pouvez exécuter la `get commande pour obtenir des volumes subordonnés. Pour plus d'informations, reportez-vous au lien :..../Trident-Reference/tridentctl.html[tridentctl commandes et options].

Usage:

```
tridentctl get [option]
```

Alarmes :

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

## Limites

- Trident ne peut pas empêcher les espaces de noms de destination d'écrire sur le volume partagé. Nous vous recommandons d'utiliser un verrouillage de fichiers ou d'autres processus pour éviter d'effacer les données du volume partagé.
- Vous ne pouvez pas révoquer l'accès au PVC source en supprimant les annotations ou ou shareFromNamespace en shareToNamespace supprimant la TridentVolumeReference demande de modification. Pour annuler l'accès, vous devez supprimer le PVC subalterne.
- Les snapshots, clones et la mise en miroir ne sont pas possibles sur les volumes subordonnés.

## Pour en savoir plus

Pour en savoir plus sur l'accès aux volumes multi-espaces de noms :

- Visitez "[Partage de volumes entre les espaces de noms : dites bonjour à l'accès aux volumes situés à l'échelle d'un espace de noms](#)".
- Regardez la démo sur "[NetAppTV](#)".

## Cloner des volumes entre des espaces de noms

Avec Trident, vous pouvez créer de nouveaux volumes à l'aide de volumes ou de volumes namesnapshots existants à partir d'un autre namespace situé dans le même cluster Kubernetes.

### Prérequis

Avant de cloner des volumes, assurez-vous que les systèmes back-end source et de destination sont du même type et ont la même classe de stockage.

### Démarrage rapide

Vous pouvez configurer le clonage de volumes en quelques étapes seulement.

1

#### Configurez la demande de volume persistant source pour cloner le volume

Le propriétaire de l'espace de noms source autorise l'accès aux données dans la demande de volume persistant source.

**2**

## Accorder l'autorisation de créer une demande de modification dans l'espace de noms de destination

L'administrateur de cluster accorde l'autorisation au propriétaire de l'espace de noms de destination pour créer le CR TridentVolumeReference.

**3**

## Créer TridentVolumeReference dans l'espace de noms de destination

Le propriétaire de l'espace de noms de destination crée le CR TridentVolumeReference pour faire référence au PVC source.

**4**

## Créer la demande de volume persistant de clone dans le namespace de destination

Le propriétaire de l'espace de noms de destination crée une demande de volume persistant pour cloner la demande de volume persistant à partir de l'espace de noms source.

### Configurer les espaces de noms source et de destination

Pour garantir la sécurité, le clonage de volumes entre espaces de noms nécessite la collaboration et l'action du propriétaire de l'espace de noms source, de l'administrateur du cluster et du propriétaire de l'espace de noms de destination. Le rôle utilisateur est désigné dans chaque étape.

#### Étapes

1. **Propriétaire de l'espace de noms source :** Créez le PVC (namespace1) (pvc1`dans l'espace de noms source qui accorde l'autorisation de partager avec l'espace de noms de destination (`namespace2) à l'aide de l'`cloneToNamespace`annotation.

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: pvc1
  namespace: namespace1
  annotations:
    trident.netapp.io/cloneToNamespace: namespace2
spec:
  accessModes:
    - ReadWriteMany
  storageClassName: trident-csi
  resources:
    requests:
      storage: 100Gi
```

Trident crée le volume PV et son volume de stockage back-end.



- Vous pouvez partager le PVC sur plusieurs espaces de noms à l'aide d'une liste délimitée par des virgules. Par exemple `trident.netapp.io/cloneToNamespace: namespace2, namespace3, namespace4, .`
- Vous pouvez partager sur tous les espaces de noms à l'aide de `*`. Par exemple, `trident.netapp.io/cloneToNamespace: *`
- Vous pouvez mettre à jour la demande de volume persistant pour inclure l'`'cloneToNamespace'`annotation à tout moment.

2. **Cluster admin:** Créez le rôle personnalisé et kubeconfig pour accorder l'autorisation au propriétaire de l'espace de noms de destination pour créer le CR TridentVolumeReference dans l'espace de noms de destination(namespace2).

3. **Nom de l'espace de noms de destination propriétaire :** Créez un CR TridentVolumeReference dans l'espace de noms de destination qui fait référence à l'espace de noms source pvc1 .

```
apiVersion: trident.netapp.io/v1
kind: TridentVolumeReference
metadata:
  name: my-first-tvr
  namespace: namespace2
spec:
  pvcName: pvc1
  pvcNamespace: namespacel
```

4. **Propriétaire de l'espace de noms de destination :** Créez un PVC (namespace2)(pvc2 dans l'espace de noms de destination en utilisant le `cloneFromPVC` ou `cloneFromSnapshot`, et les `cloneFromNamespace` annotations pour désigner le PVC source.

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  annotations:
    trident.netapp.io/cloneFromPVC: pvc1
    trident.netapp.io/cloneFromNamespace: namespacel
  name: pvc2
  namespace: namespace2
spec:
  accessModes:
    - ReadWriteMany
  storageClassName: trident-csi
  resources:
    requests:
      storage: 100Gi
```

## Limites

- Pour les demandes de provisionnement virtuels provisionnées à l'aide de pilotes ONTAP-nas-Economy, les clones en lecture seule ne sont pas pris en charge.

## RéPLICATION DE VOLUMES À L'AIDE DE SNAPMIRROR

Trident prend en charge les relations de mise en miroir entre un volume source sur un cluster et le volume de destination sur le cluster peering pour la réPLICATION des données à des fins de reprise après incident. Vous pouvez utiliser une définition de ressource personnalisée (CRD) avec un espace de nom pour effectuer les opérations suivantes :

- Création de relations de symétrie entre les volumes (ESV)
- Supprimez les relations de symétrie entre les volumes
- Rompez les relations de symétrie
- Promotion du volume secondaire en cas d'incident (basculements)
- Transition sans perte des applications d'un cluster à un autre (en cas de basculements ou de migrations planifiés)

### Conditions préalables à la réPLICATION

Assurez-vous que les conditions préalables suivantes sont remplies avant de commencer :

#### Clusters ONTAP

- **Trident** : Trident version 22.10 ou ultérieure doit exister sur les clusters Kubernetes source et destination qui utilisent ONTAP en tant que back-end.
- **Licences** : les licences asynchrones de SnapMirror ONTAP utilisant le bundle protection des données doivent être activées sur les clusters ONTAP source et cible. Pour plus d'informations, reportez-vous à la section "[Présentation des licences SnapMirror dans ONTAP](#)" .

#### Peering

- **Cluster et SVM** : les systèmes back-end de stockage ONTAP doivent être peering. Pour plus d'informations, reportez-vous à la section "[Présentation du cluster et de SVM peering](#)" .



S'assurer que les noms de SVM utilisés dans la relation de réPLICATION entre deux clusters ONTAP sont uniques.

- **Trident et SVM** : les SVM distants peering doivent être disponibles pour Trident sur le cluster destination.

#### Pilotes pris en charge

- La réPLICATION de volume est prise en charge pour les pilotes ontap-nas et ontap-san.

### Créer une demande de volume persistant en miroir

Suivez ces étapes et utilisez les exemples CRD pour créer une relation miroir entre les volumes principal et secondaire.

#### Étapes

1. Effectuez les étapes suivantes sur le cluster Kubernetes principal :

- a. Créez un objet StorageClass avec le trident.netapp.io/replication: true paramètre.

**Exemple**

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: csi-nas
provisioner: csi.trident.netapp.io
parameters:
  backendType: "ontap-nas"
  fsType: "nfs"
  trident.netapp.io/replication: "true"
```

- b. Créez une demande de volume persistant avec une classe de stockage précédemment créée.

**Exemple**

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: csi-nas
spec:
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 1Gi
  storageClassName: csi-nas
```

- c. Créez une demande de modification MirrorRelationship avec des informations locales.

**Exemple**

```
kind: TridentMirrorRelationship
apiVersion: trident.netapp.io/v1
metadata:
  name: csi-nas
spec:
  state: promoted
  volumeMappings:
    - localPVCName: csi-nas
```

Trident récupère les informations internes du volume et l'état de protection des données (DP) actuel du volume, puis remplit le champ d'état de MirrorRelationship.

- d. Obtenir le CR TridentMirrorRelationship pour obtenir le nom interne et la SVM du PVC.

```
kubectl get tmr csi-nas
```

```
kind: TridentMirrorRelationship
apiVersion: trident.netapp.io/v1
metadata:
  name: csi-nas
  generation: 1
spec:
  state: promoted
  volumeMappings:
  - localPVCName: csi-nas
status:
  conditions:
  - state: promoted
    localVolumeHandle:
      "datavserver:trident_pvc_3bedd23c_46a8_4384_b12b_3c38b313c1e1"
    localPVCName: csi-nas
    observedGeneration: 1
```

2. Effectuez les étapes suivantes sur le cluster Kubernetes secondaire :

- a. Créez une classe de stockage avec le paramètre trident.netapp.io/replication: true.

#### Exemple

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: csi-nas
provisioner: csi.trident.netapp.io
parameters:
  trident.netapp.io/replication: true
```

- b. Créez une demande de modification MirrorRelationship avec les informations de destination et de source.

## Exemple

```
kind: TridentMirrorRelationship
apiVersion: trident.netapp.io/v1
metadata:
  name: csi-nas
spec:
  state: established
  volumeMappings:
  - localPVCName: csi-nas
    remoteVolumeHandle:
      "datavserver:trident_pvc_3bedd23c_46a8_4384_b12b_3c38b313c1e1"
```

Trident crée une relation SnapMirror avec le nom de la stratégie de relation configurée (ou par défaut pour ONTAP) et l'initialise.

- c. Créez une demande de volume persistant avec une classe de stockage précédemment créée pour agir en tant que classe secondaire (destination SnapMirror).

## Exemple

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: csi-nas
  annotations:
    trident.netapp.io/mirrorRelationship: csi-nas
spec:
  accessModes:
  - ReadWriteMany
  resources:
    requests:
      storage: 1Gi
  storageClassName: csi-nas
```

Trident vérifie la CRD TridentMirrorRelationship et ne crée pas le volume si la relation n'existe pas. Si la relation existe, Trident s'assurera que le nouveau FlexVol volume est placé sur un SVM peering avec le SVM distant défini dans le MirrorRelationship.

## États de réPLICATION des volumes

Une relation de miroir Trident (TMR) est une relation CRD qui représente une extrémité d'une relation de réPLICATION entre les ESV. La TMR de destination a un état qui indique à Trident quel est l'état souhaité. La TMR de destination a les États suivants :

- **Établi** : le PVC local est le volume de destination d'une relation miroir, et il s'agit d'une nouvelle relation.
- **Promu** : Le PVC local est ReadWrite et montable, sans relation de miroir actuellement en vigueur.

- **Rétablissement:** Le PVC local est le volume de destination d'une relation miroir et était également auparavant dans cette relation miroir.
  - L'état rétabli doit être utilisé si le volume de destination était déjà en relation avec le volume source car il écrase le contenu du volume de destination.
  - L'état rétabli échouera si le volume n'était pas auparavant dans une relation avec la source.

## Promotion de la demande de volume persistant secondaire en cas de basculement non planifié

Effectuez l'étape suivante sur le cluster Kubernetes secondaire :

- Mettez à jour le champ `spec.state` de `TridentMirrorRelationship` vers `promoted`.

## Promotion de la demande de volume persistant secondaire lors d'un basculement planifié

Lors d'un basculement planifié (migration), effectuez les étapes suivantes pour promouvoir la demande de volume persistant secondaire :

### Étapes

1. Sur le cluster Kubernetes principal, créez un snapshot de la demande de volume persistant et attendez que le snapshot soit créé.
2. Sur le cluster Kubernetes principal, créez la CR `SnapshotInfo` pour obtenir des informations internes.

### Exemple

```
kind: SnapshotInfo
apiVersion: trident.netapp.io/v1
metadata:
  name: csi-nas
spec:
  snapshot-name: csi-nas-snapshot
```

3. Sur le cluster Kubernetes secondaire, mettez à jour le champ `spec.state` du `TridentMirrorRelationship` CR en `promu` et `spec.promotedSnapshotHandle` en tant que nom interne du snapshot.
4. Sur le cluster Kubernetes secondaire, confirmez l'état (champ `status.state`) de `TridentMirrorRelationship` à `promu`.

## Restaurer une relation de miroir après un basculement

Avant de restaurer une relation de symétrie, choisissez le côté que vous voulez faire comme nouveau principal.

### Étapes

1. Sur le cluster Kubernetes secondaire, assurez-vous que les valeurs du champ `spec.remoteVolumeHandle` du champ `TridentMirrorRelationship` sont mises à jour.
2. Sur le cluster Kubernetes secondaire, mettez à jour le champ `spec.mirror` de `TridentMirrorRelationship` sur `reestablished`.

## Opérations supplémentaires

Trident prend en charge les opérations suivantes sur les volumes principal et secondaire :

### Répliquer la demande de volume persistant primaire sur une nouvelle demande de volume secondaire

Assurez-vous que vous avez déjà un PVC primaire et un PVC secondaire.

#### Étapes

1. Supprimez les CRD PersistentVolumeClaim et TridentMirrorRelationship du cluster secondaire (destination) établi.
2. Supprimez le CRD TridentMirrorRelationship du cluster principal (source).
3. Créez un nouveau CRD TridentMirrorRelationship sur le cluster principal (source) pour le nouveau PVC secondaire (destination) que vous souhaitez établir.

### Redimensionner une PVC en miroir, principale ou secondaire

La demande de volume persistant peut être redimensionnée normalement, ONTAP étendra automatiquement les flexxols de destination si la quantité de données dépasse la taille actuelle.

### Supprimer la réPLICATION d'une demande de volume persistant

Pour supprimer la réPLICATION, effectuez l'une des opérations suivantes sur le volume secondaire actuel :

- Supprimez MirrorRelationship sur le PVC secondaire. Cela interrompt la relation de réPLICATION.
- Ou, mettez à jour le champ spec.state à *promu*.

### Suppression d'une demande de volume persistant (qui était auparavant mise en miroir)

Trident recherche les ESV répliquées et libère la relation de réPLICATION avant toute tentative de suppression du volume.

### Supprimer une TMR

La suppression d'une TMR d'un côté d'une relation symétrique entraîne la transition de la TMR restante vers l'état *promu* avant que Trident ne termine la suppression. Si la TMR sélectionnée pour la suppression est déjà à l'état *promoted*, il n'y a pas de relation miroir existante et la TMR sera supprimée et Trident promouvra la PVC locale en *ReadWrite*. Cette suppression libère les métadonnées SnapMirror pour le volume local dans ONTAP. Si ce volume est utilisé dans une relation miroir à l'avenir, il doit utiliser une nouvelle TMR avec un état de réPLICATION *établi* volume lors de la création de la nouvelle relation miroir.

### Mettre à jour les relations miroir lorsque ONTAP est en ligne

Les relations miroir peuvent être mises à jour à tout moment après leur établissement. Vous pouvez utiliser les state: promoted champs ou state: reestablished pour mettre à jour les relations. Lors de la promotion d'un volume de destination en volume ReadWrite standard, vous pouvez utiliser promotedSnapshotHandle pour spécifier un snapshot spécifique dans lequel restaurer le volume actuel.

### Mettre à jour les relations en miroir lorsque ONTAP est hors ligne

Vous pouvez utiliser un CRD pour effectuer une mise à jour SnapMirror sans que Trident ne dispose d'une connectivité directe au cluster ONTAP. Reportez-vous à l'exemple de format de TridentActionMirrorUpdate suivant :

## Exemple

```
apiVersion: trident.netapp.io/v1
kind: TridentActionMirrorUpdate
metadata:
  name: update-mirror-b
spec:
  snapshotHandle: "pvc-1234/snapshot-1234"
  tridentMirrorRelationshipName: mirror-b
```

status.state Reflète l'état du CRD TridentActionMirrorUpdate. Il peut prendre une valeur de *succeed*, *In Progress* ou *FAILED*.

## Utiliser la topologie CSI

Trident peut créer et attacher de manière sélective des volumes aux nœuds présents dans un cluster Kubernetes en utilisant le "[Fonction de topologie CSI](#)".

### Présentation

Grâce à la fonction de topologie CSI, l'accès aux volumes peut être limité à un sous-ensemble de nœuds, en fonction des régions et des zones de disponibilité. Les fournisseurs cloud permettent aujourd'hui aux administrateurs Kubernetes de frayer des nœuds basés sur une zone. Les nœuds peuvent se trouver dans différentes zones de disponibilité au sein d'une région ou entre différentes régions. Pour faciliter le provisionnement des volumes pour les charges de travail dans une architecture multi-zone, Trident utilise la topologie CSI.



En savoir plus sur la fonctionnalité topologie CSI "[ici](#)" .

Kubernetes propose deux modes de liaison de volumes :

- Avec la `VolumeBindingMode` valeur définie sur `Immediate`, Trident crée le volume sans connaissance de la topologie. La liaison de volumes et le provisionnement dynamique sont gérés au moment de la création de la demande de volume persistant. Il s'agit de la valeur par défaut `VolumeBindingMode` et convient aux clusters qui n'appliquent pas de contraintes de topologie. Les volumes persistants sont créés sans dépendance vis-à-vis des exigences de planification du pod demandeur.
- Avec la `VolumeBindingMode` valeur définie sur `WaitForFirstConsumer`, la création et la liaison d'un volume persistant pour une demande de volume persistant sont retardées jusqu'à ce qu'un pod qui utilise la demande de volume persistant soit planifié et créé. De cette façon, les volumes sont créés pour répondre aux contraintes de planification appliquées en fonction des besoins de topologie.



Le `WaitForFirstConsumer` mode de liaison ne nécessite pas d'étiquettes de topologie. Il peut être utilisé indépendamment de la fonction de topologie CSI.

### Ce dont vous avez besoin

Pour utiliser la topologie CSI, vous devez disposer des éléments suivants :

- Cluster Kubernetes exécutant une "[Version Kubernetes prise en charge](#)"

```

kubectl version
Client Version: version.Info{Major:"1", Minor:"19",
GitVersion:"v1.19.3",
GitCommit:"1e11e4a2108024935ecfcb2912226cedeadfd99df",
GitTreeState:"clean", BuildDate:"2020-10-14T12:50:19Z",
GoVersion:"go1.15.2", Compiler:"gc", Platform:"linux/amd64"}
Server Version: version.Info{Major:"1", Minor:"19",
GitVersion:"v1.19.3",
GitCommit:"1e11e4a2108024935ecfcb2912226cedeadfd99df",
GitTreeState:"clean", BuildDate:"2020-10-14T12:41:49Z",
GoVersion:"go1.15.2", Compiler:"gc", Platform:"linux/amd64"}

```

- Les nœuds du cluster doivent avoir des étiquettes qui permettent de prendre en compte la topologie (`topology.kubernetes.io/region` et `topology.kubernetes.io/zone`). Ces libellés **doivent être présents sur les nœuds du cluster** avant l'installation de Trident pour que Trident soit compatible avec la topologie.

```

kubectl get nodes -o=jsonpath='{range .items[*]}{{.metadata.name},\n{.metadata.labels}}{"\n"}{end}' | grep --color "topology.kubernetes.io"
[node1,
{"beta.kubernetes.io/arch":"amd64","beta.kubernetes.io/os":"linux","kubernetes.io/arch":"amd64","kubernetes.io/hostname":"node1","kubernetes.io/os":"linux","node-
role.kubernetes.io/master":"","topology.kubernetes.io/region":"us-
east1","topology.kubernetes.io/zone":"us-east1-a"}]
[node2,
 {"beta.kubernetes.io/arch":"amd64","beta.kubernetes.io/os":"linux","kubernetes.io/arch":"amd64","kubernetes.io/hostname":"node2","kubernetes.io/os":"linux","node-
role.kubernetes.io/worker":"","topology.kubernetes.io/region":"us-
east1","topology.kubernetes.io/zone":"us-east1-b"}]
[node3,
 {"beta.kubernetes.io/arch":"amd64","beta.kubernetes.io/os":"linux","kubernetes.io/arch":"amd64","kubernetes.io/hostname":"node3","kubernetes.io/os":"linux","node-
role.kubernetes.io/worker":"","topology.kubernetes.io/region":"us-
east1","topology.kubernetes.io/zone":"us-east1-c"}]

```

## Étape 1 : création d'un back-end conscient de la topologie

Les systèmes back-end de stockage Trident peuvent être conçus pour provisionner de manière sélective des volumes en fonction des zones de disponibilité. Chaque back-end peut porter un bloc facultatif `supportedTopologies` représentant une liste de zones et de régions prises en charge. Pour les classes de stockage qui utilisent un tel backend, un volume ne sera créé que si une application est planifiée dans une région/zone prise en charge.

Voici un exemple de définition de back-end :

#### YAML

```
---
version: 1
storageDriverName: ontap-san
backendName: san-backend-us-east1
managementLIF: 192.168.27.5
svm: iscsi_svm
username: admin
password: password
supportedTopologies:
  - topology.kubernetes.io/region: us-east1
    topology.kubernetes.io/zone: us-east1-a
  - topology.kubernetes.io/region: us-east1
    topology.kubernetes.io/zone: us-east1-b
```

#### JSON

```
{
  "version": 1,
  "storageDriverName": "ontap-san",
  "backendName": "san-backend-us-east1",
  "managementLIF": "192.168.27.5",
  "svm": "iscsi_svm",
  "username": "admin",
  "password": "password",
  "supportedTopologies": [
    {
      "topology.kubernetes.io/region": "us-east1",
      "topology.kubernetes.io/zone": "us-east1-a"
    },
    {
      "topology.kubernetes.io/region": "us-east1",
      "topology.kubernetes.io/zone": "us-east1-b"
    }
  ]
}
```

 **supportedTopologies** est utilisé pour fournir une liste de régions et de zones par back-end. Ces régions et ces zones représentent la liste des valeurs admissibles qui peuvent être fournies dans une classe de stockage. Pour les classes de stockage qui contiennent un sous-ensemble des régions et zones fournies dans un back-end, Trident crée un volume sur le back-end.

Vous pouvez également définir supportedTopologies par pool de stockage. Voir l'exemple suivant :

```
---
version: 1
storageDriverName: ontap-nas
backendName: nas-backend-us-central1
managementLIF: 172.16.238.5
svm: nfs_svm
username: admin
password: password
supportedTopologies:
  - topology.kubernetes.io/region: us-central1
    topology.kubernetes.io/zone: us-central1-a
  - topology.kubernetes.io/region: us-central1
    topology.kubernetes.io/zone: us-central1-b
storage:
  - labels:
      workload: production
    supportedTopologies:
      - topology.kubernetes.io/region: us-central1
        topology.kubernetes.io/zone: us-central1-a
  - labels:
      workload: dev
    supportedTopologies:
      - topology.kubernetes.io/region: us-central1
        topology.kubernetes.io/zone: us-central1-b
```

Dans cet exemple, les region étiquettes et zone représentent l'emplacement du pool de stockage. topology.kubernetes.io/region et topology.kubernetes.io/zone détermine d'où les pools de stockage peuvent être consommés.

## Étape 2 : définissez des classes de stockage qui prennent en compte la topologie

Les classes de stockage peuvent être définies en fonction des labels de topologie fournis aux nœuds du cluster, et contenir des informations de topologie. Cela déterminera les pools de stockage qui servent de candidats aux demandes de volume persistant faites et le sous-ensemble de nœuds qui peuvent utiliser les volumes provisionnés par Trident.

Voir l'exemple suivant :

```

apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata: null
name: netapp-san-us-east1
provisioner: csi.trident.netapp.io
volumeBindingMode: WaitForFirstConsumer
allowedTopologies:
  - matchLabelExpressions: null
  - key: topology.kubernetes.io/zone
    values:
      - us-east1-a
      - us-east1-b
  - key: topology.kubernetes.io/region
    values:
      - us-east1
parameters:
  fsType: ext4

```

Dans la définition de classe de stockage fournie ci-dessus, volumeBindingMode est définie sur WaitForFirstConsumer. Les demandes de volume persistant demandées pour cette classe de stockage ne seront pas traitées tant qu'elles ne seront pas référencées dans un pod. Et allowedTopologies fournit les zones et la région à utiliser. La netapp-san-us-east1 classe de stockage crée des ESV sur le san-backend-us-east1 back-end défini ci-dessus.

### Étape 3 : création et utilisation d'une demande de volume persistant

Une fois la classe de stockage créée et mappée à un back-end, vous pouvez désormais créer des demandes de volume persistant.

Voir l'exemple spec ci-dessous :

```

---
kind: PersistentVolumeClaim
apiVersion: v1
metadata: null
name: pvc-san
spec: null
accessModes:
  - ReadWriteOnce
resources:
  requests:
    storage: 300Mi
storageClassName: netapp-san-us-east1

```

La création d'une demande de volume persistant à l'aide de ce manifeste se traduit par les éléments suivants :

```

kubectl create -f pvc.yaml
persistentvolumeclaim/pvc-san created
kubectl get pvc
NAME      STATUS      VOLUME      CAPACITY      ACCESS MODES      STORAGECLASS
AGE
pvc-san   Pending          netapp-san-us-east1
2s

kubectl describe pvc
Name:            pvc-san
Namespace:       default
StorageClass:    netapp-san-us-east1
Status:          Pending
Volume:
Labels:          <none>
Annotations:    <none>
Finalizers:     [kubernetes.io/pvc-protection]
Capacity:
Access Modes:
VolumeMode:     Filesystem
Mounted By:    <none>
Events:
  Type  Reason           Age      From           Message
  ----  -----           ----     ----
  Normal  WaitForFirstConsumer  6s      persistentvolume-controller  waiting
for first consumer to be created before binding

```

Pour que Trident puisse créer un volume et le lier à la demande de volume persistant, utilisez la demande de volume persistant dans un pod. Voir l'exemple suivant :

```

apiVersion: v1
kind: Pod
metadata:
  name: app-pod-1
spec:
  affinity:
    nodeAffinity:
      requiredDuringSchedulingIgnoredDuringExecution:
        nodeSelectorTerms:
        - matchExpressions:
          - key: topology.kubernetes.io/region
            operator: In
            values:
            - us-east1
      preferredDuringSchedulingIgnoredDuringExecution:
      - weight: 1
        preference:
          matchExpressions:
          - key: topology.kubernetes.io/zone
            operator: In
            values:
            - us-east1-a
            - us-east1-b
  securityContext:
    runAsUser: 1000
    runAsGroup: 3000
    fsGroup: 2000
  volumes:
  - name: vol1
    persistentVolumeClaim:
      claimName: pvc-san
  containers:
  - name: sec-ctx-demo
    image: busybox
    command: [ "sh", "-c", "sleep 1h" ]
    volumeMounts:
    - name: vol1
      mountPath: /data/demo
    securityContext:
      allowPrivilegeEscalation: false

```

Ce podSpec demande à Kubernetes de planifier le pod sur les nœuds présents dans `us-east1` la région, puis de choisir parmi le nœud présent dans les `us-east1-a` zones ou `us-east1-b`.

Voir le résultat suivant :

```

kubectl get pods -o wide
NAME        READY   STATUS    RESTARTS   AGE      IP          NODE
NOMINATED NODE   READINESS GATES
app-pod-1   1/1     Running   0          19s     192.168.25.131   node2
<none>           <none>
kubectl get pvc -o wide
NAME        STATUS    VOLUME
ACCESS MODES   STORAGECLASS   AGE      VOLUMEMODE
pvc-san     Bound     pvc-ecb1e1a0-840c-463b-8b65-b3d033e2e62b   300Mi
RWO          netapp-san-us-east1  48s     Filesystem

```

## Mettez à jour les systèmes back-end pour inclure supportedTopologies

Les systèmes back-end préexistants peuvent être mis à jour pour inclure une liste d'`supportedTopologies``utilisation `tridentctl backend update. Cela n'affecte pas les volumes qui ont déjà été provisionnés et ne sera utilisé que pour les demandes de volume virtuel suivantes.

## Trouvez plus d'informations

- "[Gestion des ressources pour les conteneurs](#)"
- "[Outil de sélection de nœud](#)"
- "[Affinité et anti-affinité](#)"
- "[Teintes et tolérances](#)"

## Travailler avec des instantanés

Les copies Snapshot de volume Kubernetes de volumes persistants (PVS) permettent d'effectuer des copies instantanées de volumes. Vous pouvez créer un snapshot d'un volume créé à l'aide de Trident, importer un snapshot créé en dehors de Trident, créer un volume à partir d'un snapshot existant et restaurer des données de volume à partir de snapshots.

### Présentation

L'instantané de volume est pris en charge par `ontap-nas` , `ontap-nas-flexgroup` , `ontap-san` , `ontap-san-economy` , `solidfire-san` , `gcp-cvs` , `azure-netapp-files` , et `google-cloud-netapp-volumes` conducteurs.

### Avant de commencer

Vous devez disposer d'un contrôleur de snapshot externe et de définitions de ressources personnalisées (CRD) pour pouvoir utiliser les snapshots. Cela relève de la responsabilité de l'orchestrateur Kubernetes (par exemple : Kubeadm, GKE, OpenShift).

Si votre distribution Kubernetes n'inclut pas le contrôleur de snapshot et les CRD, reportez-vous à la [Déployer un contrôleur de snapshot de volume](#).



Ne créez pas de contrôleur de snapshot si vous créez des snapshots de volume à la demande dans un environnement GKE. GKE utilise un contrôleur de snapshot caché intégré.

## Créer un snapshot de volume

### Étapes

1. Créez un `VolumeSnapshotClass`. Pour plus d'informations, reportez-vous à "[VolumeSnapshotClass](#)" la section .
  - Le driver signale au conducteur Trident CSI.
  - `deletionPolicy` peut être `Delete` ou `Retain`. Lorsqu'il est défini sur `Retain`, le snapshot physique sous-jacent du cluster de stockage est conservé même lorsque l'`VolumeSnapshot` objet est supprimé.

#### Exemple

```
cat snap-sc.yaml
```

```
apiVersion: snapshot.storage.k8s.io/v1
kind: VolumeSnapshotClass
metadata:
  name: csi-snapclass
driver: csi.trident.netapp.io
deletionPolicy: Delete
```

2. Créer un snapshot d'une demande de volume persistant existante.

#### Exemples

- Dans cet exemple, nous allons créer un snapshot d'un volume persistant existant.

```
cat snap.yaml
```

```
apiVersion: snapshot.storage.k8s.io/v1
kind: VolumeSnapshot
metadata:
  name: pvc1-snap
spec:
  volumeSnapshotClassName: csi-snapclass
  source:
    persistentVolumeClaimName: pvc1
```

- Cet exemple crée un objet de snapshot de volume pour une demande de volume persistant nommée `pvc1` et le nom de l'instantané est défini sur `pvc1-snap`. Un `VolumeSnapshot` est similaire à une demande de volume persistant et est associé à un `VolumeSnapshotContent` objet qui représente le snapshot réel.

```
kubectl create -f snap.yaml
volumesnapshot.snapshot.storage.k8s.io/pvc1-snap created

kubectl get volumesnapshots
NAME          AGE
pvc1-snap    50s
```

- Vous pouvez identifier `VolumeSnapshotContent` l'objet du `pvc1-snap` `VolumeSnapshot` en le décrivant. Le système `Snapshot Content Name` identifie l'objet `VolumeSnapshotContent` qui sert ce snapshot. Le `Ready To Use` paramètre indique que l'instantané peut être utilisé pour créer une nouvelle demande de volume persistant.

```
kubectl describe volumesnapshots pvc1-snap
Name:           pvc1-snap
Namespace:      default
...
Spec:
  Snapshot Class Name:  pvc1-snap
  Snapshot Content Name: snapcontent-e8d8a0ca-9826-11e9-9807-
  525400f3f660
  Source:
    API Group:   ""
    Kind:        PersistentVolumeClaim
    Name:         pvc1
Status:
  Creation Time: 2019-06-26T15:27:29Z
  Ready To Use:  true
  Restore Size:  3Gi
...
```

## Créer une demande de volume persistant à partir d'un snapshot de volume

Vous pouvez utiliser `dataSource` pour créer une demande de volume persistant à l'aide d'un `VolumeSnapshot` nommé <pvc-name> source des données. Une fois la demande de volume persistant créée, elle peut être connectée à un pod et utilisée comme n'importe quel autre PVC.

 La demande de volume sera créée dans le même back-end que le volume source. Reportez-vous à la "[Base de connaissances : la création d'une demande de volume persistant à partir d'un Snapshot de volume persistant Trident ne peut pas être créée dans un autre back-end](#)".

L'exemple suivant crée la demande de volume persistant en utilisant `pvc1-snap` comme source de données.

```
cat pvc-from-snap.yaml
```

```

apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: pvc-from-snap
spec:
  accessModes:
    - ReadWriteOnce
  storageClassName: golden
  resources:
    requests:
      storage: 3Gi
  dataSource:
    name: pvcl-snap
    kind: VolumeSnapshot
    apiGroup: snapshot.storage.k8s.io

```

## Importer un instantané de volume

Trident prend en charge la "[Processus Snapshot prévisionné Kubernetes](#)" permettant à l'administrateur du cluster de créer un `VolumeSnapshotContent` objet et d'importer des snapshots créés en dehors de Trident.

### Avant de commencer

Trident doit avoir créé ou importé le volume parent du snapshot.

### Étapes

- Cluster admin:** Créez un `VolumeSnapshotContent` objet qui fait référence au snapshot back-end. Ceci lance le flux de travail de snapshot dans Trident.
  - Spécifiez le nom du snapshot back-end dans annotations comme `trident.netapp.io/internalSnapshotName: <"backend-snapshot-name">`.
  - Spécifiez `<name-of-parent-volume-in-trident>/<volume-snapshot-content-name>` dans `snapshotHandle`. Il s'agit de la seule information fournie à Trident par le snapshotter externe dans l'`'ListSnapshots'` appel.



Le système `<volumeSnapshotContentName>` ne peut pas toujours correspondre au nom du snapshot back-end en raison des contraintes de dénomination CR.

### Exemple

L'exemple suivant crée un `VolumeSnapshotContent` objet qui référence le snapshot back-end `snap-01`.

```

apiVersion: snapshot.storage.k8s.io/v1
kind: VolumeSnapshotContent
metadata:
  name: import-snap-content
  annotations:
    trident.netapp.io/internalSnapshotName: "snap-01" # This is the
name of the snapshot on the backend
spec:
  deletionPolicy: Retain
  driver: csi.trident.netapp.io
  source:
    snapshotHandle: pvc-f71223b5-23b9-4235-bbfe-e269ac7b84b0/import-
snap-content # <import PV name or source PV name>/<volume-snapshot-
content-name>
  volumeSnapshotRef:
    name: import-snap
    namespace: default

```

2. **Cluster admin:** Créez la VolumeSnapshot CR qui fait référence à l'VolumeSnapshotContent`objet. Cette opération demande l'accès à pour utiliser `VolumeSnapshot dans un espace de noms donné.

#### Exemple

L'exemple suivant crée une VolumeSnapshot demande de modification nommée qui fait référence à l'VolumeSnapshotContent nommée import-snap import-snap-content .

```

apiVersion: snapshot.storage.k8s.io/v1
kind: VolumeSnapshot
metadata:
  name: import-snap
spec:
  # volumeSnapshotClassName: csi-snapclass (not required for pre-
provisioned or imported snapshots)
  source:
    volumeSnapshotContentName: import-snap-content

```

3. **Traitement interne (aucune action requise):** le snapshotter externe reconnaît le nouveau créé VolumeSnapshotContent et exécute l'ListSnapshots`appel. Trident crée le `TridentSnapshot.

- Le snapshotter externe définit le VolumeSnapshotContent sur readyToUse et le VolumeSnapshot sur true.
- Trident renvoie readyToUse=true.

4. **Tout utilisateur :** Créez un PersistentVolumeClaim pour référencer le nouveau VolumeSnapshot, où le spec.dataSource nom (ou spec.dataSourceRef) est le VolumeSnapshot nom.

## Exemple

L'exemple suivant crée un PVC faisant référence au VolumeSnapshot import-snap .

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: pvc-from-snap
spec:
  accessModes:
    - ReadWriteOnce
  storageClassName: simple-sc
  resources:
    requests:
      storage: 1Gi
  dataSource:
    name: import-snap
    kind: VolumeSnapshot
    apiGroup: snapshot.storage.k8s.io
```

## Restaurez les données de volume à l'aide de snapshots

Le répertoire des snapshots est masqué par défaut pour faciliter la compatibilité maximale des volumes provisionnés à l'aide des ontap-nas pilotes et ontap-nas-economy. Activez le .snapshot répertoire pour restaurer directement les données à partir de snapshots.

Utilisez l'interface de ligne de commandes ONTAP de restauration de snapshot de volume pour restaurer un volume à un état enregistré dans un snapshot précédent.

```
cluster1::> volume snapshot restore -vserver vs0 -volume vol3 -snapshot
vol3_snap_archive
```



Lorsque vous restaurez une copie Snapshot, la configuration de volume existante est écrasée. Les modifications apportées aux données de volume après la création de la copie Snapshot sont perdues.

## Restauration de volumes sur place à partir d'un snapshot

Trident assure une restauration rapide des volumes sur place à partir d'un snapshot à l'aide du TridentActionSnapshotRestore système CR (TASR). Cette CR fonctionne comme une action Kubernetes impérative et ne persiste pas une fois l'opération terminée.

Trident prend en charge la restauration de snapshot sur ontap-san , ontap-san-economy ontap-nas , , ontap-nas-flexgroup, , , azure-netapp-files gcp-cvs, google-cloud-netapp-volumes et solidfire-san pilotes.

## Avant de commencer

Vous devez disposer d'une demande de volume liée et d'un instantané de volume disponible.

- Vérifiez que l'état de la demande de volume persistant est lié.

```
kubectl get pvc
```

- Vérifiez que le snapshot du volume est prêt à être utilisé.

```
kubectl get vs
```

## Étapes

1. Créez la CR TASR. Cet exemple crée une CR pour la PVC `pvc1` et l'instantané de volume `pvc1-snapshot`.



Le CR TIR doit se trouver dans un espace de nom où le PVC et le VS existent.

```
cat tasr-pvc1-snapshot.yaml
```

```
apiVersion: trident.netapp.io/v1
kind: TridentActionSnapshotRestore
metadata:
  name: trident-snap
  namespace: trident
spec:
  pvcName: pvc1
  volumeSnapshotName: pvc1-snapshot
```

2. Appliquez la CR pour effectuer une restauration à partir de l'instantané. Cet exemple permet de restaurer des données à partir d'un snapshot `pvc1`.

```
kubectl create -f tasr-pvc1-snapshot.yaml
```

```
tridentactionsnapshotrestore.trident.netapp.io/trident-snap created
```

## Résultats

Trident restaure les données à partir du snapshot. Vous pouvez vérifier l'état de la restauration des snapshots :

```
kubectl get tasr -o yaml
```

```

apiVersion: trident.netapp.io/v1
items:
- apiVersion: trident.netapp.io/v1
  kind: TridentActionSnapshotRestore
  metadata:
    creationTimestamp: "2023-04-14T00:20:33Z"
    generation: 3
    name: trident-snap
    namespace: trident
    resourceVersion: "3453847"
    uid: <uid>
  spec:
    pvcName: pvc1
    volumeSnapshotName: pvc1-snapshot
  status:
    startTime: "2023-04-14T00:20:34Z"
    completionTime: "2023-04-14T00:20:37Z"
    state: Succeeded
kind: List
metadata:
  resourceVersion: ""

```

- i**
- Dans la plupart des cas, Trident ne réessaiera pas automatiquement l'opération en cas d'échec. Vous devrez effectuer à nouveau l'opération.
  - Les utilisateurs Kubernetes sans accès administrateur peuvent avoir à obtenir l'autorisation de l'administrateur pour créer une CR ASR dans l'espace de noms de leur application.

## Supprimez un volume persistant avec les snapshots associés

Lors de la suppression d'un volume persistant avec snapshots associés, le volume Trident correspondant est mis à jour et passe à l'état « Suppression ». Supprimez les snapshots de volume pour supprimer le volume Trident.

### Déployer un contrôleur de snapshot de volume

Si votre distribution Kubernetes n'inclut pas le contrôleur de snapshot et les CRD, vous pouvez les déployer comme suit.

#### Étapes

1. Création de CRD de snapshot de volume.

```
cat snapshot-setup.sh
```

```
#!/bin/bash
# Create volume snapshot CRDs
kubectl apply -f https://raw.githubusercontent.com/kubernetes-csi/external-snapshotter/release-6.1/client/config/crd/snapshot.storage.k8s.io_volumesnapshotclasses.yaml
kubectl apply -f https://raw.githubusercontent.com/kubernetes-csi/external-snapshotter/release-6.1/client/config/crd/snapshot.storage.k8s.io_volumesnapshotcontents.yaml
kubectl apply -f https://raw.githubusercontent.com/kubernetes-csi/external-snapshotter/release-6.1/client/config/crd/snapshot.storage.k8s.io_volumesnapshots.yaml
```

## 2. Créer le contrôleur de snapshot.

```
kubectl apply -f https://raw.githubusercontent.com/kubernetes-csi/external-snapshotter/release-6.1/deploy/kubernetes/snapshot-controller/rbac-snapshot-controller.yaml
```

```
kubectl apply -f https://raw.githubusercontent.com/kubernetes-csi/external-snapshotter/release-6.1/deploy/kubernetes/snapshot-controller/setup-snapshot-controller.yaml
```



Si nécessaire, ouvrez `deploy/kubernetes/snapshot-controller/rbac-snapshot-controller.yaml` et mettez à jour namespace votre espace de noms.

## Liens connexes

- "[Snapshots de volume](#)"
- "[VolumeSnapshotClass](#)"

## **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 JOUSSANCE, 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.