# Configurer et gérer les backends

## Trident

NetApp
July 01, 2026

This PDF was generated from https://docs.netapp.com/fr-fr/trident-2602/trident-use/backends.html on July 01, 2026. Always check docs.netapp.com for the latest.

# Sommaire


Configurer et gérer les backends . . . 1
- Configurer les backends . . . 1
- Azure NetApp Files . . . 1
  - Configurer un backend Azure NetApp Files . . . 1
  - Préparez-vous à configurer un backend Azure NetApp Files . . . 5
  - Options et exemples de configuration du backend Azure NetApp Files . . . 8
- Google Cloud NetApp Volumes . . . 21
  - Configurer Google Cloud NetApp Volumes . . . 21
  - Configurer Google Cloud NetApp Volumes pour les charges de travail SAN . . . 26
  - Préparez-vous à configurer un backend Google Cloud NetApp Volumes . . . 32
  - Options et exemples de configuration du backend Google Cloud NetApp Volumes . . . 32
  - Configurer la hiérarchisation automatique pour Google Cloud NetApp Volumes . . . 45
- Configurer un backend NetApp HCI ou SolidFire . . . 48
  - Détails du pilote Element . . . 48
  - Avant de commencer . . . 49
  - Options de configuration du backend . . . 49
  - Exemple 1 : Configuration du backend pour `solidfire-san` driver avec trois types de volumes . . . 50
  - Exemple 2 : Configuration du backend et de la classe de stockage pour `solidfire-san` driver avec pools virtuels . . . 51
  - Pour plus d'informations . . . 54
- Pilotes ONTAP SAN . . . 54
  - Présentation du pilote ONTAP SAN . . . 54
  - Préparez-vous à configurer le backend avec les pilotes SAN ONTAP . . . 56
  - Options et exemples de configuration SAN ONTAP . . . 64
- Pilotes ONTAP NAS . . . 84
  - Présentation du pilote ONTAP NAS . . . 84
  - Préparez-vous à configurer un backend avec les pilotes NAS ONTAP . . . 85
  - Options et exemples de configuration NAS ONTAP . . . 98
- Amazon FSx for NetApp ONTAP . . . 123
  - Utilisez Trident avec Amazon FSx for NetApp ONTAP . . . 123
  - Créez un rôle IAM et un secret AWS . . . 126
  - Installer Trident . . . 132
  - Configurer une classe de stockage . . . 139
  - Configurer un PVC . . . 155
  - Déployer une application . . . 156
  - Déployer une application exemple . . . 156
  - Configurez le module complémentaire Trident EKS sur un cluster EKS . . . 158
- Créer des backends avec kubectl . . . 161
  - `TridentBackendConfig` . . . 161
  - Aperçu des étapes . . . 163
  - Étape 1 : Créer un secret Kubernetes . . . 163
  - Étape 2 : Créer la `TridentBackendConfig` demande de changement . . . 164

Étape 3 : Vérifiez le statut du `TridentBackendConfig` CR . . . 165
(Facultatif) Étape 4 : Obtenir plus de détails . . . 166
Gérer les backends . . . 168
Effectuez la gestion du backend avec kubectl . . . 168
Effectuez la gestion du backend avec tridentctl . . . 169
Passer d'une option de gestion du backend à une autre . . . 171

# Configurer et gérer les backends

## Configurer les backends

Un backend définit la relation entre Trident et un système de stockage. Il indique à Trident comment communiquer avec ce système de stockage et comment Trident doit provisionner des volumes à partir de celui-ci.

Trident propose automatiquement des pools de stockage provenant de backends correspondant aux exigences définies par une classe de stockage. Découvrez comment configurer le backend pour votre système de stockage.

- "Configurer un backend Azure NetApp Files"
- "Configurer un backend Google Cloud NetApp Volumes"
- "Configurer un backend NetApp HCI ou SolidFire"
- "Configurez un backend avec les pilotes NAS ONTAP ou Cloud Volumes ONTAP"
- "Configurez un backend avec les pilotes SAN ONTAP ou Cloud Volumes ONTAP"
- "Utilisez Trident avec Amazon FSx for NetApp ONTAP"

## Azure NetApp Files

### Configurer un backend Azure NetApp Files

Utilisez Azure NetApp Files comme backend pour Trident. Ce backend prend en charge les volumes NFS et SMB. Trident prend en charge les identités managées et l'identité de charge de travail pour les clusters Azure Kubernetes Service (AKS).

#### Environnements cloud Azure pris en charge

Trident prend en charge les backends Azure NetApp Files dans plusieurs environnements cloud Azure.

Les clouds Azure pris en charge incluent :

- Azure Commercial
- Azure Government (Azure Government / MAG)

Lorsque vous déployez Trident ou configurez un backend Azure NetApp Files, assurez-vous qu'Azure Resource Manager et les points de terminaison d'authentification correspondent à votre environnement cloud.

#### Vérifiez la prise en charge du pilote Azure NetApp Files

Trident fournit le pilote de stockage Azure NetApp Files suivant.

Les modes d'accès pris en charge incluent *ReadWriteOnce* (RWO), *ReadOnlyMany* (ROX), *ReadWriteMany* (RWX) et *ReadWriteOncePod* (RWOP).

| Pilote | Protocole | volumeMode | 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 à examiner

- Azure NetApp Files ne prend pas en charge les volumes inférieurs à 50 Gio. Trident crée un volume de 50 Gio lorsqu'un volume plus petit est demandé.
- Trident prend uniquement en charge les volumes SMB montés sur des pods exécutés sur des nœuds Windows.
- Les déploiements Azure NetApp Files dans les clouds Azure non commerciaux nécessitent des points de terminaison Azure Resource Manager et d'authentification spécifiques au cloud. Assurez-vous que Trident et toute configuration backend utilisent les points de terminaison appropriés pour votre environnement cloud Azure.

### Utilisez des identités gérées pour AKS

Trident prend en charge "identités gérées" pour les clusters AKS.

Si vous utilisez `tridentctl` pour créer ou gérer des backends Azure NetApp Files, assurez-vous qu'il est configuré pour le bon environnement cloud Azure.

Pour utiliser les identités gérées, vous devez avoir :

- Un cluster Kubernetes déployé à l'aide d'AKS
- Identités gérées configurées sur le cluster Kubernetes AKS
- Trident installé avec `cloudProvider` défini sur `"Azure"`

**Opérateur Trident**

Modifier `tridentorchestrator_cr.yaml` et définir `cloudProvider` sur `"Azure"`.

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

**Helm**

L'exemple suivant installe Trident et configure `cloudProvider` en utilisant la variable d'environnement `$CP` :

```
helm install trident trident-operator-100.2602.0.tgz --create-namespace
--namespace <trident-namespace> --set cloudProvider=$CP
```

**<code>tridentctl</code>**

L'exemple suivant installe Trident et définit le `cloud-provider` indicateur sur `Azure` :

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

### Utiliser l'identité de charge de travail pour AKS

L'identité de charge de travail permet aux pods Kubernetes d'accéder aux ressources Azure en s'authentifiant en tant qu'identité de charge de travail.

Si vous utilisez `tridentctl` pour créer ou gérer des backends Azure NetApp Files, assurez-vous qu'il est configuré pour le bon environnement cloud Azure.

Pour utiliser l'identité de charge de travail, vous devez avoir :

- Un cluster Kubernetes déployé à l'aide d'AKS
- Identité de la charge de travail et oidc-issuer configurés sur le cluster Kubernetes AKS
- Trident installé avec `cloudProvider` défini sur `"Azure"` et `cloudIdentity` défini sur la valeur d'identité de la charge de travail

**Opérateur Trident**

Modifier `tridentorchestrator_cr.yaml` et définir `cloudProvider` sur `"Azure"`. Définir `cloudIdentity` sur `azure.workload.identity/client-id: xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxx`.

```
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-xxxxxxxxxxx' # Edit
```

**Helm**

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-xxxxxxxxxxx'"
```

L'exemple suivant installe Trident et configure `cloudProvider` en utilisant `$CP` et configure `cloudIdentity` en utilisant `$CI` :

```
helm install trident trident-operator-100.6.0.tgz --set cloudProvider=$CP --set cloudIdentity="$CI"
```

**<code>tridentctl</code>**

Définissez les valeurs pour les indicateurs **fournisseur de cloud** et **identité cloud** à l'aide des variables d'environnement suivantes :

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

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

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

## Préparez-vous à configurer un backend Azure NetApp Files

Avant de pouvoir configurer votre backend Azure NetApp Files, vous devez vous assurer que les conditions suivantes sont remplies.

### Environnements cloud Azure pris en charge

Trident prend en charge les backends Azure NetApp Files dans plusieurs environnements cloud Azure.

Les clouds Azure pris en charge incluent :

- Azure Commercial
- Azure Government (Azure Government / MAG)

Lors de la préparation de votre environnement, assurez-vous que votre abonnement Azure, votre configuration d'identité et vos ressources Azure NetApp Files sont créés dans l'environnement cloud Azure approprié.

### 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 nécessaire pour configurer Azure NetApp Files et créer un volume NFS. Consultez "Azure : Configurer Azure NetApp Files et créer un volume NFS".

Pour configurer et utiliser un "Azure NetApp Files" backend, vous avez besoin des éléments suivants :

- `subscriptionID`, `tenantID`, `clientID`, `location` et `clientSecret` sont facultatifs lors de l'utilisation d'identités gérées sur un cluster AKS.
- `tenantID`, `clientID`, et `clientSecret` sont facultatifs lors de l'utilisation d'une identité cloud sur un cluster AKS.
- Les déploiements Azure NetApp Files dans les clouds Azure non commerciaux nécessitent des points de terminaison Azure Resource Manager et d'authentification spécifiques au cloud. Assurez-vous que Trident et toute configuration backend utilisent les points de terminaison appropriés pour votre environnement cloud Azure.

- Un pool de capacité. Consultez "Microsoft : Créer un pool de capacité pour Azure NetApp Files".
- Un sous-réseau délégué à Azure NetApp Files. Consultez "Microsoft : Déléguer un sous-réseau à Azure NetApp Files".
- `subscriptionID` à partir d'un abonnement Azure avec Azure NetApp Files activé.
- `tenantID`, `clientID`, et `clientSecret` à partir d'un "Inscription de l'application" dans Azure Active Directory disposant des autorisations suffisantes pour le service Azure NetApp Files. L'inscription de l'application doit utiliser l'une des méthodes suivantes :
  - Le rôle de propriétaire ou de contributeur "prédéfini par Azure".
  - Un "Rôle Contributor personnalisé" au niveau de l'abonnement (`assignableScopes`) avec les autorisations suivantes, limitées uniquement à ce dont Trident a besoin. 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/MountTarge
ts/read",
          "Microsoft.Network/virtualNetworks/read",
          "Microsoft.Network/virtualNetworks/subnets/read",

"Microsoft.Features/featureProviders/subscriptionFeatureRegistrat
ions/read",

"Microsoft.Features/featureProviders/subscriptionFeatureRegistrat
ions/write",

"Microsoft.Features/featureProviders/subscriptionFeatureRegistrat
ions/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 paramètre `location` est un champ obligatoire au niveau supérieur du fichier de configuration du backend. Les valeurs d'emplacement spécifiées dans les pools virtuels sont ignorées.
- Pour utiliser `Cloud Identity`, obtenez l' `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-xxxxxxxxxxx`.

**Exigences supplémentaires pour les volumes SMB**

Pour créer un volume SMB, vous devez disposer de :

- Active Directory configuré et connecté à Azure NetApp Files. Consultez "Microsoft : Créer et gérer des connexions Active Directory pour Azure NetApp Files".
- Un cluster Kubernetes avec un nœud contrôleur Linux et au moins un nœud de travail Windows exécutant Windows Server 2022. Trident prend uniquement en charge les volumes SMB montés sur des pods exécutés sur des nœuds Windows.
- Au moins un secret Trident contenant vos identifiants Active Directory afin qu'Azure NetApp Files puisse s'authentifier auprès d'Active Directory. Pour générer le 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 à "GitHub : CSI Proxy" ou "GitHub: CSI Proxy pour Windows" pour les nœuds Kubernetes exécutés sous Windows.

## Options et exemples de configuration du backend Azure NetApp Files

Découvrez les options de configuration NFS et SMB pour Azure NetApp Files et consultez des exemples de configuration.

### Options de configuration du backend

Trident utilise votre configuration backend (sous-réseau, réseau virtuel, niveau de service et emplacement) pour créer des volumes Azure NetApp Files sur des pools de capacité disponibles dans l'emplacement demandé et correspondant au niveau de service et au sous-réseau demandés.

Les backends Azure NetApp Files offrent ces options de configuration.

| Paramètre | Description | Défaut |
|---|---|---|
| `version` | Version de la configuration du backend. | Toujours 1 |
| `storageDriverName` | Nom du pilote de stockage | "azure-netapp-files" |
| `backendName` | Nom personnalisé pour le backend de stockage | Driver name + "_" + caractères aléatoires |
| `subscriptionID` | L'ID d'abonnement de votre abonnement Azure Facultatif lorsque les identités gérées sont activées sur un cluster AKS. | |
| `tenantID` | L'identifiant du locataire issu d'un enregistrement d'application. Facultatif lorsque des identités gérées ou une identité cloud sont utilisées sur un cluster AKS. | |
| `clientID` | L'identifiant client issu d'un enregistrement d'application, facultatif lorsque des identités gérées ou une identité cloud sont utilisées sur un cluster AKS. | |
| `clientSecret` | Le secret client issu d'un App Registration est facultatif lorsque des identités gérées ou une cloud identity sont utilisées sur un cluster AKS. | |
| `serviceLevel` | L'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 | Défaut |
| --- | --- | --- |
| `resourceGroups` | Liste des groupes de ressources pour filtrer les ressources découvertes | "[]" (aucun filtre) |
| `netappAccounts` | Liste des comptes NetApp pour le filtrage des ressources découvertes | "[]" (aucun filtre) |
| `capacityPools` | Liste des pools de capacité pour le filtrage des ressources découvertes | "[]" (no filter, 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 fonctionnalités VNet pour un volume, peut être `Basic` ou `Standard`. Les fonctionnalités réseau ne sont pas disponibles dans toutes les régions et peuvent devoir être activées dans un abonnement. Spécifier `networkFeatures` lorsque la fonctionnalité n'est pas activée entraîne l'échec du provisionnement du volume. | "" |
| `nfsMountOptions` | Contrôle précis des options de montage NFS. Ignoré pour les volumes SMB. Pour monter des volumes à l'aide de NFS version 4.1, incluez `nfsvers=4` dans la liste des options de montage séparées par des virgules pour choisir NFS v4.1. Les options de montage définies dans une définition de classe de stockage remplacent celles définies dans la configuration du backend. | "nfsvers=3" |
| `limitVolumeSize` | Échec de l'approvisionnement si la taille du volume demandée est supérieure à cette valeur | "" (non appliqué par défaut) |
| `debugTraceFlags` | Options de débogage à utiliser lors du dépannage. Exemple, `\{"api": false, "method": true, "discovery": true}`. N'utilisez cette option que si vous effectuez un dépannage et avez besoin d'un journal détaillé. | null |

| Paramètre | Description | Défaut |
| --- | --- | --- |
| `nasType` | Configurez la création de volumes NFS ou SMB. Les options sont `nfs`, `smb` ou null. La valeur null correspond par défaut à des volumes NFS. | `nfs` |
| `supportedTopologies` | Représente une liste de régions et de zones prises en charge par ce backend. Pour plus d'informations, consultez "Utiliser la topologie CSI". | |
| `qosType` | Représente le type de QoS : Auto ou Manual. | Automatique |
| `maxThroughput` | Définit le débit maximal autorisé en Mio/s. Pris en charge uniquement pour les pools de capacité QoS manuels. | `4 MiB/sec` |

Pour plus d'informations sur les fonctionnalités réseau, veuillez consulter "Configurer les fonctionnalités réseau pour un volume Azure NetApp Files".

**Envisagez les environnements cloud Azure (26.02)**

À partir de la version 26.02, Trident prend en charge la création et la gestion de backends Azure NetApp Files dans plusieurs environnements cloud Azure.

Les clouds Azure pris en charge incluent :

- Azure Commercial
- Azure Government (Azure Government / MAG)

Lorsque vous déployez Trident ou créez un backend Azure NetApp Files, assurez-vous que Azure Resource Manager et les points de terminaison d'authentification correspondent à votre environnement cloud. Si les points de terminaison ne correspondent pas, `tridentctl` ne peut pas s'authentifier et la création du backend échoue.

**Autorisations et ressources requises**

Si vous recevez l'erreur « Aucun pool de capacité trouvé » lors de la création d'un PVC, il est probable que l'enregistrement de votre application ne dispose pas des autorisations et ressources requises (sous-réseau, réseau virtuel, pool de capacité) associées. Si le débogage est activé, Trident enregistre les ressources Azure détectées lors de la création du backend. Vérifiez qu'un rôle approprié est utilisé.

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

Si le vNet est situé dans un groupe de ressources différent de celui du compte de stockage Azure NetApp Files (ANF), spécifiez le groupe de ressources du réseau virtuel lors de la configuration de la liste resourceGroups pour le backend.

Les `resourceGroups`, `netappAccounts`, et `capacityPools` valeurs sont des filtres qui limitent l'ensemble des ressources découvertes à celles disponibles pour ce système de stockage et peuvent être spécifiées dans n'importe quelle combinaison. Les noms pleinement qualifiés suivent ce format :

| Type | Format |
|---|---|
| Groupe de ressources | <resource group> |
| Compte NetApp | <resource group>/<netapp account> |
| Pool de capacité | <resource group>/<netapp account>/<capacity pool> |
| Réseau virtuel | <resource group>/<virtual network> |
| Sous-réseau | <resource group>/<virtual network>/<subnet> |

#### Provisionnement de volumes

Vous pouvez contrôler le provisionnement des volumes par défaut en spécifiant les options suivantes dans une section spéciale du fichier de configuration. Consultez Exemples de configurations pour plus de détails.

| Paramètre | Description | 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` | Accès au `.snapshot` répertoire | `true`, `false` (Défini explicitement). |
| `size` | La taille par défaut des nouveaux volumes | "100G" |
| `unixPermissions` | Les permissions Unix des nouveaux volumes (4 chiffres octaux). Ignoré pour les volumes SMB. | "" (fonctionnalité en avant-première, nécessite une inscription sur la liste blanche dans l'abonnement) |

### Exemples de configurations

Les exemples suivants présentent des configurations de base qui laissent la plupart des paramètres par défaut. C'est la manière la plus simple de définir un backend.

#### Configuration minimale

Il s'agit de la configuration minimale absolue du backend. Avec cette configuration, Trident découvre tous vos comptes NetApp, pools de capacité et sous-réseaux délégués à Azure NetApp Files dans l'emplacement configuré, et place les nouveaux volumes aléatoirement sur l'un de ces pools et sous-réseaux. Parce que `nasType` est omis, la `nfs` configuration par défaut s'applique et le backend provisionnera des volumes NFS.

Cette configuration est idéale lorsque vous débutez avec Azure NetApp Files et que vous faites des essais, mais en pratique vous souhaiterez définir une portée supplémentaire pour les 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 managées pour AKS

Cette configuration backend omet `subscriptionID`, `tenantID`, `clientID` et `clientSecret`, qui sont optionnels 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:
    - resource-group-1/netapp-account-1/ultra-pool
  resourceGroups:
    - resource-group-1
  netappAccounts:
    - resource-group-1/netapp-account-1
  virtualNetwork: resource-group-1/eastus-prod-vnet
  subnet: resource-group-1/eastus-prod-vnet/eastus-anf-subnet
```

#### Identité cloud pour AKS

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

```
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 spécifique du niveau de service avec filtres de pool de capacité

Cette configuration backend place les volumes dans l'emplacement `eastus` d'Azure dans un `Ultra` pool de capacité. Trident détecte automatiquement tous les sous-réseaux délégués à Azure NetApp Files dans cet emplacement et place un nouveau volume sur l'un d'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
```

**Exemple de backend avec des pools de capacité QoS manuels**

Cette configuration backend place les volumes dans l'emplacement `eastus` Azure avec des pools de capacité QoS manuels.

```
---
version: 1
storageDriverName: azure-netapp-files
backendName: anf1
location: eastus
labels:
  clusterName: test-cluster-1
  cloud: anf
  nasType: nfs
defaults:
  qosType: Manual
storage:
  - serviceLevel: Ultra
    labels:
      performance: gold
    defaults:
      maxThroughput: 10
  - serviceLevel: Premium
    labels:
      performance: silver
    defaults:
      maxThroughput: 5
  - serviceLevel: Standard
    labels:
      performance: bronze
    defaults:
      maxThroughput: 3
```

**Configuration avancée**

Cette configuration backend réduit encore la portée du placement des volumes à un seul sous-réseau et modifie également certains paramètres par défaut de 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: application-group-1/eastus-prod-vnet
subnet: application-group-1/eastus-prod-vnet/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 du pool virtuel**

Cette configuration backend définit plusieurs pools de stockage dans un seul fichier. Ceci est utile lorsque vous disposez de plusieurs pools de capacité prenant en charge différents niveaux de service et que vous souhaitez créer des classes de stockage dans Kubernetes pour les représenter. 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:
      - application-group-1/netapp-account-1/ultra-1
      - application-group-1/netapp-account-1/ultra-2
    networkFeatures: Standard
  - labels:
      performance: silver
    serviceLevel: Premium
    capacityPools:
      - application-group-1/netapp-account-1/premium-1
  - labels:
      performance: bronze
    serviceLevel: Standard
    capacityPools:
      - application-group-1/netapp-account-1/standard-1
      - application-group-1/netapp-account-1/standard-2
```

#### Configuration des topologies prises en charge

Trident facilite le provisionnement des volumes pour les charges de travail en fonction des régions et des zones de disponibilité. Le `supportedTopologies` bloc dans cette configuration backend est utilisé pour fournir une liste de régions et de zones par backend. Les valeurs de région et de zone spécifiées ici doivent correspondre aux valeurs de région et de zone des étiquettes sur 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 backend, Trident crée les volumes dans la région et la zone mentionnées. Pour plus d'informations, consultez "Utiliser la topologie CSI".

```
---
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 `StorageClass` définitions suivantes se rapportent aux pools de stockage ci-dessus.

#### Exemple de définitions utilisant `parameter.selector` field

En utilisant `parameter.selector`, vous pouvez spécifier pour chaque `StorageClass` le pool virtuel utilisé pour héberger un volume. Le volume aura les aspects définis dans le pool choisi.

```
---
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

En utilisant `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 prenant en charge les volumes SMB. `nasType: nfs` ou `nasType: null`filtres pour les pools NFS.

**Créer le backend**

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

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

Si vous utilisez un cloud Azure non commercial, assurez-vous que `tridentctl` est configuré pour utiliser Azure Resource Manager et les points de terminaison d'authentification de votre environnement cloud. Si la création du backend échoue, vérifiez la configuration de votre backend et consultez les journaux pour en déterminer la cause :

```
tridentctl logs
```

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

# Google Cloud NetApp Volumes

## Configurer Google Cloud NetApp Volumes

Vous pouvez configurer Google Cloud NetApp Volumes comme backend pour que Trident puisse provisionner du stockage pour les charges de travail Kubernetes.

**Aperçu**

Trident prend en charge Google Cloud NetApp Volumes pour les charges de travail NAS (NFS et SMB) et bloc (iSCSI).

- Les charges de travail NAS utilisent le `google-cloud-netapp-volumes` backend
- Les charges de travail par blocs (iSCSI) utilisent le `google-cloud-netapp-volumes-san` backend

Les volumes NAS offrent un stockage de fichiers et sont accessibles via les protocoles NFS ou SMB. Ces volumes prennent en charge l'accès partagé entre plusieurs pods ou nœuds.

Les volumes de blocs offrent un stockage bloc et sont accessibles comme des périphériques iSCSI connectés aux nœuds Kubernetes. Ces volumes sont utilisés lorsque les applications nécessitent un accès au niveau bloc.

Ceci s'applique aux environnements suivants :

- Trident 26.02 et versions ultérieures
- Google Kubernetes Engine (GKE) ou Red Hat OpenShift
- Pools de stockage Google Cloud NetApp Volumes

Pour configurer le stockage bloc (iSCSI), voir "Configurer le stockage bloc (iSCSI)".

**Préparez-vous à configurer**

L'identité cloud permet aux charges de travail Kubernetes d'accéder aux ressources Google Cloud en s'authentifiant en tant qu'identité de charge de travail au lieu d'utiliser des informations d'identification statiques.

Pour utiliser l'identité cloud avec Google Cloud NetApp Volumes, vous devez disposer des éléments suivants :

- Un cluster Kubernetes déployé à l'aide de Google Kubernetes Engine (GKE)
- L'identité de la charge de travail est activée sur le cluster GKE et le serveur de métadonnées est activé sur les pools de nœuds
- Un compte de service Google Cloud avec le rôle d'administrateur Google Cloud NetApp Volumes (`roles/netapp.admin`) ou un rôle personnalisé équivalent
- Trident installé avec le fournisseur de cloud défini sur `GCP` et l'annotation d'identité cloud configurée

**Opérateur Trident**

Pour installer Trident à l'aide de l'opérateur Trident, modifiez `tridentorchestrator_cr.yaml` :

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

**Helm**

Configurez le fournisseur de cloud et l'identité cloud lors de l'installation de Trident avec Helm :

```
helm install trident trident-operator-100.6.0.tgz \
  --set cloudProvider=GCP \
  --set cloudIdentity="iam.gke.io/gcp-service-account: cloudvolumes-
admin-sa@mygcpproject.iam.gserviceaccount.com"
```

**tridentctl**

Installez Trident en spécifiant le fournisseur de cloud et l'identité cloud :

```
tridentctl install \
  --cloud-provider=GCP \
  --cloud-identity="iam.gke.io/gcp-service-account: cloudvolumes-admin-
sa@mygcpproject.iam.gserviceaccount.com" \
  -n trident
```

## Configurer le stockage NAS

Pour les pools de stockage UNIFIED de Google Cloud NetApp Volumes, Trident applique des règles de nommage et de validation spécifiques à UNIFIED lors des opérations sur les volumes.

Lors de la localisation d'un volume, Trident peut évaluer plusieurs variantes de nom de volume compatibles (par exemple, les formats avec tiret et trait de soulignement) afin d'améliorer la fiabilité de l'importation et de la découverte.

#### Détails du pilote

Trident fournit le `google-cloud-netapp-volumes` driver pour provisionner le stockage NAS à partir de Google Cloud NetApp Volumes.

Le pilote prend en charge les modes d'accès suivants :

- ReadWriteOnce (RWO)
- ReadOnlyMany (ROX)
- ReadWriteMany (RWX)
- ReadWriteOncePod (RWOP)

| Pilote | Protocole | volumeMode | Modes d'accès pris en charge | Systèmes de fichiers pris en charge |
|---|---|---|---|---|
| `google-cloud-netapp-volumes` | NFS SMB | Système de fichiers | RWO, ROX, RWX, RWOP | `nfs`, `smb` |

#### Configurer un backend NAS Trident

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: gcnv-nas
  namespace: trident
spec:
  version: 1
  storageDriverName: google-cloud-netapp-volumes
  projectNumber: "<project-number>"
  location: "<region>"
  sdkTimeout: "600"
  storage:
  - labels:
      cloud: gcp
    network: "<vpc-network>"
```

#### Provisionner des volumes NAS

Les volumes NAS sont provisionnés à l'aide du `google-cloud-netapp-volumes` backend et prennent en charge les protocoles NFS et SMB.

### StorageClass for les volumes NFS

Pour provisionner des volumes NFS, définissez `nasType` sur `nfs`.

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

#### StorageClass for les volumes SMB

Pour provisionner des volumes SMB, configurez `nasType` sur `smb` et fournissez les informations d'identification.

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: gcnv-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"
allowVolumeExpansion: true
```

#### Exemple de PersistentVolumeClaim (RWX)

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

**Exemple de PersistentVolumeClaim (RWO)**

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: gcnv-nas-rwo
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 100Gi
  storageClassName: gcnv-nfs
```

Les volumes NAS utilisent `volumeMode: Filesystem`.

## Configurer Google Cloud NetApp Volumes pour les charges de travail SAN

Vous pouvez configurer Trident pour provisionner des volumes de stockage bloc à l'aide du protocole iSCSI depuis Google Cloud NetApp Volumes. Les volumes SAN sont provisionnés à partir de pools de stockage Flex Unified en utilisant le `google-cloud-netapp-volumes-san` storage driver.

Ce pilote est dédié aux charges de travail bloc et ne prend pas en charge les protocoles NAS.

Le `google-cloud-netapp-volumes-san` backend est requis pour provisionner des volumes de blocs iSCSI. Le `google-cloud-netapp-volumes` backend prend uniquement en charge les protocoles NAS et ne peut pas être utilisé pour les charges de travail SAN.

**Aperçu**

Trident prend en charge les charges de travail Google Cloud NetApp Volumes SAN (iSCSI) à l'aide du `google-cloud-netapp-volumes-san` driver.

Les volumes SAN sont provisionnés à partir de pools de stockage Flex Unified et présentés aux nœuds Kubernetes en tant que périphériques de blocs iSCSI.

Ceci s'applique aux environnements suivants :

- Trident 26.02 et versions ultérieures
- Google Kubernetes Engine (GKE) ou Red Hat OpenShift
- Google Cloud NetApp Volumes Flex Pools de stockage unifiés
- Charges de travail basées sur iSCSI

### Pools de stockage Flex Unified

Les pools de stockage Flex Unified fournissent du stockage bloc utilisant le protocole iSCSI et sont nécessaires au provisionnement SAN :

- Les pools régionaux Flex Unified sont pris en charge.
- Les pools ZONAL Flex Unified sont pris en charge à partir de Trident 26.02.1.
- Seul le niveau de service **Flex** est pris en charge pour les charges de travail SAN.

### Configurer un backend SAN Trident

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: gcnv-san
  namespace: trident
spec:
  version: 1
  storageDriverName: google-cloud-netapp-volumes-san
  projectNumber: "<project-number>"
  location: "<region>"
  sdkTimeout: "600"
  storage:
  - labels:
      cloud: gcp
      performance: flex
    network: "<vpc-network>"
    serviceLevel: Flex
```

### Créer un StorageClass

Après avoir configuré le backend SAN, créez un StorageClass qui référence le `google-cloud-netapp-volumes-san` driver.

Le type de système de fichiers est défini dans le StorageClass, pas dans le backend.

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: gcnv-san
provisioner: csi.trident.netapp.io
parameters:
  backendType: "google-cloud-netapp-volumes-san"
  fsType: "ext4"
allowVolumeExpansion: true
```

Types de systèmes de fichiers pris en charge :

- `ext4` (défaut)
- `ext3`
- `xfs`

> Le pilote SAN ne prend en charge que le niveau de service Flex et n'utilise pas les paramètres backend spécifiques au NAS tels que `exportRule`, `unixPermissions`, `nasType`, `snapshotDir`, `nfsMountOptions` ou les paramètres liés à la hiérarchisation.

### Approvisionner des volumes de blocs

#### ReadWriteOnce (RWO)

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: gcnv-san-rwo
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 100Gi
  storageClassName: gcnv-san
```

#### ReadWriteOncePod (RWOP)

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: gcnv-san-rwop
spec:
  accessModes:
    - ReadWriteOncePod
  resources:
    requests:
      storage: 100Gi
  storageClassName: gcnv-san
```

#### ReadOnlyMany (ROX)

Une méthode courante pour ROX consiste à cloner un volume ReadWriteOnce existant et à monter le clone en lecture seule.

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: gcnv-san-rox
spec:
  accessModes:
    - ReadOnlyMany
  resources:
    requests:
      storage: 100Gi
  storageClassName: gcnv-san
  dataSource:
    kind: PersistentVolumeClaim
    name: gcnv-san-rwo
```

#### ReadWriteMany (RWX) — bloc brut uniquement

ReadWriteMany n'est pris en charge que lorsque `volumeMode: Block`.

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: gcnv-san-raw-rwx
spec:
  accessModes:
    - ReadWriteMany
  volumeMode: Block
  resources:
    requests:
      storage: 100Gi
  storageClassName: gcnv-san
```

### Comportement du volume de bloc

Les volumes de blocs sont provisionnés en tant que LUN iSCSI et présentés aux nœuds Kubernetes en tant que périphériques de blocs.

Volumes de blocs :

- Utilisez le protocole iSCSI
- Prise en charge du système de fichiers et de la présentation des blocs bruts
- Sont rattachés et gérés par Trident
- Prise en charge de plusieurs modes d'accès Kubernetes

#### Modes d’accès

Les volumes de blocs provisionnés par Trident prennent en charge les modes d’accès suivants :

- `ReadWriteOnce` (RWO)
- `ReadOnlyMany` (ROX)
- `ReadWriteOncePod` (RWOP)
- `ReadWriteMany` (RWX), pris en charge uniquement lorsque `volumeMode: Block`

#### Comportement de volumeMode

Le `volumeMode`champ contrôle la façon dont un volume de bloc est exposé :

- `Filesystem` Trident formate et monte le volume.
- `Block` Trident attache le périphérique et l’expose comme un périphérique de bloc brut.

#### Opérations prises en charge

Volumes de blocs provisionnés à l’aide du pilote `google-cloud-netapp-volumes-san` prennent en charge :

- Créer
- Supprimer
- Cloner
- Instantané
- Redimensionner
- Importer

#### Comportement de surprovisionnement supplémentaire de GiB

Google Cloud NetApp Volumes block volumes incluent des métadonnées internes. Cette surcharge réduit la taille du périphérique visible par le noyau par rapport à la capacité provisionnée.

Les tests montrent :

- Environ 300 Kio de surcharge lors de la création initiale
- Jusqu’à environ 107 MiB de surcharge après un redimensionnement

Étant donné que Google Cloud NetApp Volumes n’accepte que des allocations de Gio entiers, Trident garantit que la taille utilisable du périphérique respecte ou dépasse toujours la demande PVC en :

- Arrondi de la taille demandée au Gio entier supérieur
- Ajout d’un tampon supplémentaire de 1 Gio

Exemple :

- Requête PVC : 100 Gio
- Taille provisionnée dans Google Cloud NetApp Volumes : 101 GiB

- Espace utilisable visible par l'application : au moins 100 GiB

### Exemples de pods

#### Volume de bloc monté sur le système de fichiers (RWO)

```
apiVersion: v1
kind: Pod
metadata:
  name: app-rwo
spec:
  containers:
  - name: app
    image: ubuntu:22.04
    command: ["sleep", "infinity"]
    volumeMounts:
    - name: data
      mountPath: /mnt/data
  volumes:
  - name: data
    persistentVolumeClaim:
      claimName: gcnv-san-rwo
```

#### Périphérique de bloc brut (RWX)

```
apiVersion: v1
kind: Pod
metadata:
  name: app-raw-rwx
spec:
  containers:
  - name: app
    image: ubuntu:22.04
    command: ["sleep", "infinity"]
    volumeDevices:
    - name: data
      devicePath: /dev/xda
  volumes:
  - name: data
    persistentVolumeClaim:
      claimName: gcnv-san-raw-rwx
```

### Comportement d'attachement et de montage

Pour les volumes SAN provisionnés à partir de Google Cloud NetApp Volumes :

- Trident crée un Logical Unit Number (LUN) dans un pool de stockage Flex Unified.
- Lors de la publication, Trident associe le LUN à un groupe d'hôtes par nœud.
- Lors de la mise en scène des nœuds, Trident :
  - Se connecte à la cible iSCSI
  - Découvre le LUN
  - Configure le multipath
- Si `volumeMode: Filesystem`Trident le nécessite, Trident formate le périphérique et le monte.
- Si `volumeMode: Block` Trident attache le périphérique et l'expose directement au pod sans le formater ni le monter.

> Les volumes de blocs SAN ne prennent pas en charge le verrouillage distribué ni la coordination des écritures. Lorsqu'un volume de blocs est accédé par plusieurs nœuds (ReadWriteMany avec `volumeMode: Block`), l'application ou le système de fichiers doit gérer la concurrence.

## Préparez-vous à configurer un backend Google Cloud NetApp Volumes

Avant de pouvoir configurer votre backend Google Cloud NetApp Volumes, vous devez vous assurer que les conditions suivantes sont remplies.

### Prérequis pour les volumes NFS ou SMB

Si vous utilisez Google Cloud NetApp Volumes pour la première fois ou dans un nouvel emplacement, une configuration initiale est nécessaire pour configurer Google Cloud NetApp Volumes et créer un volume NFS ou SMB. Consultez "Avant de commencer".

Assurez-vous de disposer des éléments suivants avant de configurer Google Cloud NetApp Volumes backend :

- Un compte Google Cloud configuré avec le service Google Cloud NetApp Volumes. Consultez "Google Cloud NetApp Volumes".
- Numéro de projet de votre compte Google Cloud. Consultez "Identification des projets".
- Un compte de service Google Cloud avec le rôle NetApp Volumes Admin (`roles/netapp.admin`. Consultez "Rôles et autorisations d'Identity and Access Management".
- Fichier de clé API pour votre compte GCNV. Consultez "Créer une clé de compte de service"
- Un pool de stockage. Consultez "Aperçu des pools de stockage".

Pour plus d'informations sur la configuration de l'accès à Google Cloud NetApp Volumes, consultez "Configurer l'accès à Google Cloud NetApp Volumes".

## Options et exemples de configuration du backend Google Cloud NetApp Volumes

Découvrez les options de configuration backend pour Google Cloud NetApp Volumes et consultez des exemples de configuration.

### Options de configuration du backend

Chaque backend provisionne des volumes dans une seule région Google Cloud. Pour créer des volumes dans

d’autres régions, vous pouvez définir des backends supplémentaires.

| Paramètre | Description | Défaut |
|---|---|---|
| `version` | | Toujours 1 |
| `storageDriverName` | Nom du pilote de stockage | La valeur de `storageDriverName` doit être spécifiée comme « google-cloud-netapp-volumes ». |
| `backendName` | (Facultatif) Nom personnalisé du stockage backend | Nom du driver + "_" + partie de la clé API |
| `storagePools` | Paramètre optionnel utilisé pour spécifier les pools de stockage pour la création de volumes. | |
| `projectNumber` | Numéro de projet du compte Google Cloud. La valeur se trouve sur la page d’accueil du portail Google Cloud. | |
| `location` | L’emplacement Google Cloud où Trident crée les volumes GCNV. Lors de la création de clusters Kubernetes interrégionaux, les volumes créés dans un `location` peuvent être utilisés dans des charges de travail planifiées sur des nœuds répartis dans plusieurs régions Google Cloud. Le trafic interrégional engendre des coûts supplémentaires. | |
| `apiKey` | Clé API pour le compte de service Google Cloud avec le `netapp.admin` rôle. Elle inclut le contenu au format JSON du fichier de clé privée d’un compte de service Google Cloud (copié tel quel dans le fichier de configuration du backend). 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` | L’approvisionnement échoue si la taille du volume demandée est supérieure à cette valeur. | "" (non appliqué par défaut) |
| `serviceLevel` | Le niveau de service d’un pool de stockage et de ses volumes. Les valeurs sont `flex`, `standard`, `premium` ou `extreme`. | |
| `labels` | Ensemble d’étiquettes arbitraires au format JSON à appliquer aux volumes | "" |
| `network` | Réseau Google Cloud utilisé pour les volumes GCNV. | |
| `debugTraceFlags` | Options de débogage à utiliser lors du dépannage. Exemple, `{"api":false, "method":true}`. N’utilisez cette option que si vous effectuez un dépannage et avez besoin d’un journal détaillé. | null |

| Paramètre | Description | Défaut |
| --- | --- | --- |
| `nasType` | Configurez la création de volumes NFS ou SMB. Les options sont `nfs`, `smb` ou null. La valeur null correspond par défaut à des volumes NFS. | `nfs` |
| `supportedTopologies` | Représente une liste de régions et de zones prises en charge par ce backend. Pour plus d'informations, consultez "Utiliser la topologie CSI". Par exemple : `supportedTopologies: - topology.kubernetes.io/region: asia-east1 topology.kubernetes.io/zone: asia-east1-a` | |

### Options de provisionnement de volume

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

| Paramètre | Description | Défaut |
| --- | --- | --- |
| `exportRule` | Les 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 `.snapshot` répertoire | `true`, `false` (Le comportement par défaut peut varier. À définir explicitement) « false » pour NFSv3 |
| `snapshotReserve` | Pourcentage du volume réservé aux instantanés | "" (accepter la valeur par défaut de 0) |
| `unixPermissions` | Les permissions Unix des nouveaux volumes (4 chiffres octaux). | "" |

### Exemples de configurations

Les exemples suivants présentent des configurations de base qui laissent la plupart des paramètres par défaut. C'est la manière la plus simple de définir un backend.

**Configuration minimale**

Il s'agit de la configuration minimale absolue du backend. Avec cette configuration, Trident découvre tous vos pools de stockage délégués à Google Cloud NetApp Volumes à l'emplacement configuré, et place les nouveaux volumes sur l'un de ces pools de façon aléatoire. Parce que `nasType` est omis, la `nfs` configuration par défaut s'applique et le backend provisionnera des volumes NFS.

Cette configuration est idéale lorsque vous débutez avec Google Cloud NetApp Volumes et que vous faites des essais, mais en pratique, vous pourriez avoir besoin de définir une portée supplémentaire pour les volumes que vous provisionnez.

Remplacez `<id_value>` et `<key_value>` par vos identifiants de compte de service.

```
---
apiVersion: v1
kind: Secret
metadata:
  name: backend-tbc-gcnv-secret
type: Opaque
stringData:
  private_key_id: "<id_value>"
  private_key: |
    -----BEGIN PRIVATE KEY-----
    <key_value>
    -----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
  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 le filtre StoragePools**

```
---
apiVersion: v1
kind: Secret
metadata:
  name: backend-tbc-gcnv-secret
type: Opaque
stringData:
  private_key_id: "<id_value>"
  private_key: |
    -----BEGIN PRIVATE KEY-----
    <key_value>
    -----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 du pool virtuel**

Cette configuration backend définit plusieurs pools virtuels dans un seul fichier. Les pools virtuels sont définis dans la `storage` section. Ils sont utiles lorsque vous disposez de plusieurs pools de stockage prenant en charge différents niveaux de service et que vous souhaitez créer des classes de stockage dans Kubernetes pour les représenter. Les étiquettes des pools virtuels sont utilisées pour différencier les pools. Par exemple, dans l'exemple ci-dessous, `performance` label et `serviceLevel` 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 pour 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, consultez "Pools virtuels".

```
---
apiVersion: v1
kind: Secret
metadata:
  name: backend-tbc-gcnv-secret
type: Opaque
stringData:
  private_key_id: "<id_value>"
  private_key: |
    -----BEGIN PRIVATE KEY-----
    <key_value>
    -----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 charges de travail en fonction des régions et des zones de disponibilité. Le `supportedTopologies` bloc dans cette configuration backend est utilisé pour fournir une liste de régions et de zones par backend. Les valeurs de région et de zone spécifiées ici doivent correspondre aux valeurs de région et de zone des étiquettes sur 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 backend, Trident crée les volumes dans la région et la zone mentionnées. Pour plus d'informations, consultez "Utiliser la topologie CSI".

```
---
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 ensuite ?

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

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

Pour vérifier que le backend a bien été 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 backend échoue, cela signifie qu'il y a un problème avec la configuration du backend. Vous pouvez décrire le backend à l'aide de la commande `kubectl get tridentbackendconfig <backend-name>` ou consulter les journaux pour déterminer la cause en exécutant la commande suivante :

```
tridentctl logs
```

Après avoir identifié et corrigé le problème lié au fichier de configuration, vous pouvez supprimer le backend et exécuter à nouveau la commande de création.

#### Définitions des classes de stockage

L'exemple suivant est une `StorageClass`définition de base qui fait référence au backend 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 champ `parameter.selector` :

En utilisant `parameter.selector`, vous pouvez spécifier pour chaque `StorageClass` le "pool virtuel" utilisé pour héberger un volume. Le volume aura les aspects définis dans le pool choisi.

```
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 à "Créer une classe de stockage".

#### Exemples de définitions pour les volumes SMB

En utilisant `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. Tout utilisateur/mot de passe Active Directory, avec ou sans autorisations, peut être utilisé pour le secret de l'étape du nœud.

**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 prenant en charge les volumes SMB. `nasType: nfs` ou `nasType: null`filtres pour les pools NFS.

**Exemple de définition du PVC**

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

Pour vérifier si le PVC est lié, 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 la hiérarchisation automatique pour Google Cloud NetApp Volumes

La hiérarchisation automatique est configurée via les paramètres du backend Trident et les annotations PersistentVolumeClaim lors du provisionnement des volumes. Vous pouvez configurer la hiérarchisation automatique pour Google Cloud NetApp Volumes à l'aide de Trident.

### Aperçu

La hiérarchisation automatique permet à Trident de provisionner des volumes qui déplacent automatiquement les données inactives d'un niveau de performance vers un niveau de capacité. Cela réduit le coût de stockage tout en préservant les performances pour les données fréquemment consultées.

Trident applique les paramètres de hiérarchisation automatique uniquement lors de la création du volume. Les modifications après le provisionnement ne sont pas prises en charge dans Trident 26.02.

### Concepts

#### hiérarchisation automatique

Le système de hiérarchisation automatique déplace les données rarement consultées d'un niveau de performance vers un niveau de capacité en fonction des modèles d'accès. Le déplacement des données s'effectue de manière asynchrone et n'est pas immédiat.

#### politique de hiérarchisation

La politique de hiérarchisation détermine si la hiérarchisation automatique est activée pour un volume.

Les politiques suivantes sont prises en charge : * `auto` : Active la hiérarchisation automatique en fonction des modèles d'accès * `none` : Désactive la hiérarchisation automatique

#### Jours de refroidissement

Les jours de refroidissement spécifient le nombre minimal de jours pendant lesquels un bloc de données doit rester inactif avant de pouvoir être classé par niveau. Les jours de refroidissement s'appliquent uniquement lorsque la politique de classement par niveau est définie sur `auto`.

### Modèle de configuration

#### Étendues de configuration

La hiérarchisation automatique peut être configurée à plusieurs niveaux :

- **Portée du pool de stockage** S'applique à tous les volumes provisionnés à partir du pool.
- **Portée du volume** S'applique à un seul volume par le biais des annotations PersistentVolumeClaim.

Trident détermine la configuration effective en fonction de l'endroit où chaque paramètre est défini.

#### Précédence de configuration

Lorsque le même paramètre est défini à plusieurs niveaux, Trident applique l'ordre de priorité suivant :

1. Annotations PersistentVolumeClaim
2. Configuration du backend Trident
3. Paramètres par défaut du storage pool

Les paramètres définis à un niveau de priorité supérieur remplacent les valeurs de niveau inférieur.

### fonctionnalité prise en charge dans Trident 26.02

Trident 26.02 prend en charge les fonctionnalités de hiérarchisation automatique suivantes pour Google Cloud NetApp Volumes :

- Activation ou désactivation du auto-tiering lors de l'allocation des volumes
- Définition d'une politique de hiérarchisation dans la configuration du backend Trident
- Remplacement de la politique de hiérarchisation et du nombre de jours de refroidissement par volume à l'aide d'annotations PVC
- Configuration des jours de refroidissement pour les volumes avec auto-tiering activé

### fonctionnalité non prise en charge dans Trident 26.02

Les opérations suivantes ne sont pas prises en charge :

- Modification des paramètres de hiérarchisation automatique après la création du volume
- Modification des politiques de hiérarchisation sur les volumes existants à l'aide des mises à jour Kubernetes
- Application des paramètres de hiérarchisation automatique en dehors des workflows de provisionnement gérés par Trident

### Paramètres de configuration du backend

Les paramètres suivants contrôlent le comportement de hiérarchisation automatique lorsqu'ils sont définis dans la configuration du backend Trident :

| Paramètre | Obligatoire | Description |
|---|---|---|
| tieringPolicy | Non | Politique de hiérarchisation des volumes (`auto` ou `none`) |
| tieringMinimumCoolingDays | Non | Nombre de jours d'inactivité avant le classement des données (plage : 2–183, valeur par défaut : 31) |

### Remplacements au niveau du volume à l'aide des annotations PersistentVolumeClaim

#### Annotations prises en charge

Les annotations PersistentVolumeClaim permettent de remplacer, volume par volume, les paramètres de hiérarchisation automatique.

| Annotation | Description |
|---|---|
| `trident.netapp.io/tieringPolicy` | Remplace la politique de hiérarchisation pour le volume |
| `trident.netapp.io/tieringMinimumCoolingDays` | Remplace la valeur des jours de refroidissement pour le volume |

#### Exemple : PersistentVolumeClaim avec des substitutions de hiérarchisation automatique

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: auto-tiering-pvc
  annotations:
    trident.netapp.io/tieringPolicy: auto
    trident.netapp.io/tieringMinimumCoolingDays: "45"
spec:
  accessModes:
    - ReadWriteOnce
  storageClassName: google-cloud-netapp-volumes-auto-tiering
  resources:
    requests:
      storage: 500Gi
```

#### Comportement et limitations

##### Comportement de provisionnement

- Les paramètres de hiérarchisation automatique sont évalués et appliqués uniquement lors de la création du volume.
- Trident ne réconcilie pas la configuration de hiérarchisation après le provisionnement.
- Les jours de refroidissement sont ignorés lorsque la politique de hiérarchisation est définie sur `none`.

##### Limitations de la plateforme

- La hiérarchisation automatique est prise en charge uniquement pour les volumes NAS (NFS et SMB).
- Les volumes de blocs (iSCSI) ne prennent pas en charge l'auto-tiering.
- Le pool de stockage Google Cloud NetApp Volumes doit avoir la hiérarchisation automatique activée dans Google Cloud.

##### Valeurs prises en charge

- Plage de valeurs valide `tieringMinimumCoolingDays`: 2 à 183
- Valeur par défaut : 31

# Configurer un backend NetApp HCI ou SolidFire

Apprenez à créer et à utiliser un backend Element avec votre installation Trident.

## Détails du pilote Element

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

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

| Pilote | Protocole | VolumeMode | 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 backend Element.

- Un système de stockage pris en charge qui exécute le logiciel Element.
- Identifiants d'un administrateur de cluster NetApp HCI/SolidFire ou d'un utilisateur locataire pouvant gérer les volumes.
- Tous vos nœuds de travail Kubernetes doivent disposer des outils iSCSI appropriés. Consultez "Informations de préparation du nœud de travail".

## Options de configuration du backend

Consultez le tableau suivant pour les options de configuration du backend :

| Paramètre | Description | Défaut |
|---|---|---|
| `version` | | Toujours 1 |
| `storageDriverName` | Nom du pilote de stockage | Toujours "solidfire-san" |
| `backendName` | Nom personnalisé ou le stockage backend | "solidfire_" + adresse IP de stockage (iSCSI) |
| `Endpoint` | MVIP pour le cluster SolidFire avec les identifiants du locataire | |
| `SVIP` | Adresse IP et port de stockage (iSCSI) | |
| `labels` | Ensemble d'étiquettes au format JSON arbitraires à appliquer aux volumes. | "" |
| `TenantName` | Nom du tenant à utiliser (créé s'il n'est pas trouvé) | |
| `InitiatorIFace` | Limiter le trafic iSCSI à une interface d'hôte spécifique | "default" |

| Paramètre | Description | Défaut |
| --- | --- | --- |
| `UseCHAP` | Utilisez CHAP pour authentifier iSCSI. Trident utilise CHAP. | true |
| `AccessGroups` | Liste des ID de groupes d'accès à utiliser | Recherche l'ID d'un groupe d'accès nommé "trident" |
| `Types` | Spécifications QoS | |
| `limitVolumeSize` | Échec de l'approvisionnement si la taille du volume demandée est supérieure à cette valeur | "" (non appliqué par défaut) |
| `debugTraceFlags` | Options de débogage à utiliser lors du dépannage. Exemple, {"api":false, "method":true} | null |

| AVERTISSEMENT | N'utilisez pas `debugTraceFlags` sauf si vous effectuez un dépannage et avez besoin d'un vidage détaillé du journal. |
| --- | --- |

## Exemple 1 : Configuration du backend pour `solidfire-san` driver avec trois types de volumes

Cet exemple illustre un fichier backend utilisant l'authentification CHAP et modélisant trois types de volumes avec des garanties QoS spécifiques. Vous définirez ensuite probablement des classes de stockage pour utiliser chacune d'elles à l'aide du `IOPS` storage class parameter.

```
---
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 backend et de la classe de stockage pour `solidfire-san` driver avec pools virtuels

Cet exemple montre le fichier de définition du backend configuré avec des pools virtuels ainsi que StorageClasses qui y font référence.

Trident copie les étiquettes présentes sur un pool de stockage vers le LUN de stockage principal lors du provisionnement. Pour plus de simplicité, les administrateurs de stockage peuvent définir des étiquettes par pool virtuel et regrouper les volumes par étiquette.

Dans l'exemple de fichier de définition de backend ci-dessous, des valeurs par défaut spécifiques sont définies pour tous les pools de stockage, qui définissent le `type` à Silver. Les pools virtuels sont définis dans la section `storage`. Dans cet exemple, certains pools de stockage définissent leur propre type, et certains pools 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 StorageClass suivantes se rapportent aux pools virtuels mentionnés ci-dessus. En utilisant le

champ `parameters.selector`, chaque StorageClass indique le ou les pools virtuels pouvant héberger un volume. Le volume possédera les aspects définis dans le pool virtuel choisi.

Le premier StorageClass (`solidfire-gold-four`) correspond au premier pool virtuel. Il s'agit du seul pool offrant des performances gold avec un `Volume Type QoS` de Gold. Le dernier StorageClass (`solidfire-silver`) fait référence à tout pool de stockage offrant des performances silver. Trident déterminera quel pool virtuel est sélectionné et s'assurera que l'exigence de stockage est respectée.

```
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
```

### Pour plus d'informations

- "Groupes d'accès au volume"

# Pilotes ONTAP SAN

## Présentation du pilote ONTAP SAN

Découvrez comment configurer un backend ONTAP avec les pilotes SAN ONTAP et Cloud Volumes ONTAP.

#### Détails du pilote ONTAP SAN

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

| Pilote | Protocole | volumeMode | Modes d'accès pris en charge | Systèmes de fichiers pris en charge |
|---|---|---|---|---|
| `ontap-san` | iSCSI SCSI sur FC | Bloc | RWO, ROX, RWX, RWOP | Aucun système de fichiers ; périphérique bloc brut |
| `ontap-san` | iSCSI SCSI sur FC | Système de fichiers | RWO, RWOP<br><br>ROX et RWX ne sont pas disponibles en mode volume du système de fichiers. | `xfs`, `ext3`, `ext4` |
| `ontap-san` | NVMe/TCP<br><br>Consultez Considérations supplémentaires pour NVMe/TCP. | Bloc | RWO, ROX, RWX, RWOP | Aucun système de fichiers ; périphérique bloc brut |

| Pilote | Protocole | volumeMode | Modes d'accès pris en charge | Systèmes de fichiers pris en charge |
|---|---|---|---|---|
| `ontap-san` | NVMe/TCP<br><br>Consultez Considérations supplémentaires pour NVMe/TCP. | Système de fichiers | RWO, RWOP<br><br>ROX et RWX ne sont pas disponibles en mode volume du système de fichiers. | `xfs`, `ext3`, `ext4` |
| `ontap-san-economy` | iSCSI | Bloc | RWO, ROX, RWX, RWOP | Aucun système de fichiers ; périphérique bloc brut |
| `ontap-san-economy` | iSCSI | Système de fichiers | RWO, RWOP<br><br>ROX et RWX ne sont pas disponibles en mode volume du système de fichiers. | `xfs`, `ext3`, `ext4` |

**AVERTISSEMENT**

- Utilisez `ontap-san-economy` uniquement si le nombre d'utilisations du volume persistant doit être supérieur à "limites de volume ONTAP prises en charge".
- Utilisez `ontap-nas-economy` uniquement si le nombre d'utilisations de volumes persistants est censé être supérieur à "limites de volume ONTAP prises en charge" et que le pilote `ontap-san-economy` ne peut pas être utilisé.
- N'utilisez pas `ontap-nas-economy` si vous prévoyez un besoin de protection des données, de reprise après sinistre ou de mobilité.
- NetApp ne recommande pas d'utiliser la croissance automatique FlexVol® dans tous les pilotes ONTAP, sauf ontap-san. En guise de solution de contournement, Trident prend en charge l'utilisation de la réserve de snapshots et adapte les volumes FlexVol® en conséquence.

### Autorisations de l'utilisateur

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

| | |
|---|---|
| **REMARQUE** | Si vous utilisez le `limitAggregateUsage` paramètre, des autorisations d'administrateur de cluster sont requises. Lors de l'utilisation d'Amazon FSx for NetApp ONTAP avec Trident, le `limitAggregateUsage` paramètre ne fonctionnera pas avec les comptes d'utilisateur `vsadmin` et `fsxadmin`. L'opération de configuration échouera si vous spécifiez ce paramètre. |

Bien qu'il soit possible de créer un rôle plus restrictif au sein d'ONTAP qu'un pilote Trident peut utiliser, nous ne le recommandons pas. La plupart des nouvelles versions de Trident feront appel à des API supplémentaires dont il faudrait tenir compte, ce qui rend les mises à niveau difficiles et sujettes aux erreurs.

#### Considérations supplémentaires pour NVMe/TCP

Trident prend en charge le protocole non-volatile memory express (NVMe) à l'aide du `ontap-san` driver, notamment :

- IPv6
- Instantanés 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
- Multipathing natif NVMe
- Arrêt progressif ou brutal des nœuds K8s (24.06)

Trident ne prend pas en charge :

- DH-HMAC-CHAP pris en charge nativement par NVMe
- Multipathing du device mapper (DM)
- chiffrement LUKS

| | |
|---|---|
| **REMARQUE** | NVMe est pris en charge uniquement avec les API REST ONTAP et non avec ONTAPI (ZAPI). |

## Préparez-vous à configurer le backend avec les pilotes SAN ONTAP

Comprenez les exigences et les options d'authentification pour configurer un backend ONTAP avec des pilotes SAN ONTAP.

#### Exigences

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

| | |
|---|---|
| **REMARQUE** | "Systèmes ASA r2" diffèrent des autres systèmes ONTAP (ASA, AFF et FAS) par l'implémentation de leur couche de stockage. Dans les systèmes ASA r2, des zones de disponibilité de stockage sont utilisées au lieu des agrégats. Consultez l'article de la "ce" Knowledge Base pour savoir comment affecter des agrégats aux SVM dans les systèmes ASA r2. |

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 le `ontap-san-economy` pilote.

Tous vos nœuds de travail Kubernetes doivent disposer des outils iSCSI appropriés. Consultez "Préparez le nœud de travail" pour plus de détails.

### Authentifier le backend ONTAP

Trident propose deux modes d'authentification pour un backend ONTAP.

- Authentification par identifiants : le nom d'utilisateur et le mot de passe d'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 d'ONTAP.
- Authentification par certificat : Trident peut également communiquer avec un cluster ONTAP à l'aide d'un certificat installé sur le backend. Dans ce cas, la définition du backend doit contenir les valeurs encodées en Base64 du certificat client, de la clé et, si utilisé (recommandé), du certificat de l'autorité de certification de confiance.

Vous pouvez mettre à jour les backends existants pour basculer entre les méthodes basées sur les identifiants et celles basées sur les certificats. Cependant, une seule méthode d'authentification est prise en charge à la fois. Pour passer à une autre méthode d'authentification, vous devez supprimer la méthode existante de la configuration du backend.

| | |
|---|---|
| **AVERTISSEMENT** | Si vous tentez de fournir **à la fois des informations d'identification et des certificats**, la création du backend échouera 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 identifiants

Trident requiert les identifiants d'un administrateur au niveau SVM ou au niveau cluster pour communiquer avec le backend ONTAP. Il est recommandé d'utiliser des rôles standard prédéfinis tels que `admin` ou `vsadmin`. Cela garantit la compatibilité avec les futures versions d'ONTAP qui pourraient exposer des API de fonctionnalités à utiliser par les futures versions de Trident. Un rôle de connexion de sécurité personnalisé peut être créé et utilisé avec Trident, mais cela n'est pas recommandé.

Voici un exemple de définition de backend :

**YAML**

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

**JSON**

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

Il est important de noter que la définition du backend est le seul endroit où les identifiants sont stockés en clair. Après la création du backend, les noms d'utilisateur et mots de passe sont encodés avec Base64 et stockés comme secrets Kubernetes. La création ou la mise à jour d'un backend est la seule étape qui nécessite la connaissance des identifiants. En tant que telle, il s'agit d'une opération réservée à l'administrateur Kubernetes/stockage.

#### Activer l'authentification basée sur les certificats

Les nouveaux et les anciens backends peuvent utiliser un certificat et communiquer avec le backend ONTAP. Trois paramètres sont requis dans la définition du backend.

- clientCertificate: Valeur codée en Base64 du certificat client.
- clientPrivateKey : valeur encodée en Base64 de la clé privée associée.
- trustedCACertificate : valeur encodée en Base64 du certificat CA de confiance. Si vous utilisez un CA de confiance, ce paramètre doit être fourni. Cela peut être ignoré si aucun CA de confiance n'est utilisé.

Un workflow typique implique 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 à authentifier.

```
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"
```

2. Ajoutez un certificat d'autorité de certification de confiance au cluster ONTAP. Cette opération peut déjà être effectuée par l'administrateur de stockage. Ignorez si aucune autorité de certification de confiance 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 (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
```

| | |
|---|---|
| **REMARQUE** | Après avoir exécuté cette commande, ONTAP vous invite à saisir un certificat. Collez le contenu du `k8senv.pem` fichier généré à l'étape 1, puis appuyez sur `END` pour terminer l'installation. |

4. Confirmez 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
```

5. Testez l'authentification à l'aide du certificat généré. Remplacez <ONTAP Management LIF> et <vserver name> par l'adresse IP de la LIF de gestion et le nom SVM.

```
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 d'autorité de certification 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 backend en utilisant les valeurs obtenues à l'étape précédente.

```
cat cert-backend.json
{
"version": 1,
"storageDriverName": "ontap-san",
"backendName": "SanBackend",
"managementLIF": "1.2.3.4",
"svm": "vserver_test",
"clientCertificate": "Faaaakkkkeeee...Vaaalllluuuueeee",
"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 |
+------------+----------------+--------------------------------------
+--------+---------+
```

#### Mettez à jour les méthodes d'authentification ou faites pivoter les identifiants

Vous pouvez mettre à jour un backend existant pour utiliser une méthode d'authentification différente ou pour faire pivoter ses identifiants. Cela fonctionne dans les deux sens : les backends qui utilisent un nom d'utilisateur/mot de passe peuvent être mis à jour pour utiliser des certificats ; les backends qui utilisent des certificats peuvent être mis à jour pour utiliser un nom d'utilisateur/mot de passe. Pour ce faire, vous devez supprimer la méthode d'authentification existante et ajouter la nouvelle méthode d'authentification. Ensuite, utilisez 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 |
+------------+----------------+--------------------------------------
+--------+---------+
```

| | |
|---|---|
| **REMARQUE** | Lors du renouvellement 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 backend. Lors du renouvellement des certificats, plusieurs certificats peuvent être associés à l'utilisateur. Le backend est ensuite mis à jour pour utiliser le nouveau certificat, après quoi l'ancien certificat peut être supprimé du cluster ONTAP. |

La mise à jour d'un backend n'interrompt pas l'accès aux volumes déjà créés et n'affecte pas les connexions de volumes établies ultérieurement. Une mise à jour réussie du backend indique que Trident peut communiquer avec le backend ONTAP et gérer les futures opérations sur les volumes.

#### Créer un rôle ONTAP personnalisé pour Trident

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

Reportez-vous à "Générateur de rôles personnalisés Trident" pour plus d'informations sur la création de rôles personnalisés Trident.

**Utilisation de l'interface de ligne de commande ONTAP**

1. Créez un nouveau 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. Associez le rôle à l'utilisateur :

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

**Utilisation de System Manager**

Effectuez les étapes suivantes dans ONTAP System Manager :

1. **Créer un rôle personnalisé**:
   a. Pour créer un rôle personnalisé au niveau du cluster, sélectionnez **Cluster > Settings**.

      (Ou) Pour créer un rôle personnalisé au niveau de la SVM, sélectionnez **Storage > Storage VMs > `required SVM`> Paramètres > Utilisateurs et rôles**.

   b. Sélectionnez l'icône flèche (→) à côté de **Users and Roles**.
   c. Sélectionnez **+Add** sous **Roles**.
   d. Définissez les règles du rôle et cliquez sur **Save**.
2. **Associez le rôle à l'utilisateur Trident** : + Effectuez les étapes suivantes sur la page **Utilisateurs et rôles** :
   a. Sélectionnez l'icône Ajouter **+** sous **Users**.
   b. Sélectionnez le nom d'utilisateur requis, puis sélectionnez un rôle dans le menu déroulant pour **Rôle**.
   c. Cliquez sur **Enregistrer**.

Reportez-vous aux pages suivantes pour plus d'informations :

- "Rôles personnalisés pour l'administration d'ONTAP" ou "Définir des rôles personnalisés"
- "Travailler avec les rôles et les utilisateurs"

### Authentifiez les connexions avec CHAP bidirectionnel

Trident peut authentifier les sessions iSCSI avec CHAP bidirectionnel pour les `ontap-san` et `ontap-san-economy` pilotes. Cela nécessite l'activation de l'option `useCHAP` dans la définition de votre backend. Lorsqu'elle est définie sur `true`, Trident configure la sécurité par défaut de l'initiateur du SVM sur CHAP bidirectionnel et définit le nom d'utilisateur et les secrets à partir du fichier backend. NetApp recommande

d'utiliser CHAP bidirectionnel pour authentifier les 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: rqxigXgkesIpwxyz
chapTargetUsername: iJF4heBRT0TCwxyz
chapUsername: uh2aNCLSd6cNwxyz
```

| AVERTISSEMENT | Le `useCHAP` paramètre est une option booléenne qui ne peut être configurée qu'une seule fois. Il est défini sur faux par défaut. Après l'avoir défini sur vrai, vous ne pouvez plus le définir sur faux. |
|---|---|

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

#### Comment ça marche

En définissant `useCHAP` sur true, l'administrateur de stockage demande à Trident de configurer CHAP sur le système de stockage. Cela inclut les éléments suivants :

- Configuration de CHAP sur la SVM :
  - Si le type de sécurité par défaut de l'initiateur SVM est « aucun » (défini par défaut) **et** qu'il n'y a pas de LUN préexistants déjà présents dans le volume, Trident définira le type de sécurité par défaut à `CHAP` et procédera à la configuration du nom d'utilisateur et des secrets de l'initiateur CHAP et de la cible.
  - Si la SVM contient des LUN, Trident n'activera pas CHAP sur la SVM. Cela garantit que l'accès aux LUN déjà présentes sur la SVM n'est pas restreint.
- Configuration du nom d'utilisateur et des secrets de l'initiateur et de la cible CHAP ; ces options doivent être spécifiées dans la configuration du backend (comme indiqué ci-dessus).

Après la création du backend, Trident crée une CRD correspondante `tridentbackend` et stocke les secrets CHAP ainsi que les noms d'utilisateur en tant que secrets Kubernetes. Tous les PV créés par Trident sur ce backend seront montés et attachés via CHAP.

#### Renouvelez les identifiants et mettez à jour les backends

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

| | |
|---|---|
| **AVERTISSEMENT** | 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 à l'aide de l'ONTAP CLI ou d'ONTAP System Manager, car Trident ne pourra pas prendre en compte 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": "cl9qxUpDaTeD",
    "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 restent inchangées ; elles demeureront actives si les identifiants sont mis à jour par Trident sur le SVM. Les nouvelles connexions utilisent les identifiants mis à jour et les connexions existantes demeurent actives. La déconnexion puis la reconnexion d'anciens PV entraînera leur utilisation des identifiants mis à jour.

## Options et exemples de configuration SAN ONTAP

Découvrez comment créer et utiliser des pilotes SAN ONTAP avec votre installation Trident. Cette section fournit des exemples de configuration backend et des détails pour le mappage des backends à StorageClasses. "Systèmes ASA r2" diffèrent des autres systèmes ONTAP (ASA, AFF et FAS) par l'implémentation de leur couche de stockage. Ces variations ont un impact sur l'utilisation de certains paramètres comme noté. "En

savoir plus sur les différences entre les systèmes ASA r2 et les autres systèmes ONTAP". Dans la configuration du backend Trident, il n'est pas nécessaire de préciser que votre système est ASA r2. Lorsque vous sélectionnez `ontap-san` comme `storageDriverName`, Trident détecte automatiquement les systèmes ASA r2 ou autres systèmes ONTAP. Certains paramètres de configuration du backend ne s'appliquent pas aux systèmes ASA r2, comme indiqué dans le tableau ci-dessous.

| | |
|---|---|
| **REMARQUE** | Seul le `ontap-san`pilote (avec les protocoles iSCSI, NVMe/TCP et FC) est pris en charge pour les systèmes ASA r2. |

**Options de configuration du backend**

Consultez le tableau suivant pour les options de configuration du backend :

| Paramètre | Description | Défaut |
|---|---|---|
| `version` | | Toujours 1 |
| `storageDriverName` | Nom du pilote de stockage | `ontap-san` ou `ontap-san-economy` |
| `backendName` | Nom personnalisé ou le stockage backend | Nom du pilote + "_" + dataLIF |
| `managementLIF` | Adresse IP d'une LIF de gestion de cluster ou de SVM.<br><br>Un nom de domaine complet (FQDN) peut être spécifié.<br><br>Il est possible de configurer l'utilisation d'adresses IPv6 si Trident a été installé avec l'option IPv6. Les adresses IPv6 doivent être définies entre crochets, comme `[28e8:d9fb:a825:b7bf:69a8:d02f:9e7b:3555]`.<br><br>Pour une transition en douceur MetroCluster, consultez le Exemple MetroCluster.<br><br>**REMARQUE** : Si vous utilisez les identifiants « vsadmin », `managementLIF` doit être celui de la SVM ; si vous utilisez les identifiants « admin », `managementLIF` doit être celui du cluster. | "10.0.0.1", "[2001:1234:abcd::fefe]" |

| Paramètre | Description | Défaut |
| --- | --- | --- |
| `dataLIF` | Adresse IP de l'interface logique de protocole (LIF). Il est possible de configurer l'utilisation d'adresses IPv6 si Trident a été installé avec l'option IPv6. Les adresses IPv6 doivent être définies entre crochets, comme `[28e8:d9fb:a825:b7bf:69a8:d02f:9e7b:3555]`. **Ne pas spécifier pour iSCSI.** Trident utilise "Mappage LUN sélectif ONTAP" pour découvrir les LIF iSCSI nécessaires à l'établissement d'une session multi-chemin. Un avertissement est généré si `dataLIF` est explicitement défini. **Omettre pour MetroCluster.** Voir le Exemple MetroCluster. | Dérivé par le SVM |
| `svm` | Machine virtuelle de stockage à utiliser **Omettre pour MetroCluster.** Voir le Exemple MetroCluster. | Dérivé si un SVM `managementLIF` est spécifié |
| `useCHAP` | Utilisez CHAP pour authentifier iSCSI pour les pilotes SAN ONTAP [paramètre booléen]. Définissez sur `true` pour que Trident configure et utilise CHAP bidirectionnel comme authentification par défaut pour la SVM indiquée dans le backend. Consultez "Préparez-vous à configurer le backend avec les pilotes SAN ONTAP" pour plus de détails. **Non pris en charge pour FCP ou NVMe/TCP.** | `false` |
| `chapInitiatorSecret` | Secret de l'initiateur CHAP. Obligatoire si `useCHAP=true` | "" |
| `labels` | Ensemble d'étiquettes arbitraires au format JSON à appliquer aux volumes | "" |
| `chapTargetInitiatorSecret` | Secret de l'initiateur de la cible CHAP. Obligatoire si `useCHAP=true` | "" |
| `chapUsername` | Nom d'utilisateur entrant. Obligatoire si `useCHAP=true` | "" |
| `chapTargetUsername` | Nom d'utilisateur cible. Obligatoire si `useCHAP=true` | "" |
| `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 d'autorité de certification de confiance. Facultatif. Utilisé pour l'authentification par certificat. | "" |
| `username` | Nom d'utilisateur requis pour communiquer avec le cluster ONTAP. Utilisé pour l'authentification par identifiants. Pour l'authentification Active Directory, voir "Authentifier Trident auprès d'un SVM backend à l'aide des identifiants Active Directory". | "" |

| Paramètre | Description | Défaut |
| --- | --- | --- |
| `password` | Mot de passe requis pour communiquer avec le cluster ONTAP. Utilisé pour l'authentification par identifiants. Pour l'authentification Active Directory, voir "Authentifier Trident auprès d'un SVM backend à l'aide des identifiants Active Directory". | "" |
| `svm` | Machine virtuelle de stockage à utiliser | Dérivé si un SVM `managementLIF` est spécifié |
| `storagePrefix` | Préfixe utilisé lors du provisionnement de nouveaux volumes dans la SVM. Ne peut pas être modifié ultérieurement. Pour mettre à jour ce paramètre, vous devrez créer un nouveau backend. | `trident` |
| `aggregate` | Agrégat pour le provisionnement (facultatif ; s'il est défini, il doit être attribué à la SVM). Pour le `ontap-nas-flexgroup` driver, cette option est ignorée. S'il n'est pas attribué, n'importe quel agrégat disponible peut être utilisé pour provisionner un FlexGroup volume.<br>**REMARQUE** : Lorsqu'un agrégat est mis à jour dans SVM, il est automatiquement mis à jour dans Trident par interrogation de SVM, sans qu'il soit nécessaire de redémarrer le contrôleur Trident. Si vous avez configuré un agrégat spécifique dans Trident pour provisionner des volumes, si l'agrégat est renommé ou déplacé hors de la SVM, le backend passera en état d'échec dans Trident lors de l'interrogation de l'agrégat SVM. Vous devez soit changer l'agrégat pour un qui est présent sur la SVM, soit le supprimer complètement pour remettre le backend en ligne.<br>**Ne pas spécifier pour les systèmes ASA r2**. | "" |
| `limitAggregateUsage` | L'approvisionnement échoue si l'utilisation dépasse ce pourcentage. Si vous utilisez un Amazon FSx for NetApp ONTAP backend, ne spécifiez pas `limitAggregateUsage`. Les `fsxadmin` et `vsadmin` fournis ne contiennent pas les autorisations requises pour récupérer l'utilisation agrégée et la limiter à l'aide de Trident. **Ne pas spécifier pour les systèmes ASA r2**. | "" (non appliqué par défaut) |

| Paramètre | Description | Défaut |
| --- | --- | --- |
| `limitVolumeSize` | L'approvisionnement échoue si la taille du volume demandée dépasse 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` | Options de débogage à utiliser lors du dépannage. Exemple, {"api":false, "method":true} n'utilisez-les que si vous effectuez un dépannage et avez besoin d'un journal détaillé. | `null` |
| `useREST` | Paramètre booléen à utiliser pour les API REST ONTAP.<br>`` `useREST` Lorsqu'il est défini sur `true`, Trident utilise les API REST ONTAP pour communiquer avec le backend ; lorsqu'il est défini sur `false`, Trident utilise les appels ONTAPI (ZAPI) pour communiquer avec le backend. Cette fonctionnalité nécessite ONTAP 9.11.1 ou une version ultérieure. De plus, le rôle de connexion ONTAP utilisé doit avoir accès à l'application `ontapi`. Cela est satisfait par les rôles prédéfinis `vsadmin` et `cluster-admin`. À partir de la version Trident 24.06 et ONTAP 9.15.1 ou version ultérieure, `useREST` est défini sur `true` par défaut ; modifiez `useREST` sur `false` pour utiliser les appels ONTAPI (ZAPI). ``<br>`useREST` est entièrement compatible NVMe/TCP.<br>**REMARQUE** : NVMe est pris en charge uniquement avec les API REST ONTAP et non avec ONTAPI (ZAPI).<br>**Si spécifié, toujours définir sur `true` pour les systèmes ASA r2**. | `true` pour ONTAP 9.15.1 ou version ultérieure, sinon `false`. |

| Paramètre | Description | Défaut |
| --- | --- | --- |
| `sanType` | Utilisez pour sélectionner `iscsi` pour iSCSI, `nvme` pour NVMe/TCP ou `fcp` pour SCSI sur Fibre Channel (FC). | `iscsi` si vide |
| `formatOptions` | Utilisez `formatOptions` pour spécifier des arguments de ligne de commandes pour la commande `mkfs`, qui seront appliqués chaque fois qu'un volume est formaté. Cela vous permet de formater le volume selon vos préférences. Assurez-vous de spécifier les formatOptions similaires à celles des options de la commande mkfs, à l'exclusion du chemin du périphérique. Exemple : « -E nodiscard »<br><br>**Pris en charge pour `ontap-san` et `ontap-san-economy` pilotes avec le protocole iSCSI. En outre, pris en charge pour les systèmes ASA r2 lors de l'utilisation des protocoles iSCSI et NVMe/TCP.** | |
| `limitVolumePoolSize` | Taille maximale requise de FlexVol lors de l'utilisation de LUN dans le backend ontap-san-economy. | "" (non appliqué par défaut) |
| `denyNewVolumePools` | Limite `ontap-san-economy` les backends à la création de nouveaux volumes FlexVol® pour contenir leurs LUN. Seuls les FlexVol® préexistants sont utilisés pour le provisionnement de nouveaux PV. | |

#### Recommandations d'utilisation de formatOptions

Trident recommande les options suivantes pour accélérer le processus de mise en forme :

- **-E nodiscard (ext3, ext4) :** Ne pas tenter de supprimer des blocs lors de la création du système de fichiers (la suppression initiale de blocs est utile sur les disques SSD et les systèmes de stockage à provisionnement fin). Cette option remplace l'option obsolète « -K » et s'applique aux systèmes de fichiers ext3 et ext4.
- **-K (xfs) :** Ne pas tenter de supprimer des blocs lors de la création du système de fichiers (mkfs). Cette option s'applique au système de fichiers xfs.

#### Authentifier Trident auprès d'un SVM backend à l'aide des identifiants Active Directory

Vous pouvez configurer Trident pour qu'il s'authentifie auprès d'une SVM backend à l'aide d'identifiants Active Directory (AD). Avant qu'un compte AD puisse accéder à la SVM, vous devez configurer l'accès des contrôleurs de domaine AD au cluster ou à la SVM. Pour l'administration du cluster avec un compte AD, vous devez créer un tunnel de domaine. Consultez "Configurer l'accès au contrôleur de domaine Active Directory dans ONTAP" pour plus de détails.

**étapes**

1. Configurer les paramètres du système de noms de domaine (DNS) pour une SVM backend :

```
vserver services dns create -vserver <svm_name> -dns-servers
<dns_server_ip1>,<dns_server_ip2>
```

2. Exécutez la commande suivante pour créer un compte d'ordinateur pour la SVM dans Active Directory :

```
vserver active-directory create -vserver DataSVM -account-name ADSERVER1
-domain demo.netapp.com
```

3. Utilisez cette commande pour créer un utilisateur ou un groupe AD afin de gérer le cluster ou la SVM

```
security login create -vserver <svm_name> -user-or-group-name
<ad_user_or_group> -application <application> -authentication-method domain
-role vsadmin
```

4. Dans le fichier de configuration backend de Trident, définissez les paramètres `username` et `password` sur le nom d'utilisateur ou de groupe AD et le mot de passe, respectivement.

### Options de configuration backend pour le provisionnement des volumes

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

| Paramètre | Description | Défaut |
|---|---|---|
| `spaceAllocation` | Allocation d'espace pour les LUNs | "true" **Si spécifié, définissez sur `true` pour les systèmes ASA r2**. |
| `spaceReserve` | Mode de réservation d'espace : « aucun » (fin) ou « volume » (épais). **À définir sur `none` pour les systèmes ASA r2**. | "none" |
| `snapshotPolicy` | Stratégie d'instantané à utiliser. **À définir sur `none` pour les systèmes ASA r2**. | "none" |
| `qosPolicy` | Groupe de règles QoS à attribuer aux volumes créés. Choisissez l'un de qosPolicy ou adaptiveQosPolicy par pool de stockage/backend. L'utilisation des groupes de règles QoS avec Trident requiert 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 individuellement à chaque composant. Un groupe de règles QoS partagé impose une limite au débit total de toutes les charges de travail. | "" |
| `adaptiveQosPolicy` | Groupe de règles QoS adaptatives à attribuer aux volumes créés. Choisissez l'un des deux, qosPolicy ou adaptiveQosPolicy, par pool de stockage/backend | "" |
| `snapshotReserve` | Pourcentage du volume réservé aux snapshots. **Ne pas spécifier pour les systèmes ASA r2**. | "0" si `snapshotPolicy` est "aucun", sinon "" |
| `splitOnClone` | Séparer un clone de son parent lors de sa création | "false" |

| Paramètre | Description | Défaut |
| --- | --- | --- |
| `encryption` | Activez le chiffrement de volume NetApp (NVE) sur le nouveau volume ; la valeur par défaut est `false`. NVE doit être sous licence et activé sur le cluster pour utiliser cette option. Si NAE est activé sur le backend, tout volume provisionné dans Trident sera activé pour NAE. Pour plus d'informations, consultez : "Comment Trident fonctionne avec NVE et NAE". | "false" **Si spécifié, définissez sur `true` pour les systèmes ASA r2**. |
| `luksEncryption` | Activez le chiffrement LUKS. Consultez "Utilisez Linux Unified Key Setup (LUKS)". | "" **Définir sur `false` pour les systèmes ASA r2**. |
| `tieringPolicy` | Politique de hiérarchisation à utiliser « none » **Ne pas spécifier pour les systèmes ASA r2**. | |
| `nameTemplate` | Modèle pour créer des noms de volumes 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'
```

| | |
|---|---|
| **REMARQUE** | Pour tous les volumes créés à l'aide du `ontap-san` driver, Trident ajoute 10 % de capacité supplémentaire à la FlexVol pour intégrer les métadonnées LUN. La LUN sera provisionnée avec la taille exacte demandée par l'utilisateur dans le PVC. Trident ajoute 10 % à la FlexVol (affiché comme taille disponible dans ONTAP). Les utilisateurs obtiendront désormais la capacité utilisable qu'ils ont demandée. Cette modification empêche également les LUN de devenir en lecture seule, sauf si l'espace disponible est entièrement utilisé. Ceci ne s'applique pas à ontap-san-economy. |

Pour les backends qui définissent `snapshotReserve`, Trident calcule la taille des volumes comme suit :

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

Le chiffre 1,1 correspond aux 10 % supplémentaires que Trident ajoute au FlexVol pour prendre en compte les métadonnées LUN. Pour `snapshotReserve` = 5 %, et une demande PVC = 5 Gio, la taille totale du volume est de 5,79 Gio et la taille disponible est de 5,5 Gio. La commande `volume show` devrait afficher des résultats similaires à cet exemple :

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

#### Exemples de configuration minimale

Les exemples suivants présentent des configurations de base qui laissent la plupart des paramètres par défaut. C'est la manière la plus simple de définir un backend.

| | |
|---|---|
| **REMARQUE** | Si vous utilisez Amazon FSx sur NetApp ONTAP avec Trident, NetApp recommande de spécifier les noms DNS des LIF au lieu des adresses IP. |

#### Exemple SAN ONTAP

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

```
---
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 backend pour éviter d'avoir à mettre à jour manuellement la définition du backend après le basculement et le retour en arrière pendant "Réplication et récupération de SVM".

Pour une transition et un retour en arrière sans heurt, spécifiez le SVM en utilisant `managementLIF` et omettez 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 par certificat**

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

```
---
version: 1
storageDriverName: ontap-san
backendName: DefaultSANBackend
managementLIF: 10.0.0.1
svm: svm_iscsi
useCHAP: true
chapInitiatorSecret: cl9qxIm36DKyawxy
chapTargetInitiatorSecret: rqxigXgkesIpwxyz
chapTargetUsername: iJF4heBRT0TCwxyz
chapUsername: uh2aNCLSd6cNwxyz
clientCertificate: ZXR0ZXJwYXB...ICMgJ3BhcGVyc2
clientPrivateKey: vciwKIyAgZG...0cnksIGRlc2NyaX
trustedCACertificate: zcyBbaG...b3Igb3duIGNsYXNz
```

### Exemples CHAP bidirectionnels

Ces exemples créent un backend avec `useCHAP` défini sur `true`.

#### Exemple ONTAP SAN CHAP

```
---
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 d'économie ONTAP SAN CHAP

```
---
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

Vous devez disposer d'une SVM configurée avec NVMe sur votre backend ONTAP. Il s'agit d'une configuration 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 disposer d'une SVM configurée avec FC sur votre backend ONTAP. Il s'agit d'une configuration de base du backend 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 du backend avec nameTemplate**

```
---
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}}"
```

**formatOptions exemple pour le pilote ontap-san-economy**

```
---
version: 1
storageDriverName: ontap-san-economy
managementLIF: ""
svm: svm1
username: ""
password: "!"
storagePrefix: whelk_
debugTraceFlags:
  method: true
  api: true
defaults:
  formatOptions: -E nodiscard
```

### Exemples de backends avec pools virtuels

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

Trident définit les étiquettes de provisionnement dans le champ « Commentaires ». Les commentaires sont définis sur le volume FlexVol. Trident copie toutes les étiquettes présentes sur un pool virtuel vers le volume de stockage lors du provisionnement. Pour plus de simplicité, les administrateurs de stockage peuvent définir des étiquettes par pool virtuel et regrouper les volumes par étiquette.

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

### Exemple 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: mysqldb
      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"
```

### Map backends vers StorageClasses

Les définitions suivantes de StorageClass se réfèrent à Exemples de backends avec pools virtuels. En utilisant le champ `parameters.selector`, chaque StorageClass indique quels pools virtuels peuvent être utilisés pour héberger un volume. Le volume possédera les aspects définis dans le pool virtuel choisi.

- Le `protection-gold` StorageClass sera associé au premier pool virtuel dans le `ontap-san` backend. Il s'agit du seul pool offrant une protection de niveau or.

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

- Le `protection-not-gold` StorageClass sera associé aux deuxième et troisième pools virtuels dans `ontap-san` backend. 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"
```

- Le `app-mysqldb` StorageClass correspondra au troisième pool virtuel dans le `ontap-san-economy` backend. Il s'agit du seul pool offrant une configuration de pool de stockage pour l'app 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"
```

- Le `protection-silver-creditpoints-20k` StorageClass sera associé au deuxième pool virtuel dans `ontap-san` backend. Il s'agit du seul pool 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"
```

- Le `creditpoints-5k` StorageClass sera associé au troisième pool virtuel dans le `ontap-san` backend et au quatrième pool virtuel dans le `ontap-san-economy` backend. Ce sont les seuls pools proposant 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"
```

- Le `my-test-app-sc` StorageClass sera associé au `testAPP` pool virtuel dans le `ontap-san` pilote avec `sanType: nvme`. C'est le seul pool offrant `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 déterminera quel pool virtuel est sélectionné et s'assurera que l'exigence de stockage est respectée.

# Pilotes ONTAP NAS

## Présentation du pilote ONTAP NAS

Découvrez comment configurer un backend ONTAP avec les pilotes NAS ONTAP et Cloud Volumes ONTAP.

### Détails du pilote ONTAP NAS

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

| Pilote | Protocole | volumeMode | 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` |

| Pilote | Protocole | volumeMode | Modes d'accès pris en charge | Systèmes de fichiers pris en charge |
|---|---|---|---|---|
| `ontap-nas-flexgroup` | NFS SMB | Système de fichiers | RWO, ROX, RWX, RWOP | "", `nfs`, `smb` |

**AVERTISSEMENT**

- Utilisez `ontap-san-economy` uniquement si le nombre d'utilisations du volume persistant doit être supérieur à "limites de volume ONTAP prises en charge".
- Utilisez `ontap-nas-economy` uniquement si le nombre d'utilisations de volumes persistants est censé être supérieur à "limites de volume ONTAP prises en charge" et que le pilote `ontap-san-economy` ne peut pas être utilisé.
- N'utilisez pas `ontap-nas-economy` si vous prévoyez un besoin de protection des données, de reprise après sinistre ou de mobilité.
- NetApp ne recommande pas d'utiliser la croissance automatique FlexVol® dans tous les pilotes ONTAP, sauf ontap-san. En guise de solution de contournement, Trident prend en charge l'utilisation de la réserve de snapshots et adapte les volumes FlexVol® en conséquence.

#### Autorisations de l'utilisateur

Trident doit être exécuté en tant qu'administrateur ONTAP ou SVM, généralement en utilisant l' `admin`utilisateur cluster ou un `vsadmin`utilisateur SVM, ou un utilisateur avec un nom différent ayant le même rôle.

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

**REMARQUE**

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

Bien qu'il soit possible de créer un rôle plus restrictif au sein d'ONTAP qu'un pilote Trident peut utiliser, nous ne le recommandons pas. La plupart des nouvelles versions de Trident feront appel à des API supplémentaires dont il faudrait tenir compte, ce qui rend les mises à niveau difficiles et sujettes aux erreurs.

## Préparez-vous à configurer un backend avec les pilotes NAS ONTAP

Comprenez les exigences, les options d'authentification et les règles d'export pour configurer un backend ONTAP avec les pilotes ONTAP NAS. À compter de la version 25.10, NetApp Trident prend en charge "NetApp système de stockage AFX". Les systèmes de stockage NetApp AFX diffèrent des autres systèmes ONTAP (ASA, AFF et FAS) dans l'implémentation de leur couche de stockage. Dans la configuration du backend Trident, il n'est pas nécessaire de préciser que votre système est AFX. Lorsque

vous sélectionnez `ontap-nas` comme `storageDriverName`, Trident détecte automatiquement les systèmes AFX.

| | |
|---|---|
| **REMARQUE** | Seul le `ontap-nas` driver (avec le protocole NFS) est pris en charge pour les systèmes AFX; le protocole SMB n'est pas pris en charge. |

#### Exigences

- Pour tous les backends ONTAP, Trident exige qu'au moins un agrégat soit affecté 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` driver et une classe Bronze qui utilise le `ontap-nas-economy` driver.
- Tous vos nœuds de travail Kubernetes doivent disposer des outils NFS appropriés. Consultez "ici" pour plus de détails.
- Trident prend uniquement en charge les volumes SMB montés sur des pods exécutés sur des nœuds Windows. Consultez Préparez-vous à provisionner des volumes SMB pour plus de détails.

#### Authentifier le backend ONTAP

Trident propose deux modes d'authentification pour un backend ONTAP.

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

Vous pouvez mettre à jour les backends existants pour basculer entre les méthodes basées sur les identifiants et celles basées sur les certificats. Cependant, une seule méthode d'authentification est prise en charge à la fois. Pour passer à une autre méthode d'authentification, vous devez supprimer la méthode existante de la configuration du backend.

| | |
|---|---|
| **AVERTISSEMENT** | Si vous tentez de fournir **à la fois des informations d'identification et des certificats**, la création du backend échouera 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 identifiants

Trident requiert les identifiants d'un administrateur au niveau SVM ou au niveau cluster pour communiquer avec le backend ONTAP. Il est recommandé d'utiliser des rôles standard prédéfinis tels que `admin` ou `vsadmin`. Cela garantit la compatibilité avec les futures versions d'ONTAP qui pourraient exposer des API de fonctionnalités à utiliser par les futures versions de Trident. Un rôle de connexion de sécurité personnalisé peut être créé et utilisé avec Trident, mais cela n'est pas recommandé.

Voici un exemple de définition de backend :

**YAML**

```yaml
---
version: 1
backendName: ExampleBackend
storageDriverName: ontap-nas
managementLIF: 10.0.0.1
dataLIF: 10.0.0.2
svm: svm_nfs
credentials:
  name: secret-backend-creds
```

**JSON**

```json
{
  "version": 1,
  "backendName": "ExampleBackend",
  "storageDriverName": "ontap-nas",
  "managementLIF": "10.0.0.1",
  "dataLIF": "10.0.0.2",
  "svm": "svm_nfs",
  "credentials": {
        "name": "secret-backend-creds"
    }
}
```

Il est important de noter que la définition du backend est le seul endroit où les identifiants sont stockés en clair. Après la création du backend, les noms d'utilisateur et mots de passe sont encodés avec Base64 et stockés comme secrets Kubernetes. La création/la mise à jour d'un backend est la seule étape nécessitant la connaissance des identifiants. En tant que telle, il s'agit d'une opération réservée à l'administrateur Kubernetes/stockage.

#### Activer l'authentification par certificat

Les nouveaux et les anciens backends peuvent utiliser un certificat et communiquer avec le backend ONTAP. Trois paramètres sont requis dans la définition du backend.

- clientCertificate: Valeur codée en Base64 du certificat client.
- clientPrivateKey : valeur encodée en Base64 de la clé privée associée.
- trustedCACertificate : valeur encodée en Base64 du certificat CA de confiance. Si vous utilisez un CA de confiance, ce paramètre doit être fourni. Cela peut être ignoré si aucun CA de confiance n'est utilisé.

Un workflow typique implique 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 à authentifier.

```
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. Cette opération peut déjà être effectuée par l'administrateur de stockage. Ignorez si aucune autorité de certification de confiance 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 (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. Confirmez 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 du certificat généré. Remplacez <ONTAP Management LIF> et <vserver name> par l'adresse IP de la LIF de gestion et le nom SVM. Vous devez vous assurer que le LIF a sa stratégie de service 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 d'autorité de certification 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 backend en utilisant les valeurs obtenues à 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": "Faaaakkkkeeee...Vaaalllluuuueeee",
"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 |
+------------+----------------+--------------------------------------
+--------+---------+
```

#### Mettez à jour les méthodes d'authentification ou faites pivoter les identifiants

Vous pouvez mettre à jour un backend existant pour utiliser une méthode d'authentification différente ou pour faire pivoter ses identifiants. Cela fonctionne dans les deux sens : les backends qui utilisent un nom d'utilisateur/mot de passe peuvent être mis à jour pour utiliser des certificats ; les backends qui utilisent des certificats peuvent être mis à jour pour utiliser un nom d'utilisateur/mot de passe. Pour ce faire, vous devez supprimer la méthode d'authentification existante et ajouter la nouvelle méthode d'authentification. Ensuite, utilisez 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 |
+------------+----------------+--------------------------------------
+--------+---------+
```

**REMARQUE** Lors du renouvellement 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 backend. Lors du renouvellement des certificats, plusieurs certificats peuvent être associés à l'utilisateur. Le backend est ensuite mis à jour pour utiliser le nouveau certificat, après quoi l'ancien certificat peut être supprimé du cluster ONTAP.

La mise à jour d'un backend n'interrompt pas l'accès aux volumes déjà créés et n'affecte pas les connexions de volumes établies ultérieurement. Une mise à jour réussie du backend indique que Trident peut communiquer avec le backend ONTAP et gérer les futures opérations sur les volumes.

#### Créer un rôle ONTAP personnalisé pour Trident

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

Reportez-vous à "Générateur de rôles personnalisés Trident" pour plus d'informations sur la création de rôles personnalisés Trident.

**Utilisation de l'interface de ligne de commande ONTAP**

1. Créez un nouveau 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. Associez le rôle à l'utilisateur :

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

**Utilisation de System Manager**

Effectuez les étapes suivantes dans ONTAP System Manager :

1. **Créer un rôle personnalisé**:
   a. Pour créer un rôle personnalisé au niveau du cluster, sélectionnez **Cluster > Settings**.

      (Ou) Pour créer un rôle personnalisé au niveau de la SVM, sélectionnez **Storage > Storage VMs > `required SVM`> Paramètres > Utilisateurs et rôles**.

   b. Sélectionnez l'icône flèche (→) à côté de **Users and Roles**.
   c. Sélectionnez **+Add** sous **Roles**.
   d. Définissez les règles du rôle et cliquez sur **Save**.
2. **Associez le rôle à l'utilisateur Trident** : + Effectuez les étapes suivantes sur la page **Utilisateurs et rôles** :
   a. Sélectionnez l'icône Ajouter **+** sous **Users**.
   b. Sélectionnez le nom d'utilisateur requis, puis sélectionnez un rôle dans le menu déroulant pour **Rôle**.
   c. Cliquez sur **Enregistrer**.

Reportez-vous aux pages suivantes pour plus d'informations :

- "Rôles personnalisés pour l'administration d'ONTAP" ou "Définir des rôles personnalisés"
- "Travailler avec les rôles et les utilisateurs"

### Gérer les règles d'export NFS

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

Trident propose deux options lors de l'utilisation des règles d'export :

- Trident peut gérer dynamiquement la règle d'export ; dans ce mode de fonctionnement, l'administrateur de stockage spécifie une liste de blocs CIDR représentant les adresses IP admissibles. Trident ajoute automatiquement à la règle d'export les adresses IP des nœuds concernés qui appartiennent à ces plages lors de la publication. Sinon, si aucun CIDR n'est spécifié, toutes les adresses IP unicast à portée globale trouvées sur le nœud auquel le volume est publié seront ajoutées à la règle d'export.
- Les administrateurs de stockage peuvent créer une règle d'export et y ajouter des règles manuellement. Trident utilise la règle d'export par défaut, sauf si un autre nom de règle d'export est spécifié dans la configuration.

#### Gérer dynamiquement les règles d'export

Trident offre la possibilité de gérer dynamiquement les règles d'export pour les backends ONTAP. Cela donne à l'administrateur de stockage la possibilité de spécifier un espace d'adresses autorisé pour les adresses IP des nœuds de travail, plutôt que de définir manuellement des règles explicites. Cela simplifie grandement la gestion des règles d'export ; les modifications apportées à la règle d'export 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 de travail qui montent des volumes et dont les adresses IP se trouvent dans la plage spécifiée, ce qui permet une gestion fine et automatisée.

| | |
|---|---|
| **REMARQUE** | N'utilisez pas la traduction d'adresses réseau (NAT) lorsque vous utilisez des règles d'export dynamiques. Avec la NAT, le contrôleur de stockage voit l'adresse NAT du frontal et non l'adresse IP réelle de l'hôte, donc l'accès sera refusé si aucune correspondance n'est trouvée dans les règles d'export. |

#### Exemple

Deux options de configuration doivent être utilisées. Voici un exemple de définition de backend :

```
---
version: 1
storageDriverName: ontap-nas-economy
backendName: ontap_nas_auto_export
managementLIF: 192.168.0.135
svm: svm1
username: vsadmin
password: password
autoExportCIDRs:
  - 192.168.0.0/24
autoExportPolicy: true
```

| | |
|---|---|
| **REMARQUE** | Lorsque vous utilisez cette fonctionnalité, vous devez vous assurer que la jonction racine de votre SVM possède une règle d'export précédemment créée avec une règle d'export autorisant le bloc CIDR du nœud (par exemple, la règle d'export par défaut). Suivez toujours la bonne pratique recommandée par NetApp pour dédier une SVM à Trident. |

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

- `autoExportPolicy` est défini sur `true`. Cela indique que Trident crée une règle d'export pour chaque

volume provisionné avec ce backend pour la `svm1` SVM et gère l'ajout et la suppression de règles à l'aide de blocs d'adresses `autoexportCIDRs`. Jusqu'à ce qu'un volume soit attaché à un nœud, le volume utilise une règle d'export vide sans aucune règle afin d'empêcher tout accès non désiré à ce volume. Lorsqu'un volume est publié sur un nœud, Trident crée une règle d'export portant le même nom que le qtree sous-jacent contenant l'adresse IP du nœud dans le bloc CIDR spécifié. Ces adresses IP seront également ajoutées à la règle d'export utilisée par le volume FlexVol parent.

  - Par exemple :
    - UUID de backend 403b5326-8482-40db-96d0-d83fb3f4daec
    - `autoExportPolicy` défini sur `true`
    - préfixe de stockage `trident`
    - UUID PVC a79bcf5f-7b6d-4a40-9876-e2551f159c1c
    - Le qtree nommé trident_pvc_a79bcf5f_7b6d_4a40_9876_e2551f159c1c crée une règle d'export pour le FlexVol nommé `trident-403b5326-8482-40db96d0-d83fb3f4daec`, une règle d'export pour le qtree nommé `trident_pvc_a79bcf5f_7b6d_4a40_9876_e2551f159c1c`, et une règle d'export vide nommée `trident_empty` sur la SVM. Les règles de la règle d'export du FlexVol seront un sur-ensemble de toutes les règles contenues dans les règles d'export du qtree. La règle d'export vide sera réutilisée par tous les volumes qui ne sont pas attachés.

- `autoExportCIDRs` contient une liste de blocs d'adresses. Ce champ est facultatif et il est défini par défaut sur ["0.0.0.0/0", "::/0"]. S'il n'est pas défini, Trident ajoute toutes les adresses unicast à portée globale trouvées sur les nœuds de travail avec des publications.

Dans cet exemple, l' `192.168.0.0/24`espace d'adresses est fourni. Cela indique que les adresses IP des nœuds Kubernetes qui se trouvent dans cette plage d'adresses avec des publications seront ajoutées à la règles d'export que Trident crée. Lorsque Trident enregistre un nœud sur lequel il s'exécute, il récupère les adresses IP du nœud et les vérifie par rapport aux blocs d'adresses fournis dans `autoExportCIDRs`. Au moment de la publication, après avoir filtré les adresses IP, Trident crée les règles d'export pour les adresses IP clientes du nœud auquel il publie.

Vous pouvez mettre à jour `autoExportPolicy` et `autoExportCIDRs` pour les backends après leur création. Vous pouvez ajouter de nouveaux CIDR pour un backend géré automatiquement ou supprimer les CIDR existants. Faites attention lors de la suppression de CIDR afin de garantir que les connexions existantes ne soient pas interrompues. Vous pouvez également choisir de désactiver `autoExportPolicy` pour un backend et revenir à une règle d'export créée manuellement. Cela nécessitera de définir le paramètre `exportPolicy` dans la configuration de votre backend.

Après que Trident a créé ou mis à jour un backend, vous pouvez vérifier le backend en utilisant `tridentctl` ou le 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 règles d'export afin de supprimer les règles d'accès correspondantes au nœud. En supprimant cette adresse IP de nœud des règles d'export des backends gérés, Trident empêche les montages non autorisés, sauf si cette adresse IP est réutilisée par un nouveau nœud dans le cluster.

Pour les backends existants, la mise à jour du backend avec `tridentctl update backend` garantit que Trident gère automatiquement les règles d'export. Cela crée deux nouvelles règles d'export nommées d'après l'UUID du backend et le nom du qtree lorsqu'elles sont nécessaires. Les volumes présents sur le backend utiliseront les nouvelles règles d'export après avoir été démontés puis remontés.

| | |
|---|---|
| **REMARQUE** | La suppression d'un backend avec des règles d'export gérées automatiquement supprimera la règle d'export créée dynamiquement. Si le backend est recréé, il est traité comme un nouveau backend et cela entraînera la création d'une nouvelle règle d'export. |

Si l'adresse IP d'un nœud en production est modifiée, vous devez redémarrer le pod Trident sur ce nœud. Trident mettra alors à jour la règle d'export pour les backends qu'il gère afin de refléter ce changement d'adresse IP.

#### Préparez-vous à provisionner des volumes SMB

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

| | |
|---|---|
| **AVERTISSEMENT** | Vous devez configurer les protocoles NFS et SMB/CIFS sur la SVM pour créer un `ontap-nas-economy` volume SMB pour les clusters ONTAP sur site. L'absence de configuration de l'un ou l'autre de ces protocoles entraînera l'échec de la création du volume SMB. |

| | |
|---|---|
| **REMARQUE** | `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.

- Un cluster Kubernetes avec un nœud contrôleur Linux et au moins un nœud de travail Windows exécutant Windows Server 2022. Trident prend uniquement en charge les volumes SMB montés sur des pods exécutés sur des nœuds Windows.
- Au moins un secret Trident contenant vos identifiants Active Directory. Pour générer le 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 à "GitHub : CSI Proxy" ou "GitHub: CSI Proxy pour Windows" pour les nœuds Kubernetes exécutés sous Windows.

**Étapes**

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

   | | |
   |---|---|
   | **REMARQUE** | 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 "Microsoft Management Console" Shared Folders snap-in, 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 commande ONTAP :

   a. Si nécessaire, créez la structure du chemin d'accès du 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 à la SVM spécifiée :

      ```
      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
      ```

**REMARQUE** Reportez-vous à "Créer un partage SMB" pour plus de détails.

2. Lors de la création du backend, vous devez configurer les éléments suivants pour spécifier les volumes SMB. Pour toutes les options de configuration du backend FSx pour ONTAP, reportez-vous à "Options de configuration et exemples pour FSx for ONTAP".

| 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 pour permettre à Trident de créer le partage SMB ; ou vous pouvez laisser le paramètre vide pour empêcher l'accès partagé aux volumes. Ce paramètre est facultatif pour ONTAP sur site. Ce paramètre est obligatoire pour Amazon FSx for ONTAP backends et ne peut pas être vide. | `smb-share` |
| `nasType` | **Doit être défini sur `smb`.** Si nul, la valeur par défaut est `nfs`. | `smb` |
| `securityStyle` | Style de sécurité pour les nouveaux volumes. **Doit être défini sur `ntfs` ou `mixed` pour les volumes SMB.** | `ntfs` or `mixed` pour les volumes SMB |
| `unixPermissions` | Mode pour les nouveaux volumes. **Doit rester vide pour les volumes SMB.** | "" |

#### Activer le SMB sécurisé

À partir de la version 25.06, NetApp Trident prend en charge le provisionnement sécurisé des volumes SMB créés à l'aide de `ontap-nas` et `ontap-nas-economy` serveurs backend. Lorsque le protocole SMB sécurisé est activé, vous pouvez fournir un accès contrôlé aux partages SMB pour les utilisateurs et groupes d'utilisateurs Active Directory (AD) à l'aide des listes de contrôle d'accès (ACL).

**Points à retenir**

- L'importation `ontap-nas-economy` de volumes n'est pas prise en charge.
- Seuls les clones en lecture seule sont pris en charge pour `ontap-nas-economy` volumes.
- Si le protocole SMB sécurisé est activé, Trident ignorera le partage SMB mentionné dans le backend.
- La mise à jour de l'annotation PVC, de l'annotation de classe de stockage et du champ backend ne met pas à jour l'ACL du partage SMB.
- La liste de contrôle d'accès (ACL) de partage SMB spécifiée dans l'annotation du PVC cloné aura priorité sur celles du PVC source.
- Veillez à fournir des utilisateurs AD valides lors de l'activation du protocole SMB sécurisé. Les utilisateurs non valides ne seront pas ajoutés à l'ACL.
- Si vous fournissez le même utilisateur AD dans le backend, la storage class et le PVC avec des autorisations différentes, la priorité des autorisations sera : PVC, storage class, puis backend.
- Le protocole SMB sécurisé est pris en charge pour `ontap-nas` les importations de volumes gérés et n'est pas applicable aux importations de volumes non gérés.

**Étapes**

1. Spécifiez adAdminUser dans TridentBackendConfig comme indiqué dans l'exemple suivant :

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-ontap
  namespace: trident
spec:
  version: 1
  storageDriverName: ontap-nas
  managementLIF: 10.193.176.x
  svm: svm0
  useREST: true
  defaults:
    adAdminUser: tridentADtest
  credentials:
    name: backend-tbc-ontap-invest-secret
```

2. Ajoutez l'annotation dans la classe de stockage.

   Ajoutez l' `trident.netapp.io/smbShareAdUser`annotation à la classe de stockage pour activer SMB sécurisé sans faute. La valeur utilisateur spécifiée pour l'annotation `trident.netapp.io/smbShareAdUser` doit être identique au nom d'utilisateur indiqué dans le `smbcreds` secret. Vous pouvez choisir l'une des options suivantes pour `smbShareAdUserPermission` : `full_control`, `change`, ou `read`. L'autorisation par défaut est `full_control`.

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontap-smb-sc
  annotations:
    trident.netapp.io/smbShareAdUserPermission: change
    trident.netapp.io/smbShareAdUser: tridentADuser
parameters:
  backendType: ontap-nas
  csi.storage.k8s.io/node-stage-secret-name: smbcreds
  csi.storage.k8s.io/node-stage-secret-namespace: trident
  trident.netapp.io/nasType: smb
provisioner: csi.trident.netapp.io
reclaimPolicy: Delete
volumeBindingMode: Immediate
```

1. Créer un PVC.

L'exemple suivant crée un PVC :

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: my-pvc4
  namespace: trident
  annotations:
    trident.netapp.io/snapshotDirectory: "true"
    trident.netapp.io/smbShareAccessControl: |
      read:
        - tridentADtest
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: ontap-smb-sc
```

## Options et exemples de configuration NAS ONTAP

Apprenez à créer et à utiliser des pilotes NAS ONTAP avec votre installation Trident. Cette section fournit des exemples de configuration backend et des détails pour le mappage des backends à StorageClasses. À compter de la version 25.10, NetApp Trident prend en charge "NetApp AFX systèmes de stockage". NetApp AFX les systèmes de stockage diffèrent des autres systèmes basés sur ONTAP (ASA, AFF et FAS) dans l'implémentation de leur couche de stockage.

| | |
|---|---|
| **REMARQUE** | Seul le `ontap-nas` driver (avec le protocole NFS) est pris en charge pour les systèmes NetApp AFX ; le protocole SMB n'est pas pris en charge. |

### Options de configuration du backend

Dans la configuration du backend Trident, il n'est pas nécessaire de préciser que votre système est un NetApp AFX storage system. Lorsque vous sélectionnez `ontap-nas` comme `storageDriverName`, Trident détecte automatiquement le système de stockage AFX. Certains paramètres de configuration du backend ne sont pas applicables aux systèmes de stockage AFX.

Le tableau suivant présente les options de configuration du backend :

| Paramètre | Description | Défaut |
|---|---|---|
| `version` | | Toujours 1 |

| Paramètre | Description | Défaut |
| --- | --- | --- |
| `storageDriverName` | Nom du pilote de stockage<br>**REMARQUE** Pour les systèmes NetApp AFX, seul `ontap-nas` est pris en charge. | `ontap-nas`, `ontap-nas-economy`, ou `ontap-nas-flexgroup` |
| `backendName` | Nom personnalisé ou le stockage backend | Nom du pilote + "_" + dataLIF |
| `managementLIF` | Adresse IP d'un cluster ou d'une LIF de gestion SVM. Un nom de domaine complet (FQDN) peut être spécifié. Il est possible de configurer l'utilisation d'adresses IPv6 si Trident a été installé avec l'option IPv6. Les adresses IPv6 doivent être définies entre crochets, comme `[28e8:d9fb:a825:b7bf:69a8:d02f:9e7b:3555]`. Pour une transition en douceur MetroCluster, consultez le Exemple MetroCluster. | "10.0.0.1", "[2001:1234:abcd::fefe]" |
| `dataLIF` | Adresse IP de l'interface logique de protocole (LIF). NetApp recommande de spécifier `dataLIF`. Si elle n'est pas fournie, Trident récupère les dataLIF depuis la 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 à répartition de charge (round-robin) entre plusieurs dataLIF. Peut être modifié après la configuration initiale. Consultez . Il est possible de configurer l'utilisation d'adresses IPv6 si Trident a été installé avec l'option IPv6. Les adresses IPv6 doivent être définies entre crochets, comme `[28e8:d9fb:a825:b7bf:69a8:d02f:9e7b:3555]`. **Omettre pour MetroCluster.** Voir le Exemple MetroCluster. | Adresse spécifiée ou dérivée de la SVM, si non spécifiée (non recommandé) |
| `svm` | Machine virtuelle de stockage à utiliser **Omettre pour MetroCluster.** Voir le Exemple MetroCluster. | Dérivé si un SVM `managementLIF` est spécifié |
| `autoExportPolicy` | Activer la création et la mise à jour automatiques des règles d'export [Booléen]. En utilisant les options `autoExportPolicy` et `autoExportCIDRs`, Trident peut gérer les règles d'export automatiquement. | false |
| `autoExportCIDRs` | Liste des CIDR pour filtrer les adresses IP des nœuds Kubernetes lorsque `autoExportPolicy` est activé. En utilisant les options `autoExportPolicy` et `autoExportCIDRs`, Trident peut gérer les règles d'export automatiquement. | ["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 | Défaut |
| --- | --- | --- |
| `clientPrivateKey` | Valeur encodée en Base64 de la clé privée du client. Utilisé pour l'authentification par certificat | "" |
| `trustedCACertificate` | Valeur encodée en Base64 du certificat d'autorité de certification de confiance. Facultatif. Utilisé pour l'authentification par certificat | "" |
| `username` | Nom d'utilisateur pour se connecter au cluster/SVM. Utilisé pour l'authentification par identifiants. Pour l'authentification Active Directory, voir "Authentifier Trident auprès d'un SVM backend à l'aide des identifiants Active Directory". | |
| `password` | Mot de passe pour se connecter au cluster/SVM. Utilisé pour l'authentification par identifiants. Pour l'authentification Active Directory, voir "Authentifier Trident auprès d'un SVM backend à l'aide des identifiants Active Directory". | |
| `storagePrefix` | Préfixe utilisé lors du provisionnement de nouveaux volumes dans la SVM. Ne peut pas être modifié après sa définition<br>**REMARQUE** : Lors de l'utilisation d'ontap-nas-economy et d'un storagePrefix de 24 caractères ou plus, les qtrees n'auront pas le préfixe de stockage intégré, bien qu'il soit présent dans le nom du volume. | "Trident" |

| Paramètre | Description | Défaut |
| --- | --- | --- |
| `aggregate` | Agrégat pour le provisionnement (facultatif ; s'il est défini, il doit être attribué à la SVM). Pour le `ontap-nas-flexgroup` driver, cette option est ignorée. S'il n'est pas attribué, n'importe quel agrégat disponible peut être utilisé pour provisionner un FlexGroup volume.<br>**REMARQUE** : Lorsqu'un agrégat est mis à jour dans SVM, il est automatiquement mis à jour dans Trident par interrogation de SVM, sans qu'il soit nécessaire de redémarrer le contrôleur Trident. Si vous avez configuré un agrégat spécifique dans Trident pour provisionner des volumes, si l'agrégat est renommé ou déplacé hors de la SVM, le backend passera en état d'échec dans Trident lors de l'interrogation de l'agrégat SVM. Vous devez soit changer l'agrégat pour un qui est présent sur la SVM, soit le supprimer complètement pour remettre le backend en ligne.<br>**Ne pas spécifier pour les systèmes de stockage AFX.** | "" |
| `limitAggregateUsage` | L'approvisionnement échoue si l'utilisation dépasse ce pourcentage. **Ne s'applique pas à Amazon FSx pour ONTAP**. **Ne pas spécifier pour les systèmes de stockage AFX.** | "" (non appliqué par défaut) |

| Paramètre | Description | Défaut |
| --- | --- | --- |
| flexgroupAggregateList | Liste des agrégats à provisionner (facultatif ; si défini, doit être affecté au SVM). Tous les agrégats affectés au SVM sont utilisés pour provisionner un FlexGroup volume. Pris en charge par le pilote de stockage **ontap-nas-flexgroup**.<br>**REMARQUE** : Lorsque la liste d'agrégats est mise à jour dans SVM, la liste est mise à jour automatiquement dans Trident par interrogation de SVM, sans qu'il soit nécessaire de redémarrer le contrôleur Trident. Lorsque vous avez configuré une liste d'agrégats spécifique dans Trident pour le provisionnement de volumes, si la liste d'agrégats est renommée ou déplacée hors de SVM, le backend passera à l'état d'échec dans Trident lors de l'interrogation de l'agrégat SVM. Vous devez soit modifier la liste d'agrégats pour en choisir une présente sur le SVM, soit la supprimer complètement afin de remettre le backend en ligne. | "" |
| `limitVolumeSize` | L'approvisionnement échoue si la taille du volume demandée dépasse cette valeur. | "" (non appliqué par défaut) |
| `debugTraceFlags` | Options de débogage à utiliser lors du dépannage. Exemple, {"api":false, "method":true} Ne pas utiliser `debugTraceFlags` sauf si vous effectuez un dépannage et avez besoin d'un journal détaillé. | null |
| `nasType` | Configurez la création de volumes NFS ou SMB. Les options sont `nfs`, `smb` ou null. La valeur null correspond par défaut à des volumes NFS. **Si spécifié, définissez toujours sur `nfs` pour les systèmes de stockage AFX**. | `nfs` |

| Paramètre | Description | Défaut |
| --- | --- | --- |
| `nfsMountOptions` | Liste d'options de montage NFS séparées par des virgules. Les options de montage pour les volumes persistants Kubernetes sont généralement spécifiées dans les classes de stockage, mais si aucune option de montage n'est spécifiée dans une classe de stockage, Trident utilisera les options de montage spécifiées dans le fichier de configuration du backend de stockage. Si aucune option de montage n'est spécifiée ni dans la classe de stockage ni dans le fichier de configuration, Trident n'appliquera aucune option de montage au volume persistant associé. | "" |
| `qtreesPerFlexvol` | Nombre maximal de Qtrees par FlexVol, 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 pour permettre à Trident de créer le partage SMB ; ou vous pouvez laisser le paramètre vide pour empêcher l'accès partagé aux volumes. Ce paramètre est facultatif pour ONTAP sur site. Ce paramètre est obligatoire pour Amazon FSx for ONTAP backends et ne peut pas être vide. | `smb-share` |
| `useREST` | Paramètre booléen à utiliser pour les API REST ONTAP. `useREST` Lorsqu'il est défini sur `true`, Trident utilise les API REST ONTAP pour communiquer avec le backend ; lorsqu'il est défini sur `false`, Trident utilise les appels ONTAPI (ZAPI) pour communiquer avec le backend. Cette fonctionnalité nécessite ONTAP 9.11.1 ou une version ultérieure. De plus, le rôle de connexion ONTAP utilisé doit avoir accès à l'application `ontapi`. Cela est satisfait par les rôles prédéfinis `vsadmin` et `cluster-admin`. À partir de la version Trident 24.06 et ONTAP 9.15.1 ou version ultérieure, `useREST` est défini sur `true` par défaut ; modifiez `useREST` sur `false` pour utiliser les appels ONTAPI (ZAPI). **Si spécifié, définissez toujours sur `true` pour les systèmes de stockage AFX**. | `true` pour ONTAP 9.15.1 ou version ultérieure, sinon `false`. |
| `limitVolumePoolSize` | Taille maximale requise de FlexVol lors de l'utilisation de Qtrees dans le backend ontap-nas-economy. | "" (non appliqué par défaut) |
| `denyNewVolumePools` | Empêche `ontap-nas-economy` les serveurs backend de créer de nouveaux volumes FlexVol pour y stocker leurs Qtrees. Seuls les FlexVol® préexistants sont utilisés pour le provisionnement de nouveaux PV. | |

| Paramètre | Description | Défaut |
|---|---|---|
| adAdminUser | Utilisateur ou groupe d'utilisateurs administrateur Active Directory disposant d'un accès complet aux partages SMB. Utilisez ce paramètre pour accorder des droits d'administrateur sur le partage SMB avec un contrôle total. | |

### Options de configuration backend pour le provisionnement des volumes

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

| Paramètre | Description | Défaut |
|---|---|---|
| spaceAllocation | Allocation d'espace pour les Qtrees | "true" |
| spaceReserve | Mode de réservation d'espace ; "none" (fin) ou "volume" (épais) | "none" |
| snapshotPolicy | Stratégie de snapshot à utiliser | "none" |
| qosPolicy | Groupe de règles QoS à attribuer aux volumes créés. Choisissez l'un des deux, qosPolicy ou adaptiveQosPolicy, par pool de stockage/backend | "" |
| adaptiveQosPolicy | Groupe de règles QoS adaptatives à attribuer aux volumes créés. Choisissez l'un de qosPolicy ou adaptiveQosPolicy par pool de stockage/backend. Non pris en charge par ontap-nas-economy. | "" |
| snapshotReserve | Pourcentage du volume réservé aux instantanés | "0" si `snapshotPolicy` est "aucun", sinon "" |
| 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`. NVE doit être sous licence et activé sur le cluster pour utiliser cette option. Si NAE est activé sur le backend, tout volume provisionné dans Trident sera activé pour NAE. Pour plus d'informations, consultez : "Comment Trident fonctionne avec NVE et NAE". | "false" |
| tieringPolicy | Politique de hiérarchisation à utiliser « none » | |
| unixPermissions | Mode pour les nouveaux volumes | "777" pour les volumes NFS; vide (non applicable) pour les volumes SMB |
| snapshotDir | Contrôle l'accès au `.snapshot` répertoire | `true`, `false` (Défini explicitement). |
| exportPolicy | Règles d'export à utiliser | "default" |

| Paramètre | Description | Défaut |
|---|---|---|
| `securityStyle` | Style de sécurité pour les nouveaux volumes. NFS prend en charge `mixed` et `unix` styles de sécurité. SMB prend en charge `mixed` et `ntfs` styles de sécurité. | La valeur par défaut de NFS est `unix`. La valeur par défaut de SMB est `ntfs`. |
| `nameTemplate` | Modèle pour créer des noms de volumes personnalisés. | "" |

| | |
|---|---|
| **REMARQUE** | L'utilisation des groupes de règles QoS avec Trident requiert 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 individuellement à chaque composant. Un groupe de règles QoS partagé impose une limite au 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 désormais un nouveau calcul afin de garantir que le FlexVol soit correctement dimensionné avec le pourcentage de snapshotReserve et le PVC. Lorsque l'utilisateur demande un PVC, Trident crée le FlexVol initial avec plus d'espace en utilisant le nouveau calcul.

Ce calcul garantit que l'utilisateur reçoit l'espace inscriptible qu'il a demandé dans le PVC, et non moins que ce qu'il a demandé. Avant la version v21.07, lorsque l'utilisateur demandait un PVC (par exemple, 5 Gio), avec le snapshotReserve à 50 pour cent, il n'obtenait que 2,5 Gio d'espace inscriptible. Cela s'explique par le fait que ce que l'utilisateur demandait était le volume entier et `snapshotReserve` est un pourcentage de celui-ci. Avec Trident 21.07, ce que l'utilisateur demande est l'espace inscriptible et Trident définit le nombre `snapshotReserve` comme le pourcentage du volume entier. Cela ne s'applique pas à `ontap-nas-economy`. Voir l'exemple suivant pour comprendre comment cela fonctionne :

Le calcul est le suivant :

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

Pour snapshotReserve = 50 %, et une demande PVC = 5 Gio, la taille totale du volume est de 5 / 0,5 = 10 Gio et la taille disponible est de 5 Gio, ce qui correspond à la demande de l'utilisateur dans la demande PVC. La commande `volume show` devrait afficher des résultats similaires à cet exemple :

Les backends existants issus d'installations précédentes provisionneront 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 pour que la modification soit prise en compte. Par exemple, un PVC de 2 Gio avec `snapshotReserve=50` aboutissait auparavant à un volume offrant 1 Gio d'espace accessible en écriture. En redimensionnant le volume à 3 Gio, par exemple, l'application disposera de 3 Gio d'espace accessible en écriture sur un volume de 6 Gio.

**Exemples de configuration minimale**

Les exemples suivants présentent des configurations de base qui laissent la plupart des paramètres par défaut. C'est la manière la plus simple de définir un backend.

| | |
|---|---|
| **REMARQUE** | Si vous utilisez Amazon FSx sur NetApp ONTAP avec Trident, la recommandation est de spécifier les noms DNS des LIF au lieu des 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 backend pour éviter d'avoir à mettre à jour manuellement la définition du backend après le basculement et le retour en arrière pendant "Réplication et récupération de SVM".

Pour une transition et un retour en arrière sans heurt, spécifiez le SVM en utilisant `managementLIF` et omettez les `dataLIF` et `svm` paramètres. 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 par certificat

Voici un exemple de configuration minimale du backend. `clientCertificate`, `clientPrivateKey`, et `trustedCACertificate` (facultatif, si vous utilisez une autorité de certification de confiance) sont renseignés dans `backend.json` et prennent respectivement les valeurs encodées en base64 du certificat client, de la clé privée et du certificat de l'autorité de certification de confiance.

```
---
version: 1
backendName: DefaultNASBackend
storageDriverName: ontap-nas
managementLIF: 10.0.0.1
dataLIF: 10.0.0.15
svm: nfs_svm
clientCertificate: ZXR0ZXJwYXB...ICMgJ3BhcGVyc2
clientPrivateKey: vciwKIyAgZG...0cnksIGRlc2NyaX
trustedCACertificate: zcyBbaG...b3Igb3duIGNsYXNz
storagePrefix: myPrefix_
```

#### Exemple de règles d'export automatique

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

```
---
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 montre `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'utilisation d'Amazon FSx for 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: ZXR0ZXJwYXB...ICMgJ3BhcGVyc2
clientPrivateKey: vciwKIyAgZG...0cnksIGRlc2NyaX
trustedCACertificate: zcyBbaG...b3Igb3duIGNsYXNz
storagePrefix: myPrefix_
```

#### Exemple de configuration du backend avec nameTemplate

```
---
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 backends avec pools virtuels

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

Trident définit les étiquettes de provisionnement dans le champ « Commentaires ». Les commentaires sont définis sur FlexVol pour `ontap-nas` ou sur 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 simplicité, les administrateurs de stockage peuvent définir des étiquettes par pool virtuel et regrouper les volumes par étiquette.

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

**Exemple 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 NAS ONTAP FlexGroup**

```
---
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"
```

### Map backends vers StorageClasses

Les définitions de StorageClass suivantes font référence à Exemples de backends avec pools virtuels. En utilisant le champ `parameters.selector`, chaque StorageClass indique quels pools virtuels peuvent être utilisés pour héberger un volume. Le volume possédera les aspects définis dans le pool virtuel choisi.

- Le `protection-gold` StorageClass sera associé au premier et au deuxième pool virtuel dans le `ontap-nas-flexgroup` backend. Ce sont les seuls pools offrant une protection de niveau or.

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

- Le `protection-not-gold` StorageClass sera associé au troisième et au quatrième pool virtuel dans le `ontap-nas-flexgroup` backend. 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"
```

- Le `app-mysqldb` StorageClass correspondra au quatrième pool virtuel dans le `ontap-nas` backend. Il s'agit du seul pool offrant une configuration de pool de stockage pour les applications 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"
```

- Le `protection-silver-creditpoints-20k` StorageClass sera associé au troisième pool virtuel dans le `ontap-nas-flexgroup` backend. Il s'agit du seul pool 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"
```

- Le `creditpoints-5k` StorageClass correspondra au troisième pool virtuel dans le `ontap-nas` backend et au deuxième pool virtuel dans le `ontap-nas-economy` backend. Ce sont les seuls pools proposant 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 déterminera quel pool virtuel est sélectionné et s'assurera que l'exigence de stockage est respectée.

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

Vous pouvez modifier le dataLIF après la configuration initiale en exécutant la commande suivante pour fournir le nouveau fichier JSON backend avec le dataLIF mis à jour.

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

| REMARQUE | Si des PVC sont connectés à un ou plusieurs pods, vous devez mettre hors service tous les pods correspondants, puis les remettre en service afin que la nouvelle dataLIF prenne effet. |
|---|---|

### Exemples de SMB sécurisés

#### Configuration du backend avec le pilote ontap-nas

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-ontap-nas
  namespace: trident
spec:
  version: 1
  storageDriverName: ontap-nas
  managementLIF: 10.0.0.1
  svm: svm2
  nasType: smb
  defaults:
    adAdminUser: tridentADtest
  credentials:
    name: backend-tbc-ontap-invest-secret
```

#### Configuration du backend avec le pilote ontap-nas-economy

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-ontap-nas
  namespace: trident
spec:
  version: 1
  storageDriverName: ontap-nas-economy
  managementLIF: 10.0.0.1
  svm: svm2
  nasType: smb
  defaults:
    adAdminUser: tridentADtest
  credentials:
    name: backend-tbc-ontap-invest-secret
```

**Configuration du backend avec pool de stockage**

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-ontap-nas
  namespace: trident
spec:
  version: 1
  storageDriverName: ontap-nas
  managementLIF: 10.0.0.1
  svm: svm0
  useREST: false
  storage:
  - labels:
      app: msoffice
    defaults:
      adAdminUser: tridentADuser
  nasType: smb
  credentials:
    name: backend-tbc-ontap-invest-secret
```

**Exemple de classe de stockage avec le pilote ontap-nas**

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontap-smb-sc
  annotations:
    trident.netapp.io/smbShareAdUserPermission: change
    trident.netapp.io/smbShareAdUser: tridentADtest
parameters:
  backendType: ontap-nas
  csi.storage.k8s.io/node-stage-secret-name: smbcreds
  csi.storage.k8s.io/node-stage-secret-namespace: trident
  trident.netapp.io/nasType: smb
provisioner: csi.trident.netapp.io
reclaimPolicy: Delete
volumeBindingMode: Immediate
```

| REMARQUE | Assurez-vous d'ajouter `annotations` pour activer le SMB sécurisé. Le SMB sécurisé ne fonctionne pas sans les annotations, quelles que soient les configurations définies dans le Backend ou le PVC. |
|---|---|

**Exemple de classe de stockage avec le pilote ontap-nas-economy**

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontap-smb-sc
  annotations:
    trident.netapp.io/smbShareAdUserPermission: change
    trident.netapp.io/smbShareAdUser: tridentADuser3
parameters:
  backendType: ontap-nas-economy
  csi.storage.k8s.io/node-stage-secret-name: smbcreds
  csi.storage.k8s.io/node-stage-secret-namespace: trident
  trident.netapp.io/nasType: smb
provisioner: csi.trident.netapp.io
reclaimPolicy: Delete
volumeBindingMode: Immediate
```

**Exemple de PVC avec un seul utilisateur AD**

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: my-pvc4
  namespace: trident
  annotations:
    trident.netapp.io/smbShareAccessControl: |
      change:
        - tridentADtest
      read:
        - tridentADuser
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: ontap-smb-sc
```

**Exemple de PVC avec plusieurs utilisateurs AD**

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: my-test-pvc
  annotations:
    trident.netapp.io/smbShareAccessControl: |
      full_control:
        - tridentTestuser
        - tridentuser
        - tridentTestuser1
        - tridentuser1
      change:
        - tridentADuser
        - tridentADuser1
        - tridentADuser4
        - tridentTestuser2
      read:
        - tridentTestuser2
        - tridentTestuser3
        - tridentADuser2
        - tridentADuser3
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
```

# Amazon FSx for NetApp ONTAP

### Utilisez Trident avec Amazon FSx for NetApp ONTAP

"Amazon FSx for NetApp ONTAP" est un service AWS entièrement géré qui exécute des systèmes de fichiers alimentés par le système d'exploitation de stockage NetApp ONTAP. Il fournit les fonctionnalités, les performances et l'administration d'ONTAP avec l'évolutivité et la simplicité opérationnelle d'AWS. Un système de fichiers est la ressource principale dans Amazon FSx et est analogue à un cluster ONTAP sur site. Chaque système de fichiers contient une ou plusieurs machines virtuelles de stockage (SVM), et chaque SVM contient un ou plusieurs volumes qui stockent des fichiers et des répertoires. Cette intégration permet aux clusters Kubernetes exécutés dans Amazon Elastic Kubernetes Service (EKS) de provisionner des volumes persistants basés sur ONTAP pour les charges de travail bloc et fichier.

### Exigences

En plus de "Exigences de Trident", pour intégrer FSx for ONTAP avec Trident, vous avez besoin de :

- Un cluster Amazon EKS existant ou un cluster Kubernetes autogéré avec `kubectl` installé.
- Un système de fichiers Amazon FSx for NetApp ONTAP existant et une machine virtuelle de stockage (SVM) accessibles depuis les nœuds de travail de votre cluster.
- Nœuds de travail préparés pour "NFS ou iSCSI".

| | |
|---|---|
| **REMARQUE** | Assurez-vous de suivre les étapes de préparation des nœuds requises pour Amazon Linux et Ubuntu "Amazon Machine Images" (AMIs) en fonction de votre type d'AMI EKS. |

### Considérations

- Volumes SMB :
  - Les volumes SMB sont pris en charge à l'aide du pilote `ontap-nas` uniquement.
  - Les volumes SMB ne sont pas pris en charge avec l'add-on Trident EKS.
  - Trident prend uniquement en charge les volumes SMB montés sur des pods exécutés sur des nœuds Windows. Consultez "Préparez-vous à provisionner des volumes SMB" pour plus de détails.
- Avant Trident 24.02, les volumes créés sur les systèmes de fichiers Amazon FSx avec les sauvegardes automatiques activées ne pouvaient pas être supprimés par Trident. Pour éviter ce problème dans Trident 24.02 ou version ultérieure, spécifiez le `fsxFilesystemID`, AWS `apiRegion`, AWS `apikey`, et AWS `secretKey` dans le fichier de configuration du backend pour AWS FSx for ONTAP.

| | |
|---|---|
| **REMARQUE** | Si vous spécifiez un rôle IAM à Trident, vous pouvez omettre de spécifier les champs `apiRegion`, `apiKey`, et `secretKey` à Trident explicitement. Pour plus d'informations, consultez "Options de configuration et exemples pour FSx for ONTAP". |

#### Utilisation simultanée de Trident SAN/iSCSI et du pilote EBS-CSI

Si vous prévoyez d'utiliser des pilotes ontap-san (par exemple, iSCSI) avec AWS (EKS, ROSA, EC2 ou toute autre instance), la configuration multipath requise sur les nœuds peut entrer en conflit avec le pilote CSI Amazon Elastic Block Store (EBS). Pour garantir que le multipath fonctionne sans interférer avec les disques EBS sur le même nœud, vous devez exclure EBS de votre configuration multipath. Cet exemple montre un fichier `multipath.conf` qui inclut les paramètres Trident requis tout en excluant les disques EBS du multipath :

```
defaults {
    find_multipaths no
}
blacklist {
    device {
        vendor "NVME"
        product "Amazon Elastic Block Store"
    }
}
```

### Authentification

Trident propose deux modes d'authentification.

- Authentification par identifiants (recommandée) : stocke les identifiants en toute sécurité dans AWS Secrets Manager. Vous pouvez utiliser l `fsxadmin` utilisateur de votre système de fichiers ou l `vsadmin` utilisateur configuré pour votre SVM.

| | |
|---|---|
| **AVERTISSEMENT** | Trident s'attend à être exécuté en tant qu' `vsadmin`utilisateur SVM ou en tant qu'utilisateur portant un nom différent mais ayant le même rôle. Amazon FSx for NetApp ONTAP dispose d'un `fsxadmin` utilisateur qui est un remplacement limité de l'utilisateur du cluster ONTAP `admin`. Nous recommandons fortement d'utiliser `vsadmin` avec Trident. |

- Basé sur un certificat : Trident communiquera avec la SVM de 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 l'authentification pour votre type de pilote :

- "Authentification NAS ONTAP"
- "Authentification SAN ONTAP"

### Amazon Machine Images (AMIs) testées

Le cluster EKS prend en charge différents systèmes d'exploitation, mais AWS a optimisé certaines images machine Amazon (AMIs) pour les conteneurs et EKS. Les AMIs suivantes ont été testées avec NetApp Trident 25.02.

| AMI | NAS | NAS-economy | iSCSI | iSCSI-economy |
|---|---|---|---|---|
| AL2023_x86_64_STANDARD | 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_ARM_64 | Oui** | Oui | S/O | S/O |

- * Impossible de supprimer le PV sans redémarrer le nœud
- ** Ne fonctionne pas avec NFSv3 avec Trident version 25.02.

| | |
|---|---|
| **REMARQUE** | Si l'AMI que vous recherchez ne figure pas dans cette liste, cela ne signifie pas qu'elle n'est pas prise en charge ; cela signifie simplement qu'elle n'a pas été testée. Cette liste sert de guide pour les AMI dont le fonctionnement est avéré. |

**Tests effectués avec**:

- EKS version : 1.32
- Méthode d'installation : Helm 25.06 et en tant qu'AWS add-On 25.06
- Pour NAS, les protocoles NFSv3 et NFSv4.1 ont été testés.
- Pour le SAN, seul l'iSCSI a été testé, pas le NVMe-oF.

**Tests effectués** :

- Créer : Storage Class, pvc, pod
- Supprimer : pod, pvc (standard, qtree/lun – économique, NAS avec sauvegarde AWS)

**Pour plus d'informations**

- "Documentation Amazon FSx for NetApp ONTAP"
- "Article de blog sur Amazon FSx for NetApp ONTAP"

## Créez un rôle IAM et un secret AWS

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

| | |
|---|---|
| **REMARQUE** | Pour vous authentifier à l'aide d'un rôle AWS IAM, vous devez disposer d'un cluster Kubernetes déployé à l'aide d'EKS. |

### Créer un secret AWS Secrets Manager

Étant donné que Trident utilisera des API sur un vserver FSx pour gérer le stockage pour vous, il aura besoin d'identifiants pour le faire. La méthode sécurisée pour transmettre ces identifiants consiste à utiliser un secret AWS Secrets Manager. Par conséquent, si vous n'en possédez pas déjà un, vous devrez créer un secret AWS Secrets Manager contenant les identifiants 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"\
  --secret-string
"{\"username\":\"vsadmin\",\"password\":\"<svmpassword>\"}"
```

**Créer une stratégie IAM**

Trident a également besoin des autorisations AWS pour fonctionner correctement. Par conséquent, vous devez créer une stratégie qui donne à Trident les autorisations dont il a besoin.

Les exemples suivants créent une stratégie IAM à l'aide de l'AWS CLI :

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

**Exemple de JSON de stratégie** :

```
{
  "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 une identité de pod ou un rôle IAM pour l'association du compte de service (IRSA)

Vous pouvez configurer un compte de service Kubernetes pour qu'il assume un rôle AWS Identity and Access Management (IAM) avec EKS Pod Identity ou IAM role for Service account association (IRSA). Tous les Pods configurés pour utiliser le compte de service peuvent alors accéder à tout service AWS auquel le rôle a des autorisations d'accès.

**Identité du pod**

Les associations d'identité de pod Amazon EKS offrent la possibilité de gérer les informations d'identification de vos applications, de la même manière que les profils d'instance Amazon EC2 fournissent des informations d'identification aux instances Amazon EC2.

**Installez Pod Identity sur votre cluster EKS** :

Vous pouvez créer une identité de pod via la console AWS ou en utilisant la commande AWS CLI suivante :

```
aws eks create-addon --cluster-name <EKS_CLUSTER_NAME> --addon-name
eks-pod-identity-agent
```

Pour plus d'informations, consultez "Configurer l'agent d'identité du pod Amazon EKS".

**Créer trust-relationship.json**:

Créez trust-relationship.json pour permettre au Service Principal EKS d'assumer ce rôle pour l'identité du pod. Créez ensuite un rôle avec cette stratégie d'approbation :

```
aws iam create-role \
  --role-name fsxn-csi-role --assume-role-policy-document file://trust-
relationship.json \
  --description "fsxn csi pod identity role"
```

**Fichier trust-relationship.json**:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "pods.eks.amazonaws.com"
      },
      "Action": [
        "sts:AssumeRole",
        "sts:TagSession"
      ]
    }
  ]
}
```

**Associez la stratégie de rôle au rôle IAM**:

Associez la stratégie de rôle de l'étape précédente au rôle IAM qui a été créé :

```
aws iam attach-role-policy \
  --policy-arn arn:aws:iam::aws:111122223333:policy/fsxn-csi-policy \
  --role-name fsxn-csi-role
```

**Créer une association d'identité de pod** :

Créer une association d'identité de pod entre le rôle IAM et le compte de service Trident (trident-controller)

```
aws eks create-pod-identity-association \
  --cluster-name <EKS_CLUSTER_NAME> \
  --role-arn arn:aws:iam::111122223333:role/fsxn-csi-role \
  --namespace trident --service-account trident-controller
```

**Rôle IAM pour l'association de compte de service (IRSA)**

**Utilisation de l'interface de ligne de commande 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 fichier `trust-relationship.json` :

- **<account_id>** - Votre ID de compte AWS
- **<oidc_provider>** - L'OIDC de votre cluster EKS. Vous pouvez obtenir le oidc_provider en exécutant :

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

**Associez le rôle IAM à la stratégie IAM**:

Une fois le rôle créé, associez la stratégie (qui a été créée à l'étape ci-dessus) au rôle à l'aide de cette commande :

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

**Vérifiez que le fournisseur OICD est associé**:

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

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

Si le résultat est vide, utilisez la commande suivante pour associer IAM OIDC à votre cluster :

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

**Si vous utilisez eksctl**, utilisez l'exemple suivant pour créer 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
```

## Installer Trident

Trident simplifie la gestion du stockage Amazon FSx for NetApp ONTAP dans Kubernetes afin de permettre à vos développeurs et administrateurs de se concentrer sur le déploiement des applications. Vous pouvez installer Trident en utilisant l'une des méthodes suivantes :

- Helm
- Module complémentaire EKS

Si vous souhaitez utiliser la fonctionnalité de snapshot, installez l'add-on CSI snapshot controller. Consultez "Activer la fonctionnalité de snapshot pour les volumes CSI" pour plus d'informations.

#### Installez Trident via helm

**Identité du pod**

1. Ajoutez le dépôt Trident Helm :

```
helm repo add netapp-trident https://netapp.github.io/trident-helm-chart
```

2. Installez Trident en utilisant l'exemple suivant :

```
helm install trident-operator netapp-trident/trident-operator --version 100.2502.1 --namespace trident --create-namespace
```

Vous pouvez utiliser la commande `helm list` pour consulter les détails de l'installation tels que le nom, l'espace de noms, le chart, 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
```

**Association de compte de service (IRSA)**

1. Ajoutez le dépôt Trident Helm :

```
helm repo add netapp-trident https://netapp.github.io/trident-helm-chart
```

2. Définissez les valeurs pour **fournisseur de cloud** et **cloud identity** :

```
helm install trident-operator netapp-trident/trident-operator
--version 100.2502.1  \
--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 la commande `helm list` pour consulter les détails de l'installation tels que le nom, l'espace de noms, le chart, 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-
100.2510.0    25.10.0
```

**REMARQUE**

Si vous prévoyez d'utiliser iSCSI, assurez-vous que iSCSI est activé sur votre machine cliente. Si vous utilisez AL2023 Worker node OS, vous pouvez automatiser l'installation du client iSCSI en ajoutant le paramètre node prep lors de l'installation avec helm :

```
helm install trident-operator netapp-trident/trident-
operator --version 100.2502.1 --namespace trident --create
-namespace --set nodePrep={iscsi}
```

### Installez Trident via le module complémentaire EKS

L'add-on Trident EKS inclut les derniers correctifs de sécurité, les corrections de bogues et est validé par AWS pour fonctionner avec Amazon EKS. L'add-on EKS vous permet de garantir de manière cohérente que vos clusters Amazon EKS sont sécurisés et stables, et de réduire la quantité de travail nécessaire pour installer, configurer et mettre à jour les add-ons.

#### Prérequis

Assurez-vous de disposer des éléments suivants avant de configurer le module complémentaire Trident pour AWS EKS :

- Un compte de cluster Amazon EKS avec abonnement complémentaire
- Autorisations AWS pour la AWS marketplace :
  `"aws-marketplace:ViewSubscriptions",`
  `"aws-marketplace:Subscribe",`
  `"aws-marketplace:Unsubscribe`
- Type d'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 for NetApp ONTAP existant

#### Activez le module complémentaire Trident pour AWS

**Console de gestion**

1. Ouvrez la console Amazon EKS à 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 **Modules complémentaires** puis sélectionnez **Obtenir plus de modules complémentaires**.
5. Suivez ces étapes pour sélectionner le module complémentaire :
   a. Faites défiler vers le bas jusqu'à la section **AWS Marketplace add-ons** et tapez **"Trident"** dans la zone de recherche.
   b. Cochez la case à cocher située dans le coin supérieur droit de la boîte Trident by NetApp.
   c. Sélectionnez **Next**.
6. Sur la page des paramètres **Configure selected add-ons**, procédez comme suit :

   **REMARQUE** | **Ignorez ces étapes si vous utilisez l'association Pod Identity.**

   a. Sélectionnez la **Version** que vous souhaitez utiliser.
   b. Si vous utilisez l'authentification IRSA, assurez-vous de définir les valeurs de configuration disponibles dans les paramètres de configuration optionnels :
      - Sélectionnez la **Version** que vous souhaitez utiliser.
      - Suivez le **schéma de configuration du module complémentaire** et définissez le paramètre **configurationValues** dans la section **Valeurs de configuration** sur le role-arn que vous avez créé à l'étape précédente (la valeur doit être au format suivant) :

```
{

  "cloudIdentity": "'eks.amazonaws.com/role-arn: <role ARN>'",
  "cloudProvider": "AWS"

}
```

   +
   Si vous sélectionnez Override comme méthode de résolution des conflits, un ou plusieurs des paramètres de l'add-on existant peuvent être remplacés par les paramètres de l'add-on 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 résultant pour résoudre le conflit. Avant de sélectionner cette option, assurez-vous que l'add-on Amazon EKS ne gère pas des paramètres que vous devez gérer vous-même.

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

   Une fois l'installation du logiciel complémentaire terminée, vous voyez votre logiciel complémentaire installé.

**AWS CLI**

**1. Créez le `add-on.json` fichier** :

**Pour l'identité du pod, utilisez le format suivant**:

**REMARQUE** Utilisez le

```
{
  "clusterName": "<eks-cluster>",
  "addonName": "netapp_trident-operator",
  "addonVersion": "v25.6.0-eksbuild.1",
}
```

**Pour l'authentification IRSA, utilisez le format suivant**:

```
{
  "clusterName": "<eks-cluster>",
  "addonName": "netapp_trident-operator",
  "addonVersion": "v25.6.0-eksbuild.1",
  "serviceAccountRoleArn": "<role ARN>",
  "configurationValues": {
    "cloudIdentity": "'eks.amazonaws.com/role-arn: <role ARN>'",
    "cloudProvider": "AWS"
  }
}
```

**REMARQUE** Remplacez `<role ARN>` par l'ARN du rôle qui a été créé à l'étape précédente.

**2. Installez le logiciel complémentaire Trident EKS.**

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

**eksctl**

La commande suivante permet d'installer le module complémentaire Trident EKS :

```
eksctl create addon --name netapp_trident-operator --cluster
<cluster_name> --force
```

**Mettre à jour le module complémentaire Trident EKS**

**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 logiciel complémentaire NetApp Trident CSI.
4. Sélectionnez l'onglet **Add-ons**.
5. Sélectionnez **Trident par NetApp** puis sélectionnez **Modifier**.
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 optionnels** et modifiez-les si nécessaire.
   c. Sélectionnez **Save changes**.

**AWS CLI**

L'exemple suivant met à jour le add-on EKS :

```
aws eks update-addon --cluster-name <eks_cluster_name> --addon-name
netapp_trident-operator --addon-version v25.6.0-eksbuild.1 \
  --service-account-role-arn <role-ARN> --resolve-conflict preserve \
  --configuration-values "{\"cloudIdentity\":
\"'eks.amazonaws.com/role-arn: <role ARN>'\"}"
```

**eksctl**

- Vérifiez la version actuelle de votre add-on Trident CSI FSxN. 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    v25.6.0-eksbuild.1    ACTIVE    0
{"cloudIdentity":"'eks.amazonaws.com/role-arn:
arn:aws:iam::139763910815:role/AmazonEKS_FSXN_CSI_DriverRole'"}
```

- Mettez à jour le logiciel complémentaire avec la version renvoyée sous UPDATE AVAILABLE dans le résultat de l'étape précédente.

  ```
  eksctl update addon --name netapp_trident-operator --version
  v25.6.0-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 ; vous recevez un message d'erreur pour vous aider à résoudre le conflit. Avant de spécifier cette option, assurez-vous que le module complémentaire Amazon EKS ne gère pas de paramètres que vous devez gérer, car ces paramètres sont écrasés avec cette option. Pour plus d'informations sur les autres options de ce paramètre, consultez "Modules complémentaires". Pour plus d'informations sur la gestion des champs Kubernetes d'Amazon EKS, consultez "Gestion des champs Kubernetes".

#### Désinstallez/supprimez le module complémentaire Trident EKS

Vous avez deux options pour supprimer un add-on Amazon EKS :

- **Conserver le logiciel complémentaire sur votre cluster** – Cette option supprime la gestion de tous les paramètres par Amazon EKS. Elle supprime également la capacité d'Amazon EKS à vous notifier des mises à jour et à mettre à jour automatiquement le logiciel complémentaire Amazon EKS après que vous ayez initié une mise à jour. Cependant, elle conserve le logiciel complémentaire sur votre cluster. Cette option fait du logiciel complémentaire une installation autogérée, plutôt qu'un logiciel complémentaire Amazon EKS. Avec cette option, il n'y a pas d'interruption pour le logiciel complémentaire. Conservez l' `--preserve`option dans la commande pour préserver le logiciel complémentaire.
- **Supprimez entièrement le logiciel complémentaire de votre cluster** – NetApp recommande de supprimer le module complémentaire Amazon EKS de votre cluster uniquement s'il n'y a aucune ressource sur votre cluster qui en dépend. Supprimez l'option `--preserve` du `delete`commande pour supprimer le logiciel complémentaire.

| | |
|---|---|
| **REMARQUE** | Si le logiciel complémentaire a un compte IAM associé, le compte IAM n'est pas supprimé. |

**Console de gestion**

1. Ouvrez la console Amazon EKS à 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 logiciel complémentaire Trident CSI NetApp.
4. Sélectionnez l'onglet **Add-ons** puis sélectionnez **Trident par NetApp**.*
5. Sélectionnez **Remove**.
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 logiciel complémentaire, sélectionnez **Conserver sur le cluster**. Procédez ainsi si vous souhaitez conserver le logiciel complémentaire sur votre cluster afin de pouvoir gérer vous-même tous les paramètres du logiciel complémentaire.
   b. Saisissez **netapp_trident-operator**.
   c. Sélectionnez **Remove**.

**AWS CLI**

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
```

**eksctl**

La commande suivante désinstalle le module complémentaire Trident EKS :

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

## Configurer une classe de stockage

La "Objet Kubernetes StorageClass" identifie un provisionneur et lui indique comment provisionner les volumes. Cette section montre comment configurer un objet Kubernetes StorageClass qui spécifie Trident comme provisionneur.

### Créer un objet StorageClass

Lorsque vous créez un StorageClass pour FSx for ONTAP, Trident créera automatiquement la configuration backend.

| | |
|---|---|
| **REMARQUE** | Si vous souhaitez configurer manuellement le stockage backend, veuillez vous référer à la section [create-a-kubernetes-storageclass-without-automatic-backend-configuration] pour savoir comment créer séparément le backend Trident et la classe de stockage. |

#### Spécifiez les paramètres requis StorageClass

Les trois paramètres suivants doivent être définis lors de la création d'un StorageClass :

| Paramètre | Obligatoire | Type | Description |
|---|---|---|---|
| `fsxFilesystemID` | Oui | chaîne | ID du système de fichiers FSx for NetApp ONTAP |
| `storageDriverName` | Oui | chaîne | Pilote de stockage Trident (par exemple, `ontap-nas` ou `ontap-san`) |
| `credentialsName` | Oui | chaîne | Nom du secret Kubernetes qui contient les informations d'identification FSx for ONTAP |

#### Spécifiez les paramètres optionnels

Vous pouvez transmettre des paramètres backend optionnels via le StorageClass. Définissez toutes les valeurs optionnelles sous forme de chaînes dans la section StorageClass `parameters`. Pour une liste complète des paramètres du backend, voir : "Configuration du backend FSx for NetApp ONTAP".

#### Exemples de fichiers de configuration StorageClass.

L'exemple suivant montre un StorageClass qui déclenche une configuration automatique du backend.

**YAML**

```yaml
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontap-fsx-demo
  annotations:
    description: "Demo StorageClass for FSx for NetApp ONTAP"
provisioner: csi.trident.netapp.io
parameters:
  fsxFilesystemID: "fs-0abc123"
  storageDriverName: "ontap-nas"
  credentialsName: trident-fsx-credentials
allowVolumeExpansion: true
reclaimPolicy: Delete
volumeBindingMode: Immediate
```

**JSON**

```json
{
  "apiVersion": "storage.k8s.io/v1",
  "kind": "StorageClass",
  "metadata": {
    "name": "ontap-fsx-demo",
    "annotations": {
      "description": "Demo StorageClass for FSx for NetApp ONTAP"
    }
  },
  "provisioner": "csi.trident.netapp.io",
  "parameters": {
    "fsxFilesystemID": "fs-0abc123",
    "storageDriverName": "ontap-nas",
    "credentialsName": "trident-fsx-credentials"
  },
  "allowVolumeExpansion": true,
  "reclaimPolicy": "Delete",
  "volumeBindingMode": "Immediate"
}
```

#### Créez le StorageClass

Une fois que vous avez créé votre fichier de configuration, exécutez la commande suivante pour créer la classe de stockage.

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

Vous devriez maintenant voir une classe de stockage **basic-csi** dans Kubernetes et Trident, et Trident devrait avoir découvert les pools sur le backend.

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

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

Après avoir appliqué le StorageClass, Trident crée automatiquement le backend. Vous pouvez ensuite créer des PersistentVolumeClaims qui référencent ce StorageClass.

#### Vérifier l'état de la configuration du backend

Trident enregistre le résultat de la création du backend dans les annotations StorageClass.

| Annotation | Description |
|---|---|
| `trident.netapp.io/configuratorStatus` | Résultat de la configuration (`Success` ou `Failure`) |
| `trident.netapp.io/configuratorMessage` | Message d'état ou d'erreur détaillé |
| `trident.netapp.io/configuratorName` | Nom de la ressource du configurateur interne |
| `trident.netapp.io/managed` | Indique que le StorageClass est géré par Trident |
| `trident.netapp.io/additionalStoragePools` | Pools de stockage créés pour ce backend |

Pour vérifier l'état, exécutez :

```
kubectl get storageclass ontap-fsx-demo -o yaml
```

Confirmez que `trident.netapp.io/configuratorStatus` est défini sur `Success`. Si la valeur est `Failure`, examinez `trident.netapp.io/configuratorMessage` pour l'erreur.

#### Ajouter des systèmes de fichiers FSxN supplémentaires

Si vous avez besoin de stockage supplémentaire tout en continuant à utiliser le même StorageClass, ajoutez des identifiants de système de fichiers FSxN supplémentaires.

Modifiez le StorageClass et ajoutez l'annotation suivante :

```
metadata:
  annotations:
    trident.netapp.io/additionalFsxNFileSystemID: '["fs-
xxxxxxxxxxxxxxxxxx"]'
```

Une fois la modification appliquée, Trident met à jour la configuration du backend et met à jour les annotations StorageClass.

### Considérations et limitations opérationnelles

- La suppression d'un StorageClass qui utilise la configuration automatique du backend supprime généralement le backend Trident associé. Cela peut perturber la connectivité du stockage et interrompre les charges de travail en cours. Validez l'impact avant de supprimer un StorageClass géré.
- La configuration automatique du backend est prise en charge uniquement pour AWS FSx pour NetApp ONTAP.

#### Créer un StorageClass Kubernetes sans configuration automatique du backend

Si vous souhaitez créer le backend Trident et le StorageClass séparément, suivez ces étapes.

### Comprendre le fonctionnement de la configuration automatique du backend

Trident déduit la configuration du backend à partir de la définition de la StorageClass. Lorsque vous appliquez la StorageClass, Trident valide les paramètres requis, crée le backend et annote la StorageClass avec le statut.

Trident crée le VolumeSnapshotClass une seule fois. Trident réutilise le même VolumeSnapshotClass pour les StorageClasses suivants.

#### Créer le backend Trident

Pour créer un backend Trident, 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, la SVM à utiliser et la méthode d'authentification. L'exemple suivant montre comment définir un stockage basé sur NAS et utiliser un secret AWS pour stocker les informations d'identification 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"
    }
  }
}
```

### Détails du pilote FSx for ONTAP

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

| Nom du pilote | Description |
|---|---|
| `ontap-san` | Chaque PV provisionné est un LUN au sein de son propre volume Amazon FSx for NetApp ONTAP. Recommandé pour le stockage bloc. |
| `ontap-nas` | Chaque PV provisionné est un volume Amazon FSx for NetApp ONTAP complet. Recommandé pour NFS et SMB. |
| `ontap-san-economy` | Chaque PV provisionné est un LUN avec un nombre configurable de LUN par Amazon FSx for NetApp ONTAP volume. |
| `ontap-nas-economy` | Chaque PV provisionné est un qtree, avec un nombre configurable de qtrees par volume Amazon FSx for NetApp ONTAP. |
| `ontap-nas-flexgroup` | Chaque PV provisionné est un volume Amazon FSx for NetApp ONTAP FlexGroup complet. |

Pour plus de détails sur le conducteur, consultez "Pilotes NAS" et "Pilotes SAN".

### Créer le backend

Après avoir créé le fichier de configuration, exécutez les commandes suivantes pour créer et valider la configuration du backend Trident (TBC) :

- Créez une configuration backend Trident (TBC) à partir d'un 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 backend trident (TBC) a été créée avec succès :

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

```
NAME                          BACKEND NAME          BACKEND UUID
PHASE   STATUS

backend-tbc-ontap-nas         tbc-ontap-nas         933e0071-66ce-4324-
b9ff-f96d916ac5e9   Bound   Success
```

Pour plus d'informations sur les autres options de configuration, consultez la [Backend-advanced-configuration-and-examples] section ci-dessous.

#### Configurer une classe de stockage sans configuration automatique du backend

Voici des exemples de configurations de classes de stockage à utiliser avec Trident et FSx pour ONTAP.

### Classe de stockage pour NFS

Vous pouvez utiliser cet exemple pour configurer StorageClass pour des volumes utilisant NFS (reportez-vous à la section Attributs Trident ci-dessous pour la liste complète des attributs) :

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

### Classe de stockage pour iSCSI

Utilisez cet exemple pour configurer StorageClass pour des volumes utilisant iSCSI :

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

### Classe de stockage utilisant NFSv3 et AWS Bottlerocket

Pour provisionner des volumes NFSv3 sur AWS Bottlerocket, ajoutez les `mountOptions` requis 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
```

#### Attributs de Trident StorageClass

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

| Attribut | Type | Valeurs | Offre | Demande | Soutenu par |
|---|---|---|---|---|---|
| médias[1] | chaîne | hdd, hybride, ssd | Le pool contient des médias de ce type ; hybride signifie les deux | Type de média spécifié | ontap-nas, ontap-nas-economy, ontap-nas-flexgroup, ontap-san, solidfire-san |
| provisioningType | chaîne | mince, épais | Pool prend en charge cette méthode de provisionnement | Méthode de provisionnement spécifiée | épais : all ontap ; mince : all ontap & solidfire-san |
| backendType | chaîne | ontap-nas, ontap-nas-economy, ontap-nas-flexgroup, ontap-san, solidfire-san, azure-netapp-files, ontap-san-economy | Pool appartient à ce type de backend | Backend spécifié | Tous les drivers |
| instantanés | bool | vrai, faux | Pool prend en charge les volumes avec snapshots | Volume avec instantanés activés | ontap-nas, ontap-san, solidfire-san |

| Attribut | Type | Valeurs | Offre | Demande | Soutenu par |
|---|---|---|---|---|---|
| clones | bool | vrai, faux | Pool prend en charge le clonage des volumes | Volume avec clones activés | ontap-nas, ontap-san, solidfire-san |
| chiffrement | bool | vrai, faux | Pool prend en charge les volumes chiffrés | Volume avec chiffrement activé | ontap-nas, ontap-nas-economy, ontap-nas-flexgroups, ontap-san |
| Op E/S par sec | int | entier positif | Pool est capable de garantir des IOPS dans cette plage | Volume garanti ces IOPS | solidfire-san |

[1] : Non pris en charge par ONTAP Select ou les systèmes FSx for ONTAP

Consultez "Objets Kubernetes et Trident" pour plus de détails sur la manière dont les classes de stockage interagissent avec le `PersistentVolumeClaim` et les paramètres permettant de contrôler la façon dont Trident provisionne les volumes.

### Créer la classe de stockage

Une fois que vous avez configuré le StorageClass, vous pouvez le créer dans Kubernetes.

#### Étapes

1. Il s'agit d'un objet Kubernetes, utilisez donc `kubectl` pour le créer dans Kubernetes.

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

2. Vous devriez maintenant voir une classe de stockage **basic-csi** dans Kubernetes et Trident, et Trident devrait avoir découvert les pools sur le backend.

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

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

### Provisionner des volumes SMB

Vous pouvez provisionner des volumes SMB à l'aide du `ontap-nas` driver. Cependant, pour ce faire, vous devez effectuer ces étapes : "Préparez-vous à provisionner des volumes SMB".

### Configuration avancée du backend et exemples

Consultez le tableau suivant pour les options de configuration du backend :

| 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 le stockage backend | Nom du pilote + "_" + dataLIF |
| `managementLIF` | Adresse IP d'un cluster ou d'une LIF de gestion SVM. Un nom de domaine complet (FQDN) peut être spécifié. Il est possible de configurer l'utilisation d'adresses IPv6 si Trident a été installé avec l'option IPv6. Les adresses IPv6 doivent être définies entre crochets, par exemple : [28e8:d9fb:a825:b7bf:69a8:d02f:9e7b:3555]. Si vous fournissez le `fsxFilesystemID` sous le champ `aws`, il n'est pas nécessaire de fournir le `managementLIF` car Trident récupère les informations SVM `managementLIF` depuis AWS. Vous devez donc fournir les identifiants d'un utilisateur sous la SVM (par exemple : vsadmin) et l'utilisateur doit avoir le rôle `vsadmin`. | "10.0.0.1", "[2001:1234:abcd::fefe]" |

| Paramètre | Description | Exemple |
| --- | --- | --- |
| `dataLIF` | Adresse IP de l'interface logique de protocole (LIF). **Pilotes ONTAP NAS** : NetApp recommande de spécifier dataLIF. Si elle n'est pas fournie, Trident récupère les dataLIF depuis la 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 à répartition de charge (round-robin) entre plusieurs dataLIF. Peut être modifié après la configuration initiale. **Pilotes ONTAP SAN** : ne pas spécifier pour iSCSI. Trident utilise ONTAP Selective LUN Map pour découvrir les LIF iSCSI nécessaires à l'établissement d'une session multipath. Un avertissement est généré si dataLIF est explicitement défini. Il est possible de configurer l'utilisation d'adresses IPv6 si Trident a été installé avec l'option IPv6. Les adresses IPv6 doivent être définies entre crochets, par exemple : [28e8:d9fb:a825:b7bf:69a8:d02f:9e7b:3555]. | |
| `autoExportPolicy` | Activer la création et la mise à jour automatiques des règles d'export [Booléen]. En utilisant les options `autoExportPolicy` et `autoExportCIDRs`, Trident peut gérer les règles d'export automatiquement. | `false` |
| `autoExportCIDRs` | Liste des CIDR pour filtrer les adresses IP des nœuds Kubernetes lorsque `autoExportPolicy` est activé. En utilisant les options `autoExportPolicy` et `autoExportCIDRs`, Trident peut gérer les règles d'export automatiquement. | "["0.0.0.0/0", "::/0"]" |
| `labels` | Ensemble d'étiquettes arbitraires au format JSON à appliquer aux volumes | "" |

| Paramètre | Description | Exemple |
| --- | --- | --- |
| `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 d'autorité de certification de confiance. Facultatif. Utilisé pour l'authentification par certificat. | "" |
| `username` | Nom d'utilisateur pour se connecter au cluster ou à la SVM. Utilisé pour l'authentification par identifiants. Par exemple, vsadmin. | |
| `password` | Mot de passe permettant de se connecter au cluster ou à la SVM. Utilisé pour l'authentification par identifiants. | |
| `svm` | Machine virtuelle de stockage à utiliser | Dérivé si un LIF de gestion SVM est spécifié. |
| `storagePrefix` | Préfixe utilisé lors du provisionnement de nouveaux volumes dans la SVM. Ne peut pas être modifié après création. Pour mettre à jour ce paramètre, vous devrez créer un nouveau backend. | `trident` |
| `limitAggregateUsage` | **Ne pas spécifier pour Amazon FSx for NetApp ONTAP.** Les `fsxadmin` et `vsadmin` fournis ne contiennent pas les autorisations requises pour récupérer l'utilisation agrégée et la limiter à l'aide de Trident. | Ne pas utiliser. |
| `limitVolumeSize` | L'approvisionnement échoue si la taille du volume demandée dépasse cette valeur. Limite également la taille maximale des volumes qu'il gère pour les qtrees et les LUN, et l' `qtreesPerFlexvol`option permet de personnaliser le nombre maximal de qtrees par volume FlexVol | "" (non appliqué par défaut) |
| `lunsPerFlexvol` | Nombre maximal de LUN par volume FlexVol, doit être compris entre 50 et 200. SAN uniquement. | "100" |

| Paramètre | Description | Exemple |
| --- | --- | --- |
| `debugTraceFlags` | Options de débogage à utiliser lors du dépannage. Exemple, {"api":false, "method":true} Ne pas utiliser `debugTraceFlags` sauf si vous effectuez un dépannage et avez besoin d'un journal détaillé. | null |
| `nfsMountOptions` | Liste d'options de montage NFS séparées par des virgules. Les options de montage pour les volumes persistants Kubernetes sont généralement spécifiées dans les classes de stockage, mais si aucune option de montage n'est spécifiée dans une classe de stockage, Trident utilisera les options de montage spécifiées dans le fichier de configuration du backend de stockage. Si aucune option de montage n'est spécifiée ni dans la classe de stockage ni dans le fichier de configuration, Trident n'appliquera aucune option de montage au volume persistant associé. | "" |
| `nasType` | Configurez la création de volumes NFS ou SMB. Les options sont `nfs`, `smb` ou null. **Doit être défini sur `smb` pour les volumes SMB.** La valeur null correspond par défaut à des volumes NFS. | `nfs` |
| `qtreesPerFlexvol` | Nombre maximal de Qtrees par volume FlexVol, doit être compris dans la plage [50, 300] | `"200"` |
| `smbShare` | Vous pouvez spécifier l'un des éléments suivants : 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 les backends Amazon FSx for ONTAP. | `smb-share` |

| Paramètre | Description | Exemple |
|---|---|---|
| `useREST` | Paramètre booléen à utiliser pour les API REST ONTAP. Lorsqu'il est défini sur `true`, Trident utilisera les API REST ONTAP pour communiquer avec le backend. Cette fonctionnalité nécessite ONTAP 9.11.1 ou une version ultérieure. De plus, le rôle de connexion ONTAP utilisé doit avoir accès à l'application `ontap`. Cela est satisfait par les rôles prédéfinis `vsadmin` et `cluster-admin`. | `false` |
| `aws` | Vous pouvez spécifier les éléments suivants dans le fichier de configuration pour AWS FSx for ONTAP : - `fsxFilesystemID` : Spécifiez l'ID du système de fichiers AWS FSx. - `apiRegion` : Nom de la région de l'API AWS. - `apikey` : Clé d'API AWS. - `secretKey` : Clé secrète AWS. | `""`<br>`""`<br>`""` |
| `credentials` | Spécifiez les informations d'identification de la SVM FSx à stocker dans AWS Secrets Manager. - `name` : Nom de ressource Amazon (ARN) du secret, qui contient les informations d'identification de la SVM. - `type` : Définissez sur `awsarn`. Consultez "Créer un secret AWS Secrets Manager" pour plus d'informations. | |

### Options de configuration backend pour le provisionnement des volumes

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

| Paramètre | Description | Défaut |
|---|---|---|
| `spaceAllocation` | Allocation d'espace pour les LUNs | `true` |
| `spaceReserve` | Mode de réservation d'espace ; "none" (fin) ou "volume" (épais) | `none` |
| `snapshotPolicy` | Stratégie de snapshot à utiliser | `none` |

| Paramètre | Description | Défaut |
| --- | --- | --- |
| `qosPolicy` | Groupe de règles QoS à attribuer aux volumes créés. Choisissez l'un des qosPolicy ou adaptiveQosPolicy par pool de stockage ou backend. L'utilisation des groupes de règles QoS avec Trident requiert 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 individuellement à chaque composant. Un groupe de règles QoS partagé impose une limite au débit total de toutes les charges de travail. | "" |
| `adaptiveQosPolicy` | Groupe de règles QoS adaptatives à attribuer aux volumes créés. Choisissez l'un des qosPolicy ou adaptiveQosPolicy par pool de stockage ou backend. Non pris en charge par ontap-nas-economy. | "" |
| `snapshotReserve` | Pourcentage du volume réservé aux 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`. NVE doit être sous licence et activé sur le cluster pour utiliser cette option. Si NAE est activé sur le backend, tout volume provisionné dans Trident sera activé pour NAE. Pour plus d'informations, consultez : "Comment Trident fonctionne avec NVE et NAE". | `false` |
| `luksEncryption` | Activez le chiffrement LUKS. Consultez "Utilisez Linux Unified Key Setup (LUKS)". SAN uniquement. | "" |
| `tieringPolicy` | Politique de hiérarchisation à utiliser `none` | |
| `unixPermissions` | Mode pour les nouveaux volumes. **Laisser vide pour les volumes SMB.** | "" |

| Paramètre | Description | Défaut |
| --- | --- | --- |
| `securityStyle` | Style de sécurité pour les nouveaux volumes. NFS prend en charge `mixed` et `unix` styles de sécurité. SMB prend en charge `mixed` et `ntfs` styles de sécurité. | La valeur par défaut de NFS est `unix`. La valeur par défaut de SMB est `ntfs`. |

## Configurer un PVC

Cette section explique comment créer un PersistentVolumeClaim (PVC) qui utilise la StorageClass Kubernetes configurée pour demander un PV. En cas de succès, vous pourrez ensuite monter le PV sur un pod.

### Créer le PVC

Un "*PersistentVolumeClaim*" (PVC) est une demande d'accès au PersistentVolume sur le cluster. Le PVC peut être configuré pour demander un espace de stockage d'une certaine taille ou un certain mode d'accès. En utilisant la StorageClass associée, l'administrateur du cluster peut contrôler plus que la taille et le mode d'accès du PersistentVolume—comme les performances ou le niveau de service.

Après avoir créé le backend Trident et le StorageClass, vous pouvez créer un PVC. Après la création du PVC, vous pouvez monter le volume dans un pod.

#### Exemples de manifestes

Les exemples suivants illustrent les options de configuration de base du PVC.

#### PVC avec accès RWX

Cet exemple montre un PVC de base avec un accès RWX qui est associé à un StorageClass nommé `basic-csi`.

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

#### Exemple d'utilisation de PVC avec iSCSI

Cet exemple montre un PVC de base pour iSCSI avec accès RWO associé à un StorageClass nommé `protection-gold`.

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: pvc-san
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: protection-gold
```

#### Créer un PVC

**Étapes**

1. Créez le 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
```

Consultez "Objets Kubernetes et Trident" pour plus de détails sur la manière dont les classes de stockage interagissent avec le `PersistentVolumeClaim` et les paramètres permettant de contrôler la façon dont Trident provisionne les volumes.

## Déployer une application

Une fois la classe de stockage et le PVC créés, vous pouvez monter le PV sur un pod. Cette section présente la commande et la configuration d'exemple pour associer le PV à un pod.

## Déployer une application exemple

**Étapes**

1. Montez le volume dans un pod.

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

Ces exemples montrent des 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
```

**REMARQUE** Vous pouvez surveiller la progression en utilisant `kubectl get pod --watch`.

2. 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  1.1G
320K  1.0G   1%   /my/mount/path
```

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

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

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

NetApp Trident simplifie la gestion du stockage Amazon FSx for NetApp ONTAP dans Kubernetes afin de permettre à vos développeurs et administrateurs de se concentrer sur le déploiement des applications. L'add-on NetApp Trident EKS inclut les derniers correctifs de sécurité, les corrections de bogues, et est validé par AWS pour fonctionner avec Amazon EKS. L'add-on EKS vous permet de garantir de manière cohérente que vos clusters Amazon EKS sont sécurisés et stables, et de réduire la quantité de travail nécessaire pour installer, configurer et mettre à jour les add-ons.

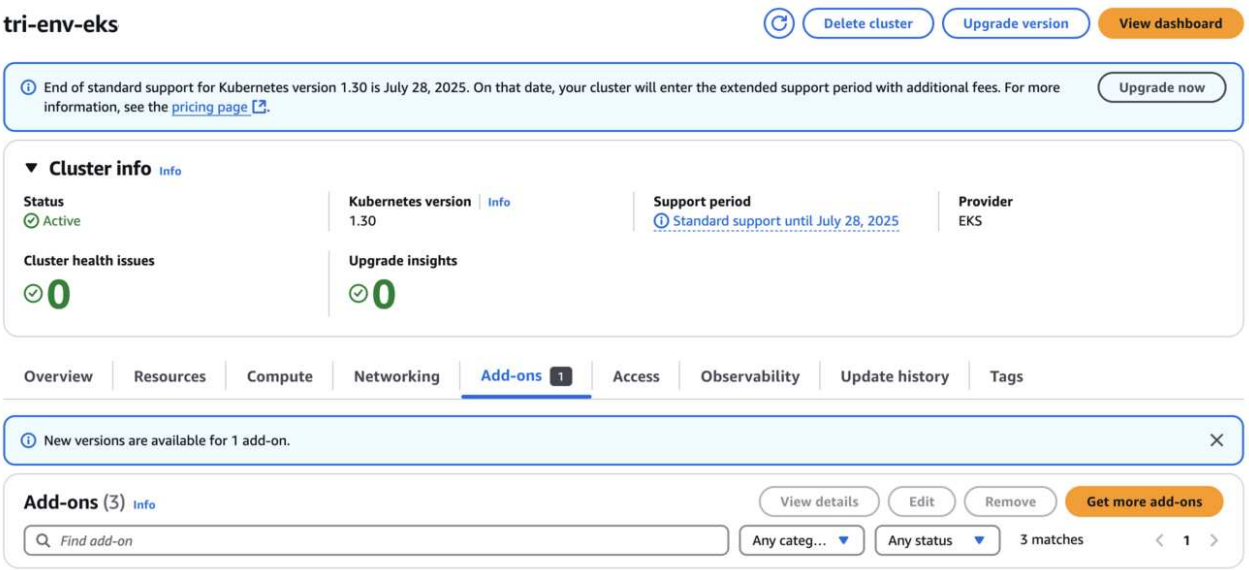
### Prérequis

Assurez-vous de disposer des éléments suivants avant de configurer le module complémentaire Trident pour AWS EKS :

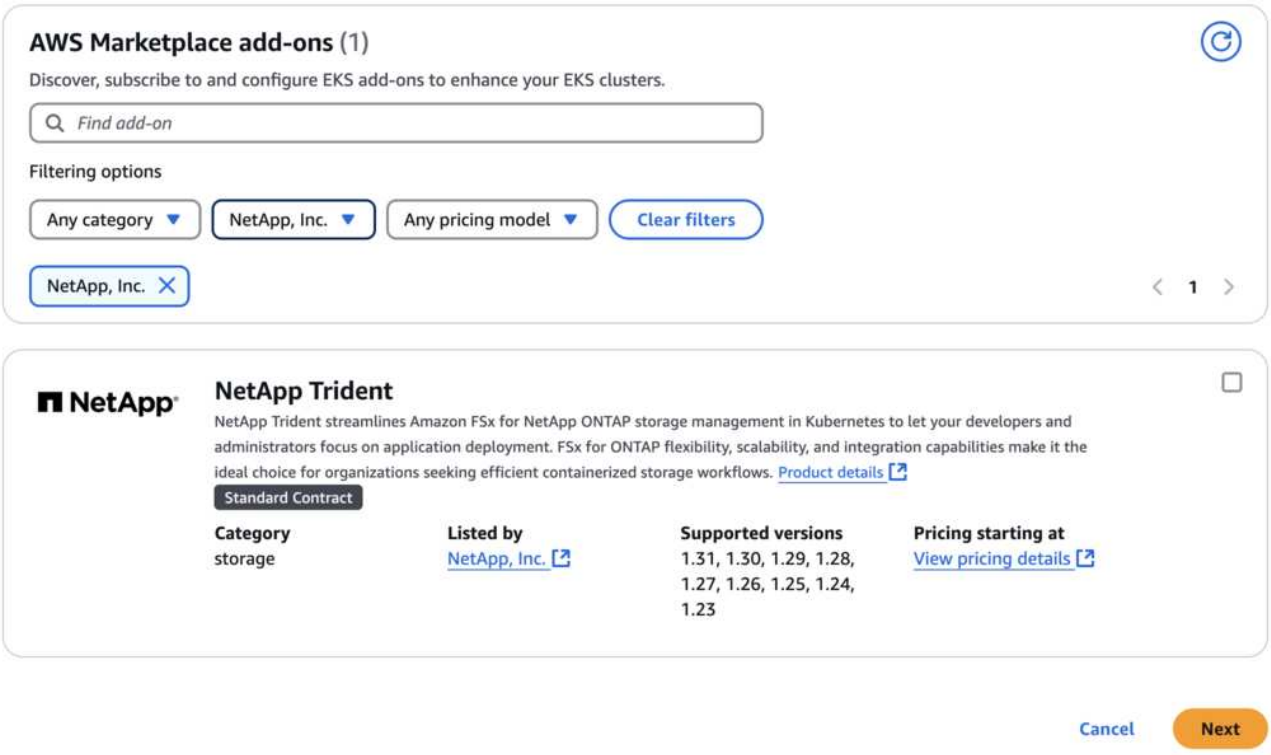
- Un compte de cluster Amazon EKS disposant des autorisations nécessaires pour gérer les modules complémentaires. Consultez "Modules complémentaires Amazon EKS".
- Autorisations AWS pour la AWS marketplace :
  ```
  "aws-marketplace:ViewSubscriptions",
  "aws-marketplace:Subscribe",
  "aws-marketplace:Unsubscribe
  ```
- Type d'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 for NetApp ONTAP existant

### Étapes

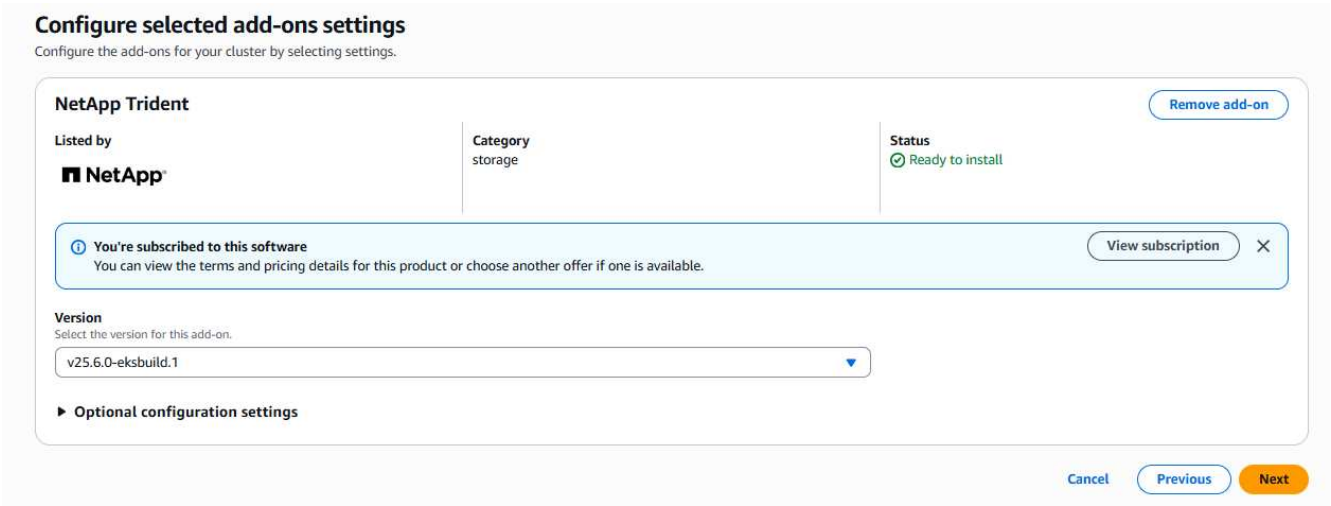
1. Veillez à créer un rôle IAM et un secret AWS pour permettre aux pods EKS d'accéder aux ressources AWS. Pour obtenir des instructions, consultez "Créez un rôle IAM et un secret AWS".
2. Sur votre cluster Kubernetes EKS, accédez à l'onglet **Add-ons**.



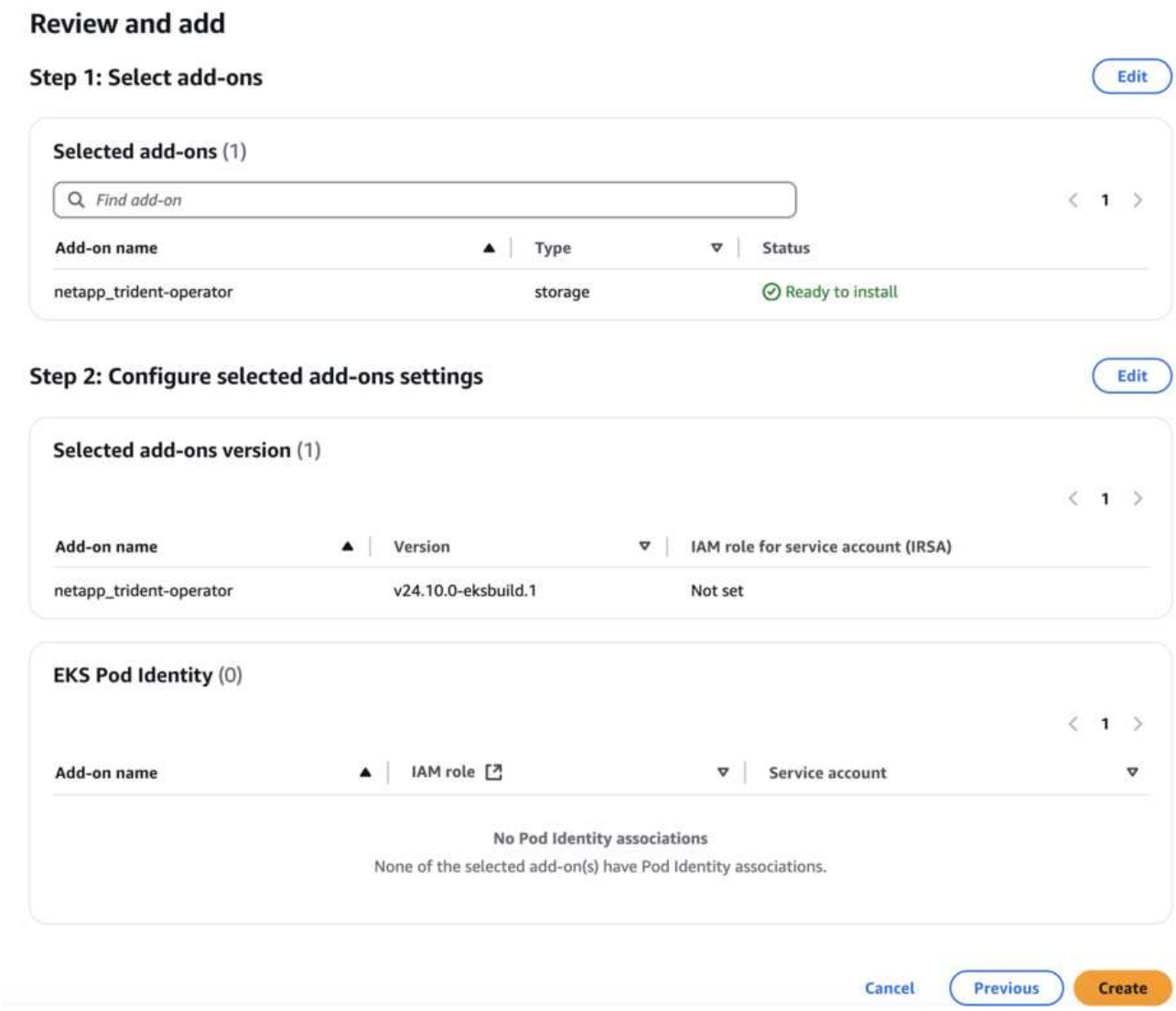

3. Accédez à **AWS Marketplace add-ons** et choisissez la catégorie *storage*.



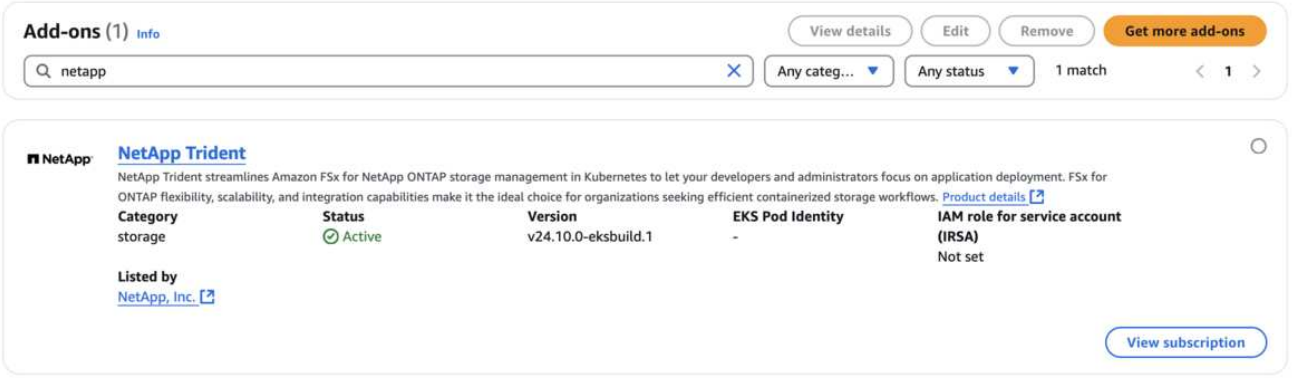

4. Repérez **NetApp Trident** et cochez la case de l'extension Trident, puis cliquez sur **Suivant**.
5. Choisissez la version souhaitée du module complémentaire.




6. Configurez les paramètres du module complémentaire requis.




7. Si vous utilisez IRSA (IAM roles for service account), reportez-vous aux étapes de configuration supplémentaires "ici".
8. Sélectionnez **Create**.
9. Vérifiez que le statut du module complémentaire est *Active*.




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 configurez le stockage backend. Pour plus d'informations, consultez "Configurer le backend de stockage".

**Installez/désinstallez l'add-on Trident EKS à l'aide de la CLI**

**Installez le module complémentaire NetApp Trident EKS à l'aide de l'interface de ligne de commande :**

La commande suivante installe le module complémentaire Trident EKS :
`eksctl create addon --cluster clusterName --name netapp_trident-operator --version v25.6.0-eksbuild.1` (avec une version dédiée)

L'exemple de commande suivant installe le Trident EKS add-on version 25.6.1 :
`eksctl create addon --cluster clusterName --name netapp_trident-operator --version v25.6.1-eksbuild.1` (avec une version dédiée)

L'exemple de commande suivant installe le Trident EKS add-on version 25.6.2 :
`eksctl create addon --cluster clusterName --name netapp_trident-operator --version v25.6.2-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
```

# Créer des backends avec kubectl

Un backend définit la relation entre Trident et un système de stockage. Il indique à Trident comment communiquer avec ce système de stockage et comment Trident doit provisionner des volumes à partir de celui-ci. Après l'installation de Trident, l'étape suivante consiste à créer un backend. La `TridentBackendConfig` Custom Resource Definition (CRD) permet de créer et de gérer des backends Trident directement via l'interface Kubernetes. Vous pouvez le faire en utilisant `kubectl` ou l'outil CLI équivalent pour votre distribution Kubernetes.

## `TridentBackendConfig`

`TridentBackendConfig` (`tbc`, `tbconfig`, `tbackendconfig`) est une CRD frontale avec espace de noms qui vous permet de gérer les backends Trident à l'aide de `kubectl`. Les administrateurs Kubernetes et de stockage peuvent désormais créer et gérer des backends directement via l'interface de ligne de commande Kubernetes sans avoir besoin d'un utilitaire en ligne de commande dédié (`tridentctl`).

Lors de la création d'un `TridentBackendConfig` objet, les événements suivants se produisent :

- Un backend est créé automatiquement par Trident en fonction de la configuration que vous fournissez. Ceci est représenté en interne par une `TridentBackend` (`tbe`, `tridentbackend`) CR.

- Le `TridentBackendConfig` est lié de manière unique à un `TridentBackend` qui a été créé par Trident.

Chacune `TridentBackendConfig` maintient une correspondance un-à-un avec une `TridentBackend`. La première est l'interface fournie à l'utilisateur pour concevoir et configurer les backends ; la seconde est la façon dont Trident représente l'objet backend réel.

| | |
|---|---|
| **AVERTISSEMENT** | `TridentBackend` Les CR sont créés automatiquement par Trident. Vous **ne devez pas** les modifier. Si vous souhaitez mettre à jour les backends, faites-le en modifiant l'objet `TridentBackendConfig`. |

Voir l'exemple suivant pour le format du `TridentBackendConfig` 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 dans le répertoire "trident-installer" pour des configurations types pour la plateforme/le service de stockage souhaité.

Le `spec` prend des paramètres de configuration spécifiques au backend. Dans cet exemple, le backend utilise le `ontap-san` storage driver et utilise les paramètres de configuration qui sont présentés ici. Pour la liste des options de configuration pour votre storage driver souhaité, consultez le "Informations de configuration du backend pour votre pilote de stockage".

La `spec` section inclut également `credentials` et `deletionPolicy` des champs, qui sont nouvellement introduits dans le `TridentBackendConfig` CR :

- `credentials`: Ce paramètre est obligatoire et contient les identifiants utilisés pour l'authentification auprès du système/service de stockage. Ceci est défini sur un Secret Kubernetes créé par l'utilisateur. Les identifiants ne peuvent pas être transmis en clair et cela entraînera 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 backend associé. Il s'agit de la valeur par défaut.
  - `retain` : Lorsqu'un `TridentBackendConfig` CR est supprimé, la définition du backend reste présente et peut être gérée avec `tridentctl`. La configuration de la politique de suppression sur `retain` permet aux utilisateurs de revenir à une version antérieure (pré-21.04) et de conserver les

backends créés. La valeur de ce champ peut être mise à jour après qu'un `TridentBackendConfig` est créé.

| | |
|---|---|
| **REMARQUE** | 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 l'objet `TridentBackendConfig` (metadata.name). Il est recommandé de définir explicitement les noms des backends à l'aide de `spec.backendName`. |
| **ASTUCE** | Les backends qui ont été créés avec `tridentctl` n'ont pas d'objet `TridentBackendConfig` associé. Vous pouvez choisir de gérer ces backends avec `kubectl` en créant un CR `TridentBackendConfig`. Il faut veiller à spécifier des paramètres de configuration identiques (tels que `spec.backendName`, `spec.storagePrefix`, `spec.storageDriverName`, etc.). Trident associera automatiquement le nouvel objet `TridentBackendConfig` au backend préexistant. |

## Aperçu des étapes

Pour créer un nouveau backend en utilisant `kubectl`, vous devez procéder comme suit :

1. Créez un "Secret Kubernetes". Le secret contient les identifiants dont Trident a besoin pour communiquer avec le cluster/service de stockage.
2. Créez un `TridentBackendConfig` objet. Celui-ci 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 statut en utilisant `kubectl get tbc <tbc-name> -n <trident-namespace>` et recueillir des informations supplémentaires.

## Étape 1 : Créer un secret Kubernetes

Créez un secret contenant les identifiants d'accès pour le backend. Ceci 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 qui doivent figurer dans le Secret pour chaque plateforme de stockage :

| Description des champs secrets de la plateforme de stockage | Secret | Description des champs |
|---|---|---|
| Azure NetApp Files | clientID | L'identifiant client issu de l'enregistrement d'une application |
| Element (NetApp HCI/SolidFire) | Point de terminaison | MVIP pour le cluster SolidFire avec les identifiants du locataire |
| ONTAP | nom d'utilisateur | Nom d'utilisateur pour se connecter au cluster/SVM. Utilisé pour l'authentification basée sur les identifiants |
| ONTAP | mot de passe | Mot de passe pour se connecter au cluster/SVM. Utilisé pour l'authentification basée sur les identifiants |
| ONTAP | clientPrivateKey | Valeur encodée en Base64 de la clé privée du client. Utilisée pour l'authentification basée sur un certificat |
| ONTAP | chapUsername | Nom d'utilisateur entrant. Requis si useCHAP=true. Pour `ontap-san` et `ontap-san-economy` |
| ONTAP | chapInitiatorSecret | Secret de l'initiateur CHAP. Requis si useCHAP=true. Pour `ontap-san` et `ontap-san-economy` |
| ONTAP | chapTargetUsername | Nom d'utilisateur cible. Requis si useCHAP=true. Pour `ontap-san` et `ontap-san-economy` |
| ONTAP | chapTargetInitiatorSecret | Secret de l'initiateur de la cible CHAP. Requis si useCHAP=true. Pour `ontap-san` et `ontap-san-economy` |

Le secret créé à cette étape sera référencé dans le champ `spec.credentials` de l'objet `TridentBackendConfig` qui est créé à l'étape suivante.

## Étape 2 : Créer la `TridentBackendConfig` demande de changement

Vous êtes maintenant prêt à créer votre `TridentBackendConfig` CR. Dans cet exemple, un backend qui utilise le `ontap-san` driver est créé à l'aide de l' `TridentBackendConfig` objet présenté 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 : Vérifiez le statut du `TridentBackendConfig` CR

Maintenant que vous avez créé le `TridentBackendConfig` CR, vous pouvez vérifier l'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 backend a été créé avec succès et lié au `TridentBackendConfig` CR.

La phase peut prendre l'une des valeurs suivantes :

- `Bound` : Le `TridentBackendConfig`CR est associé à un backend, et ce backend contient `configRef`défini sur l' `TridentBackendConfig`uid du CR.
- `Unbound` : Représenté à l'aide de `""`. L'objet `TridentBackendConfig` n'est pas lié à un backend. Tous les nouveaux CR créés `TridentBackendConfig` sont par défaut dans cette phase. Après le changement de phase, il ne peut plus revenir à l'état Unbound.
- `Deleting` : Le `TridentBackendConfig`CR `deletionPolicy` a été configuré pour être supprimé. Lorsque le `TridentBackendConfig`CR est supprimé, il passe à l'état Deleting.
  - Si aucune revendication de volume persistant (PVC) n'existe sur le backend, la suppression du `TridentBackendConfig` entraînera Trident à supprimer le backend ainsi que le `TridentBackendConfig` CR.
  - Si une ou plusieurs PVC sont présentes sur le backend, il passe en état de suppression. Le `TridentBackendConfig` CR entre ensuite également en phase de suppression. Le backend et

`TridentBackendConfig` ne sont supprimés qu'après la suppression de toutes les PVC.

- `Lost` : Le backend associé au `TridentBackendConfig` CR a été supprimé accidentellement ou intentionnellement et le `TridentBackendConfig` CR possède toujours une référence vers le backend supprimé. Le `TridentBackendConfig` CR peut toujours être supprimé, quelle que soit la valeur de `deletionPolicy`.
- `Unknown` : Trident ne parvient pas à déterminer l'état ou l'existence du backend associé au `TridentBackendConfig` CR. Par exemple, si le serveur API ne répond pas ou si le `tridentbackends.trident.netapp.io` CRD est manquant. Cela peut nécessiter une intervention.

À ce stade, le backend est créé avec succès ! Plusieurs opérations supplémentaires peuvent être gérées, telles que "mises à jour du backend et suppressions du backend".

## (Facultatif) Étape 4 : Obtenir plus de détails

Vous pouvez exécuter la commande suivante pour obtenir plus d'informations sur votre backend :

```
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   ontap-san-backend   8d24fce7-6f60-4d4a-8ef6-
bab2699e6ab8   Bound   Success   ontap-san        delete
```

De plus, vous pouvez également obtenir un dump 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 l' `backendName` et le `backendUUID` du backend qui a été créé en réponse à la `TridentBackendConfig` CR. Le `lastOperationStatus` champ représente le statut de la dernière opération de la `TridentBackendConfig` CR, qui peut être déclenchée par l'utilisateur (par exemple, si l'utilisateur a modifié quelque chose dans `spec`) ou déclenchée par Trident (par exemple, lors des redémarrages de Trident). Il peut être soit Success soit Failed. `phase` représente le statut de la relation entre la `TridentBackendConfig` CR et le backend. Dans l'exemple ci-dessus, `phase` a la valeur Bound, ce qui signifie que la `TridentBackendConfig` CR est associée au backend.

Vous pouvez exécuter la commande `kubectl -n trident describe tbc <tbc-cr-name>` pour obtenir les détails des journaux d'événements.

| | |
|---|---|
| **AVERTISSEMENT** | Vous ne pouvez pas mettre à jour ou supprimer un backend qui contient un objet associé `TridentBackendConfig` en utilisant `tridentctl`. Pour comprendre les étapes nécessaires pour passer de `tridentctl` à `TridentBackendConfig`, "voir ici". |

# Gérer les backends

## Effectuez la gestion du backend avec kubectl

Découvrez comment effectuer des opérations de gestion backend en utilisant `kubectl`.

### Supprimer un backend

En supprimant un `TridentBackendConfig`, vous indiquez à Trident de supprimer ou de conserver les backends (selon `deletionPolicy`). Pour supprimer un backend, assurez-vous que `deletionPolicy` est défini sur delete. Pour supprimer uniquement le `TridentBackendConfig`, assurez-vous que `deletionPolicy` est défini sur retain. Cela garantit que le backend 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 qui étaient utilisés par `TridentBackendConfig`. L'utilisateur Kubernetes est responsable du nettoyage des secrets. Il faut faire attention lors de la suppression des secrets. Vous ne devez supprimer les secrets que s'ils ne sont pas utilisés par les backends.

### Afficher les backends 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 une liste de tous les backends existants. Cette liste inclura également les backends qui ont été créés avec `tridentctl`.

### Mettre à jour un backend

Il peut exister plusieurs raisons de mettre à jour un backend :

- Les identifiants d'accès au système de stockage ont changé. Pour mettre à jour les identifiants, le Secret Kubernetes utilisé dans l' `TridentBackendConfig`objet doit être mis à jour. Trident mettra automatiquement à jour le backend avec les identifiants les plus récents fournis. Exécutez la commande suivante pour mettre à jour le Secret Kubernetes :

  ```
  kubectl apply -f <updated-secret-file.yaml> -n trident
  ```

- Les paramètres (tels que le nom de la ONTAP SVM utilisée) doivent être mis à jour.
  - Vous pouvez mettre à jour `TridentBackendConfig` les objets directement via Kubernetes à l'aide de la commande suivante :

```
kubectl apply -f <updated-backend-file.yaml>
```

- Vous pouvez également apporter des modifications à la `TridentBackendConfig`CR existante à l'aide de la commande suivante :

```
kubectl edit tbc <tbc-name> -n trident
```

**REMARQUE**

- En cas d'échec d'une mise à jour du backend, le backend conserve sa dernière configuration connue. Vous pouvez consulter les journaux pour en 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 de mise à jour.

## Effectuez la gestion du backend avec tridentctl

Découvrez comment effectuer des opérations de gestion backend en utilisant `tridentctl`.

### Créer un backend

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

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

Si la création du backend échoue, quelque chose n'allait pas avec la configuration du backend. Vous pouvez consulter les journaux pour en déterminer la cause en exécutant la commande suivante :

```
tridentctl logs -n trident
```

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

### Supprimer un backend

Pour supprimer un backend de Trident, procédez comme suit :

1. Récupérer le nom du backend :

```
tridentctl get backend -n trident
```

2. Supprimez le backend :

```
tridentctl delete backend <backend-name> -n trident
```

| REMARQUE | Si Trident a provisionné des volumes et des instantanés à partir de ce backend qui existent encore, la suppression du backend empêche le provisionnement de nouveaux volumes par celui-ci. Le backend continuera d'exister dans un état « Deleting ». |
|---|---|

#### Afficher les backends existants

Pour afficher les backends que Trident connaît, procédez comme suit :

- Pour obtenir un résumé, 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 backend

Après avoir créé un nouveau fichier de configuration, exécutez la commande suivante :

```
tridentctl update backend <backend-name> -f <backend-file> -n trident
```

Si la mise à jour du backend échoue, cela signifie qu'il y a un problème avec la configuration du backend ou que vous avez tenté une mise à jour invalide. Vous pouvez consulter les journaux pour en déterminer la cause en exécutant la commande suivante :

```
tridentctl logs -n trident
```

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

#### Identifiez les classes de stockage qui utilisent un backend

Voici un exemple du type de questions auxquelles vous pouvez répondre avec le JSON que `tridentctl` génère 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 backends qui ont été créés en utilisant `TridentBackendConfig`.

## Passer d'une option de gestion du backend à une autre

Découvrez les différentes manières de gérer les backends dans Trident.

### Options pour la gestion des backends

Avec l'introduction de `TridentBackendConfig`, les administrateurs disposent désormais de deux méthodes distinctes pour gérer les backends. Cela soulève les questions suivantes :

- Les backends créés à l'aide de `tridentctl` peuvent-ils être gérés avec `TridentBackendConfig` ?
- Les backends créés à l'aide de `TridentBackendConfig` peuvent-ils être gérés à l'aide de `tridentctl` ?

### Gérer `tridentctl` les backends à l'aide de `TridentBackendConfig`

Cette section couvre les étapes nécessaires pour gérer les backends qui ont été créés à l'aide de `tridentctl` directement via l'interface Kubernetes en créant des `TridentBackendConfig` objets.

Cela s'appliquera aux scénarios suivants :

- Les backends préexistants, qui n'ont pas de `TridentBackendConfig` parce qu'ils ont été créés avec `tridentctl`.
- De nouveaux backends ont été créés avec `tridentctl`, tandis que d'autres `TridentBackendConfig` objets existent.

Dans les deux cas, les backends resteront présents, Trident planifiant les volumes et opérant dessus. Les administrateurs ont alors deux choix :

- Continuez à utiliser `tridentctl` pour gérer les backends qui ont été créés avec.
- Liez les backends créés à l'aide de `tridentctl` à un nouvel objet `TridentBackendConfig`. Cela signifie que les backends seront gérés à l'aide de `kubectl` et non `tridentctl`.

Pour gérer un backend préexistant à l'aide de `kubectl`, vous devrez créer un `TridentBackendConfig` qui se lie au backend existant. Voici un aperçu de la façon dont cela fonctionne :

1. Créez un secret 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. Celui-ci 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 backend existant.

#### Étape 0 : Identifier le backend

Pour créer une `TridentBackendConfig` qui se lie à un backend existant, vous devrez obtenir la configuration du backend. Dans cet exemple, supposons qu'un backend ait 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 : Créer un secret Kubernetes

Créez un Secret qui contient les identifiants pour le backend, 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 un `TridentBackendConfig` CR

L'étape suivante consiste à créer un `TridentBackendConfig` CR qui se liera automatiquement à la `ontap-nas-backend` préexistante (comme dans cet exemple). Assurez-vous que les conditions suivantes sont remplies :

- Le même nom de backend est défini dans `spec.backendName`.
- Les paramètres de configuration sont identiques à ceux du backend d'origine.
- Les pools virtuels (le cas échéant) doivent conserver le même ordre que dans le backend d'origine.
- Les informations d'identification sont fournies via un Secret Kubernetes et non en texte clair.

Dans ce cas, le `TridentBackendConfig` ressemblera à ceci :

```
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 : Vérifiez le statut du `TridentBackendConfig` CR

Après la `TridentBackendConfig` création, sa phase doit être `Bound`. Il doit également refléter le même nom de backend et le même UUID que le backend 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 backend sera désormais entièrement géré à l'aide de l' `tbc-ontap-nas-backend` `TridentBackendConfig`objet.

### Gérer `TridentBackendConfig` les backends à l'aide de `tridentctl`

```
`tridentctl` peut être utilisé pour lister les backends qui ont été créés
à l'aide de  `TridentBackendConfig`. De plus, les administrateurs peuvent
également choisir de gérer entièrement ces backends via  `tridentctl` en
supprimant  `TridentBackendConfig` et en s'assurant que
`spec.deletionPolicy` est défini sur  `retain`.
```

#### Étape 0 : Identifier le backend

Par exemple, supposons que le backend suivant ait 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 |
+-------------------+----------------
+--------------------------------------+--------+---------+
```

D'après le résultat, on constate 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`. Celle-ci doit être définie sur `retain`. Cela garantit que lorsqu'une `TridentBackendConfig` CR est supprimée, la définition backend sera toujours présente et pourra 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
```

| REMARQUE | Ne passez pas à l'étape suivante à moins que `deletionPolicy` soit défini sur `retain`. |
|---|---|

**Étape 2 : Supprimer le `TridentBackendConfig` CR**

La dernière étape consiste à supprimer le `TridentBackendConfig` CR. Après avoir confirmé que le `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 l `TridentBackendConfig`objet, Trident le supprime simplement sans supprimer le backend lui-même.

**Informations sur le copyright**

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

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

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

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

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

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

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

**Informations sur les marques commerciales**

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