# Installer Trident Protect

## Trident

NetApp
January 15, 2026

# Sommaire

# Installer Trident Protect

## Exigences de Trident Protect

Commencez par vérifier l'état de préparation de votre environnement opérationnel, de vos clusters d'applications, de vos applications et de vos licences. Assurez-vous que votre environnement répond à ces exigences pour déployer et utiliser Trident Protect.

### Compatibilité avec les clusters Kubernetes de Trident Protect

Trident Protect est compatible avec une large gamme d'offres Kubernetes entièrement gérées et autogérées, notamment :

- Amazon Elastic Kubernetes Service (EKS)
- Moteur Google Kubernetes (GKE)
- Service Kubernetes Microsoft Azure (AKS)
- Red Hat OpenShift
- SUSE Rancher
- Portefeuille VMware Tanzu
- Kubernetes en amont

- Les sauvegardes Trident Protect sont prises en charge uniquement sur les nœuds de calcul Linux. Les nœuds de calcul Windows ne sont pas pris en charge pour les opérations de sauvegarde.
- Assurez-vous que le cluster sur lequel vous installez Trident Protect est configuré avec un contrôleur de snapshots en cours d'exécution et les CRD associés. Pour installer un contrôleur de snapshots, reportez-vous à la documentation. "ces instructions" .

### Compatibilité du système de stockage Trident Protect

Trident Protect prend en charge les systèmes de stockage suivants :

- Amazon FSx for NetApp ONTAP
- Cloud Volumes ONTAP
- baies de stockage ONTAP
- Google Cloud NetApp Volumes
- Azure NetApp Files

Assurez-vous que votre système de stockage réponde aux exigences suivantes :

- Assurez-vous que le stockage NetApp connecté au cluster utilise Trident 24.02 ou une version plus récente (Trident 24.10 est recommandé).
- Assurez-vous de disposer d'un système de stockage NetApp ONTAP .
- Assurez-vous d'avoir configuré un compartiment de stockage d'objets pour stocker les sauvegardes.
- Créez les espaces de noms d'application que vous prévoyez d'utiliser pour les applications ou les

opérations de gestion des données d'application. Trident Protect ne crée pas ces espaces de noms pour vous ; si vous spécifiez un espace de noms inexistant dans une ressource personnalisée, l'opération échouera.

## Exigences pour les volumes de l'économie NAS

Trident Protect prend en charge les opérations de sauvegarde et de restauration sur les volumes NAS économiques. Les snapshots, les clones et la réplication SnapMirror vers des volumes nas-economy ne sont actuellement pas pris en charge. Vous devez activer un répertoire de snapshots pour chaque volume nas-economy que vous prévoyez d'utiliser avec Trident Protect.

Certaines applications ne sont pas compatibles avec les volumes qui utilisent un répertoire de snapshots. Pour ces applications, vous devez masquer le répertoire des instantanés en exécutant la commande suivante sur le système de stockage ONTAP :

```
nfs modify -vserver <svm> -v3-hide-snapshot enabled
```

Vous pouvez activer le répertoire de snapshots en exécutant la commande suivante pour chaque volume nas-economy, en remplaçant `<volume-UUID>` avec l'UUID du volume que vous souhaitez modifier :

```
tridentctl update volume <volume-UUID> --snapshot-dir=true --pool-level
=true -n trident
```

Vous pouvez activer par défaut les répertoires de snapshots pour les nouveaux volumes en configurant l'option de configuration du backend Trident . `snapshotDir` à `true` . Les volumes existants ne sont pas affectés.

## Protection des données avec les machines virtuelles KubeVirt

Trident Protect 24.10 et 24.10.1 et versions ultérieures ont un comportement différent lorsque vous protégez des applications exécutées sur des machines virtuelles KubeVirt. Pour les deux versions, vous pouvez activer ou désactiver le gel et le dégel du système de fichiers pendant les opérations de protection des données.

Lors des opérations de restauration, tout `VirtualMachineSnapshots` Les éléments créés pour une machine virtuelle (VM) ne sont pas restaurés.

**Trident Protect 24.10**

Trident Protect 24.10 ne garantit pas automatiquement un état cohérent pour les systèmes de fichiers de machines virtuelles KubeVirt lors des opérations de protection des données. Si vous souhaitez protéger les données de votre machine virtuelle KubeVirt à l'aide de Trident Protect 24.10, vous devez activer manuellement la fonctionnalité de gel/dégel des systèmes de fichiers avant l'opération de protection des données. Cela garantit que les systèmes de fichiers sont dans un état cohérent.

Vous pouvez configurer Trident Protect 24.10 pour gérer le gel et le dégel du système de fichiers de la machine virtuelle lors des opérations de protection des données."configuration de la virtualisation" puis en utilisant la commande suivante :

```
kubectl set env deployment/trident-protect-controller-manager
NEPTUNE_VM_FREEZE=true -n trident-protect
```

**Trident Protect 24.10.1 et versions ultérieures**

À partir de Trident Protect 24.10.1, Trident Protect gèle et débloque automatiquement les systèmes de fichiers KubeVirt lors des opérations de protection des données. Vous pouvez désactiver ce comportement automatique à l'aide de la commande suivante :

```
kubectl set env deployment/trident-protect-controller-manager
NEPTUNE_VM_FREEZE=false -n trident-protect
```

## Exigences pour la réplication SnapMirror

La réplication NetApp SnapMirror est disponible pour une utilisation avec Trident Protect pour les solutions ONTAP suivantes :

- Clusters NetApp FAS, AFF et ASA sur site
- ONTAP Select NetApp ONTAP
- NetApp Cloud Volumes ONTAP
- Amazon FSx for NetApp ONTAP

### Configuration requise pour la réplication SnapMirror dans un cluster ONTAP

Assurez-vous que votre cluster ONTAP répond aux exigences suivantes si vous prévoyez d'utiliser la réplication SnapMirror :

- * NetApp Trident* : NetApp Trident doit exister à la fois sur les clusters Kubernetes source et de destination qui utilisent ONTAP comme backend. Trident Protect prend en charge la réplication avec la technologie NetApp SnapMirror utilisant des classes de stockage reposant sur les pilotes suivants :
  - `ontap-nas`: NFS
  - `ontap-san`: iSCSI
  - `ontap-san`: FC
  - `ontap-san`: NVMe/TCP (nécessite au minimum la version ONTAP 9.15.1)
- **Licences** : Les licences asynchrones ONTAP SnapMirror utilisant le module de protection des données doivent être activées sur les clusters ONTAP source et de destination. Se référer à "Présentation des licences SnapMirror dans ONTAP" pour plus d'informations.

  À partir d' ONTAP 9.10.1, toutes les licences sont fournies sous forme de fichier de licence NetApp (NLF), qui est un fichier unique permettant d'activer plusieurs fonctionnalités. Se référer à"Licences incluses avec ONTAP One" pour plus d'informations.

  Seule la protection asynchrone SnapMirror est prise en charge.

#### Considérations relatives au peering pour la réplication SnapMirror

Si vous prévoyez d'utiliser le peering de stockage backend, assurez-vous que votre environnement réponde aux exigences suivantes :

- **Cluster et SVM** : Les backends de stockage ONTAP doivent être appariés. Se référer à "Aperçu du peering de clusters et de SVM" pour plus d'informations.

  Assurez-vous que les noms SVM utilisés dans la relation de réplication entre deux clusters ONTAP sont uniques.

- * NetApp Trident et SVM * : Les SVM distants appariés doivent être disponibles pour NetApp Trident sur le cluster de destination.
- **Systèmes de stockage backend gérés** : Vous devez ajouter et gérer des systèmes de stockage backend ONTAP dans Trident Protect pour créer une relation de réplication.

#### Configuration Trident / ONTAP pour la réplication SnapMirror

Trident Protect exige que vous configuriez au moins un système de stockage dorsal prenant en charge la réplication pour les clusters source et de destination. Si les clusters source et de destination sont identiques, l'application de destination doit utiliser un système de stockage différent de celui de l'application source pour une résilience optimale.

#### Configuration requise pour la réplication SnapMirror dans un cluster Kubernetes

Assurez-vous que vos clusters Kubernetes répondent aux exigences suivantes :

- **Accessibilité à AppVault** : Les clusters source et de destination doivent avoir un accès réseau pour lire et écrire dans AppVault pour la réplication des objets d'application.
- **Connectivité réseau** : Configurez les règles de pare-feu, les autorisations de compartiment et les listes d'adresses IP autorisées pour permettre la communication entre les deux clusters et AppVault via les réseaux WAN.

  De nombreux environnements d'entreprise mettent en œuvre des politiques de pare-feu strictes sur les connexions WAN. Vérifiez ces exigences réseau auprès de votre équipe d'infrastructure avant de configurer la réplication.

# Installez et configurez Trident Protect.

Si votre environnement répond aux exigences de Trident Protect, vous pouvez suivre ces étapes pour installer Trident Protect sur votre cluster. Vous pouvez obtenir Trident Protect auprès de NetApp ou l'installer à partir de votre propre registre privé. L'installation à partir d'un registre privé est utile si votre cluster ne peut pas accéder à Internet.

## Installer Trident Protect

**Installez Trident Protect de NetApp**

**Étapes**

1. Ajouter le dépôt Trident Helm :

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

2. Utilisez Helm pour installer Trident Protect. Remplacer `<name-of-cluster>` avec un nom de cluster, qui sera attribué au cluster et utilisé pour identifier les sauvegardes et les instantanés du cluster :

```
helm install trident-protect netapp-trident-protect/trident-protect
--set clusterName=<name-of-cluster> --version 100.2506.0 --create
-namespace --namespace trident-protect
```

**Installez Trident Protect à partir d'un registre privé**

Vous pouvez installer Trident Protect à partir d'un registre d'images privé si votre cluster Kubernetes ne peut pas accéder à Internet. Dans ces exemples, remplacez les valeurs entre crochets par les informations provenant de votre environnement :

**Étapes**

1. Téléchargez les images suivantes sur votre machine locale, mettez à jour les étiquettes, puis envoyez-les vers votre registre privé :

```
netapp/controller:25.06.0
netapp/restic:25.06.0
netapp/kopia:25.06.0
netapp/trident-autosupport:25.06.0
netapp/exechook:25.06.0
netapp/resourcebackup:25.06.0
netapp/resourcerestore:25.06.0
netapp/resourcedelete:25.06.0
bitnami/kubectl:1.30.2
kubebuilder/kube-rbac-proxy:v0.16.0
```

Par exemple:

```
docker pull netapp/controller:25.06.0
```

```
docker tag netapp/controller:25.06.0 <private-registry-
url>/controller:25.06.0
```

```
docker push <private-registry-url>/controller:25.06.0
```

2. Créez l'espace de noms système Trident Protect :

```
kubectl create ns trident-protect
```

3. Connectez-vous au registre :

```
helm registry login <private-registry-url> -u <account-id> -p <api-token>
```

4. Créez un secret d'extraction à utiliser pour l'authentification du registre privé :

```
kubectl create secret docker-registry regcred --docker-username=<registry-username> --docker-password=<api-token> -n trident-protect --docker-server=<private-registry-url>
```

5. Ajouter le dépôt Trident Helm :

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

6. Créez un fichier nommé `protectValues.yaml` . Assurez-vous qu'il contienne les paramètres Trident Protect suivants :

```
---
image:
  registry: <private-registry-url>
imagePullSecrets:
  - name: regcred
controller:
  image:
    registry: <private-registry-url>
rbacProxy:
  image:
    registry: <private-registry-url>
crCleanup:
  imagePullSecrets:
    - name: regcred
webhooksCleanup:
  imagePullSecrets:
    - name: regcred
```

7. Utilisez Helm pour installer Trident Protect. Remplacer `<name_of_cluster>` avec un nom de cluster, qui sera attribué au cluster et utilisé pour identifier les sauvegardes et les instantanés du cluster :

```
helm install trident-protect netapp-trident-protect/trident-protect
--set clusterName=<name_of_cluster> --version 100.2506.0 --create
-namespace --namespace trident-protect -f protectValues.yaml
```

# Installez le plugin CLI Trident Protect

Vous pouvez utiliser le plugin en ligne de commande Trident Protect, qui est une extension de Trident. `tridentctl` utilitaire, pour créer et interagir avec les ressources personnalisées (CR) de Trident Protect.

## Installez le plugin CLI Trident Protect

Avant d'utiliser l'utilitaire en ligne de commande, vous devez l'installer sur la machine que vous utilisez pour accéder à votre cluster. Suivez ces étapes, selon que votre machine utilise un processeur x64 ou ARM .

**Télécharger le plugin pour les processeurs Linux AMD64**

**Étapes**

1. Téléchargez le plugin CLI Trident Protect :

```
curl -L -o tridentctl-protect https://github.com/NetApp/tridentctl-protect/releases/download/25.06.0/tridentctl-protect-linux-amd64
```

**Télécharger le plugin pour les processeurs Linux ARM64**

**Étapes**

1. Téléchargez le plugin CLI Trident Protect :

```
curl -L -o tridentctl-protect https://github.com/NetApp/tridentctl-protect/releases/download/25.06.0/tridentctl-protect-linux-arm64
```

**Télécharger le plugin pour les processeurs Mac AMD64**

**Étapes**

1. Téléchargez le plugin CLI Trident Protect :

```
curl -L -o tridentctl-protect https://github.com/NetApp/tridentctl-protect/releases/download/25.06.0/tridentctl-protect-macos-amd64
```

**Télécharger le plugin pour les processeurs Mac ARM64**

**Étapes**

1. Téléchargez le plugin CLI Trident Protect :

```
curl -L -o tridentctl-protect https://github.com/NetApp/tridentctl-protect/releases/download/25.06.0/tridentctl-protect-macos-arm64
```

1. Activer les permissions d'exécution pour le fichier binaire du plugin :

```
chmod +x tridentctl-protect
```

2. Copiez le fichier binaire du plugin à un emplacement défini dans votre variable PATH. Par exemple, `/usr/bin` ou `/usr/local/bin` (Vous pourriez avoir besoin de privilèges élevés) :

```
cp ./tridentctl-protect /usr/local/bin/
```

3. Vous pouvez également copier le fichier binaire du plugin dans un emplacement de votre répertoire personnel. Dans ce cas, il est recommandé de s'assurer que l'emplacement fait partie de votre variable PATH :

```
cp ./tridentctl-protect ~/bin/
```

Copier le plugin dans un emplacement figurant dans votre variable PATH vous permet de l'utiliser en tapant : `tridentctl-protect` ou `tridentctl protect` de n'importe quel endroit.

## Consultez l'aide du plugin Trident CLI

Vous pouvez utiliser les fonctionnalités d'aide intégrées au plugin pour obtenir une aide détaillée sur ses capacités :

**Étapes**

1. Utilisez la fonction d'aide pour consulter les instructions d'utilisation :

```
tridentctl-protect help
```

## Activer la saisie semi-automatique des commandes

Une fois le plugin CLI Trident Protect installé, vous pouvez activer la saisie semi-automatique pour certaines commandes.

**Activer la saisie semi-automatique pour le shell Bash**

**Étapes**

1. Télécharger le script de complétion :

```
curl -L -O https://github.com/NetApp/tridentctl-
protect/releases/download/25.06.0/tridentctl-completion.bash
```

2. Créez un nouveau répertoire dans votre répertoire personnel pour y placer le script :

```
mkdir -p ~/.bash/completions
```

3. Déplacez le script téléchargé vers le `~/.bash/completions` annuaire:

```
mv tridentctl-completion.bash ~/.bash/completions/
```

4. Ajoutez la ligne suivante à la `~/.bashrc` fichier dans votre répertoire personnel :

```
source ~/.bash/completions/tridentctl-completion.bash
```

**Activer la saisie semi-automatique pour le shell Z**

**Étapes**

1. Télécharger le script de complétion :

```
curl -L -O https://github.com/NetApp/tridentctl-
protect/releases/download/25.06.0/tridentctl-completion.zsh
```

2. Créez un nouveau répertoire dans votre répertoire personnel pour y placer le script :

```
mkdir -p ~/.zsh/completions
```

3. Déplacez le script téléchargé vers le `~/.zsh/completions` annuaire:

```
mv tridentctl-completion.zsh ~/.zsh/completions/
```

4. Ajoutez la ligne suivante à la `~/.zprofile` fichier dans votre répertoire personnel :

```
source ~/.zsh/completions/tridentctl-completion.zsh
```

**Résultat**

Lors de votre prochaine connexion à l'interpréteur de commandes, vous pourrez utiliser la saisie semi-automatique des commandes grâce au plugin tridentctl-protect.

# Personnaliser l'installation de Trident Protect

Vous pouvez personnaliser la configuration par défaut de Trident Protect pour répondre aux exigences spécifiques de votre environnement.

## Spécifiez les limites de ressources du conteneur Trident Protect

Vous pouvez utiliser un fichier de configuration pour spécifier les limites de ressources des conteneurs Trident Protect après l'installation de Trident Protect. La définition de limites de ressources vous permet de contrôler la quantité de ressources du cluster consommées par les opérations de Trident Protect.

**Étapes**

1. Créez un fichier nommé `resourceLimits.yaml` .
2. Renseignez le fichier avec les options de limite de ressources pour les conteneurs Trident Protect en fonction des besoins de votre environnement.

   Le fichier de configuration d'exemple suivant présente les paramètres disponibles et contient les valeurs par défaut pour chaque limite de ressources :

```
---
jobResources:
  defaults:
    limits:
      cpu: 8000m
      memory: 10000Mi
      ephemeralStorage: ""
    requests:
      cpu: 100m
      memory: 100Mi
      ephemeralStorage: ""
  resticVolumeBackup:
    limits:
      cpu: ""
      memory: ""
      ephemeralStorage: ""
    requests:
      cpu: ""
      memory: ""
      ephemeralStorage: ""
  resticVolumeRestore:
    limits:
      cpu: ""
      memory: ""
```

```
      ephemeralStorage: ""
    requests:
      cpu: ""
      memory: ""
      ephemeralStorage: ""
  kopiaVolumeBackup:
    limits:
      cpu: ""
      memory: ""
      ephemeralStorage: ""
    requests:
      cpu: ""
      memory: ""
      ephemeralStorage: ""
  kopiaVolumeRestore:
    limits:
      cpu: ""
      memory: ""
      ephemeralStorage: ""
    requests:
      cpu: ""
      memory: ""
      ephemeralStorage: ""
```

3. Appliquez les valeurs de `resourceLimits.yaml` déposer:

```
helm upgrade trident-protect -n trident-protect netapp-trident-
protect/trident-protect -f resourceLimits.yaml --reuse-values
```

## Personnaliser les contraintes de contexte de sécurité

Vous pouvez utiliser un fichier de configuration pour modifier les contraintes de contexte de sécurité OpenShift (SCC) pour les conteneurs Trident Protect après avoir installé Trident Protect. Ces contraintes définissent les restrictions de sécurité des pods dans un cluster Red Hat OpenShift.

**Étapes**

1. Créez un fichier nommé `sccconfig.yaml` .
2. Ajoutez l'option SCC au fichier et modifiez les paramètres en fonction des besoins de votre environnement.

   L'exemple suivant illustre les valeurs par défaut des paramètres de l'option SCC :

```
scc:
  create: true
  name: trident-protect-job
  priority: 1
```

Ce tableau décrit les paramètres de l'option SCC :

| Paramètre | Description | Défaut |
|---|---|---|
| créer | Détermine si une ressource SCC peut être créée. Une ressource SCC ne sera créée que si `scc.create` est réglé sur `true` et le processus d'installation de Helm identifie un environnement OpenShift. Si vous n'utilisez pas OpenShift, ou si `scc.create` est réglé sur `false` Aucune ressource SCC ne sera créée. | true |
| nom | Spécifie le nom du SCC. | trident-protect-emploi |
| priorité | Définit la priorité du SCC. Les SCC ayant des valeurs de priorité plus élevées sont évaluées avant celles ayant des valeurs plus faibles. | 1 |

3. Appliquez les valeurs de `sccconfig.yaml` déposer:

```
helm upgrade trident-protect netapp-trident-protect/trident-protect -f
sccconfig.yaml --reuse-values
```

Cela remplacera les valeurs par défaut par celles spécifiées dans le `sccconfig.yaml` déposer.

## Configurer les paramètres supplémentaires du graphique de barre de Trident Protect

Vous pouvez personnaliser les paramètres AutoSupport et le filtrage des espaces de noms pour répondre à vos besoins spécifiques. Le tableau suivant décrit les paramètres de configuration disponibles :

| Paramètre | Type | Description |
| --- | --- | --- |
| autoSupport.proxy | chaîne | Configure une URL proxy pour les connexions NetApp AutoSupport . Utilisez ceci pour acheminer les téléchargements de bundles de support via un serveur proxy. Exemple: `http://my.proxy.url` . |
| autoSupport.insecure | booléen | Ignore la vérification TLS pour les connexions proxy AutoSupport lorsque cette option est activée. `true` . À utiliser uniquement pour les connexions proxy non sécurisées. (défaut: `false` ) |
| autoSupport.activé | booléen | Active ou désactive les téléchargements quotidiens des modules Trident Protect AutoSupport . Lorsqu'il est réglé sur `false` Les téléchargements quotidiens programmés sont désactivés, mais vous pouvez toujours générer manuellement des ensembles de support. (défaut: `true` ) |
| restaurerSkipNamespaceAnnotations | chaîne | Liste séparée par des virgules d'annotations d'espace de noms à exclure des opérations de sauvegarde et de restauration. Vous permet de filtrer les espaces de noms en fonction des annotations. |
| restaurerSkipNamespaceLabels | chaîne | Liste séparée par des virgules d'étiquettes d'espace de noms à exclure des opérations de sauvegarde et de restauration. Vous permet de filtrer les espaces de noms en fonction des étiquettes. |

Vous pouvez configurer ces options à l'aide d'un fichier de configuration YAML ou d'indicateurs de ligne de commande :

**Utiliser le fichier YAML**

**Étapes**

1. Créez un fichier de configuration et nommez-le `values.yaml` .
2. Dans le fichier que vous avez créé, ajoutez les options de configuration que vous souhaitez personnaliser.

```
autoSupport:
  enabled: false
  proxy: http://my.proxy.url
  insecure: true
restoreSkipNamespaceAnnotations: "annotation1,annotation2"
restoreSkipNamespaceLabels: "label1,label2"
```

3. Après avoir rempli le `values.yaml` Fichier contenant les valeurs correctes, appliquez le fichier de configuration :

```
helm upgrade trident-protect -n trident-protect netapp-trident-
protect/trident-protect -f values.yaml --reuse-values
```

**Utiliser l'indicateur CLI**

**Étapes**

1. Utilisez la commande suivante avec le `--set` indicateur permettant de spécifier des paramètres individuels :

```
helm upgrade trident-protect -n trident-protect netapp-trident-
protect/trident-protect \
  --set autoSupport.enabled=false \
  --set autoSupport.proxy=http://my.proxy.url \
  --set restoreSkipNamespaceAnnotations="annotation1,annotation2" \
  --set restoreSkipNamespaceLabels="label1,label2" \
  --reuse-values
```

## Limiter les pods Trident Protect à des nœuds spécifiques

Vous pouvez utiliser la contrainte de sélection de nœuds nodeSelector de Kubernetes pour contrôler quels nœuds sont éligibles pour exécuter des pods Trident Protect, en fonction des étiquettes de nœud. Par défaut, Trident Protect est limité aux nœuds exécutant Linux. Vous pouvez personnaliser davantage ces contraintes en fonction de vos besoins.

**Étapes**

1. Créez un fichier nommé `nodeSelectorConfig.yaml` .

2. Ajoutez l'option nodeSelector au fichier et modifiez ce dernier pour ajouter ou modifier les étiquettes des nœuds afin de les adapter aux besoins de votre environnement. Par exemple, le fichier suivant contient la restriction par défaut du système d'exploitation, mais cible également une région et un nom d'application spécifiques :

```
nodeSelector:
  kubernetes.io/os: linux
  region: us-west
  app.kubernetes.io/name: mysql
```

3. Appliquez les valeurs de `nodeSelectorConfig.yaml` déposer:

```
helm upgrade trident-protect -n trident-protect netapp-trident-
protect/trident-protect -f nodeSelectorConfig.yaml --reuse-values
```

Cela remplace les restrictions par défaut par celles que vous avez spécifiées dans le `nodeSelectorConfig.yaml` déposer.