Installer Trident Protect

Trident

NetApp
July 01, 2026

Sommaire

Installer Trident Protect

Exigences de Trident Protect

Commencez par vérifier la conformité 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 exploiter Trident Protect.

Compatibilité de Trident Protect avec les clusters Kubernetes

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

- Amazon Elastic Kubernetes Service (EKS)
- Google Kubernetes Engine (GKE)
- Microsoft Azure Kubernetes Service (AKS)
- Red Hat OpenShift
- SUSE Harvester 1.7.0 (ONTAP iSCSI)
- SUSE Rancher
- VMware Tanzu Portfolio
- 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 "ces instructions".
- Assurez-vous qu'au moins un VolumeSnapshotClass existe. Pour plus d'informations, consultez "VolumeSnapshotClass".
- Helm 4.x ou une version ultérieure est nécessaire pour installer Trident Protect.

Compatibilité du système de stockage Trident Protect

Trident Protect prend en charge les backends 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 stockage backend 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 NetApp ONTAP stockage backend.
- 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 nas-economy

Trident Protect prend en charge les opérations de sauvegarde et de restauration sur les volumes nas-economy. Les snapshots, les clones et la réplication SnapMirror vers les 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 utilisant un répertoire de snapshots. Pour ces applications, vous devez masquer le répertoire de snapshots 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>` par 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 définissant l'option de configuration du backend Trident `snapshotDir` sur `true`. Les volumes existants ne sont pas affectés.

Protection des données avec les machines virtuelles KubeVirt

Trident Protect offre des fonctionnalités de gel et de dégel du système de fichiers pour les machines virtuelles KubeVirt lors des opérations de protection des données afin de garantir la cohérence des données. La méthode de configuration et le comportement par défaut des opérations de gel des machines virtuelles varient selon les versions de Trident Protect, les versions plus récentes proposant une configuration simplifiée via les paramètres du chart Helm.

Lors des opérations de restauration, tout `VirtualMachineSnapshots` créé pour une machine virtuelle (VM) n'est pas restauré.

Trident Protect 25.10 et versions ultérieures

Trident Protect gèle et dégèle automatiquement les systèmes de fichiers KubeVirt lors des opérations de protection des données afin de garantir la cohérence. À partir de Trident Protect 25.10, vous pouvez désactiver ce comportement à l'aide du paramètre `vm.freeze` lors de l'installation du chart Helm. Le paramètre est activé par défaut.

```
helm install ... --set vm.freeze=false ...
```

Trident Protect 24.10.1 à 25.06

À partir de Trident Protect 24.10.1, Trident Protect gèle et dégèle 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
```

Trident Protect 24.10

Trident Protect 24.10 n'assure pas automatiquement un état cohérent pour les systèmes de fichiers KubeVirt des machines virtuelles lors des opérations de protection des données. Si vous souhaitez protéger les données de vos machines virtuelles KubeVirt avec Trident Protect 24.10, vous devez activer manuellement la fonctionnalité de gel/dégel pour les 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 en "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
```

Exigences pour la réplication SnapMirror

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

- Systèmes NetApp FAS, AFF et ASA sur site. La réplication SnapMirror avec Trident protect n'est actuellement pas prise en charge pour les systèmes ASA r2.
- NetApp ONTAP Select
- NetApp Cloud Volumes ONTAP
- Amazon FSx for NetApp ONTAP

Exigences du cluster ONTAP pour la réplication SnapMirror

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

- **NetApp Trident** : NetApp Trident doit exister 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 en utilisant des classes de stockage reposant sur les pilotes suivants :
  - `ontap-nas` : NFS
  - `ontap-san` : iSCSI
  - `ontap-san` : FC
  - `ontap-san`: NVMe/TCP (nécessite la version minimale ONTAP 9.15.1)
- **Licences** : Les licences asynchrones ONTAP SnapMirror utilisant le Data Protection bundle doivent être activées sur les clusters ONTAP source et de destination. Consultez "SnapMirror aperçu des licences dans ONTAP" pour plus d'informations.

  À compter de 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. Consultez "Licences incluses avec ONTAP One" pour plus d'informations.

  > Seule la protection asynchrone SnapMirror est prise en charge.

Considérations de peering pour la réplication SnapMirror

Assurez-vous que votre environnement répond aux exigences suivantes si vous prévoyez d'utiliser le peering de stockage backend :

- **Cluster et SVM** : Les backends de stockage ONTAP doivent être appariés. Consultez "Aperçu du peering de cluster 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 backends de stockage ONTAP dans Trident Protect pour créer une relation de réplication.

Configuration de Trident / ONTAP pour la réplication SnapMirror

Trident Protect exige que vous configuriez au moins un système de stockage 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.

Exigences d'un cluster Kubernetes pour la réplication SnapMirror

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

- **AppVault accessibilité** : Les clusters source et de destination doivent avoir un accès réseau pour lire

depuis et écrire vers le 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 à travers les WAN.

> De nombreux environnements d'entreprise appliquent des politiques de pare-feu strictes sur les connexions WAN. Vérifiez ces exigences réseau avec 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 depuis votre propre registre privé. L'installation depuis un registre privé est utile si votre cluster ne peut pas accéder à Internet.

Installer Trident Protect

Installez Trident Protect depuis NetApp

Étapes

1. Ajoutez 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. Remplacez `<name-of-cluster>` par le nom du cluster, qui sera attribué au cluster et utilisé pour identifier les sauvegardes et instantanés du cluster :

```
helm install trident-protect netapp-trident-protect/trident-protect
--set clusterName=<name-of-cluster> --version 100.2602.0 --create
-namespace --namespace trident-protect
```

3. Facultativement, pour activer la journalisation de débogage (recommandée pour le dépannage), utilisez :

```
helm install trident-protect netapp-trident-protect/trident-protect
--set clusterName=<name-of-cluster> --set logLevel=debug --version
100.2602.0 --create-namespace --namespace trident-protect
```

La journalisation de débogage aide le support NetApp à résoudre les problèmes sans nécessiter de modification du niveau de journalisation ni de reproduction du problème.

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 de votre environnement :

Étapes

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

```
docker.io/netapp/controller:26.02.0
docker.io/netapp/restic:26.02.0
docker.io/netapp/kopia:26.02.0
docker.io/netapp/kopiablockrestore:26.02.0
docker.io/netapp/trident-autosupport:26.02.0
docker.io/netapp/exechook:26.02.0
docker.io/netapp/resourcebackup:26.02.0
docker.io/netapp/resourcerestore:26.02.0
docker.io/netapp/resourcedelete:26.02.0
docker.io/netapp/trident-protect-utils:v1.0.0
```

Par exemple :

```
docker pull docker.io/netapp/controller:26.02.0
```

```
docker tag docker.io/netapp/controller:26.02.0 <private-registry-url>/controller:26.02.0
```

```
docker push <private-registry-url>/controller:26.02.0
```

> Pour obtenir le graphique Helm, téléchargez d'abord le graphique Helm sur une machine ayant accès à Internet à l'aide de `helm pull trident-protect --version 100.2602.0 --repo https://netapp.github.io/trident-protect-helm-chart`, puis copiez le fichier `trident-protect-100.2602.0.tgz` résultant dans votre environnement hors ligne et installez-le en utilisant `helm install trident-protect ./trident-protect-100.2602.0.tgz` au lieu de la référence du dépôt à l'étape finale.

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. Ajoutez 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 :

```
---
imageRegistry: <private-registry-url>
imagePullSecrets:
  - name: regcred
```

> Les `imageRegistry` et `imagePullSecrets` valeurs s'appliquent à toutes les images des composants, y compris `resourcebackup` et `resourcerestore`. Si vous envoyez des images vers un chemin de dépôt spécifique dans votre registre (par exemple, `example.com:443/my-repo`), indiquez le chemin complet dans le champ du registre. Cela garantira que toutes les images sont extraites de `<private-registry-url>/<image-name>:<tag>`.

7. Utilisez Helm pour installer Trident Protect. Remplacez `<name_of_cluster>` par le nom du cluster, qui sera attribué au cluster et utilisé pour identifier les sauvegardes et instantanés du cluster :

```
helm install trident-protect netapp-trident-protect/trident-protect
--set clusterName=<name_of_cluster> --version 100.2602.0 --create
-namespace --namespace trident-protect -f protectValues.yaml
```

8. Facultativement, pour activer la journalisation de débogage (recommandée pour le dépannage), utilisez :

```
helm install trident-protect netapp-trident-protect/trident-protect
--set clusterName=<name-of-cluster> --set logLevel=debug --version
100.2602.0 --create-namespace --namespace trident-protect -f
protectValues.yaml
```

La journalisation de débogage aide le support NetApp à résoudre les problèmes sans nécessiter de modification du niveau de journalisation ni de reproduction du problème.

Pour plus d'options de configuration du chart Helm, y compris les paramètres AutoSupport et le filtrage des espaces de noms, reportez-vous à "Personnaliser l'installation de Trident Protect".

Installez le plugin CLI Trident Protect

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

Installez le plugin CLI Trident Protect

Avant d'utiliser l'utilitaire de ligne de commandes, 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/26.02.0/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/26.02.0/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/26.02.0/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/26.02.0/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 dans un emplacement défini dans votre variable PATH. Par exemple, `/usr/bin` ou `/usr/local/bin` (des privilèges élevés peuvent être nécessaires) :

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

3. Vous pouvez également copier le fichier binaire du plugin dans un emplacement de votre répertoire local. Dans ce cas, il est recommandé de s'assurer que cet emplacement figure dans 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` depuis n'importe quel emplacement.

Afficher l'aide du plugin Trident CLI

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

Étapes

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

```
tridentctl-protect help
```

Activer la saisie semi-automatique des commandes

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

Activer la saisie semi-automatique pour le shell Bash

Étapes

1. Créez le script de complétion :

```
tridentctl-protect completion bash > tridentctl-completion.bash
```

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

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

3. Déplacez le script téléchargé dans le répertoire `~/.bash/completions` :

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

4. Ajoutez la ligne suivante au fichier `~/.bashrc` dans votre répertoire local :

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

Activer la saisie semi-automatique pour le Z shell

Étapes

1. Créez le script de complétion :

```
tridentctl-protect completion zsh > tridentctl-completion.zsh
```

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

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

3. Déplacez le script téléchargé dans le répertoire `~/.zsh/completions` :

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

4. Ajoutez la ligne suivante au fichier `~/.zprofile` dans votre répertoire local :

```
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 avec le 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 pour les conteneurs Trident Protect après avoir installé Trident Protect. La définition de limites de ressources vous permet de contrôler la quantité de ressources du cluster consommée 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.

   L'exemple de fichier de configuration 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 du fichier `resourceLimits.yaml` :

```
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 (SCCs) pour les conteneurs Trident Protect après avoir installé Trident Protect. Ces contraintes définissent les restrictions de sécurité pour les 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 selon les besoins de votre environnement.

   L'exemple suivant montre les valeurs par défaut des paramètres pour 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 sera créée uniquement si `scc.create` est définie sur `true` et si le processus d'installation Helm identifie un environnement OpenShift. Si ce n'est pas sur OpenShift, ou si `scc.create` est définie sur `false`, aucune ressource SCC ne sera créée.	true
nom	Spécifie le nom du SCC.	trident-protect-job
priorité	Définit la priorité du SCC. Les SCC ayant une valeur de priorité plus élevée sont évalués avant ceux ayant une valeur plus faible.	1

3. Appliquez les valeurs du fichier `sccconfig.yaml` :

```
helm upgrade trident-protect -n 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` fichier.

Configurer des paramètres supplémentaires du chart Helm 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 de proxy pour NetApp AutoSupport connexions. Utilisez ceci pour acheminer les chargements de bundles de support via un serveur proxy. Exemple : `http://my.proxy.url`.

Paramètre	Type	Description
autoSupport.insecure	booléen	Désactive la vérification TLS pour AutoSupport proxy connections lorsque défini sur `true`. À utiliser uniquement pour les connexions proxy non sécurisées. (par défaut : `false`)
AutoSupport.enabled	booléen	Active ou désactive les chargements quotidiens de bundles Trident Protect AutoSupport. Lorsqu'elle est définie sur `false`, les chargements quotidiens planifiés sont désactivés, mais vous pouvez toujours générer manuellement des bundles de support. (par défaut : `true`)
restoreSkipNamespaceAnnotations	chaîne	Liste d'annotations d'espace de noms, séparées par des virgules, à exclure des opérations de sauvegarde et de restauration. Permet de filtrer les espaces de noms en fonction des annotations.
restoreSkipNamespaceLabels	chaîne	Liste séparée par des virgules d'étiquettes d'espace de noms à exclure des opérations de sauvegarde et de restauration. 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 de paramètres de ligne de commandes :

Utiliser un 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 fichier `values.yaml` avec 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
```

Utilisez le paramètre CLI

Étapes

1. Utilisez la commande suivante avec le `--set` paramètre pour 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-string
restoreSkipNamespaceAnnotations="{annotation1,annotation2}" \
  --set-string restoreSkipNamespaceLabels="{label1,label2}" \
  --reuse-values
```

Limiter les pods Trident Protect à des nœuds spécifiques

Vous pouvez utiliser la contrainte de sélection de nœuds Kubernetes nodeSelector pour contrôler quels nœuds sont autorisés à 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 selon vos besoins.

Étapes

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

2. Ajoutez l'option nodeSelector au fichier et modifiez le fichier pour ajouter ou modifier les étiquettes des nœuds afin de les restreindre selon les besoins de votre environnement. Par exemple, le fichier suivant contient la restriction OS par défaut, mais cible également une région spécifique et un nom d'application :

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

3. Appliquez les valeurs du fichier `nodeSelectorConfig.yaml` :

```
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` fichier.