



Gérer Trident Protect

Trident

NetApp
October 30, 2026

Sommaire

- Gérer Trident Protect 1
 - Gérer l'autorisation Trident et le contrôle d'accès 1
 - Exemple : gérer l'accès pour deux groupes d'utilisateurs 1
- Surveiller les ressources de Trident Protect 7
 - Étape 1 : Installer les outils de surveillance 7
 - Étape 2 : Configurer les outils de surveillance pour qu'ils fonctionnent ensemble 9
 - Étape 3 : Configurer les alertes et les destinations d'alerte 10
- Générer un bundle de support Trident Protect 11
 - Surveiller et récupérer le bundle de support 12
- Mettre à niveau Trident Protect 13
 - Étape 1 : Sélectionnez une version 13
 - Étape 2 : Mise à niveau de Trident Protect 14

Gérer Trident Protect

Gérer l'autorisation Trident et le contrôle d'accès

Trident Protect utilise le modèle Kubernetes de contrôle d'accès basé sur les rôles (RBAC). Par défaut, Trident Protect fournit un espace de noms système unique et son compte de service par défaut associé. Si vous avez une organisation avec de nombreux utilisateurs ou des besoins de sécurité spécifiques, vous pouvez utiliser les fonctionnalités RBAC de Trident Protect pour obtenir un contrôle plus granulaire sur l'accès aux ressources et aux espaces de noms.

L'administrateur du cluster a toujours accès aux ressources dans l' `trident-protect` namespace par défaut, et peut également accéder aux ressources dans tous les autres namespaces. Pour contrôler l'accès aux ressources et aux applications, vous devez créer des namespaces supplémentaires et ajouter des ressources et des applications à ces namespaces.

Notez qu'aucun utilisateur ne peut créer des CR de gestion des données d'application dans l' `trident-protect` espace de noms par défaut. Vous devez créer des CR de gestion des données d'application dans un espace de noms d'application (comme bonne pratique, créez les CR de gestion des données d'application dans le même espace de noms que leur application associée).

Seuls les administrateurs devraient avoir accès aux objets de ressources personnalisés privilégiés de Trident Protect, notamment :



- **AppVault** : Nécessite des identifiants de compartiment
- **AutoSupportBundle** : Collecte des métriques, des journaux et d'autres données sensibles Trident Protect
- **AutoSupportBundleSchedule**: Gère les calendriers de collecte des journaux

En tant que bonne pratique, utilisez le RBAC pour limiter l'accès aux objets privilégiés aux administrateurs.

Pour plus d'informations sur la manière dont le RBAC régule l'accès aux ressources et aux espaces de noms, reportez-vous à la ["Documentation RBAC Kubernetes"](#).

Pour plus d'informations sur les comptes de service, veuillez vous référer au ["Documentation du compte de service Kubernetes"](#).

Exemple : gérer l'accès pour deux groupes d'utilisateurs

Par exemple, une organisation dispose d'un administrateur du cluster, d'un groupe d'utilisateurs ingénieurs et d'un groupe d'utilisateurs marketing. L'administrateur du cluster effectuerait les tâches suivantes pour créer un environnement où le groupe d'ingénierie et le groupe marketing ont chacun accès uniquement aux ressources attribuées à leurs espaces de noms respectifs.

Étape 1 : Créer un espace de noms pour contenir les ressources de chaque groupe

La création d'un espace de noms vous permet de séparer logiquement les ressources et de mieux contrôler qui a accès à ces ressources.

Étapes

1. Créer un espace de noms pour le groupe d'ingénierie :

```
kubectl create ns engineering-ns
```

2. Créez un espace de noms pour le groupe marketing :

```
kubectl create ns marketing-ns
```

Étape 2 : Créez de nouveaux comptes de service pour interagir avec les ressources dans chaque espace de noms

Chaque nouvel espace de noms que vous créez est fourni avec un compte de service par défaut, mais vous devez créer un compte de service pour chaque groupe d'utilisateurs afin de pouvoir diviser davantage les privilèges entre les groupes à l'avenir si nécessaire.

Étapes

1. Créer un compte de service pour le groupe d'ingénierie :

```
apiVersion: v1
kind: ServiceAccount
metadata:
  name: eng-user
  namespace: engineering-ns
```

2. Créer un compte de service pour le groupe marketing :

```
apiVersion: v1
kind: ServiceAccount
metadata:
  name: mkt-user
  namespace: marketing-ns
```

Étape 3 : Créez un secret pour chaque nouveau service account

Un secret de compte de service est utilisé pour s'authentifier avec le compte de service et peut facilement être supprimé et recréé en cas de compromission.

Étapes

1. Créez un secret pour le compte du service d'ingénierie :

```
apiVersion: v1
kind: Secret
metadata:
```

```

annotations:
  kubernetes.io/service-account.name: eng-user
name: eng-user-secret
namespace: engineering-ns
type: kubernetes.io/service-account-token

```

2. Créez un secret pour le compte du service marketing :

```

apiVersion: v1
kind: Secret
metadata:
  annotations:
    kubernetes.io/service-account.name: mkt-user
  name: mkt-user-secret
  namespace: marketing-ns
  type: kubernetes.io/service-account-token

```

Étape 4 : Créez un objet RoleBinding pour lier l'objet ClusterRole à chaque nouveau compte de service

Un objet ClusterRole par défaut est créé lors de l'installation de Trident Protect. Vous pouvez lier ce ClusterRole au compte de service en créant et en appliquant un objet RoleBinding.

Étapes

1. Liez le ClusterRole au compte de service d'ingénierie :

```

apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: engineering-ns-tenant-rolebinding
  namespace: engineering-ns
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: trident-protect-tenant-cluster-role
subjects:
- kind: ServiceAccount
  name: eng-user
  namespace: engineering-ns

```

2. Liez le ClusterRole au compte de service marketing :

```

apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:

```

```

name: marketing-ns-tenant-rolebinding
namespace: marketing-ns
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: trident-protect-tenant-cluster-role
subjects:
- kind: ServiceAccount
  name: mkt-user
  namespace: marketing-ns

```

Étape 5 : Tester les autorisations

Testez que les autorisations sont correctes.

Étapes

1. Confirmez que les utilisateurs ingénieurs peuvent accéder aux ressources d'ingénierie :

```

kubectl auth can-i --as=system:serviceaccount:engineering-ns:eng-user
get applications.protect.trident.netapp.io -n engineering-ns

```

2. Confirmez que les utilisateurs ingénieurs ne peuvent pas accéder aux ressources marketing :

```

kubectl auth can-i --as=system:serviceaccount:engineering-ns:eng-user
get applications.protect.trident.netapp.io -n marketing-ns

```

Étape 6 : Accorder l'accès aux objets AppVault

Pour effectuer des tâches de gestion des données telles que les sauvegardes et les instantanés, l'administrateur du cluster doit accorder l'accès aux objets AppVault à chaque utilisateur.

Étapes

1. Créez et appliquez un fichier YAML de combinaison AppVault et secret qui accorde à un utilisateur l'accès à un AppVault. Par exemple, le CR suivant accorde l'accès à un AppVault à l'utilisateur `eng-user` :

```

apiVersion: v1
data:
  accessKeyID: <ID_value>
  secretAccessKey: <key_value>
kind: Secret
metadata:
  name: appvault-for-eng-user-only-secret
  namespace: trident-protect
type: Opaque
---
```

```

apiVersion: protect.trident.netapp.io/v1
kind: AppVault
metadata:
  name: appvault-for-eng-user-only
  namespace: trident-protect # Trident Protect system namespace
spec:
  providerConfig:
    azure:
      accountName: ""
      bucketName: ""
      endpoint: ""
    gcp:
      bucketName: ""
      projectID: ""
    s3:
      bucketName: testbucket
      endpoint: 192.168.0.1:30000
      secure: "false"
      skipCertValidation: "true"
  providerCredentials:
    accessKeyID:
      valueFromSecret:
        key: accessKeyID
        name: appvault-for-eng-user-only-secret
    secretAccessKey:
      valueFromSecret:
        key: secretAccessKey
        name: appvault-for-eng-user-only-secret
  providerType: GenericS3

```

2. Créez et appliquez un Role CR pour permettre aux administrateurs du cluster d'accorder l'accès à des ressources spécifiques dans un espace de noms. Par exemple :

```

apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  name: eng-user-appvault-reader
  namespace: trident-protect
rules:
- apiGroups:
  - protect.trident.netapp.io
  resourceNames:
  - appvault-for-enguser-only
  resources:
  - appvaults

```

```
verbs:
- get
```

3. Créez et appliquez un CR RoleBinding pour lier les autorisations à l'utilisateur eng-user. Par exemple :

```
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: eng-user-read-appvault-binding
  namespace: trident-protect
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: Role
  name: eng-user-appvault-reader
subjects:
- kind: ServiceAccount
  name: eng-user
  namespace: engineering-ns
```

4. Vérifiez que les autorisations sont correctes.

a. Tentative de récupération des informations d'objet AppVault pour tous les espaces de noms :

```
kubectl get appvaults -n trident-protect
--as=system:serviceaccount:engineering-ns:eng-user
```

Vous devriez voir une sortie similaire à la suivante :

```
Error from server (Forbidden): appvaults.protect.trident.netapp.io is
forbidden: User "system:serviceaccount:engineering-ns:eng-user"
cannot list resource "appvaults" in API group
"protect.trident.netapp.io" in the namespace "trident-protect"
```

b. Vérifiez si l'utilisateur peut obtenir les informations AppVault auxquelles il a désormais accès :

```
kubectl auth can-i --as=system:serviceaccount:engineering-ns:eng-user
get appvaults.protect.trident.netapp.io/appvault-for-eng-user-only -n
trident-protect
```

Vous devriez voir une sortie similaire à la suivante :

```
yes
```

Résultat

Les utilisateurs auxquels vous avez accordé des autorisations AppVault doivent pouvoir utiliser les objets AppVault autorisés pour les opérations de gestion des données d'application, et ne doivent pas pouvoir accéder à des ressources en dehors des espaces de noms attribués ni créer de nouvelles ressources auxquelles ils n'ont pas accès.

Surveiller les ressources de Trident Protect

Vous pouvez utiliser les outils open source kube-state-metrics, Prometheus et Alertmanager pour surveiller la santé des ressources protégées par Trident Protect.

Le service kube-state-metrics génère des métriques à partir des communications avec l'API Kubernetes. L'utiliser avec Trident Protect expose des informations utiles sur l'état des ressources dans votre environnement.

Prometheus est un ensemble d'outils permettant d'ingérer les données générées par kube-state-metrics et de les présenter sous forme d'informations facilement lisibles sur ces objets. Ensemble, kube-state-metrics et Prometheus fournissent un moyen de surveiller la santé et l'état des ressources que vous gérez avec Trident Protect.

Alertmanager est un service qui ingère les alertes envoyées par des outils tels que Prometheus et les achemine vers des destinations que vous configurez.



Les configurations et les conseils fournis dans ces étapes ne sont que des exemples ; vous devez les adapter à votre environnement. Consultez la documentation officielle suivante pour obtenir des instructions et une assistance spécifiques :

- ["documentation kube-state-metrics"](#)
- ["Documentation Prometheus"](#)
- ["Documentation d'Alertmanager"](#)

Étape 1 : Installer les outils de surveillance

Pour activer la surveillance des ressources dans Trident Protect, vous devez installer et configurer kube-state-metrics, Prometheus, et Alertmanager.

Installer kube-state-metrics

Vous pouvez installer kube-state-metrics à l'aide de Helm.

Étapes

1. Ajoutez le graphique Helm kube-state-metrics. Par exemple :

```
helm repo add prometheus-community https://prometheus-  
community.github.io/helm-charts  
helm repo update
```

2. Appliquez le CRD ServiceMonitor Prometheus au cluster :

```
kubectl apply -f https://raw.githubusercontent.com/prometheus-operator/prometheus-operator/main/example/prometheus-operator-crd/monitoring.coreos.com_servicemonitors.yaml
```

3. Créez un fichier de configuration pour le Helm chart (par exemple, metrics-config.yaml). Vous pouvez personnaliser l'exemple de configuration suivant pour l'adapter à votre environnement :

metrics-config.yaml : configuration du Helm chart kube-state-metrics

```
---
extraArgs:
  # Collect only custom metrics
  - --custom-resource-state-only=true

customResourceState:
  enabled: true
  config:
    kind: CustomResourceStateMetrics
    spec:
      resources:
      - groupVersionKind:
          group: protect.trident.netapp.io
          kind: "Backup"
          version: "v1"
        labelsFromPath:
          backup_uid: [metadata, uid]
          backup_name: [metadata, name]
          creation_time: [metadata, creationTimestamp]
      metrics:
      - name: backup_info
        help: "Exposes details about the Backup state"
        each:
          type: Info
          info:
            labelsFromPath:
              appVaultReference: ["spec", "appVaultRef"]
              appReference: ["spec", "applicationRef"]
  rbac:
    extraRules:
    - apiGroups: ["protect.trident.netapp.io"]
      resources: ["backups"]
      verbs: ["list", "watch"]

# Collect metrics from all namespaces
namespaces: ""
```

```
# Ensure that the metrics are collected by Prometheus
prometheus:
  monitor:
    enabled: true
```

4. Installez kube-state-metrics en déployant le chart Helm. Par exemple :

```
helm install custom-resource -f metrics-config.yaml prometheus-
community/kube-state-metrics --version 5.21.0
```

5. Configurez kube-state-metrics pour générer des métriques pour les ressources personnalisées utilisées par Trident Protect en suivant les instructions dans le ["Documentation sur la ressource personnalisée kube-state-metrics"](#).

Installer Prometheus

Vous pouvez installer Prometheus en suivant les instructions dans le ["Documentation Prometheus"](#).

Installer Alertmanager

Vous pouvez installer Alertmanager en suivant les instructions dans le ["Documentation d'Alertmanager"](#).

Étape 2 : Configurer les outils de surveillance pour qu'ils fonctionnent ensemble

Après avoir installé les outils de surveillance, vous devez les configurer pour qu'ils fonctionnent ensemble.

Étapes

1. Intégrez kube-state-metrics à Prometheus. Modifiez le fichier de configuration Prometheus (`prometheus.yaml`) et ajoutez les informations du service kube-state-metrics. Par exemple :

prometheus.yaml: intégration du service kube-state-metrics avec Prometheus

```
---
apiVersion: v1
kind: ConfigMap
metadata:
  name: prometheus-config
  namespace: trident-protect
data:
  prometheus.yaml: |
    global:
      scrape_interval: 15s
    scrape_configs:
      - job_name: 'kube-state-metrics'
        static_configs:
          - targets: ['kube-state-metrics.trident-protect.svc:8080']
```

2. Configurez Prometheus pour acheminer les alertes vers Alertmanager. Modifiez le fichier de configuration Prometheus (`prometheus.yaml`) et ajoutez la section suivante :

prometheus.yaml : Envoyer des alertes à Alertmanager

```
alerting:
  alertmanagers:
    - static_configs:
      - targets:
        - alertmanager.trident-protect.svc:9093
```

Résultat

Prometheus peut désormais collecter des métriques depuis kube-state-metrics et peut envoyer des alertes à Alertmanager. Vous êtes maintenant prêt à configurer les conditions qui déclenchent une alerte et où les alertes doivent être envoyées.

Étape 3 : Configurer les alertes et les destinations d'alerte

Après avoir configuré les outils pour fonctionner ensemble, vous devez configurer le type d'informations qui déclenchent les alertes et l'endroit où les alertes doivent être envoyées.

Exemple d'alerte : échec de la sauvegarde

L'exemple suivant définit une alerte critique qui se déclenche lorsque l'état de la ressource personnalisée de sauvegarde est défini sur `Error` pendant 5 secondes ou plus. Vous pouvez personnaliser cet exemple pour l'adapter à votre environnement et inclure cet extrait YAML dans votre fichier de configuration

`prometheus.yaml` :

rules.yaml : Définir une alerte Prometheus pour les sauvegardes ayant échoué

```
rules.yaml: |
  groups:
    - name: fail-backup
      rules:
        - alert: BackupFailed
          expr: kube_customresource_backup_info{status="Error"}
          for: 5s
          labels:
            severity: critical
          annotations:
            summary: "Backup failed"
            description: "A backup has failed."
```

Configurez Alertmanager pour envoyer des alertes vers d'autres canaux

Vous pouvez configurer Alertmanager pour envoyer des notifications à d'autres canaux, tels que le courrier électronique, PagerDuty, Microsoft Teams ou d'autres services de notification, en spécifiant la configuration correspondante dans le `alertmanager.yaml` fichier.

L'exemple suivant configure Alertmanager pour envoyer des notifications à un canal Slack. Pour adapter cet exemple à votre environnement, remplacez la valeur de la `api_url` clé par l'URL du webhook Slack que vous utilisez :

alertmanager.yaml : Envoyer des alertes à un canal Slack

```
data:
  alertmanager.yaml: |
    global:
      resolve_timeout: 5m
    route:
      receiver: 'slack-notifications'
    receivers:
      - name: 'slack-notifications'
        slack_configs:
          - api_url: '<your-slack-webhook-url>'
            channel: '#failed-backups-channel'
            send_resolved: false
```

Générer un bundle de support Trident Protect

Trident Protect permet aux administrateurs de générer des bundles contenant des informations utiles à NetApp Support, notamment les journaux, les métriques et les informations de topologie des clusters et des applications gérés. Si vous êtes connecté à Internet, vous pouvez charger des bundles de support sur le NetApp Support Site (NSS) à l'aide d'un fichier de ressource personnalisé (CR).

Créez un bundle de support à l'aide d'un CR

Étapes

1. Créez le fichier de ressource personnalisée (CR) et nommez-le (par exemple, `trident-protect-support-bundle.yaml`).
2. Configurez les attributs suivants :
 - **metadata.name** : (*Obligatoire*) Le nom de cette ressource personnalisée; choisissez un nom unique et pertinent pour votre environnement.
 - **spec.triggerType** : (*Obligatoire*) Détermine si le support bundle est généré immédiatement ou planifié. La génération planifiée du support bundle a lieu à 12h00 UTC. Valeurs possibles :
 - Programmé
 - Manuel
 - **spec.uploadEnabled** : (*Facultatif*) Contrôle si le bundle de support doit être téléchargé sur le site de support NetApp après sa génération. Si non spécifié, la valeur par défaut est `false`. Valeurs possibles :
 - `true`
 - faux (default)

- **spec.dataWindowStart** : (*Facultatif*) Une chaîne de date au format RFC 3339 qui spécifie la date et l'heure auxquelles la fenêtre des données incluses dans le bundle de support doit commencer. Si elle n'est pas spécifiée, la valeur par défaut est il y a 24 heures. La date de fenêtre la plus ancienne que vous pouvez spécifier est il y a 7 jours.

Exemple YAML :

```
---
apiVersion: protect.trident.netapp.io/v1
kind: AutoSupportBundle
metadata:
  name: trident-protect-support-bundle
spec:
  triggerType: Manual
  uploadEnabled: true
  dataWindowStart: 2024-05-05T12:30:00Z
```

3. Après avoir rempli le `trident-protect-support-bundle.yaml` fichier avec les valeurs correctes, appliquez le CR :

```
kubectl apply -f trident-protect-support-bundle.yaml -n trident-protect
```

Créez un bundle de support à l'aide de la ligne de commandes

Étapes

1. Créez le bundle de support en remplaçant les valeurs entre crochets par les informations de votre environnement. Le `trigger-type` détermine si le bundle est créé immédiatement ou si l'heure de création est dictée par la planification, et peut être `Manual` ou `Scheduled`. Le paramètre par défaut est `Manual`.

Par exemple :

```
tridentctl-protect create autosupportbundle <my-bundle-name>
--trigger-type <trigger-type> -n trident-protect
```

Surveiller et récupérer le bundle de support

Après avoir créé un bundle de support en utilisant l'une ou l'autre méthode, vous pouvez suivre la progression de sa génération et le récupérer sur votre système local.

Étapes

1. Attendez que le `status.generationState` atteigne l'`Completed` état. Vous pouvez suivre la progression de la génération avec la commande suivante :

```
kubectl get autosupportbundle trident-protect-support-bundle -n trident-protect
```

2. Récupérez le support bundle sur votre système local. Obtenez la commande de copie à partir du bundle AutoSupport terminé :

```
kubectl describe autosupportbundle trident-protect-support-bundle -n trident-protect
```

Trouvez la commande `kubectl cp` dans le résultat et exécutez-la en remplaçant l'argument de destination par votre répertoire local préféré.

Mettre à niveau Trident Protect

Vous pouvez mettre à jour Trident Protect vers la dernière version pour bénéficier de nouvelles fonctionnalités ou de correctifs de bugs.



- Lors de la mise à niveau depuis la version 24.10, les instantanés en cours d'exécution pendant la mise à niveau peuvent échouer. Cet échec n'empêche pas la création future d'instantanés, qu'ils soient manuels ou planifiés. Si un instantané échoue pendant la mise à niveau, vous pouvez créer un nouvel instantané manuellement pour garantir la protection de votre application.
- Pour éviter d'éventuels dysfonctionnements, vous pouvez désactiver toutes les planifications de snapshots avant la mise à niveau et les réactiver ensuite. Cependant, cela entraîne l'absence de snapshots planifiés pendant la période de mise à niveau.
- Pour les installations de registre privé, assurez-vous que le Helm chart et les images requis pour la version cible sont disponibles dans votre registre privé, et vérifiez que vos valeurs Helm personnalisées sont compatibles avec la nouvelle version du chart. Pour plus d'informations, consultez ["Installez Trident Protect à partir d'un registre privé"](#).

Étape 1 : Sélectionnez une version

Les versions de Trident Protect suivent une convention de nommage basée sur la date `YY.MM`, où « YY » est les deux derniers chiffres de l'année et « MM » est le mois. Les versions dot suivent une `YY.MM.X` convention, où « X » est le niveau de correctif. Vous sélectionnez la version vers laquelle effectuer la mise à niveau en fonction de la version à partir de laquelle vous effectuez la mise à niveau.

- Vous pouvez effectuer une mise à niveau directe vers n'importe quelle version cible située dans un intervalle de quatre versions maximum par rapport à votre version installée. Par exemple, vous pouvez effectuer une mise à niveau directe de 24.10 (ou de toute version dérivée de 24.10) vers 25.10.
- Si vous effectuez une mise à niveau à partir d'une version en dehors de la fenêtre de quatre versions, procédez à une mise à niveau en plusieurs étapes. Utilisez les instructions de mise à niveau pour la ["version antérieure"](#) à partir de laquelle vous effectuez la mise à niveau afin de passer à la version la plus récente qui respecte la fenêtre de quatre versions. Par exemple, si vous utilisez la version 24.10 et souhaitez passer à la version 26.06 :

- a. Première mise à jour de 24.10 à 25.02.
- b. Ensuite, passez de 25.02 à 26.06.

Étape 2 : Mise à niveau de Trident Protect

Pour mettre à niveau Trident Protect, procédez comme suit.

Étapes

1. Mettre à jour le dépôt Trident Helm :

```
helm repo update
```

2. Mettez à niveau les CRD Trident Protect :



Cette étape est nécessaire si vous effectuez une mise à niveau depuis une version antérieure à la 25.06, car les CRD sont désormais inclus dans le chart Helm Trident Protect.

- a. Exécutez cette commande pour transférer la gestion des CRD de `trident-protect-crds` vers `trident-protect` :

```
kubectl get crd | grep protect.trident.netapp.io | awk '{print $1}' |  
xargs -I {} kubectl patch crd {} --type merge -p '{"metadata":  
{"annotations":{"meta.helm.sh/release-name": "trident-protect"}}}'
```

- b. Exécutez cette commande pour supprimer le secret Helm pour le `trident-protect-crds` chart :



Ne désinstallez pas le `trident-protect-crds` chart à l'aide de Helm, car cela pourrait supprimer vos CRDs et toutes les données associées.

```
kubectl delete secret -n trident-protect -l name=trident-protect-  
crds,owner=helm
```

3. Mise à niveau de Trident Protect :

```
helm upgrade trident-protect netapp-trident-protect/trident-protect  
--version 100.2606.0 --namespace trident-protect
```



Vous pouvez configurer le niveau de journalisation lors de la mise à niveau en ajoutant `--set logLevel=debug` à la commande de mise à niveau. Le niveau de journalisation par défaut est `warn`. La journalisation de débogage est recommandée pour le dépannage, car elle aide NetApp support à diagnostiquer les problèmes sans nécessiter de modification du niveau de journalisation ni de reproduction du problème.

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 DÉLICTELLE (Y COMPRIS LA NÉGLIGENCE OU AUTRE) DÉCOULANT DE L'UTILISATION DE CE LOGICIEL, MÊME SI LA SOCIÉTÉ A ÉTÉ INFORMÉE DE LA POSSIBILITÉ DE TELS DOMMAGES.

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

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

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

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

Informations sur les marques commerciales

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