



# **Protégez vos applications**

## **Astra Control Center**

NetApp

November 21, 2023

# Sommaire

- Protégez vos applications ..... 1
  - Présentation de la protection ..... 1
  - Protéger les applications avec les snapshots et les sauvegardes ..... 1
  - Restaurez les applications ..... 5
  - Cloner et migrer les applications ..... 6
  - Gérer les crochets d'exécution de l'application ..... 8

# Protégez vos applications

## Présentation de la protection

Vous pouvez créer des sauvegardes, des clones, des copies Snapshot et des règles de protection pour vos applications à l'aide d'Astra Control Center. La sauvegarde de vos applications aide vos services et vos données associées à être aussi disponibles que possible. En cas d'incident, la restauration à partir d'une sauvegarde permet une restauration complète d'une application et de ses données, avec une interruption minimale. Les sauvegardes, les clones et les snapshots contribuent à vous protéger contre les menaces classiques, comme les ransomwares, la perte accidentelle de données et les incidents environnementaux. ["Découvrez les types de protection des données disponibles dans Astra Control Center et le moment de les utiliser"](#).

## Workflow de protection des applications

Vous pouvez utiliser l'exemple de flux de travail suivant pour commencer à protéger vos applications.

### [Une seule] Sauvegardez toutes les applications

Pour être sûr que vos applications sont immédiatement protégées, ["créez une sauvegarde manuelle de toutes les applications"](#).

### [Deux] Configurez une stratégie de protection pour chaque application

Pour automatiser les sauvegardes et snapshots futurs, ["configurez une stratégie de protection pour chaque application"](#). Par exemple, vous pouvez commencer avec des sauvegardes hebdomadaires et des snapshots quotidiens, et en conserver un mois pour les deux. Il est fortement recommandé d'automatiser les sauvegardes et les snapshots avec une règle de protection par rapport aux sauvegardes et snapshots manuels.

### [Trois] Facultatif : ajustez les règles de protection

À mesure que les applications et leurs modèles d'utilisation évoluent, ajustez les règles de protection selon les besoins pour bénéficier d'une protection optimale.

### [Quatre] En cas d'incident, restaurez vos applications

En cas de perte de données, vous pouvez effectuer une restauration par ["restauration de la dernière sauvegarde"](#) d'abord pour chaque application. Vous pouvez alors restaurer le dernier snapshot (si disponible).

## Protéger les applications avec les snapshots et les sauvegardes

Protégez vos applications en effectuant des copies Snapshot et des sauvegardes à l'aide d'une stratégie de protection automatisée ou ad hoc. Vous pouvez utiliser l'interface utilisateur Astra ou ["API de contrôle Astra"](#) pour protéger les applications.



Si vous utilisez Helm pour déployer des applications, Astra Control Center requiert Helm version 3. La gestion et le clonage des applications déployées avec Helm 3 (ou mises à niveau de Helm 2 à Helm 3) sont entièrement pris en charge. Les applications déployées avec Helm 2 ne sont pas prises en charge.



Lorsque vous créez un projet d'hébergement d'une application sur un cluster OpenShift, un UID SecurityContext est attribué au projet (ou à l'espace de noms Kubernetes). Pour permettre à Astra Control Center de protéger votre application et de la déplacer vers un autre cluster ou projet dans OpenShift, vous devez ajouter des règles qui permettent à l'application de s'exécuter comme un UID. Par exemple, les commandes OpenShift CLI suivantes octroient les règles appropriées à une application WordPress.

```
oc new-project wordpress
oc adm policy add-scc-to-group anyuid system:serviceaccounts:wordpress
oc adm policy add-scc-to-user privileged -z default -n wordpress
```

## Configurer une règle de protection

Une règle de protection protège une application en créant des snapshots, des sauvegardes ou les deux à un calendrier défini. Vous pouvez choisir de créer des snapshots et des sauvegardes toutes les heures, tous les jours, toutes les semaines et tous les mois, et vous pouvez spécifier le nombre de copies à conserver. Par exemple, une règle de protection peut créer des sauvegardes hebdomadaires et des snapshots quotidiens, et conserver les sauvegardes et les snapshots pendant un mois. La fréquence de création des snapshots et des sauvegardes et la durée de conservation dépendent des besoins de votre entreprise.

### Étapes

1. Sélectionnez **applications**, puis le nom d'une application.
2. Sélectionnez **protection des données**.
3. Sélectionnez **configurer la stratégie de protection**.
4. Définissez un planning de protection en choisissant le nombre de snapshots et de sauvegardes pour conserver toutes les heures, tous les jours, toutes les semaines et tous les mois.

Vous pouvez définir les horaires horaires, quotidiens, hebdomadaires et mensuels simultanément. Un programme ne s'active pas tant que vous n'avez pas défini de niveau de rétention.

L'exemple suivant illustre quatre planifications de protection : toutes les heures, tous les jours, toutes les semaines et tous les mois pour les snapshots et les sauvegardes.

**Configure protection policy**

STEP 1/2: DETAILS

✕

PROTECTION SCHEDULE

🕒 Hourly

Every hour on the 0th minute, keep the last 4 snapshots

🕒 Daily

Daily at 02:00 (UTC), keep the last 15 snapshots

🕒 Weekly

Weekly on Mondays at 02:00 (UTC), keep the last 26 snapshots

🕒 Monthly

Every 1st of the month at 02:00 (UTC), keep the last 12 backups

● Hourly

● Daily

● **Weekly**

● Monthly

Select Weekday(s) (optional)

Monday X

Time (UTC) (optional)

02:00

– Snapshots to keep +

26

– Backups to keep +

0

BACKUP DESTINATION

Bucket

ntp-nautilus-bucket-10 - ntp-nautilus-bucket-10

Default

OVERVIEW

Schedule and retention

Define a policy to continuously protect your application on a schedule and configure a retention count to get started.

For select stateful applications, expect I/O to pause for a short time during a backup or snapshot operation.

Read more in [Protection policies](#)

Application

cattle-logging

Namespace

cattle-logging

Cluster

se-openlab-astra-enterprise-05-se-openlab-astra-enterprise-05-mstr-1

Cancel

Review →

5. Sélectionnez **Revue**.

6. Sélectionnez **définir la stratégie de protection**.

## Résultat

Astra Control Center implémente la règle de protection des données en créant et en conservant des snapshots et des sauvegardes à l'aide du calendrier et de la règle de conservation que vous avez définis.

## Créer un snapshot

Vous pouvez créer un snapshot à la demande à tout moment.

### Étapes

1. Sélectionnez **applications**.
2. Sélectionnez la liste déroulante dans la colonne **actions** pour l'application souhaitée.
3. Sélectionnez **instantané**.
4. Personnalisez le nom de l'instantané, puis sélectionnez **Review**.
5. Examinez le résumé de l'instantané et sélectionnez **instantané**.

## Résultat

Le processus d'instantané commence. Un instantané a réussi lorsque l'état est **disponible** dans la colonne **actions** de la page **protection des données > snapshots**.

## Créer une sauvegarde

Vous pouvez également sauvegarder une application à tout moment.



Les compartiments S3 du centre de contrôle Astra n'indiquent pas la capacité disponible. Avant de sauvegarder ou de cloner des applications gérées par Astra Control Center, vérifiez les informations de compartiment dans le système de gestion ONTAP ou StorageGRID.

## Étapes

1. Sélectionnez **applications**.
2. Sélectionnez la liste déroulante dans la colonne **actions** pour l'application souhaitée.
3. Sélectionnez **Backup**.
4. Personnaliser le nom de la sauvegarde.
5. Choisissez de sauvegarder l'application à partir d'un snapshot existant. Si vous sélectionnez cette option, vous pouvez choisir parmi une liste de snapshots existants.
6. Choisissez une destination pour la sauvegarde en sélectionnant dans la liste des compartiments de stockage.
7. Sélectionnez **Revue**.
8. Passez en revue le résumé des sauvegardes et sélectionnez **Backup**.

## Résultat

Astra Control Center crée une sauvegarde de l'application.



Si votre réseau est en panne ou anormalement lent, une opération de sauvegarde risque d'être terminée. Ceci entraîne l'échec de la sauvegarde.



Il est impossible d'arrêter une sauvegarde en cours d'exécution. Si vous devez supprimer la sauvegarde, attendez qu'elle soit terminée, puis suivez les instructions de la section [Supprimer les sauvegardes](#). Pour supprimer une sauvegarde défectueuse, "[Utilisez l'API de contrôle Astra](#)".



Après une opération de protection des données (clonage, sauvegarde, restauration) et après le redimensionnement du volume persistant, il y a vingt minutes de retard avant que la nouvelle taille du volume ne s'affiche dans l'interface utilisateur. La protection des données fonctionne avec succès en quelques minutes et vous pouvez utiliser le logiciel de gestion pour le système back-end pour confirmer la modification de la taille du volume.

## Afficher les snapshots et les sauvegardes

Vous pouvez afficher les instantanés et les sauvegardes d'une application à partir de l'onglet protection des données.

## Étapes

1. Sélectionnez **applications**, puis le nom d'une application.
2. Sélectionnez **protection des données**.

Les snapshots s'affichent par défaut.

3. Sélectionnez **backups** pour afficher la liste des sauvegardes.

## Supprimer les instantanés

Supprimez les snapshots programmés ou à la demande dont vous n'avez plus besoin.

### Étapes

1. Sélectionnez **applications**, puis le nom d'une application.
2. Sélectionnez **protection des données**.
3. Sélectionnez la liste déroulante dans la colonne **actions** pour l'instantané souhaité.
4. Sélectionnez **Supprimer instantané**.
5. Tapez le mot "supprimer" pour confirmer la suppression, puis sélectionnez **Oui, Supprimer l'instantané**.

### Résultat

Astra Control Center supprime le snapshot.

## Supprimer les sauvegardes

Supprimez les sauvegardes planifiées ou à la demande qui ne vous sont plus nécessaires.



Il est impossible d'arrêter une sauvegarde en cours d'exécution. Si vous devez supprimer la sauvegarde, attendez qu'elle soit terminée, puis suivez ces instructions. Pour supprimer une sauvegarde défaillante, ["Utilisez l'API de contrôle Astra"](#).

1. Sélectionnez **applications**, puis le nom d'une application.
2. Sélectionnez **protection des données**.
3. Sélectionnez **backups**.
4. Sélectionnez la liste déroulante dans la colonne **actions** pour la sauvegarde souhaitée.
5. Sélectionnez **Supprimer sauvegarde**.
6. Tapez le mot "supprimer" pour confirmer la suppression, puis sélectionnez **Oui, Supprimer sauvegarde**.

### Résultat

Astra Control Center supprime la sauvegarde.

## Restaurez les applications

Astra Control peut restaurer votre application à partir d'un snapshot ou d'une sauvegarde. La restauration d'un snapshot existant est plus rapide lors de la restauration d'une application sur le même cluster. Vous pouvez utiliser l'interface utilisateur de contrôle Astra ou ["API de contrôle Astra"](#) pour restaurer des applications.



Si vous utilisez Helm pour déployer des applications, Astra Control Center requiert Helm version 3. La gestion et le clonage des applications déployées avec Helm 3 (ou mises à niveau de Helm 2 à Helm 3) sont entièrement pris en charge. Les applications déployées avec Helm 2 ne sont pas prises en charge.



Si vous effectuez une restauration sur un autre cluster, assurez-vous que le cluster utilise le même mode d'accès aux volumes persistants (par exemple, ReadWriteMany). L'opération de restauration échoue si le mode d'accès au volume persistant de destination est différent.



Lorsque vous créez un projet d'hébergement d'une application sur un cluster OpenShift, un UID SecurityContext est attribué au projet (ou à l'espace de noms Kubernetes). Pour permettre à Astra Control Center de protéger votre application et de la déplacer vers un autre cluster ou projet dans OpenShift, vous devez ajouter des règles qui permettent à l'application de s'exécuter comme un UID. Par exemple, les commandes OpenShift CLI suivantes octroient les règles appropriées à une application WordPress.

```
oc new-project wordpress
oc adm policy add-scc-to-group anyuid system:serviceaccounts:wordpress
oc adm policy add-scc-to-user privileged -z default -n wordpress
```

## Étapes

1. Sélectionnez **applications**, puis le nom d'une application.
2. Sélectionnez **protection des données**.
3. Si vous souhaitez effectuer une restauration à partir d'un instantané, conservez l'icône **snapshots** sélectionnée. Sinon, sélectionnez l'icône **backups** pour restaurer à partir d'une sauvegarde.
4. Sélectionnez la liste déroulante dans la colonne **actions** pour l'instantané ou la sauvegarde à partir duquel vous souhaitez restaurer.
5. Sélectionnez **Restaurer l'application**.
6. **Détails de restauration** : spécifiez les détails de l'application restaurée. Par défaut, le cluster et l'espace de noms actuels apparaissent. Laissez ces valeurs intactes pour restaurer une application sur place, ce qui rétablit sa version antérieure. Modifiez ces valeurs si vous souhaitez restaurer vers un autre cluster ou espace de noms.
  - Entrez un nom et un espace de noms pour l'application.
  - Choisissez le cluster de destination de l'application.
  - Sélectionnez **Revue**.
7. **Résumé de restauration** : consultez les détails de l'action de restauration, tapez "Restaurer", puis sélectionnez **Restaurer**.

## Résultat

Astra Control Center restaure l'application en fonction des informations que vous avez fournies. Si vous avez restauré l'application sur place, le contenu des volumes persistants existants est remplacé par le contenu des volumes persistants de l'application restaurée.



Après une opération de protection des données (clonage, sauvegarde, restauration) et après le redimensionnement du volume persistant, il y a vingt minutes de retard avant que la nouvelle taille du volume ne s'affiche dans l'interface utilisateur. La protection des données fonctionne avec succès en quelques minutes et vous pouvez utiliser le logiciel de gestion pour le système back-end pour confirmer la modification de la taille du volume.

# Cloner et migrer les applications

Clonez une application existante pour créer une application dupliquée sur le même



cluster Kubernetes ou sur un autre cluster. Le clonage peut être utile pour déplacer des applications et du stockage d'un cluster Kubernetes vers un autre. Par exemple, il peut être intéressant de déplacer les workloads dans un pipeline ci/CD et entre les espaces de noms Kubernetes. Vous pouvez utiliser l'interface utilisateur Astra ou "[API de contrôle Astra](#)" clonage et migration des applications.



Si vous déployez une application avec une classe de stockage définie de manière explicite et que vous devez cloner l'application, le cluster cible doit avoir la classe de stockage spécifiée à l'origine. Le clonage d'une application avec une classe de stockage explicitement définie sur un cluster ne disposant pas de la même classe de stockage échouera.



Si vous clonez une instance déployée par l'opérateur de Jenkins ci, vous devez restaurer manuellement les données persistantes. Il s'agit d'une limitation du modèle de déploiement de l'application.



Si vous clonez une application entre les clusters, les clusters source et destination doivent être de la même distribution qu'OpenShift. Par exemple, si vous clonez une application depuis un cluster OpenShift 4.7, utilisez un cluster de destination qui est également OpenShift 4.7.

Lorsque Astra Control Center clone une application, il crée un clone de la configuration des applications et du stockage persistant.



Les compartiments S3 du centre de contrôle Astra n'indiquent pas la capacité disponible. Avant de sauvegarder ou de cloner des applications gérées par Astra Control Center, vérifiez les informations de compartiment dans le système de gestion ONTAP ou StorageGRID.



Lorsque vous créez un projet d'hébergement d'une application sur un cluster OpenShift, un UID SecurityContext est attribué au projet (ou à l'espace de noms Kubernetes). Pour permettre à Astra Control Center de protéger votre application et de la déplacer vers un autre cluster ou projet dans OpenShift, vous devez ajouter des règles qui permettent à l'application de s'exécuter comme un UID. Par exemple, les commandes OpenShift CLI suivantes octroient les règles appropriées à une application WordPress.

```
oc new-project wordpress
oc adm policy add-scc-to-group anyuid system:serviceaccounts:wordpress
oc adm policy add-scc-to-user privileged -z default -n wordpress
```

## Ce dont vous avez besoin

Pour cloner les applications vers un autre cluster, il vous faut un compartiment par défaut. Lorsque vous ajoutez votre premier compartiment, il devient le compartiment par défaut.

## Étapes

1. Sélectionnez **applications**.
2. Effectuez l'une des opérations suivantes :
  - Sélectionnez la liste déroulante dans la colonne **actions** pour l'application souhaitée.
  - Sélectionnez le nom de l'application souhaitée et sélectionnez la liste déroulante d'état en haut à droite de la page.
3. Sélectionnez **Clone**.

#### 4. **Détails du clone** : spécifiez les détails du clone :

- Entrez un nom.
- Entrez un namespace pour le clone.
- Choisissez un cluster de destination pour le clone.
- Indiquez si vous souhaitez créer le clone à partir d'un snapshot ou d'une sauvegarde existant. Si vous ne sélectionnez pas cette option, Astra Control Center crée le clone à partir de l'état actuel de l'application.

#### 5. **Source** : si vous choisissez de cloner à partir d'un snapshot ou d'une sauvegarde existant, choisissez le snapshot ou la sauvegarde que vous souhaitez utiliser.

#### 6. Sélectionnez **Revue**.

#### 7. **Résumé du clone** : consultez les détails sur le clone et sélectionnez **Clone**.

### Résultat

Astra Control Center clone cette application en fonction des informations que vous avez fournies. L'opération de clonage est réussie lorsque le nouveau clone d'application est dans `Available`. Indiquez la page **applications**.



Après une opération de protection des données (clonage, sauvegarde, restauration) et après le redimensionnement du volume persistant, il y a vingt minutes de retard avant que la nouvelle taille du volume ne s'affiche dans l'interface utilisateur. La protection des données fonctionne avec succès en quelques minutes et vous pouvez utiliser le logiciel de gestion pour le système back-end pour confirmer la modification de la taille du volume.

## Gérer les crochets d'exécution de l'application

Un crochet d'exécution est un script personnalisé que vous pouvez exécuter avant ou après un instantané d'une application gérée. Par exemple, si vous disposez d'une application de base de données, vous pouvez utiliser des crochets d'exécution pour interrompre toutes les transactions de base de données avant un instantané et reprendre les transactions une fois l'instantané terminé. Les snapshots sont ainsi cohérents au niveau des applications.

### Crochets d'exécution par défaut et expressions régulières

Pour certaines applications, Astra Control est doté de crochets d'exécution par défaut fournis par NetApp qui gèrent les opérations de gel et de dégel avant et après les snapshots. Astra Control utilise des expressions régulières pour faire correspondre l'image de conteneur d'une application à ces applications :

- MariaDB
  - Expression régulière correspondante : `\bmariadb\b`
- MySQL
  - Expression régulière correspondante : `\bmysql\b`
- PostgreSQL
  - Expression régulière correspondante : `\bpostgres\b`

S'il y a une correspondance, les crochets d'exécution par défaut fournis par NetApp pour cette application

apparaissent dans la liste des crochets d'exécution actifs de l'application, et ces crochets s'exécutent automatiquement lorsque des instantanés de cette application sont effectués. Si l'une de vos applications personnalisées possède un nom d'image similaire qui correspond à l'une des expressions régulières (et que vous ne souhaitez pas utiliser les crochets d'exécution par défaut), vous pouvez modifier le nom de l'image, ou désactivez le crochet d'exécution par défaut pour cette application et utilisez plutôt un crochet personnalisé.

Vous ne pouvez pas supprimer ou modifier les crochets d'exécution par défaut.

## Remarques importantes sur les crochets d'exécution personnalisés

Lors de la planification de crochets d'exécution pour vos applications, tenez compte des points suivants.

- Astra Control nécessite que les crochets d'exécution soient écrits au format de scripts shell exécutables.
- La taille du script est limitée à 128 Ko.
- Astra Control utilise les paramètres de crochet d'exécution et tout critère de correspondance pour déterminer quels crochets s'appliquent à un instantané.
- Toutes les défaillances de crochet d'exécution sont des défaillances logicielles ; d'autres crochets et l'instantané sont toujours tentés même en cas de défaillance d'un crochet. Cependant, lorsqu'un crochet échoue, un événement d'avertissement est enregistré dans le journal des événements de la page **activité**.
- Pour créer, modifier ou supprimer des crochets d'exécution, vous devez être un utilisateur disposant des autorisations propriétaire, administrateur ou membre.
- Si l'exécution d'un crochet d'exécution prend plus de 25 minutes, le crochet échoue, créant une entrée de journal d'événements avec un code retour « N/A ». Tout instantané affecté expire et sera marqué comme ayant échoué, avec une entrée du journal des événements qui en résulte indiquant le délai d'attente.



Puisque les crochets d'exécution réduisent souvent ou désactivent complètement la fonctionnalité de l'application contre laquelle ils sont en cours d'exécution, vous devez toujours essayer de réduire le temps d'exécution de vos crochets d'exécution personnalisés.

Lors de l'exécution d'un instantané, les événements de hook d'exécution ont lieu dans l'ordre suivant :

1. Tous les crochets d'exécution par défaut fournis par NetApp s'exécutent sur les conteneurs appropriés.
2. Tous les crochets d'exécution pré-snapshot personnalisés applicables sont exécutés sur les conteneurs appropriés. Vous pouvez créer et exécuter autant de crochets pré-snapshot personnalisés que vous le souhaitez, mais l'ordre d'exécution de ces crochets avant que le snapshot ne soit ni garanti ni configurable.
3. Le snapshot est effectué.
4. Tous les crochets d'exécution post-snapshot personnalisés applicables sont exécutés sur les conteneurs appropriés. Vous pouvez créer et exécuter autant de crochets post-snapshot personnalisés que vous le souhaitez, mais l'ordre d'exécution de ces crochets après le snapshot n'est ni garanti ni configurable.
5. Tous les crochets d'exécution post-snapshot par défaut fournis par NetApp s'exécutent sur les conteneurs appropriés.



Vous devez toujours tester vos scripts d'exécution avant de les activer dans un environnement de production. Vous pouvez utiliser la commande 'kubectl exec' pour tester aisément les scripts. Une fois que vous avez activé les crochets d'exécution dans un environnement de production, testez les snapshots obtenus pour vous assurer qu'ils sont cohérents. Pour ce faire, vous pouvez cloner l'application dans un espace de noms temporaire, restaurer le snapshot, puis tester l'application.

## Afficher les crochets d'exécution existants

Vous pouvez afficher les crochets d'exécution d'une application personnalisés ou par défaut fournis par NetApp.

### Étapes

1. Accédez à **applications**, puis sélectionnez le nom d'une application gérée.
2. Sélectionnez l'onglet **crochets d'exécution**.

Vous pouvez afficher tous les crochets d'exécution activés ou désactivés dans la liste résultante. Vous pouvez voir l'état d'un crochet, sa source et le moment où il est exécuté (pré ou post-instantané). Pour afficher les journaux d'événements entourant les crochets d'exécution, accédez à la page **activité** dans la zone de navigation de gauche.

## Créer un crochet d'exécution personnalisé

Vous pouvez créer un crochet d'exécution personnalisé pour une application. Voir ["Exemples de crochet d'exécution"](#) pour des exemples de crochet. Vous devez disposer d'autorisations propriétaire, administrateur ou membre pour créer des crochets d'exécution.



Lorsque vous créez un script de shell personnalisé à utiliser comme crochet d'exécution, n'oubliez pas de spécifier le shell approprié au début du fichier, sauf si vous exécutez des commandes linux ou fournissez le chemin complet à un exécutable.

### Étapes

1. Sélectionnez **applications**, puis le nom d'une application gérée.
2. Sélectionnez l'onglet **crochets d'exécution**.
3. Sélectionnez **Ajouter un nouveau crochet**.
4. Dans la zone **Détails du crochet**, selon le moment où le crochet doit être exécuté, choisissez **Préinstantané** ou **Post-instantané**.
5. Saisissez un nom unique pour le crochet.
6. (Facultatif) saisissez les arguments à transmettre au crochet pendant l'exécution, en appuyant sur la touche entrée après chaque argument que vous entrez pour enregistrer chacun.
7. Dans la zone **Images conteneur**, si le crochet doit être exécuté sur toutes les images de conteneur contenues dans l'application, activez la case à cocher **appliquer à toutes les images de conteneur**. Si, à la place, le crochet ne doit agir que sur une ou plusieurs images de conteneur spécifiées, entrez les noms d'image de conteneur dans le champ **noms d'image de conteneur à associer**.
8. Dans la zone **script**, effectuez l'une des opérations suivantes :
  - Charger un script personnalisé.
    - i. Sélectionnez l'option **Télécharger le fichier**.
    - ii. Accédez à un fichier et téléchargez-le.
    - iii. Donnez un nom unique au script.
    - iv. (Facultatif) Entrez toutes les notes que les autres administrateurs doivent connaître au sujet du script.
  - Coller dans un script personnalisé à partir du presse-papiers.
    - i. Sélectionnez l'option **Coller dans le presse-papiers**.

- ii. Sélectionnez le champ de texte et collez le texte du script dans le champ.
- iii. Donnez un nom unique au script.
- iv. (Facultatif) Entrez toutes les notes que les autres administrateurs doivent connaître au sujet du script.

9. Sélectionnez **Ajouter crochet**.

## Désactivez un crochet d'exécution

Vous pouvez désactiver un crochet d'exécution si vous souhaitez l'empêcher temporairement de s'exécuter avant ou après un instantané d'une application. Vous devez disposer d'autorisations propriétaire, administrateur ou membre pour désactiver les crochets d'exécution.

### Étapes

1. Sélectionnez **applications**, puis le nom d'une application gérée.
2. Sélectionnez l'onglet **crochets d'exécution**.
3. Sélectionnez la liste déroulante **actions** pour un crochet que vous souhaitez désactiver.
4. Sélectionnez **Désactiver**.

## Supprimer un crochet d'exécution

Vous pouvez supprimer entièrement un crochet d'exécution si vous n'en avez plus besoin. Vous devez disposer d'autorisations propriétaire, administrateur ou membre pour supprimer les crochets d'exécution.

### Étapes

1. Sélectionnez **applications**, puis le nom d'une application gérée.
2. Sélectionnez l'onglet **crochets d'exécution**.
3. Sélectionnez la liste déroulante **actions** pour un crochet que vous souhaitez supprimer.
4. Sélectionnez **Supprimer**.

## Exemples de crochet d'exécution

Utilisez les exemples suivants pour avoir une idée de la structure de vos crochets d'exécution. Vous pouvez utiliser ces crochets comme modèles ou comme scripts de test.

### Exemple de réussite simple

Voici un exemple de crochet simple qui réussit et écrit un message sur une sortie standard et une erreur standard.

```
#!/bin/sh

# success_sample.sh
#
# A simple noop success hook script for testing purposes.
#
```

```

# args: None
#

#
# Writes the given message to standard output
#
# $* - The message to write
#
msg() {
    echo "$*"
}

#
# Writes the given information message to standard output
#
# $* - The message to write
#
info() {
    msg "INFO: $*"
}

#
# Writes the given error message to standard error
#
# $* - The message to write
#
error() {
    msg "ERROR: $*" 1>&2
}

#
# main
#

# log something to stdout
info "running success_sample.sh"

# exit with 0 to indicate success
info "exit 0"
exit 0

```

### Exemple de réussite simple (version bash)

Voici un exemple de crochet simple qui réussit et écrit un message sur une sortie standard et une erreur standard, écrit pour bash.

```
#!/bin/bash

# success_sample.bash
#
# A simple noop success hook script for testing purposes.
#
# args: None

#
# Writes the given message to standard output
#
# $* - The message to write
#
msg() {
    echo "$*"
}

#
# Writes the given information message to standard output
#
# $* - The message to write
#
info() {
    msg "INFO: $*"
}

#
# Writes the given error message to standard error
#
# $* - The message to write
#
error() {
    msg "ERROR: $*" 1>&2
}

#
# main
#
```

```
# log something to stdout
info "running success_sample.bash"

# exit with 0 to indicate success
info "exit 0"
exit 0
```

### Exemple de réussite simple (version zsh)

Voici un exemple de crochet simple qui réussit et écrit un message sur une sortie standard et une erreur standard, écrite pour le shell Z.

```
#!/bin/zsh

# success_sample.zsh
#
# A simple noop success hook script for testing purposes.
#
# args: None
#

#
# Writes the given message to standard output
#
# $* - The message to write
#
msg() {
    echo "$*"
}

#
# Writes the given information message to standard output
#
# $* - The message to write
#
info() {
    msg "INFO: $*"
}

#
# Writes the given error message to standard error
#
# $* - The message to write
#
```



```

error() {
    msg "ERROR: $" 1>&2
}

#
# main
#

# log something to stdout
info "running success_sample.zsh"

# exit with 0 to indicate success
info "exit 0"
exit 0

```

### Exemple de réussite avec les arguments

L'exemple suivant montre comment utiliser des args dans un crochet.

```

#!/bin/sh

# success_sample_args.sh
#
# A simple success hook script with args for testing purposes.
#
# args: Up to two optional args that are echoed to stdout
#
# Writes the given message to standard output
#
# $* - The message to write
#
msg() {
    echo "$*"
}

#
# Writes the given information message to standard output
#
# $* - The message to write
#
info() {
    msg "INFO: $"
}

```

```

#
# Writes the given error message to standard error
#
# $* - The message to write
#
error() {
    msg "ERROR: $*" 1>&2
}

#
# main
#

# log something to stdout
info "running success_sample_args.sh"

# collect args
arg1=$1
arg2=$2

# output args and arg count to stdout
info "number of args: $#"
```

```

info "arg1 ${arg1}"
info "arg2 ${arg2}"

# exit with 0 to indicate success
info "exit 0"
exit 0

```

### Exemple de crochet pré-instantané/post-instantané

L'exemple suivant montre comment le même script peut être utilisé à la fois pour un pré-snapshot et un crochet post-snapshot.

```

#!/bin/sh

# success_sample_pre_post.sh
#
# A simple success hook script example with an arg for testing purposes
# to demonstrate how the same script can be used for both a prehook and
# posthook
#
# args: [pre|post]

```

```

# unique error codes for every error case
ebase=100
eusage=$((ebase+1))
ebadstage=$((ebase+2))
epre=$((ebase+3))
epost=$((ebase+4))

#
# Writes the given message to standard output
#
# $* - The message to write
#
msg() {
    echo "$*"
}

#
# Writes the given information message to standard output
#
# $* - The message to write
#
info() {
    msg "INFO: $*"
}

#
# Writes the given error message to standard error
#
# $* - The message to write
#
error() {
    msg "ERROR: $*" 1>&2
}

#
# Would run prehook steps here
#
prehook() {
    info "Running noop prehook"
    return 0
}

#

```

```

# Would run posthook steps here
#
posthook() {
    info "Running noop posthook"
    return 0
}

#
# main
#

# check arg
stage=$1
if [ -z "${stage}" ]; then
    echo "Usage: $0 <pre|post>"
    exit ${eusage}
fi

if [ "${stage}" != "pre" ] && [ "${stage}" != "post" ]; then
    echo "Invalid arg: ${stage}"
    exit ${ebadstage}
fi

# log something to stdout
info "running success_sample_pre_post.sh"

if [ "${stage}" = "pre" ]; then
    prehook
    rc=$?
    if [ ${rc} -ne 0 ]; then
        error "Error during prehook"
    fi
fi

if [ "${stage}" = "post" ]; then
    posthook
    rc=$?
    if [ ${rc} -ne 0 ]; then
        error "Error during posthook"
    fi
fi

exit ${rc}

```

## Exemple de panne

L'exemple suivant montre comment vous pouvez gérer les défaillances d'un crochet.

```
#!/bin/sh

# failure_sample_arg_exit_code.sh
#
# A simple failure hook script for testing purposes.
#
# args: [the exit code to return]
#

#
# Writes the given message to standard output
#
# $* - The message to write
#
msg() {
    echo "$*"
}

#
# Writes the given information message to standard output
#
# $* - The message to write
#
info() {
    msg "INFO: $*"
}

#
# Writes the given error message to standard error
#
# $* - The message to write
#
error() {
    msg "ERROR: $*" 1>&2
}

#
# main
#
```

```
# log something to stdout
info "running failure_sample_arg_exit_code.sh"

argexitcode=$1

# log to stderr
error "script failed, returning exit code ${argexitcode}"

# exit with specified exit code
exit ${argexitcode}
```

## Exemple détaillé d'échec

L'exemple suivant montre comment gérer les défaillances d'un crochet, avec une consignation plus détaillée.

```
#!/bin/sh

# failure_sample_verbose.sh
#
# A simple failure hook script with args for testing purposes.
#
# args: [The number of lines to output to stdout]

#
# Writes the given message to standard output
#
# $* - The message to write
#
msg() {
    echo "$*"
}

#
# Writes the given information message to standard output
#
# $* - The message to write
#
info() {
    msg "INFO: $*"
}

#
# Writes the given error message to standard error
#
```

```

# $* - The message to write
#
error() {
    msg "ERROR: $" 1>&2
}

#
# main
#

# log something to stdout
info "running failure_sample_verbose.sh"

# output arg value to stdout
linecount=$1
info "line count ${linecount}"

# write out a line to stdout based on line count arg
i=1
while [ "$i" -le ${linecount} ]; do
    info "This is line ${i} from failure_sample_verbose.sh"
    i=$(( i + 1 ))
done

error "exiting with error code 8"
exit 8

```

## Échec avec un exemple de code de sortie

L'exemple suivant illustre l'échec d'un crochet avec un code de sortie.

```

#!/bin/sh

# failure_sample_arg_exit_code.sh
#
# A simple failure hook script for testing purposes.
#
# args: [the exit code to return]
#

#
# Writes the given message to standard output
#

```

```

# $* - The message to write
#
msg() {
    echo "$*"
}

#
# Writes the given information message to standard output
#
# $* - The message to write
#
info() {
    msg "INFO: $*"
}

#
# Writes the given error message to standard error
#
# $* - The message to write
#
error() {
    msg "ERROR: $*" 1>&2
}

#
# main
#

# log something to stdout
info "running failure_sample_arg_exit_code.sh"

argexitcode=$1

# log to stderr
error "script failed, returning exit code ${argexitcode}"

# exit with specified exit code
exit ${argexitcode}

```

### Exemple de succès après échec

L'exemple suivant illustre l'échec d'un crochet lors de sa première exécution, mais la réussite après la seconde course.



```
#!/bin/sh

# failure_then_success_sample.sh
#
# A hook script that fails on initial run but succeeds on second run for
testing purposes.
#
# Helpful for testing retry logic for post hooks.
#
# args: None
#

#
# Writes the given message to standard output
#
# $* - The message to write
#
msg() {
    echo "$*"
}

#
# Writes the given information message to standard output
#
# $* - The message to write
#
info() {
    msg "INFO: $*"
}

#
# Writes the given error message to standard error
#
# $* - The message to write
#
error() {
    msg "ERROR: $*" 1>&2
}

#
# main
#

# log something to stdout
```

```
info "running failure_success sample.sh"

if [ -e /tmp/hook-test.junk ] ; then
    info "File does exist. Removing /tmp/hook-test.junk"
    rm /tmp/hook-test.junk
    info "Second run so returning exit code 0"
    exit 0
else
    info "File does not exist. Creating /tmp/hook-test.junk"
    echo "test" > /tmp/hook-test.junk
    error "Failed first run, returning exit code 5"
    exit 5
fi
```

## Informations sur le copyright

Copyright © 2023 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.