Restauration des applications

Trident

NetApp
February 02, 2026

Sommaire

Restauration des applications

Restaurez les applications à l'aide de Trident Protect

Vous pouvez utiliser Trident Protect pour restaurer votre application à partir d'un instantané ou d'une sauvegarde. La restauration à partir d'un instantané existant sera plus rapide lors de la restauration de l'application sur le même cluster.

- Lorsque vous restaurez une application, tous les crochets d'exécution configurés pour l'application sont restaurés avec l'application. Si un hook d'exécution post-restauration est présent, il s'exécute automatiquement dans le cadre de l'opération de restauration.
- La restauration à partir d'une sauvegarde vers un espace de noms différent ou vers l'espace de noms d'origine est prise en charge pour les volumes qtree. En revanche, la restauration à partir d'un snapshot vers un espace de noms différent ou vers l'espace de noms d'origine n'est pas prise en charge pour les volumes qtree.
- Vous pouvez utiliser des paramètres avancés pour personnaliser les opérations de restauration. Pour en savoir plus, consultez "Utilisez les paramètres de restauration avancés de Trident Protect".

Restauration d'une sauvegarde vers un autre espace de noms

Lorsque vous restaurez une sauvegarde dans un espace de noms différent à l'aide d'une ressource personnalisée BackupRestore, Trident Protect restaure l'application dans un nouvel espace de noms et crée une ressource personnalisée d'application pour l'application restaurée. Pour protéger l'application restaurée, créez des sauvegardes ou des instantanés à la demande, ou établissez un calendrier de protection.

- La restauration d'une sauvegarde dans un espace de noms différent avec des ressources existantes ne modifie aucune ressource qui partage des noms avec ceux de la sauvegarde. Pour restaurer toutes les ressources de la sauvegarde, supprimez et recréez l'espace de noms cible ou restaurez la sauvegarde dans un nouvel espace de noms.
- Lorsque vous utilisez une modification de configuration (CR) pour restaurer un nouvel espace de noms, vous devez créer manuellement l'espace de noms de destination avant d'appliquer la CR. Trident Protect crée automatiquement des espaces de noms uniquement lors de l'utilisation de l'interface de ligne de commande (CLI).

Avant de commencer

Assurez-vous que l'expiration du jeton de session AWS suffit pour toutes les opérations de restauration s3 à long terme. Si le jeton expire pendant l'opération de restauration, l'opération peut échouer.

- Pour plus d'informations sur la vérification de l'expiration du jeton de session en cours, reportez-vous "Documentation de l'API AWS" au.
- Pour plus d'informations sur les identifiants avec les ressources AWS, consultez le "Documentation AWS IAM".

Lorsque vous restaurez des sauvegardes à l'aide de Kopia comme outil de transfert de données, vous pouvez éventuellement spécifier des annotations dans le CR ou utiliser l'interface de ligne de commande pour contrôler le comportement du stockage temporaire utilisé par Kopia. Se référer à "Documentation Kopia" pour plus d'informations sur les options que vous pouvez configurer. Utilisez le `tridentctl-protect create --help` commande pour plus d'informations sur la spécification des annotations avec l'interface de ligne de commande Trident Protect.

Utiliser une CR

Étapes

1. Créez le fichier de ressource personnalisée (CR) et nommez-le `trident-protect-backup-restore-cr.yaml`.
2. Dans le fichier que vous avez créé, configurez les attributs suivants :
   - **metadata.name**: (*required*) le nom de cette ressource personnalisée; choisissez un nom unique et sensible pour votre environnement.
   - **Spec.appArchivePath** : chemin d'accès dans AppVault où sont stockés le contenu de la sauvegarde. Vous pouvez utiliser la commande suivante pour trouver ce chemin :

     ```
     kubectl get backups <BACKUP_NAME> -n my-app-namespace -o
     jsonpath='{.status.appArchivePath}'
     ```

   - **Spec.appVaultRef**: (*required*) Nom de l'AppVault où sont stockés le contenu de la sauvegarde.
   - **spec.namespaceMapping**: mappage de l'espace de noms source de l'opération de restauration sur l'espace de noms de destination. Remplacez `my-source-namespace` et `my-destination-namespace` par des informations provenant de votre environnement.

     ```
     ---
     apiVersion: protect.trident.netapp.io/v1
     kind: BackupRestore
     metadata:
       name: my-cr-name
       namespace: my-destination-namespace
     spec:
       appArchivePath: my-backup-path
       appVaultRef: appvault-name
       namespaceMapping: [{"source": "my-source-namespace",
     "destination": "my-destination-namespace"}]
     ```

3. (*Facultatif*) si vous devez sélectionner uniquement certaines ressources de l'application à restaurer, ajoutez un filtrage qui inclut ou exclut les ressources marquées avec des étiquettes particulières :

   > Trident Protect sélectionne automatiquement certaines ressources en raison de leur relation avec les ressources que vous sélectionnez. Par exemple, si vous sélectionnez une ressource de revendication de volume persistant et qu'elle possède un pod associé, Trident Protect restaurera également le pod associé.

   - **ResourceFilter.resourceSelectionCriteria**: (Requis pour le filtrage) utiliser `Include` ou `Exclude` inclure ou exclure une ressource définie dans resourceMatchers. Ajoutez les paramètres resourceMatchers suivants pour définir les ressources à inclure ou à exclure :
     - **ResourceFilter.resourceMatchers** : un tableau d'objets resourceMatcher. Si vous définissez plusieurs éléments dans ce tableau, ils correspondent en tant qu'opération OU et les champs de chaque élément (groupe, type, version) correspondent en tant qu'opération ET.

- **ResourceMatchers[].group**: (*Optional*) Groupe de la ressource à filtrer.
- **ResourceMatchers[].kind**: (*Optional*) Type de la ressource à filtrer.
- **ResourceMatchers[].version**: (*Optional*) version de la ressource à filtrer.
- **ResourceMatchers[].names**: (*Optional*) noms dans le champ Kubernetes metadata.name de la ressource à filtrer.
- **ResourceMatchers[].namespaces**: (*Optional*) Namespaces dans le champ Kubernetes metadata.name de la ressource à filtrer.
- **ResourceMatchers[].labelSelectors**: (*Optional*) chaîne de sélecteur de libellé dans le champ Kubernetes metadata.name de la ressource, comme défini dans le "Documentation Kubernetes". Par exemple : `"trident.netapp.io/os=linux"`.

  Par exemple :

```
spec:
  resourceFilter:
    resourceSelectionCriteria: "Include"
    resourceMatchers:
      - group: my-resource-group-1
        kind: my-resource-kind-1
        version: my-resource-version-1
        names: ["my-resource-names"]
        namespaces: ["my-resource-namespaces"]
        labelSelectors: ["trident.netapp.io/os=linux"]
      - group: my-resource-group-2
        kind: my-resource-kind-2
        version: my-resource-version-2
        names: ["my-resource-names"]
        namespaces: ["my-resource-namespaces"]
        labelSelectors: ["trident.netapp.io/os=linux"]
```

4. Une fois que vous avez rempli le `trident-protect-backup-restore-cr.yaml` fichier avec les valeurs correctes, appliquez la CR :

```
kubectl apply -f trident-protect-backup-restore-cr.yaml
```

Utiliser l'interface de ligne de commande

Étapes

1. Restaurez la sauvegarde dans un espace de noms différent, en remplaçant les valeurs entre parenthèses par les informations de votre environnement. L' `namespace-mapping`argument utilise des espaces de noms séparés par deux-points pour mapper les espaces de noms source aux espaces de noms de destination corrects dans le format `source1:dest1,source2:dest2`. Par exemple :

```
tridentctl-protect create backuprestore <my_restore_name> \
--backup <backup_namespace>/<backup_to_restore> \
--namespace-mapping <source_to_destination_namespace_mapping> \
-n <application_namespace>
```

Restaurer à partir d'une sauvegarde vers l'espace de noms d'origine

Vous pouvez à tout moment restaurer une sauvegarde dans l'espace de noms d'origine.

Avant de commencer

Assurez-vous que l'expiration du jeton de session AWS suffit pour toutes les opérations de restauration s3 à long terme. Si le jeton expire pendant l'opération de restauration, l'opération peut échouer.

- Pour plus d'informations sur la vérification de l'expiration du jeton de session en cours, reportez-vous "Documentation de l'API AWS" au.
- Pour plus d'informations sur les identifiants avec les ressources AWS, consultez le "Documentation AWS IAM".

> Lorsque vous restaurez des sauvegardes à l'aide de Kopia comme outil de transfert de données, vous pouvez éventuellement spécifier des annotations dans le CR ou utiliser l'interface de ligne de commande pour contrôler le comportement du stockage temporaire utilisé par Kopia. Se référer à "Documentation Kopia" pour plus d'informations sur les options que vous pouvez configurer. Utilisez le `tridentctl-protect create --help` commande pour plus d'informations sur la spécification des annotations avec l'interface de ligne de commande Trident Protect.

Utiliser une CR

Étapes

1. Créez le fichier de ressource personnalisée (CR) et nommez-le `trident-protect-backup-ipr-cr.yaml`.
2. Dans le fichier que vous avez créé, configurez les attributs suivants :
   - **metadata.name**: (*required*) le nom de cette ressource personnalisée; choisissez un nom unique et sensible pour votre environnement.
   - **Spec.appArchivePath** : chemin d'accès dans AppVault où sont stockés le contenu de la sauvegarde. Vous pouvez utiliser la commande suivante pour trouver ce chemin :

     ```
     kubectl get backups <BACKUP_NAME> -n my-app-namespace -o
     jsonpath='{.status.appArchivePath}'
     ```

   - **Spec.appVaultRef**: (*required*) Nom de l'AppVault où sont stockés le contenu de la sauvegarde.

     Par exemple :

```
---
apiVersion: protect.trident.netapp.io/v1
kind: BackupInplaceRestore
metadata:
  name: my-cr-name
  namespace: my-app-namespace
spec:
  appArchivePath: my-backup-path
  appVaultRef: appvault-name
```

3. (*Facultatif*) si vous devez sélectionner uniquement certaines ressources de l'application à restaurer, ajoutez un filtrage qui inclut ou exclut les ressources marquées avec des étiquettes particulières :

   Trident Protect sélectionne automatiquement certaines ressources en raison de leur relation avec les ressources que vous sélectionnez. Par exemple, si vous sélectionnez une ressource de revendication de volume persistant et qu'elle possède un pod associé, Trident Protect restaurera également le pod associé.

   - **ResourceFilter.resourceSelectionCriteria**: (Requis pour le filtrage) utiliser `Include` ou `Exclude` inclure ou exclure une ressource définie dans resourceMatchers. Ajoutez les paramètres resourceMatchers suivants pour définir les ressources à inclure ou à exclure :
     - **ResourceFilter.resourceMatchers** : un tableau d'objets resourceMatcher. Si vous définissez plusieurs éléments dans ce tableau, ils correspondent en tant qu'opération OU et les champs de chaque élément (groupe, type, version) correspondent en tant qu'opération ET.
       - **ResourceMatchers[].group**: (*Optional*) Groupe de la ressource à filtrer.
       - **ResourceMatchers[].kind**: (*Optional*) Type de la ressource à filtrer.

- **ResourceMatchers[].version**: (*Optional*) version de la ressource à filtrer.
- **ResourceMatchers[].names**: (*Optional*) noms dans le champ Kubernetes metadata.name de la ressource à filtrer.
- **ResourceMatchers[].namespaces**: (*Optional*) Namespaces dans le champ Kubernetes metadata.name de la ressource à filtrer.
- **ResourceMatchers[].labelSelectors**: (*Optional*) chaîne de sélecteur de libellé dans le champ Kubernetes metadata.name de la ressource, comme défini dans le "Documentation Kubernetes". Par exemple : `"trident.netapp.io/os=linux"`.

  Par exemple :

```
spec:
  resourceFilter:
    resourceSelectionCriteria: "Include"
    resourceMatchers:
      - group: my-resource-group-1
        kind: my-resource-kind-1
        version: my-resource-version-1
        names: ["my-resource-names"]
        namespaces: ["my-resource-namespaces"]
        labelSelectors: ["trident.netapp.io/os=linux"]
      - group: my-resource-group-2
        kind: my-resource-kind-2
        version: my-resource-version-2
        names: ["my-resource-names"]
        namespaces: ["my-resource-namespaces"]
        labelSelectors: ["trident.netapp.io/os=linux"]
```

4. Une fois que vous avez rempli le `trident-protect-backup-ipr-cr.yaml` fichier avec les valeurs correctes, appliquez la CR :

```
kubectl apply -f trident-protect-backup-ipr-cr.yaml
```

Utiliser l'interface de ligne de commande

Étapes

1. Restaurez la sauvegarde dans l'espace de noms d'origine en remplaçant les valeurs entre parenthèses par les informations de votre environnement. L'`backup`argument utilise un nom d'espace de noms et un nom de sauvegarde au format `<namespace>/<name>. Par exemple :

```
tridentctl-protect create backupinplacerestore <my_restore_name> \
--backup <namespace/backup_to_restore> \
-n <application_namespace>
```

Restauration à partir d'une sauvegarde sur un autre cluster

Vous pouvez restaurer une sauvegarde sur un autre cluster en cas de problème avec le cluster d'origine.

- Lorsque vous restaurez des sauvegardes à l'aide de Kopia comme outil de transfert de données, vous pouvez éventuellement spécifier des annotations dans le CR ou utiliser l'interface de ligne de commande pour contrôler le comportement du stockage temporaire utilisé par Kopia. Se référer à "Documentation Kopia" pour plus d'informations sur les options que vous pouvez configurer. Utilisez le `tridentctl-protect create --help` commande pour plus d'informations sur la spécification des annotations avec l'interface de ligne de commande Trident Protect.
- Lorsque vous utilisez une modification de configuration (CR) pour restaurer un nouvel espace de noms, vous devez créer manuellement l'espace de noms de destination avant d'appliquer la CR. Trident Protect crée automatiquement des espaces de noms uniquement lors de l'utilisation de l'interface de ligne de commande (CLI).

Avant de commencer

Assurez-vous que les conditions préalables suivantes sont remplies :

- Le cluster de destination possède Trident Protect installé.
- Le cluster de destination a accès au chemin de compartiment du même AppVault que le cluster source, où la sauvegarde est stockée.
- Assurez-vous que votre environnement local peut se connecter au compartiment de stockage d'objets défini dans la ressource personnalisée AppVault lors de l'exécution. `tridentctl-protect get appvaultcontent` commande. Si des restrictions réseau empêchent l'accès, exécutez plutôt l'interface de ligne de commande Trident Protect depuis un pod sur le cluster de destination.
- Assurez-vous que l'expiration du jeton de session AWS suffit pour toutes les opérations de restauration à long terme. Si le jeton expire pendant l'opération de restauration, l'opération peut échouer.
  - Pour plus d'informations sur la vérification de l'expiration du jeton de session en cours, reportez-vous "Documentation de l'API AWS" au.
  - Pour plus d'informations sur les identifiants avec les ressources AWS, consultez le "Documentation de l'AWS".

Étapes

1. Vérifiez la disponibilité de la ressource personnalisée AppVault sur le cluster de destination à l'aide du plugin CLI Trident Protect :

```
tridentctl-protect get appvault --context <destination_cluster_name>
```

   Assurez-vous que l'espace de noms destiné à la restauration d'application existe sur le cluster de destination.

2. Afficher le contenu de la sauvegarde de l'AppVault disponible à partir du cluster de destination :

```
tridentctl-protect get appvaultcontent <appvault_name> \
--show-resources backup \
--show-paths \
--context <destination_cluster_name>
```

L'exécution de cette commande affiche les sauvegardes disponibles dans le AppVault, y compris leurs clusters d'origine, les noms d'applications correspondants, les horodatages et les chemins d'archivage.

Exemple de sortie :

```
+-------------+-----------+--------+-----------------
+--------------------------+-------------+
|   CLUSTER   |    APP    |  TYPE  |      NAME       |        TIMESTAMP
|    PATH     |
+-------------+-----------+--------+-----------------
+--------------------------+-------------+
| production1 | wordpress | backup | wordpress-bkup-1| 2024-10-30
08:37:40 (UTC)| backuppath1 |
| production1 | wordpress | backup | wordpress-bkup-2| 2024-10-30
08:37:40 (UTC)| backuppath2 |
+-------------+-----------+--------+-----------------
+--------------------------+-------------+
```

3. Restaurez l'application sur le cluster de destination à l'aide du nom AppVault et du chemin d'archivage :

Utiliser une CR

1. Créez le fichier de ressource personnalisée (CR) et nommez-le `trident-protect-backup-restore-cr.yaml`.
2. Dans le fichier que vous avez créé, configurez les attributs suivants :
   - **metadata.name**: (*required*) le nom de cette ressource personnalisée; choisissez un nom unique et sensible pour votre environnement.
   - **Spec.appVaultRef**: (*required*) Nom de l'AppVault où sont stockés le contenu de la sauvegarde.
   - **Spec.appArchivePath** : chemin d'accès dans AppVault où sont stockés le contenu de la sauvegarde. Vous pouvez utiliser la commande suivante pour trouver ce chemin :

     ```
     kubectl get backups <BACKUP_NAME> -n my-app-namespace -o
     jsonpath='{.status.appArchivePath}'
     ```

     Si BackupRestore CR n'est pas disponible, vous pouvez utiliser la commande mentionnée à l'étape 2 pour afficher le contenu de la sauvegarde.

   - **spec.namespaceMapping**: mappage de l'espace de noms source de l'opération de restauration sur l'espace de noms de destination. Remplacez `my-source-namespace` et `my-destination-namespace` par des informations provenant de votre environnement.

     Par exemple :

```
apiVersion: protect.trident.netapp.io/v1
kind: BackupRestore
metadata:
  name: my-cr-name
  namespace: my-destination-namespace
spec:
  appVaultRef: appvault-name
  appArchivePath: my-backup-path
  namespaceMapping: [{"source": "my-source-namespace", "
destination": "my-destination-namespace"}]
```

3. Une fois que vous avez rempli le `trident-protect-backup-restore-cr.yaml` fichier avec les valeurs correctes, appliquez la CR :

```
kubectl apply -f trident-protect-backup-restore-cr.yaml
```

Utiliser l'interface de ligne de commande

1. Utilisez la commande suivante pour restaurer l'application, en remplaçant les valeurs entre parenthèses par les informations de votre environnement. L'argument namespace-mapping utilise des espaces de noms séparés par deux points pour mapper les espaces de noms source aux

espaces de noms de destination corrects au format source1:dest1,source2:dest2. Par exemple :

```
tridentctl-protect create backuprestore <restore_name> \
--namespace-mapping <source_to_destination_namespace_mapping> \
--appvault <appvault_name> \
--path <backup_path> \
--context <destination_cluster_name> \
-n <application_namespace>
```

Restauration d'un snapshot vers un autre espace de noms

Vous pouvez restaurer des données à partir d'un instantané à l'aide d'un fichier de ressources personnalisé (CR), soit dans un espace de noms différent, soit dans l'espace de noms source d'origine. Lorsque vous restaurez un instantané dans un espace de noms différent à l'aide d'une ressource personnalisée SnapshotRestore, Trident Protect restaure l'application dans un nouvel espace de noms et crée une ressource personnalisée d'application pour l'application restaurée. Pour protéger l'application restaurée, créez des sauvegardes ou des instantanés à la demande, ou établissez un calendrier de protection.

- SnapshotRestore prend en charge le `spec.storageClassMapping` attribut, mais uniquement lorsque les classes de stockage source et de destination utilisent le même backend de stockage. Si vous tentez de restaurer un `StorageClass` qui utilise un backend de stockage différent, l'opération de restauration échouera.
- Lorsque vous utilisez une modification de configuration (CR) pour restaurer un nouvel espace de noms, vous devez créer manuellement l'espace de noms de destination avant d'appliquer la CR. Trident Protect crée automatiquement des espaces de noms uniquement lors de l'utilisation de l'interface de ligne de commande (CLI).

Avant de commencer

Assurez-vous que l'expiration du jeton de session AWS suffit pour toutes les opérations de restauration s3 à long terme. Si le jeton expire pendant l'opération de restauration, l'opération peut échouer.

- Pour plus d'informations sur la vérification de l'expiration du jeton de session en cours, reportez-vous "Documentation de l'API AWS" au.
- Pour plus d'informations sur les identifiants avec les ressources AWS, consultez le "Documentation AWS IAM".

Utiliser une CR

Étapes

1. Créez le fichier de ressource personnalisée (CR) et nommez-le `trident-protect-snapshot-restore-cr.yaml`.
2. Dans le fichier que vous avez créé, configurez les attributs suivants :
   - **metadata.name**: (*required*) le nom de cette ressource personnalisée; choisissez un nom unique et sensible pour votre environnement.
   - **Spec.appVaultRef**: (*required*) le nom du AppVault dans lequel le contenu de l'instantané est stocké.
   - **Spec.appArchivePath** : chemin d'accès dans AppVault où sont stockés le contenu de l'instantané. Vous pouvez utiliser la commande suivante pour trouver ce chemin :

     ```
     kubectl get snapshots <SNAPHOT_NAME> -n my-app-namespace -o
     jsonpath='{.status.appArchivePath}'
     ```

   - **spec.namespaceMapping**: mappage de l'espace de noms source de l'opération de restauration sur l'espace de noms de destination. Remplacez `my-source-namespace` et `my-destination-namespace` par des informations provenant de votre environnement.

     ```
     ---
     apiVersion: protect.trident.netapp.io/v1
     kind: SnapshotRestore
     metadata:
       name: my-cr-name
       namespace: my-app-namespace
     spec:
       appVaultRef: appvault-name
       appArchivePath: my-snapshot-path
       namespaceMapping: [{"source": "my-source-namespace",
     "destination": "my-destination-namespace"}]
     ```

3. (*Facultatif*) si vous devez sélectionner uniquement certaines ressources de l'application à restaurer, ajoutez un filtrage qui inclut ou exclut les ressources marquées avec des étiquettes particulières :

   > Trident Protect sélectionne automatiquement certaines ressources en raison de leur relation avec les ressources que vous sélectionnez. Par exemple, si vous sélectionnez une ressource de revendication de volume persistant et qu'elle possède un pod associé, Trident Protect restaurera également le pod associé.

   - **ResourceFilter.resourceSelectionCriteria**: (Requis pour le filtrage) utiliser `Include` ou `Exclude` inclure ou exclure une ressource définie dans resourceMatchers. Ajoutez les paramètres resourceMatchers suivants pour définir les ressources à inclure ou à exclure :
     - **ResourceFilter.resourceMatchers** : un tableau d'objets resourceMatcher. Si vous définissez plusieurs éléments dans ce tableau, ils correspondent en tant qu'opération OU et les champs

de chaque élément (groupe, type, version) correspondent en tant qu'opération ET.

- **ResourceMatchers[].group**: (*Optional*) Groupe de la ressource à filtrer.
- **ResourceMatchers[].kind**: (*Optional*) Type de la ressource à filtrer.
- **ResourceMatchers[].version**: (*Optional*) version de la ressource à filtrer.
- **ResourceMatchers[].names**: (*Optional*) noms dans le champ Kubernetes metadata.name de la ressource à filtrer.
- **ResourceMatchers[].namespaces**: (*Optional*) Namespaces dans le champ Kubernetes metadata.name de la ressource à filtrer.
- **ResourceMatchers[].labelSelectors**: (*Optional*) chaîne de sélecteur de libellé dans le champ Kubernetes metadata.name de la ressource, comme défini dans le "Documentation Kubernetes". Par exemple : `"trident.netapp.io/os=linux"`.

  Par exemple :

```
spec:
  resourceFilter:
    resourceSelectionCriteria: "Include"
    resourceMatchers:
      - group: my-resource-group-1
        kind: my-resource-kind-1
        version: my-resource-version-1
        names: ["my-resource-names"]
        namespaces: ["my-resource-namespaces"]
        labelSelectors: ["trident.netapp.io/os=linux"]
      - group: my-resource-group-2
        kind: my-resource-kind-2
        version: my-resource-version-2
        names: ["my-resource-names"]
        namespaces: ["my-resource-namespaces"]
        labelSelectors: ["trident.netapp.io/os=linux"]
```

4. Une fois que vous avez rempli le `trident-protect-snapshot-restore-cr.yaml` fichier avec les valeurs correctes, appliquez la CR :

```
kubectl apply -f trident-protect-snapshot-restore-cr.yaml
```

Utiliser l'interface de ligne de commande

Étapes

1. Restaurez l'instantané dans un autre espace de noms, en remplaçant les valeurs entre parenthèses par les informations de votre environnement.

   - L'`snapshot`argument utilise un nom d'espace de noms et un nom d'instantané au format `<namespace>/<name>.
   - L'`namespace-mapping`argument utilise des espaces de noms séparés par

deux-points pour mapper les espaces de noms source aux espaces de noms de destination corrects dans le format `source1:dest1,source2:dest2.

Par exemple :

```
tridentctl-protect create snapshotrestore <my_restore_name> \
--snapshot <namespace/snapshot_to_restore> \
--namespace-mapping <source_to_destination_namespace_mapping> \
-n <application_namespace>
```

Restaurer à partir d'un snapshot vers l'espace de noms d'origine

Vous pouvez à tout moment restaurer un snapshot dans l'espace de noms d'origine.

Avant de commencer

Assurez-vous que l'expiration du jeton de session AWS suffit pour toutes les opérations de restauration s3 à long terme. Si le jeton expire pendant l'opération de restauration, l'opération peut échouer.

- Pour plus d'informations sur la vérification de l'expiration du jeton de session en cours, reportez-vous "Documentation de l'API AWS" au.
- Pour plus d'informations sur les identifiants avec les ressources AWS, consultez le "Documentation AWS IAM".

Utiliser une CR

Étapes

1. Créez le fichier de ressource personnalisée (CR) et nommez-le `trident-protect-snapshot-ipr-cr.yaml`.
2. Dans le fichier que vous avez créé, configurez les attributs suivants :
   - **metadata.name**: (*required*) le nom de cette ressource personnalisée; choisissez un nom unique et sensible pour votre environnement.
   - **Spec.appVaultRef**: (*required*) le nom du AppVault dans lequel le contenu de l'instantané est stocké.
   - **Spec.appArchivePath** : chemin d'accès dans AppVault où sont stockés le contenu de l'instantané. Vous pouvez utiliser la commande suivante pour trouver ce chemin :

```
kubectl get snapshots <SNAPSHOT_NAME> -n my-app-namespace -o
jsonpath='{.status.appArchivePath}'
```

```
---
apiVersion: protect.trident.netapp.io/v1
kind: SnapshotInplaceRestore
metadata:
  name: my-cr-name
  namespace: my-app-namespace
spec:
  appVaultRef: appvault-name
  appArchivePath: my-snapshot-path
```

3. (*Facultatif*) si vous devez sélectionner uniquement certaines ressources de l'application à restaurer, ajoutez un filtrage qui inclut ou exclut les ressources marquées avec des étiquettes particulières :

> Trident Protect sélectionne automatiquement certaines ressources en raison de leur relation avec les ressources que vous sélectionnez. Par exemple, si vous sélectionnez une ressource de revendication de volume persistant et qu'elle possède un pod associé, Trident Protect restaurera également le pod associé.

   - **ResourceFilter.resourceSelectionCriteria**: (Requis pour le filtrage) utiliser `Include` ou `Exclude` inclure ou exclure une ressource définie dans resourceMatchers. Ajoutez les paramètres resourceMatchers suivants pour définir les ressources à inclure ou à exclure :
     - **ResourceFilter.resourceMatchers** : un tableau d'objets resourceMatcher. Si vous définissez plusieurs éléments dans ce tableau, ils correspondent en tant qu'opération OU et les champs de chaque élément (groupe, type, version) correspondent en tant qu'opération ET.
       - **ResourceMatchers[].group**: (*Optional*) Groupe de la ressource à filtrer.
       - **ResourceMatchers[].kind**: (*Optional*) Type de la ressource à filtrer.
       - **ResourceMatchers[].version**: (*Optional*) version de la ressource à filtrer.

- **ResourceMatchers[].names**: (*Optional*) noms dans le champ Kubernetes metadata.name de la ressource à filtrer.
- **ResourceMatchers[].namespaces**: (*Optional*) Namespaces dans le champ Kubernetes metadata.name de la ressource à filtrer.
- **ResourceMatchers[].labelSelectors**: (*Optional*) chaîne de sélecteur de libellé dans le champ Kubernetes metadata.name de la ressource, comme défini dans le "Documentation Kubernetes". Par exemple : `"trident.netapp.io/os=linux"`.

  Par exemple :

```
spec:
  resourceFilter:
    resourceSelectionCriteria: "Include"
    resourceMatchers:
      - group: my-resource-group-1
        kind: my-resource-kind-1
        version: my-resource-version-1
        names: ["my-resource-names"]
        namespaces: ["my-resource-namespaces"]
        labelSelectors: ["trident.netapp.io/os=linux"]
      - group: my-resource-group-2
        kind: my-resource-kind-2
        version: my-resource-version-2
        names: ["my-resource-names"]
        namespaces: ["my-resource-namespaces"]
        labelSelectors: ["trident.netapp.io/os=linux"]
```

4. Une fois que vous avez rempli le `trident-protect-snapshot-ipr-cr.yaml` fichier avec les valeurs correctes, appliquez la CR :

```
kubectl apply -f trident-protect-snapshot-ipr-cr.yaml
```

Utiliser l'interface de ligne de commande

Étapes

1. Restaurez l'instantané dans l'espace de noms d'origine en remplaçant les valeurs entre parenthèses par les informations de votre environnement. Par exemple :

```
tridentctl-protect create snapshotinplacerestore <my_restore_name> \
--snapshot <namespace/snapshot_to_restore> \
-n <application_namespace>
```

Vérifiez l'état d'une opération de restauration

Vous pouvez utiliser la ligne de commande pour vérifier l'état d'une opération de restauration en cours, terminée ou ayant échoué.

Étapes

1. Utilisez la commande suivante pour récupérer le statut de l'opération de restauration en remplaçant les valeurs entre crochets par des informations de votre environnement :

```
kubectl get backuprestore -n <namespace_name> <my_restore_cr_name> -o
jsonpath='{.status}'
```

Utilisez les paramètres de restauration avancés de Trident Protect

Vous pouvez personnaliser les opérations de restauration à l'aide de paramètres avancés tels que les annotations, les paramètres d'espace de noms et les options de stockage pour répondre à vos besoins spécifiques.

Annotations et étiquettes de namespace pendant les opérations de restauration et de basculement

Lors des opérations de restauration et de basculement, les libellés et les annotations dans l'espace de noms de destination correspondent aux libellés et aux annotations dans l'espace de noms source. Des étiquettes ou des annotations provenant de l'espace de noms source qui n'existent pas dans l'espace de noms de destination sont ajoutées et toutes les étiquettes ou annotations qui existent déjà sont écrasées pour correspondre à la valeur de l'espace de noms source. Les libellés ou annotations qui existent uniquement dans l'espace de noms de destination restent inchangés.

> Si vous utilisez Red Hat OpenShift, il est important de noter le rôle essentiel des annotations d'espace de noms dans les environnements OpenShift. Les annotations d'espace de noms garantissent que les pods restaurés adhèrent aux autorisations et aux configurations de sécurité appropriées définies par les contraintes de contexte de sécurité (SCC) OpenShift et peuvent accéder aux volumes sans problèmes d'autorisation. Pour plus d'informations, reportez-vous à la"Documentation sur les contraintes de contexte de sécurité OpenShift" .

Vous pouvez empêcher l'écrasement d'annotations spécifiques dans l'espace de noms de destination en configurant la variable d'environnement Kubernetes `RESTORE_SKIP_NAMESPACE_ANNOTATIONS` avant d'effectuer l'opération de restauration ou de basculement. Par exemple :

```
helm upgrade trident-protect -n trident-protect netapp-trident-
protect/trident-protect \
  --set-string
restoreSkipNamespaceAnnotations="{<annotation_key_to_skip_1>,<annotation_k
ey_to_skip_2>}" \
  --reuse-values
```

Lors d'une opération de restauration ou de basculement, toutes les annotations et étiquettes d'espace de noms spécifiées dans `restoreSkipNamespaceAnnotations` et `restoreSkipNamespaceLabels` sont exclues de l'opération de restauration ou de basculement. Assurez-vous que ces paramètres sont configurés lors de l'installation initiale de Helm. Pour en savoir plus, consultez "Configurer les paramètres supplémentaires du graphique de barre de Trident Protect".

Si vous avez installé l'application source à l'aide de Helm avec le `--create-namespace` Le drapeau, un traitement spécial est accordé au `name` Légende. Lors du processus de restauration ou de basculement, Trident Protect copie cette étiquette dans l'espace de noms de destination, mais met à jour la valeur avec la valeur de l'espace de noms de destination si la valeur de la source correspond à l'espace de noms source. Si cette valeur ne correspond pas à l'espace de noms source, elle est copiée dans l'espace de noms de destination sans modification.

Exemple

L'exemple suivant présente un espace de noms source et de destination, chacun avec des annotations et des libellés différents. Vous pouvez voir l'état de l'espace de noms de destination avant et après l'opération, ainsi que la manière dont les annotations et les étiquettes sont combinées ou écrasées dans l'espace de noms de destination.

Avant l'opération de restauration ou de basculement

Le tableau suivant illustre l'état de l'exemple d'espaces de noms source et de destination avant l'opération de restauration ou de basculement :

Espace de noms	Annotations	Étiquettes
Espace de noms ns-1 (source)	• annotation.one/key : « updatedvalue » • annotation.deux/touche : « vrai »	• environnement=production • conformité = hipaa • name=ns-1
Espace de noms ns-2 (destination)	• annotation.un/touche : « vrai » • annotation.trois/touche : « false »	• role=base de données

Après l'opération de restauration

Le tableau suivant illustre l'état de l'exemple d'espace de noms de destination après une opération de restauration ou de basculement. Certaines clés ont été ajoutées, d'autres ont été écrasées et le `name` libellé a été mis à jour pour correspondre à l'espace de noms de destination :

Espace de noms	Annotations	Étiquettes
Espace de noms ns-2 (destination)	• annotation.one/key : « updatedvalue » • annotation.deux/touche : « vrai » • annotation.trois/touche : « false »	• name=ns-2 • conformité = hipaa • environnement=production • role=base de données

Champs pris en charge

Cette section décrit les champs supplémentaires disponibles pour les opérations de restauration.

Mappage des classes de stockage

Le `spec.storageClassMapping` L'attribut définit un mappage d'une classe de stockage présente dans l'application source vers une nouvelle classe de stockage sur le cluster cible. Vous pouvez l'utiliser lors de la migration d'applications entre des clusters avec différentes classes de stockage ou lors du changement du backend de stockage pour les opérations BackupRestore.

Exemple:

```
storageClassMapping:
  - destination: "destinationStorageClass1"
    source: "sourceStorageClass1"
  - destination: "destinationStorageClass2"
    source: "sourceStorageClass2"
```

Annotations prises en charge

Cette section répertorie les annotations prises en charge pour configurer différents comportements du système. Si une annotation n'est pas explicitement définie par l'utilisateur, le système utilisera la valeur par défaut.

Annotation	Type	Description	Valeur par défaut
protect.trident.netapp.io/data-mover-timeout-sec	chaîne	Le temps maximal (en secondes) autorisé pour que le fonctionnement du moteur de transfert de données soit bloqué.	"300"
protect.trident.netapp.io/kopia-content-cache-size-limit-mb	chaîne	La limite de taille maximale (en mégaoctets) pour le cache de contenu Kopia.	"1000"
protect.trident.netapp.io/pvc-bind-timeout-sec	chaîne	Délai maximal (en secondes) d'attente pour que les PersistentVolumeClaims (PVC) nouvellement créées atteignent le `Bound` phase précédant l'échec des opérations. S'applique à tous les types de restauration CR (BackupRestore, BackupInplaceRestore, SnapshotRestore, SnapshotInplaceRestore). Utilisez une valeur plus élevée si votre système de stockage ou votre cluster nécessite souvent plus de temps.	"1200" (20 minutes)