Restaurer les applications

Trident

NetApp
July 01, 2026

Sommaire

Restaurer les applications . . . 1
- Restaurez les applications à l'aide de Trident Protect . . . 1
  - Restaurer à partir d'une sauvegarde vers un espace de noms différent . . . 1
  - Restaurer à partir d'une sauvegarde vers l'espace de noms d'origine . . . 5
  - Restaurer à partir d'une sauvegarde vers un cluster différent . . . 8
  - Restaurer à partir d'un instantané vers un espace de noms différent . . . 11
  - Restaurer à partir d'un instantané vers l'espace de noms d'origine . . . 14
  - Vérifiez l'état d'une opération de restauration . . . 16
- Utilisez les paramètres de restauration avancés de Trident Protect . . . 17
  - Annotations et étiquettes d'espace de noms lors des opérations de restauration et de basculement . . . 17
  - Champs pris en charge . . . 19
  - Annotations prises en charge . . . 19

Restaurer les 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.

- Lors de la restauration d'une application, tous les points d'exécution configurés pour l'application sont restaurés avec l'application. Si un point 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. Cependant, la restauration à partir d'un instantané 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 les paramètres avancés pour personnaliser les opérations de restauration. Pour en savoir plus, consultez "Utilisez les paramètres de restauration avancés de Trident Protect".

Restaurer à partir d'une sauvegarde vers un espace de noms différent

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

- La restauration d'une sauvegarde dans un espace de noms différent contenant des ressources existantes ne modifiera pas les ressources portant le même nom que celles 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.
- Lors de l'utilisation d'un CR pour restaurer dans un nouvel espace de noms, vous devez créer manuellement l'espace de noms de destination avant d'appliquer le CR. Trident Protect crée automatiquement les espaces de noms uniquement lors de l'utilisation du CLI.

Avant de commencer

Assurez-vous que la durée de validité du jeton de session AWS est suffisante pour toute opération de restauration s3 de longue durée. Si le jeton expire pendant l'opération de restauration, l'opération peut échouer.

- Consultez la "Documentation AWS API" pour plus d'informations sur la vérification de l'expiration du jeton de session actuel.
- Consultez la "Documentation AWS IAM" pour plus d'informations sur les identifiants relatifs aux ressources AWS.

Lorsque vous restaurez des sauvegardes en utilisant Kopia comme moteur de déplacement de données, vous pouvez éventuellement spécifier des annotations dans le CR ou en utilisant la CLI pour contrôler le comportement du stockage temporaire utilisé par Kopia. Reportez-vous à l' "Documentation Kopia" pour plus d'informations sur les options que vous pouvez configurer. Utilisez la commande `tridentctl-protect create --help` pour plus d'informations sur la spécification des annotations avec la CLI Trident Protect.

Utilisez un 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**: (*Obligatoire*) Le nom de cette ressource personnalisée; choisissez un nom unique et pertinent pour votre environnement.
   - **spec.appArchivePath** : Le chemin à l'intérieur de AppVault où le contenu de la sauvegarde est stocké. 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** : (*Obligatoire*) Le nom du AppVault où les contenus de sauvegarde sont stockés.
   - **spec.destinationApplicationName** : (*Facultatif*) Le nom de l'application restaurée. Si ce nom est fourni, l'application restaurée utilise ce nom. Si ce nom n'est pas fourni, l'application restaurée utilise le nom de l'application source.
   - **spec.namespaceMapping** : La correspondance de l'espace de noms source de l'opération de restauration avec l'espace de noms de destination. Remplacez `my-source-namespace` et `my-destination-namespace` par les informations 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
       destinationApplicationName: my-new-app-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 fonction de leur relation avec les ressources que vous sélectionnez. Par exemple, si vous sélectionnez une ressource de type revendication de volume persistant et qu'elle possède un pod associé, Trident Protect restaurera également le pod associé.

   - **resourceFilter.resourceSelectionCriteria** : (Obligatoire pour le filtrage) Utilisez `Include` ou

`Exclude` pour 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 selon une opération OU, et les champs à l'intérieur de chaque élément (group, kind, version) correspondent selon une opération ET.
  - **resourceMatchers[].group**: (*Optionnel*) Groupe de la ressource à filtrer.
  - **resourceMatchers[].kind**: (*Optionnel*) Type de ressource à filtrer.
  - **resourceMatchers[].version**: (*Optionnel*) Version de la ressource à filtrer.
  - **resourceMatchers[].names**: (*Optionnel*) Noms dans le champ Kubernetes metadata.name de la ressource à filtrer.
  - **resourceMatchers[].namespaces**: (*Optionnel*) Espaces de noms dans le champ metadata.name de Kubernetes de la ressource à filtrer.
  - **resourceMatchers[].labelSelectors** : (*Optionnel*) Chaîne de sélection d'étiquette dans le champ metadata.name de la ressource Kubernetes tel que 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. Après avoir rempli le fichier `trident-protect-backup-restore-cr.yaml` avec les valeurs correctes, appliquez le CR :

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

Utilisez la ligne de commandes (CLI)

Étapes

1. Restaurez la sauvegarde dans un espace de noms différent, en remplaçant les valeurs entre crochets par les informations de votre environnement. L' `namespace-mapping`argument utilise des

espaces de noms séparés par des deux-points pour faire correspondre les espaces de noms source aux espaces de noms de destination corrects au 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> \
--destination-app-name<custom_app_name>\
-n <application_namespace>
```

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

Vous pouvez restaurer une sauvegarde dans l'espace de noms d'origine à tout moment. Lorsque vous effectuez une restauration sur place, Trident Protect gère automatiquement les planifications de protection et les opérations en cours afin d'éviter les points de récupération invalides :

- Tous les plans de protection activés pour l'application sont désactivés avant le début de la restauration. Cela empêche les sauvegardes planifiées ou les instantanés de s'exécuter pendant la restauration des ressources de l'application.
- Après la restauration réussie, seules les planifications qui étaient activées avant la restauration sont réactivées. Les planifications qui étaient déjà désactivées restent désactivées.
- Toute opération de sauvegarde ou de capture instantanée en cours est annulée avant le début de la restauration. Si une opération n'est pas annulée dans les 5 minutes, la restauration se poursuit et un avertissement est consigné dans le statut du CR de restauration.

Avant de commencer

Assurez-vous que la durée de validité du jeton de session AWS est suffisante pour toute opération de restauration s3 de longue durée. Si le jeton expire pendant l'opération de restauration, l'opération peut échouer.

- Consultez la "Documentation AWS API" pour plus d'informations sur la vérification de l'expiration du jeton de session actuel.
- Consultez la "Documentation AWS IAM" pour plus d'informations sur les identifiants relatifs aux ressources AWS.

> Lorsque vous restaurez des sauvegardes en utilisant Kopia comme moteur de déplacement de données, vous pouvez éventuellement spécifier des annotations dans le CR ou en utilisant la CLI pour contrôler le comportement du stockage temporaire utilisé par Kopia. Reportez-vous à l' "Documentation Kopia" pour plus d'informations sur les options que vous pouvez configurer. Utilisez la commande `tridentctl-protect create --help` pour plus d'informations sur la spécification des annotations avec la CLI Trident Protect.

Utilisez un 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**: (*Obligatoire*) Le nom de cette ressource personnalisée; choisissez un nom unique et pertinent pour votre environnement.
   - **spec.appArchivePath** : Le chemin à l'intérieur de AppVault où le contenu de la sauvegarde est stocké. 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** : (*Obligatoire*) Le nom du AppVault où les contenus de sauvegarde sont stockés.

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

   - **resourceFilter.resourceSelectionCriteria** : (Obligatoire pour le filtrage) Utilisez `Include` ou `Exclude` pour 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 selon une opération OU, et les champs à l'intérieur de chaque élément (group, kind, version) correspondent selon une opération ET.
       - **resourceMatchers[].group**: (*Optionnel*) Groupe de la ressource à filtrer.
       - **resourceMatchers[].kind**: (*Optionnel*) Type de ressource à filtrer.

- **resourceMatchers[].version**: (*Optionnel*) Version de la ressource à filtrer.
- **resourceMatchers[].names**: (*Optionnel*) Noms dans le champ Kubernetes metadata.name de la ressource à filtrer.
- **resourceMatchers[].namespaces**: (*Optionnel*) Espaces de noms dans le champ metadata.name de Kubernetes de la ressource à filtrer.
- **resourceMatchers[].labelSelectors** : (*Optionnel*) Chaîne de sélection d'étiquette dans le champ metadata.name de la ressource Kubernetes tel que 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. Après avoir rempli le fichier `trident-protect-backup-ipr-cr.yaml` avec les valeurs correctes, appliquez le CR :

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

Utilisez la ligne de commandes (CLI)

Étapes

1. Restaurez la sauvegarde dans l'espace de noms d'origine, en remplaçant les valeurs entre crochets par les informations de votre environnement. L'`backup`argument utilise un 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>
```

Restaurer à partir d'une sauvegarde vers un cluster différent

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

- Lorsque vous restaurez des sauvegardes en utilisant Kopia comme moteur de déplacement de données, vous pouvez éventuellement spécifier des annotations dans le CR ou en utilisant la CLI pour contrôler le comportement du stockage temporaire utilisé par Kopia. Reportez-vous à l' "Documentation Kopia" pour plus d'informations sur les options que vous pouvez configurer. Utilisez la commande `tridentctl-protect create --help` pour plus d'informations sur la spécification des annotations avec la CLI Trident Protect.
- Lors de l'utilisation d'un CR pour restaurer dans un nouvel espace de noms, vous devez créer manuellement l'espace de noms de destination avant d'appliquer le CR. Trident Protect crée automatiquement les espaces de noms uniquement lors de l'utilisation du CLI.

Avant de commencer

Assurez-vous que les conditions préalables suivantes sont remplies :

- Le cluster de destination a Trident Protect installé.
- Le cluster de destination a accès au chemin du 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 le CR AppVault lors de l'exécution de la commande `tridentctl-protect get appvaultcontent`. Si des restrictions réseau empêchent l'accès, exécutez la CLI Trident Protect depuis un pod sur le cluster de destination à la place.
- Assurez-vous que la durée de validité du jeton de session AWS soit suffisante pour toute opération de restauration de longue durée. Si le jeton expire pendant l'opération de restauration, l'opération peut échouer.
  - Consultez la "Documentation AWS API" pour plus d'informations sur la vérification de l'expiration du jeton de session actuel.
  - Consultez la "Documentation AWS" pour plus d'informations sur les identifiants relatifs aux ressources AWS.

Étapes

1. Vérifiez que la AppVault CR existe sur le cluster de destination à l'aide du plugin CLI Trident Protect :

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

   Si la AppVault CR n'existe pas sur le cluster de destination, créez-la en suivant les étapes dans "Utilisez les objets Trident Protect AppVault pour gérer les compartiments".

2. Consultez le contenu de la sauvegarde disponible de AppVault sur le cluster de destination, et notez `appArchivePath` de la sauvegarde que vous souhaitez restaurer :

```
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 des applications correspondantes, les horodatages et les chemins d'accès aux archives.

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 en utilisant le nom AppVault et le chemin d'accès à l'archive :

> Lors de l'utilisation d'une CR, assurez-vous que l'espace de noms destiné à la restauration de l'application existe sur le cluster de destination.

Utilisez un 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**: (*Obligatoire*) Le nom de cette ressource personnalisée; choisissez un nom unique et pertinent pour votre environnement.
   - **spec.appVaultRef** : (*Obligatoire*) Le nom du AppVault où les contenus de sauvegarde sont stockés.
   - **spec.appArchivePath** : (Obligatoire) Le chemin à l'intérieur de AppVault où le contenu de la sauvegarde est stocké. Utilisez la commande de l'étape 2 pour afficher le contenu de la sauvegarde et trouver `appArchivePath` la sauvegarde que vous souhaitez restaurer.
   - **spec.destinationApplicationName** : (*Facultatif*) Le nom de l'application restaurée. Si ce nom est fourni, l'application restaurée utilise ce nom. Si ce nom n'est pas fourni, l'application restaurée utilise le nom de l'application source.
   - **spec.namespaceMapping** : La correspondance de l'espace de noms source de l'opération de restauration avec l'espace de noms de destination. Remplacez `my-source-namespace` et `my-destination-namespace` par les informations 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
  destinationApplicationName: my-new-app-name
  namespaceMapping: [{"source": "my-source-namespace", "
destination": "my-destination-namespace"}]
```

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

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

Utilisez la ligne de commandes (CLI)

1. Utilisez la commande suivante pour restaurer l'application, en remplaçant les valeurs entre crochets par les informations de votre environnement. L'argument namespace-mapping utilise des espaces de noms séparés par des deux-points pour associer les espaces de noms source aux espaces de noms de destination correspondants, 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> \
--destination-app-name <custom_app_name> \
--context <destination_cluster_name> \
-n <application_namespace>
```

Restaurer à partir d'un instantané vers un espace de noms différent

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

- SnapshotRestore prend en charge l' `spec.storageClassMapping`attribut, mais uniquement lorsque les classes de stockage source et de destination utilisent le même système de stockage. Si vous tentez de restaurer vers une classe de stockage `StorageClass` qui utilise un système de stockage différent, l'opération de restauration échouera.
- Lors de l'utilisation d'un CR pour restaurer dans un nouvel espace de noms, vous devez créer manuellement l'espace de noms de destination avant d'appliquer le CR. Trident Protect crée automatiquement les espaces de noms uniquement lors de l'utilisation du CLI.

Avant de commencer

Assurez-vous que la durée de validité du jeton de session AWS est suffisante pour toute opération de restauration s3 de longue durée. Si le jeton expire pendant l'opération de restauration, l'opération peut échouer.

- Consultez la "Documentation AWS API" pour plus d'informations sur la vérification de l'expiration du jeton de session actuel.
- Consultez la "Documentation AWS IAM" pour plus d'informations sur les identifiants relatifs aux ressources AWS.

Utilisez un 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**: (*Obligatoire*) Le nom de cette ressource personnalisée; choisissez un nom unique et pertinent pour votre environnement.
   - **spec.appVaultRef** : (*Obligatoire*) Le nom du AppVault où le contenu de l'instantané est stocké.
   - **spec.appArchivePath** : Le chemin à l'intérieur de AppVault où les contenus de l'instantané sont stockés. Vous pouvez utiliser la commande suivante pour trouver ce chemin :

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

   - **spec.destinationApplicationName** : (*Facultatif*) Le nom de l'application restaurée. Si ce nom est fourni, l'application restaurée utilise ce nom. Si ce nom n'est pas fourni, l'application restaurée utilise le nom de l'application source.
   - **spec.namespaceMapping** : La correspondance de l'espace de noms source de l'opération de restauration avec l'espace de noms de destination. Remplacez `my-source-namespace` et `my-destination-namespace` par les informations 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 fonction de leur relation avec les ressources que vous sélectionnez. Par exemple, si vous sélectionnez une ressource de type revendication de volume persistant et qu'elle possède un pod associé, Trident Protect restaurera également le pod associé.

   - **resourceFilter.resourceSelectionCriteria** : (Obligatoire pour le filtrage) Utilisez `Include` ou `Exclude` pour 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 selon une opération OU, et les champs à l'intérieur de chaque élément (group, kind, version) correspondent selon une opération ET.
  - **resourceMatchers[].group**: (*Optionnel*) Groupe de la ressource à filtrer.
  - **resourceMatchers[].kind**: (*Optionnel*) Type de ressource à filtrer.
  - **resourceMatchers[].version**: (*Optionnel*) Version de la ressource à filtrer.
  - **resourceMatchers[].names**: (*Optionnel*) Noms dans le champ Kubernetes metadata.name de la ressource à filtrer.
  - **resourceMatchers[].namespaces**: (*Optionnel*) Espaces de noms dans le champ metadata.name de Kubernetes de la ressource à filtrer.
  - **resourceMatchers[].labelSelectors** : (*Optionnel*) Chaîne de sélection d'étiquette dans le champ metadata.name de la ressource Kubernetes tel que 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. Après avoir rempli le fichier `trident-protect-snapshot-restore-cr.yaml` avec les valeurs correctes, appliquez le CR :

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

Utilisez la ligne de commandes (CLI)

Étapes

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

   ◦ L'`snapshot`argument utilise un espace de noms et un nom d'instantané au

format `<namespace>/<name>`.

- L'`namespace-mapping`argument utilise des espaces de noms séparés par des deux-points pour faire correspondre les espaces de noms source aux espaces de noms de destination corrects au 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> \
--destination-app-name <custom_app_name> \
-n <application_namespace>
```

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

Vous pouvez restaurer un instantané dans l'espace de noms d'origine à tout moment. Lorsque vous effectuez une restauration sur place, Trident Protect gère automatiquement les planifications de protection et les opérations en cours afin d'éviter les points de récupération invalides :

- Tous les plans de protection activés pour l'application sont désactivés avant le début de la restauration. Cela empêche les sauvegardes planifiées ou les instantanés de s'exécuter pendant la restauration des ressources de l'application.
- Après la restauration réussie, seules les planifications qui étaient activées avant la restauration sont réactivées. Les planifications qui étaient déjà désactivées restent désactivées.
- Toute opération de sauvegarde ou de capture instantanée en cours est annulée avant le début de la restauration. Si une opération n'est pas annulée dans les 5 minutes, la restauration se poursuit et un avertissement est consigné dans le statut du CR de restauration.

Avant de commencer

Assurez-vous que la durée de validité du jeton de session AWS est suffisante pour toute opération de restauration s3 de longue durée. Si le jeton expire pendant l'opération de restauration, l'opération peut échouer.

- Consultez la "Documentation AWS API" pour plus d'informations sur la vérification de l'expiration du jeton de session actuel.
- Consultez la "Documentation AWS IAM" pour plus d'informations sur les identifiants relatifs aux ressources AWS.

Utilisez un 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**: (*Obligatoire*) Le nom de cette ressource personnalisée; choisissez un nom unique et pertinent pour votre environnement.
   - **spec.appVaultRef** : (*Obligatoire*) Le nom du AppVault où le contenu de l'instantané est stocké.
   - **spec.appArchivePath** : Le chemin à l'intérieur de AppVault où les contenus de l'instantané sont stockés. 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 fonction de leur relation avec les ressources que vous sélectionnez. Par exemple, si vous sélectionnez une ressource de type revendication de volume persistant et qu'elle possède un pod associé, Trident Protect restaurera également le pod associé.

   - **resourceFilter.resourceSelectionCriteria** : (Obligatoire pour le filtrage) Utilisez `Include` ou `Exclude` pour 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 selon une opération OU, et les champs à l'intérieur de chaque élément (group, kind, version) correspondent selon une opération ET.
       - **resourceMatchers[].group**: (*Optionnel*) Groupe de la ressource à filtrer.
       - **resourceMatchers[].kind**: (*Optionnel*) Type de ressource à filtrer.
       - **resourceMatchers[].version**: (*Optionnel*) Version de la ressource à filtrer.
       - **resourceMatchers[].names**: (*Optionnel*) Noms dans le champ Kubernetes metadata.name de la ressource à filtrer.

- **resourceMatchers[].namespaces**: (*Optionnel*) Espaces de noms dans le champ metadata.name de Kubernetes de la ressource à filtrer.
- **resourceMatchers[].labelSelectors** : (*Optionnel*) Chaîne de sélection d'étiquette dans le champ metadata.name de la ressource Kubernetes tel que 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. Après avoir rempli le fichier `trident-protect-snapshot-ipr-cr.yaml` avec les valeurs correctes, appliquez le CR :

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

Utilisez la ligne de commandes (CLI)

Étapes

1. Restaurez l'instantané dans l'espace de noms d'origine, en remplaçant les valeurs entre crochets 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 commandes pour vérifier l'état d'une opération de restauration qui est en cours, terminée ou ayant échoué.

Étapes

1. Utilisez la commande suivante pour récupérer l'état de l'opération de restauration, en remplaçant les valeurs entre crochets par les 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 de namespace et les options de stockage pour répondre à vos besoins spécifiques.

Annotations et étiquettes d'espace de noms lors des opérations de restauration et de basculement

Lors des opérations de restauration et de basculement, les étiquettes et annotations de l'espace de noms de destination sont mises à jour pour correspondre aux étiquettes et annotations de l'espace de noms source. Les étiquettes ou annotations 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 déjà présentes sont remplacées pour correspondre à la valeur de l'espace de noms source. Les étiquettes ou annotations qui existent uniquement dans l'espace de noms de destination restent inchangées.

> Si vous utilisez Red Hat OpenShift, il est important de noter le rôle crucial des annotations d'espace de noms dans les environnements OpenShift. Les annotations d'espace de noms garantissent que les pods restaurés respectent les permissions et les configurations de sécurité appropriées définies par les contraintes de contexte de sécurité (OpenShift SCC) et peuvent accéder aux volumes sans problème de permissions. Pour plus d'informations, consultez la "Documentation des contraintes de contexte de sécurité OpenShift".

Vous pouvez empêcher l'écrasement de certaines annotations dans l'espace de noms de destination en définissant 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, 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 des paramètres supplémentaires du chart Helm Trident Protect".

Si vous avez installé l'application source avec Helm avec le `--create-namespace` flag, un traitement spécial est appliqué à la clé de label `name`. Lors du processus de restauration ou de basculement, Trident Protect copie ce label dans l'espace de noms de destination, mais met à jour la valeur avec celle de l'espace de noms de destination si la valeur provenant 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 un espace de noms de destination, chacun possédant des annotations et des étiquettes différentes. Vous pouvez voir l'état de l'espace de noms de destination avant et après l'opération, et comment 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 des espaces de noms source et de destination de l'exemple avant l'opération de restauration ou de basculement :

Espace de noms	Annotations	Étiquettes
Espace de noms ns-1 (source)	• annotation.one/key: "updatedvalue" • annotation.two/key: "true"	• environment=production • conformité=hipaa • name=ns-1
Espace de noms ns-2 (destination)	• annotation.one/key: "true" • annotation.three/key: "false"	• rôle=base de données

Après l'opération de restauration

Le tableau suivant illustre l'état de l'espace de noms de destination après l'opération de restauration ou de basculement. Certaines clés ont été ajoutées, d'autres ont été écrasées, et l' `name`étiquette a été mise 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.two/key: "true" • annotation.three/key: "false"	• name=ns-2 • conformité=hipaa • environment=production • rôle=base de données

Champs pris en charge

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

Correspondance des classes de stockage

L' `spec.storageClassMapping`attribut définit une correspondance entre une classe de stockage présente dans l'application source et une nouvelle classe de stockage sur le cluster cible. Vous pouvez l'utiliser lors de la migration d'applications entre des clusters avec des classes de stockage différentes ou lors du changement de système 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 l'opération de déplacement de données soit bloquée.	"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	Temps maximal (en secondes) d'attente pour que tout nouveau PersistentVolumeClaims (PVC) atteigne la phase `Bound` avant que l'opération n'échoue. S'applique à tous les types de CR de restauration (BackupRestore, BackupInplaceRestore, SnapshotRestore, SnapshotInplaceRestore). Utilisez une valeur plus élevée si votre backend de stockage ou votre cluster nécessite souvent plus de temps.	"1200" (20 minutes)