Anwendungen wiederherstellen

Trident

NetApp
January 15, 2026

Inhalt

Anwendungen wiederherstellen

Anwendungen mit Trident Protect wiederherstellen

Mit Trident Protect können Sie Ihre Anwendung aus einem Snapshot oder einer Sicherung wiederherstellen. Die Wiederherstellung aus einem vorhandenen Snapshot ist schneller, wenn die Anwendung im selben Cluster wiederhergestellt wird.

- Wenn Sie eine Anwendung wiederherstellen, werden alle für die Anwendung konfigurierten Ausführungs-Hooks mit der Anwendung wiederhergestellt. Wenn ein Hook für die Nachbearbeitung der Wiederherstellung vorhanden ist, wird dieser automatisch im Rahmen des Wiederherstellungsvorgangs ausgeführt.
- Die Wiederherstellung aus einer Sicherung in einen anderen Namespace oder in den ursprünglichen Namespace wird für Qtree-Volumes unterstützt. Die Wiederherstellung von einem Snapshot in einen anderen Namespace oder in den ursprünglichen Namespace wird für Qtree-Volumes jedoch nicht unterstützt.
- Sie können erweiterte Einstellungen verwenden, um Wiederherstellungsvorgänge anzupassen. Weitere Informationen finden Sie unter "Verwenden Sie die erweiterten Trident Protect-Wiederherstellungseinstellungen.".

Wiederherstellung aus einem Backup in einen anderen Namensraum

Wenn Sie eine Sicherung mithilfe eines BackupRestore CR in einen anderen Namespace wiederherstellen, stellt Trident Protect die Anwendung in einem neuen Namespace wieder her und erstellt einen Anwendungs-CR für die wiederhergestellte Anwendung. Um die wiederhergestellte Anwendung zu schützen, erstellen Sie bei Bedarf Backups oder Snapshots oder legen Sie einen Schutzzeitplan fest.

Das Wiederherstellen einer Sicherung in einem anderen Namensraum mit bereits vorhandenen Ressourcen ändert keine Ressourcen, die denselben Namen wie die Ressourcen in der Sicherung haben. Um alle Ressourcen im Backup wiederherzustellen, müssen Sie entweder den Ziel-Namespace löschen und neu erstellen oder das Backup in einem neuen Namespace wiederherstellen.

Bevor Sie beginnen

Stellen Sie sicher, dass die Gültigkeitsdauer des AWS-Sitzungstokens für alle länger andauernden S3-Wiederherstellungsvorgänge ausreichend ist. Wenn das Token während des Wiederherstellungsvorgangs abläuft, kann der Vorgang fehlschlagen.

- Siehe die "AWS API-Dokumentation" Weitere Informationen zum Prüfen des Ablaufs des aktuellen Sitzungstokens finden Sie hier.
- Siehe die "AWS IAM-Dokumentation" Weitere Informationen zu Anmeldeinformationen für AWS-Ressourcen finden Sie hier.

Wenn Sie Sicherungen mit Kopia als Datenverschieber wiederherstellen, können Sie optional Anmerkungen im CR angeben oder die CLI verwenden, um das Verhalten des von Kopia verwendeten temporären Speichers zu steuern. Siehe die "Kopia-Dokumentation" Weitere Informationen zu den konfigurierbaren Optionen finden Sie hier. Verwenden Sie die `tridentctl-protect create --help` Befehl für weitere Informationen zum Festlegen von Annotationen mit der Trident Protect CLI.

Verwenden Sie einen CR

Schritte

1. Erstellen Sie die benutzerdefinierte Ressourcendatei (CR-Datei) und benennen Sie sie. `trident-protect-backup-restore-cr.yaml` .
2. Konfigurieren Sie in der von Ihnen erstellten Datei die folgenden Attribute:
   - **metadata.name**: (*Erforderlich*) Der Name dieser benutzerdefinierten Ressource; wählen Sie einen eindeutigen und aussagekräftigen Namen für Ihre Umgebung.
   - **spec.appArchivePath**: Der Pfad innerhalb von AppVault, in dem die Sicherungsinhalte gespeichert sind. Sie können folgenden Befehl verwenden, um diesen Pfad zu finden:

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

   - **spec.appVaultRef**: (*Erforderlich*) Der Name des AppVaults, in dem die Sicherungsinhalte gespeichert sind.
   - **spec.namespaceMapping**: Die Zuordnung des Quell-Namespace der Wiederherstellungsoperation zum Ziel-Namespace. Ersetzen `my-source-namespace` Und `my-destination-namespace` mit Informationen aus Ihrer Umgebung.

     ```
     ---
     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. (*Optional*) Falls Sie nur bestimmte Ressourcen der Anwendung für die Wiederherstellung auswählen möchten, fügen Sie Filter hinzu, die Ressourcen mit bestimmten Bezeichnungen ein- oder ausschließen:

   > Trident Protect wählt einige Ressourcen automatisch aus, weil sie mit den von Ihnen ausgewählten Ressourcen in Beziehung stehen. Wenn Sie beispielsweise eine persistente Volume-Claim-Ressource auswählen und diese einen zugehörigen Pod hat, stellt Trident Protect auch den zugehörigen Pod wieder her.

   - **resourceFilter.resourceSelectionCriteria**: (Für die Filterung erforderlich) Verwenden Sie `Include` oder `Exclude` Eine in resourceMatchers definierte Ressource ein- oder ausschließen. Fügen Sie die folgenden resourceMatchers-Parameter hinzu, um die ein- oder auszuschließenden Ressourcen zu definieren:

- **resourceFilter.resourceMatchers**: Ein Array von resourceMatcher-Objekten. Wenn Sie mehrere Elemente in diesem Array definieren, werden diese per OR-Verknüpfung abgeglichen, und die Felder innerhalb jedes Elements (Gruppe, Art, Version) werden per AND-Verknüpfung abgeglichen.
  - **resourceMatchers[].group**: (*Optional*) Gruppe der zu filternden Ressourcen.
  - **resourceMatchers[].kind**: (*Optional*) Art der zu filternden Ressource.
  - **resourceMatchers[].version**: (*Optional*) Version der zu filternden Ressource.
  - **resourceMatchers[].names**: (*Optional*) Namen im Kubernetes metadata.name Feld der zu filternden Ressource.
  - **resourceMatchers[].namespaces**: (*Optional*) Namespaces im Kubernetes metadata.name-Feld der zu filternden Ressource.
  - **resourceMatchers[].labelSelectors**: (*Optional*) Label-Selektorzeichenfolge im Kubernetes-Metadatenfeld „name“ der Ressource, wie in der Dokumentation definiert. "Kubernetes-Dokumentation" . Zum Beispiel: `"trident.netapp.io/os=linux"` .

    Beispiel:

```
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. Nachdem Sie die `trident-protect-backup-restore-cr.yaml` Datei mit den korrekten Werten, CR anwenden:

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

Verwenden der CLI

Schritte

1. Stellen Sie die Sicherung in einem anderen Namespace wieder her und ersetzen Sie die Werte in Klammern durch Informationen aus Ihrer Umgebung. Der `namespace-mapping` Das Argument verwendet durch Doppelpunkte getrennte Namensräume, um Quell-Namensräume den korrekten

Ziel-Namensräumen im folgenden Format zuzuordnen: `source1:dest1,source2:dest2` . Beispiel:

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

Wiederherstellung aus einer Sicherung in den ursprünglichen Namensraum

Sie können eine Sicherung jederzeit im ursprünglichen Namensraum wiederherstellen.

Bevor Sie beginnen

Stellen Sie sicher, dass die Gültigkeitsdauer des AWS-Sitzungstokens für alle länger andauernden S3-Wiederherstellungsvorgänge ausreichend ist. Wenn das Token während des Wiederherstellungsvorgangs abläuft, kann der Vorgang fehlschlagen.

- Siehe die "AWS API-Dokumentation" Weitere Informationen zum Prüfen des Ablaufs des aktuellen Sitzungstokens finden Sie hier.
- Siehe die "AWS IAM-Dokumentation" Weitere Informationen zu Anmeldeinformationen für AWS-Ressourcen finden Sie hier.

Wenn Sie Sicherungen mit Kopia als Datenverschieber wiederherstellen, können Sie optional Anmerkungen im CR angeben oder die CLI verwenden, um das Verhalten des von Kopia verwendeten temporären Speichers zu steuern. Siehe die "Kopia-Dokumentation" Weitere Informationen zu den konfigurierbaren Optionen finden Sie hier. Verwenden Sie die `tridentctl-protect create --help` Befehl für weitere Informationen zum Festlegen von Annotationen mit der Trident Protect CLI.

Verwenden Sie einen CR

Schritte

1. Erstellen Sie die benutzerdefinierte Ressourcendatei (CR-Datei) und benennen Sie sie. `trident-protect-backup-ipr-cr.yaml` .
2. Konfigurieren Sie in der von Ihnen erstellten Datei die folgenden Attribute:
   - **metadata.name**: (*Erforderlich*) Der Name dieser benutzerdefinierten Ressource; wählen Sie einen eindeutigen und aussagekräftigen Namen für Ihre Umgebung.
   - **spec.appArchivePath**: Der Pfad innerhalb von AppVault, in dem die Sicherungsinhalte gespeichert sind. Sie können folgenden Befehl verwenden, um diesen Pfad zu finden:

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

   - **spec.appVaultRef**: (*Erforderlich*) Der Name des AppVaults, in dem die Sicherungsinhalte gespeichert sind.

     Beispiel:

```
---
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. (*Optional*) Falls Sie nur bestimmte Ressourcen der Anwendung für die Wiederherstellung auswählen möchten, fügen Sie Filter hinzu, die Ressourcen mit bestimmten Bezeichnungen ein- oder ausschließen:

   Trident Protect wählt einige Ressourcen automatisch aus, weil sie mit den von Ihnen ausgewählten Ressourcen in Beziehung stehen. Wenn Sie beispielsweise eine persistente Volume-Claim-Ressource auswählen und diese einen zugehörigen Pod hat, stellt Trident Protect auch den zugehörigen Pod wieder her.

   - **resourceFilter.resourceSelectionCriteria**: (Für die Filterung erforderlich) Verwenden Sie `Include` oder `Exclude` Eine in resourceMatchers definierte Ressource ein- oder ausschließen. Fügen Sie die folgenden resourceMatchers-Parameter hinzu, um die ein- oder auszuschließenden Ressourcen zu definieren:
     - **resourceFilter.resourceMatchers**: Ein Array von resourceMatcher-Objekten. Wenn Sie mehrere Elemente in diesem Array definieren, werden diese per OR-Verknüpfung abgeglichen, und die Felder innerhalb jedes Elements (Gruppe, Art, Version) werden per AND-Verknüpfung abgeglichen.

- **resourceMatchers[].group**: (*Optional*) Gruppe der zu filternden Ressourcen.
- **resourceMatchers[].kind**: (*Optional*) Art der zu filternden Ressource.
- **resourceMatchers[].version**: (*Optional*) Version der zu filternden Ressource.
- **resourceMatchers[].names**: (*Optional*) Namen im Kubernetes metadata.name Feld der zu filternden Ressource.
- **resourceMatchers[].namespaces**: (*Optional*) Namespaces im Kubernetes metadata.name-Feld der zu filternden Ressource.
- **resourceMatchers[].labelSelectors**: (*Optional*) Label-Selektorzeichenfolge im Kubernetes-Metadatenfeld „name“ der Ressource, wie in der Dokumentation definiert. "Kubernetes-Dokumentation" . Zum Beispiel: `"trident.netapp.io/os=linux"` .

  Beispiel:

```
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. Nachdem Sie die `trident-protect-backup-ipr-cr.yaml` Datei mit den korrekten Werten, CR anwenden:

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

Verwenden der CLI

Schritte

1. Stellen Sie die Sicherung im ursprünglichen Namensraum wieder her und ersetzen Sie die Werte in Klammern durch Informationen aus Ihrer Umgebung. Der `backup` Das Argument verwendet einen Namespace und einen Backup-Namen im folgenden Format: `<namespace>/<name>` . Beispiel:

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

Wiederherstellung aus einem Backup auf einem anderen Cluster

Sie können eine Sicherung auf einem anderen Cluster wiederherstellen, falls es ein Problem mit dem ursprünglichen Cluster gibt.

Wenn Sie Sicherungen mit Kopia als Datenverschieber wiederherstellen, können Sie optional Anmerkungen im CR angeben oder die CLI verwenden, um das Verhalten des von Kopia verwendeten temporären Speichers zu steuern. Siehe die "Kopia-Dokumentation" Weitere Informationen zu den konfigurierbaren Optionen finden Sie hier. Verwenden Sie die `tridentctl-protect create --help` Befehl für weitere Informationen zum Festlegen von Annotationen mit der Trident Protect CLI.

Bevor Sie beginnen

Stellen Sie sicher, dass die folgenden Voraussetzungen erfüllt sind:

- Auf dem Zielcluster ist Trident Protect installiert.
- Der Zielcluster hat Zugriff auf den Bucket-Pfad desselben AppVault wie der Quellcluster, in dem das Backup gespeichert ist.
- Stellen Sie sicher, dass Ihre lokale Umgebung beim Ausführen des AppVault CR eine Verbindung zum im AppVault CR definierten Objektspeicher-Bucket herstellen kann. `tridentctl-protect get appvaultcontent` Befehl. Falls Netzwerkbeschränkungen den Zugriff verhindern, führen Sie die Trident Protect CLI stattdessen innerhalb eines Pods auf dem Zielcluster aus.
- Stellen Sie sicher, dass die Gültigkeitsdauer des AWS-Sitzungstokens für alle länger andauernden Wiederherstellungsvorgänge ausreichend ist. Wenn das Token während des Wiederherstellungsvorgangs abläuft, kann der Vorgang fehlschlagen.
  - Siehe die "AWS API-Dokumentation" Weitere Informationen zum Prüfen des Ablaufs des aktuellen Sitzungstokens finden Sie hier.
  - Siehe die "AWS-Dokumentation" Weitere Informationen zu Anmeldeinformationen für AWS-Ressourcen finden Sie hier.

Schritte

1. Überprüfen Sie die Verfügbarkeit des AppVault CR auf dem Zielcluster mithilfe des Trident Protect CLI-Plugins:

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

   Stellen Sie sicher, dass der für die Anwendungswiederherstellung vorgesehene Namespace auf dem Zielcluster vorhanden ist.

2. Zeigen Sie die Sicherungsinhalte des verfügbaren AppVaults vom Zielcluster an:

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

Durch Ausführen dieses Befehls werden die verfügbaren Backups im AppVault angezeigt, einschließlich ihrer Ursprungscluster, zugehörigen Anwendungsnamen, Zeitstempel und Archivpfade.

Beispielausgabe:

```
+-------------+-----------+--------+-----------------
+--------------------------+-------------+
|   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. Die Anwendung mithilfe des AppVault-Namens und des Archivpfads im Zielcluster wiederherstellen:

Verwenden Sie einen CR

1. Erstellen Sie die benutzerdefinierte Ressourcendatei (CR-Datei) und benennen Sie sie. `trident-protect-backup-restore-cr.yaml` .
2. Konfigurieren Sie in der von Ihnen erstellten Datei die folgenden Attribute:
   - **metadata.name**: (*Erforderlich*) Der Name dieser benutzerdefinierten Ressource; wählen Sie einen eindeutigen und aussagekräftigen Namen für Ihre Umgebung.
   - **spec.appVaultRef**: (*Erforderlich*) Der Name des AppVaults, in dem die Sicherungsinhalte gespeichert sind.
   - **spec.appArchivePath**: Der Pfad innerhalb von AppVault, in dem die Sicherungsinhalte gespeichert sind. Sie können folgenden Befehl verwenden, um diesen Pfad zu finden:

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

     Falls BackupRestore CR nicht verfügbar ist, können Sie den in Schritt 2 genannten Befehl verwenden, um den Sicherungsinhalt anzuzeigen.

   - **spec.namespaceMapping**: Die Zuordnung des Quell-Namespace der Wiederherstellungsoperation zum Ziel-Namespace. Ersetzen `my-source-namespace` Und `my-destination-namespace` mit Informationen aus Ihrer Umgebung.

     Beispiel:

```
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. Nachdem Sie die `trident-protect-backup-restore-cr.yaml` Datei mit den korrekten Werten, CR anwenden:

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

Verwenden der CLI

1. Verwenden Sie den folgenden Befehl, um die Anwendung wiederherzustellen, und ersetzen Sie dabei die Werte in Klammern durch Informationen aus Ihrer Umgebung. Das Argument namespace-

mapping verwendet durch Doppelpunkte getrennte Namensräume, um Quell-Namensräume den richtigen Ziel-Namensräumen im Format source1:dest1,source2:dest2 zuzuordnen. Beispiel:

```
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>
```

Wiederherstellung aus einem Snapshot in einen anderen Namespace

Sie können Daten aus einem Snapshot mithilfe einer benutzerdefinierten Ressourcendatei (CR-Datei) entweder in einem anderen Namensraum oder im ursprünglichen Quellnamensraum wiederherstellen. Wenn Sie einen Snapshot mithilfe eines SnapshotRestore CR in einem anderen Namespace wiederherstellen, stellt Trident Protect die Anwendung in einem neuen Namespace wieder her und erstellt einen Anwendungs-CR für die wiederhergestellte Anwendung. Um die wiederhergestellte Anwendung zu schützen, erstellen Sie bei Bedarf Backups oder Snapshots oder legen Sie einen Schutzzeitplan fest.

SnapshotRestore unterstützt die `spec.storageClassMapping` Attribut, jedoch nur dann, wenn die Quell- und Zielspeicherklassen das gleiche Speicher-Backend verwenden. Wenn Sie versuchen, einen Wiederherstellungszustand wiederherzustellen `StorageClass` Wenn ein anderes Speichersystem verwendet wird, schlägt die Wiederherstellung fehl.

Bevor Sie beginnen

Stellen Sie sicher, dass die Gültigkeitsdauer des AWS-Sitzungstokens für alle länger andauernden S3-Wiederherstellungsvorgänge ausreichend ist. Wenn das Token während des Wiederherstellungsvorgangs abläuft, kann der Vorgang fehlschlagen.

- Siehe die "AWS API-Dokumentation" Weitere Informationen zum Prüfen des Ablaufs des aktuellen Sitzungstokens finden Sie hier.
- Siehe die "AWS IAM-Dokumentation" Weitere Informationen zu Anmeldeinformationen für AWS-Ressourcen finden Sie hier.

Verwenden Sie einen CR

Schritte

1. Erstellen Sie die benutzerdefinierte Ressourcendatei (CR-Datei) und benennen Sie sie. `trident-protect-snapshot-restore-cr.yaml` .
2. Konfigurieren Sie in der von Ihnen erstellten Datei die folgenden Attribute:
   - **metadata.name**: (*Erforderlich*) Der Name dieser benutzerdefinierten Ressource; wählen Sie einen eindeutigen und aussagekräftigen Namen für Ihre Umgebung.
   - **spec.appVaultRef**: (*Erforderlich*) Der Name des AppVaults, in dem die Snapshot-Inhalte gespeichert sind.
   - **spec.appArchivePath**: Der Pfad innerhalb von AppVault, in dem die Snapshot-Inhalte gespeichert sind. Sie können folgenden Befehl verwenden, um diesen Pfad zu finden:

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

   - **spec.namespaceMapping**: Die Zuordnung des Quell-Namespace der Wiederherstellungsoperation zum Ziel-Namespace. Ersetzen `my-source-namespace` Und `my-destination-namespace` mit Informationen aus Ihrer Umgebung.

     ```
     ---
     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. (*Optional*) Falls Sie nur bestimmte Ressourcen der Anwendung für die Wiederherstellung auswählen möchten, fügen Sie Filter hinzu, die Ressourcen mit bestimmten Bezeichnungen ein- oder ausschließen:

   Trident Protect wählt einige Ressourcen automatisch aus, weil sie mit den von Ihnen ausgewählten Ressourcen in Beziehung stehen. Wenn Sie beispielsweise eine persistente Volume-Claim-Ressource auswählen und diese einen zugehörigen Pod hat, stellt Trident Protect auch den zugehörigen Pod wieder her.

   - **resourceFilter.resourceSelectionCriteria**: (Für die Filterung erforderlich) Verwenden Sie `Include` oder `Exclude` Eine in resourceMatchers definierte Ressource ein- oder ausschließen. Fügen Sie die folgenden resourceMatchers-Parameter hinzu, um die ein- oder auszuschließenden Ressourcen zu definieren:

- **resourceFilter.resourceMatchers**: Ein Array von resourceMatcher-Objekten. Wenn Sie mehrere Elemente in diesem Array definieren, werden diese per OR-Verknüpfung abgeglichen, und die Felder innerhalb jedes Elements (Gruppe, Art, Version) werden per AND-Verknüpfung abgeglichen.
  - **resourceMatchers[].group**: (*Optional*) Gruppe der zu filternden Ressourcen.
  - **resourceMatchers[].kind**: (*Optional*) Art der zu filternden Ressource.
  - **resourceMatchers[].version**: (*Optional*) Version der zu filternden Ressource.
  - **resourceMatchers[].names**: (*Optional*) Namen im Kubernetes metadata.name Feld der zu filternden Ressource.
  - **resourceMatchers[].namespaces**: (*Optional*) Namespaces im Kubernetes metadata.name-Feld der zu filternden Ressource.
  - **resourceMatchers[].labelSelectors**: (*Optional*) Label-Selektorzeichenfolge im Kubernetes-Metadatenfeld „name“ der Ressource, wie in der Dokumentation definiert. "Kubernetes-Dokumentation" . Zum Beispiel: `"trident.netapp.io/os=linux"` .

    Beispiel:

```
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. Nachdem Sie die `trident-protect-snapshot-restore-cr.yaml` Datei mit den korrekten Werten, CR anwenden:

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

Verwenden der CLI

Schritte

1. Stellen Sie den Snapshot in einem anderen Namespace wieder her und ersetzen Sie die Werte in Klammern durch Informationen aus Ihrer Umgebung.

- Der `snapshot` Das Argument verwendet einen Namespace und einen Snapshot-Namen im folgenden Format: `<namespace>/<name>` .
- Der `namespace-mapping` Das Argument verwendet durch Doppelpunkte getrennte Namensräume, um Quell-Namensräume den korrekten Ziel-Namensräumen im folgenden Format zuzuordnen: `source1:dest1,source2:dest2` .

  Beispiel:

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

Wiederherstellung aus einem Snapshot in den ursprünglichen Namensraum

Sie können jederzeit einen Snapshot im ursprünglichen Namensraum wiederherstellen.

Bevor Sie beginnen

Stellen Sie sicher, dass die Gültigkeitsdauer des AWS-Sitzungstokens für alle länger andauernden S3-Wiederherstellungsvorgänge ausreichend ist. Wenn das Token während des Wiederherstellungsvorgangs abläuft, kann der Vorgang fehlschlagen.

- Siehe die "AWS API-Dokumentation" Weitere Informationen zum Prüfen des Ablaufs des aktuellen Sitzungstokens finden Sie hier.
- Siehe die "AWS IAM-Dokumentation" Weitere Informationen zu Anmeldeinformationen für AWS-Ressourcen finden Sie hier.

Verwenden Sie einen CR

Schritte

1. Erstellen Sie die benutzerdefinierte Ressourcendatei (CR-Datei) und benennen Sie sie. `trident-protect-snapshot-ipr-cr.yaml` .
2. Konfigurieren Sie in der von Ihnen erstellten Datei die folgenden Attribute:
   - **metadata.name**: (*Erforderlich*) Der Name dieser benutzerdefinierten Ressource; wählen Sie einen eindeutigen und aussagekräftigen Namen für Ihre Umgebung.
   - **spec.appVaultRef**: (*Erforderlich*) Der Name des AppVaults, in dem die Snapshot-Inhalte gespeichert sind.
   - **spec.appArchivePath**: Der Pfad innerhalb von AppVault, in dem die Snapshot-Inhalte gespeichert sind. Sie können folgenden Befehl verwenden, um diesen Pfad zu finden:

     ```
     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. (*Optional*) Falls Sie nur bestimmte Ressourcen der Anwendung für die Wiederherstellung auswählen möchten, fügen Sie Filter hinzu, die Ressourcen mit bestimmten Bezeichnungen ein- oder ausschließen:

   > Trident Protect wählt einige Ressourcen automatisch aus, weil sie mit den von Ihnen ausgewählten Ressourcen in Beziehung stehen. Wenn Sie beispielsweise eine persistente Volume-Claim-Ressource auswählen und diese einen zugehörigen Pod hat, stellt Trident Protect auch den zugehörigen Pod wieder her.

   - **resourceFilter.resourceSelectionCriteria**: (Für die Filterung erforderlich) Verwenden Sie `Include` oder `Exclude` Eine in resourceMatchers definierte Ressource ein- oder ausschließen. Fügen Sie die folgenden resourceMatchers-Parameter hinzu, um die ein- oder auszuschließenden Ressourcen zu definieren:
     - **resourceFilter.resourceMatchers**: Ein Array von resourceMatcher-Objekten. Wenn Sie mehrere Elemente in diesem Array definieren, werden diese per OR-Verknüpfung abgeglichen, und die Felder innerhalb jedes Elements (Gruppe, Art, Version) werden per AND-Verknüpfung abgeglichen.
       - **resourceMatchers[].group**: (*Optional*) Gruppe der zu filternden Ressourcen.
       - **resourceMatchers[].kind**: (*Optional*) Art der zu filternden Ressource.

- **resourceMatchers[].version**: (*Optional*) Version der zu filternden Ressource.
- **resourceMatchers[].names**: (*Optional*) Namen im Kubernetes metadata.name Feld der zu filternden Ressource.
- **resourceMatchers[].namespaces**: (*Optional*) Namespaces im Kubernetes metadata.name-Feld der zu filternden Ressource.
- **resourceMatchers[].labelSelectors**: (*Optional*) Label-Selektorzeichenfolge im Kubernetes-Metadatenfeld „name“ der Ressource, wie in der Dokumentation definiert. "Kubernetes-Dokumentation" . Zum Beispiel: `"trident.netapp.io/os=linux"` .

  Beispiel:

```
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. Nachdem Sie die `trident-protect-snapshot-ipr-cr.yaml` Datei mit den korrekten Werten, CR anwenden:

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

Verwenden der CLI

Schritte

1. Stellen Sie den Snapshot im ursprünglichen Namespace wieder her und ersetzen Sie die Werte in Klammern durch Informationen aus Ihrer Umgebung. Beispiel:

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

Überprüfen Sie den Status eines Wiederherstellungsvorgangs

Sie können die Befehlszeile verwenden, um den Status eines Wiederherstellungsvorgangs zu überprüfen, der gerade läuft, abgeschlossen ist oder fehlgeschlagen ist.

Schritte

1. Verwenden Sie den folgenden Befehl, um den Status des Wiederherstellungsvorgangs abzurufen. Ersetzen Sie dabei die Werte in Klammern durch Informationen aus Ihrer Umgebung:

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

Verwenden Sie die erweiterten Trident Protect-Wiederherstellungseinstellungen.

Sie können Wiederherstellungsvorgänge mithilfe erweiterter Einstellungen wie Anmerkungen, Namespace-Einstellungen und Speicheroptionen an Ihre spezifischen Anforderungen anpassen.

Namespace-Annotationen und -Bezeichnungen während Wiederherstellungs- und Failover-Operationen

Bei Wiederherstellungs- und Failover-Vorgängen werden die Bezeichnungen und Annotationen im Ziel-Namensraum so angepasst, dass sie mit den Bezeichnungen und Annotationen im Quell-Namensraum übereinstimmen. Es werden Bezeichnungen oder Annotationen aus dem Quell-Namensraum hinzugefügt, die im Ziel-Namensraum nicht existieren, und alle bereits vorhandenen Bezeichnungen oder Annotationen werden überschrieben, um dem Wert aus dem Quell-Namensraum zu entsprechen. Labels oder Annotationen, die nur im Ziel-Namensraum existieren, bleiben unverändert.

> Wenn Sie Red Hat OpenShift verwenden, ist es wichtig, die entscheidende Rolle von Namespace-Annotationen in OpenShift-Umgebungen zu beachten. Namespace-Anmerkungen stellen sicher, dass wiederhergestellte Pods die entsprechenden Berechtigungen und Sicherheitskonfigurationen einhalten, die durch die OpenShift-Sicherheitskontextbeschränkungen (SCCs) definiert sind, und ohne Berechtigungsprobleme auf Volumes zugreifen können. Weitere Informationen finden Sie unter "Dokumentation zu den Sicherheitskontextbeschränkungen von OpenShift" .

Sie können verhindern, dass bestimmte Annotationen im Ziel-Namespace überschrieben werden, indem Sie die Kubernetes-Umgebungsvariable festlegen. `RESTORE_SKIP_NAMESPACE_ANNOTATIONS` bevor Sie die Wiederherstellungs- oder Failover-Operation durchführen. Beispiel:

```
helm upgrade trident-protect --set
restoreSkipNamespaceAnnotations=<annotation_key_to_skip_1>,<annotation_key
_to_skip_2> --reuse-values
```

Bei der Durchführung einer Wiederherstellungs- oder Failover-Operation werden alle in angegebenen Namespace-Annotationen und -Labels berücksichtigt. `restoreSkipNamespaceAnnotations` Und `restoreSkipNamespaceLabels` sind von der Wiederherstellungs- oder Failover-Operation ausgeschlossen. Stellen Sie sicher, dass diese Einstellungen während der ersten Helm-Installation konfiguriert werden. Weitere Informationen finden Sie unter "Konfigurieren Sie AutoSupport und Namespace-Filteroptionen".

Wenn Sie die Quellanwendung mit Helm installiert haben `--create-namespace` Flagge, besondere Behandlung wird der `name` Legende mit Beschriftung. Während des Wiederherstellungs- oder Failover-Prozesses kopiert Trident Protect diese Bezeichnung in den Ziel-Namespace, aktualisiert aber den Wert auf den Wert des Ziel-Namespace, wenn der Wert aus der Quelle mit dem Quell-Namespace übereinstimmt. Stimmt dieser Wert nicht mit dem Quell-Namespace überein, wird er unverändert in den Ziel-Namespace kopiert.

Beispiel

Das folgende Beispiel zeigt einen Quell- und einen Ziel-Namensraum, die jeweils unterschiedliche Annotationen und Bezeichnungen aufweisen. Sie können den Zustand des Ziel-Namensraums vor und nach der Operation einsehen und erkennen, wie die Annotationen und Labels im Ziel-Namensraum kombiniert oder überschrieben werden.

Vor dem Wiederherstellungs- oder Failover-Vorgang

Die folgende Tabelle veranschaulicht den Zustand der Beispiel-Quell- und Ziel-Namespaces vor der Wiederherstellungs- oder Failover-Operation:

Namensraum	Anmerkungen	Labels
Namespace ns-1 (Quelle)	• annotation.one/key: "updatedvalue" • annotation.two/key: "true"	• Umgebung=Produktion • Compliance=HIPAA • Name=ns-1
Namespace ns-2 (Ziel)	• annotation.one/key: "true" • annotation.three/key: "false"	• Rolle=Datenbank

Nach dem Wiederherstellungsvorgang

Die folgende Tabelle veranschaulicht den Zustand des Beispiel-Ziel-Namespace nach der Wiederherstellungs- oder Failover-Operation. Einige Schlüssel wurden hinzugefügt, einige wurden überschrieben, und die `name` Die Bezeichnung wurde aktualisiert, um dem Ziel-Namespace zu entsprechen:

Namensraum	Anmerkungen	Labels
Namespace ns-2 (Ziel)	• annotation.one/key: "updatedvalue" • annotation.two/key: "true" • annotation.three/key: "false"	• Name=ns-2 • Compliance=HIPAA • Umgebung=Produktion • Rolle=Datenbank

Unterstützte Felder

In diesem Abschnitt werden zusätzliche Felder beschrieben, die für Wiederherstellungsvorgänge verfügbar sind.

Speicherklassenzuordnung

Der `spec.storageClassMapping` Das Attribut definiert eine Zuordnung von einer in der Quellanwendung vorhandenen Speicherklasse zu einer neuen Speicherklasse im Zielcluster. Sie können dies verwenden, wenn Sie Anwendungen zwischen Clustern mit unterschiedlichen Speicherklassen migrieren oder wenn Sie das Speicher-Backend für BackupRestore-Vorgänge ändern.

Beispiel:

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

Unterstützte Annotationen

Dieser Abschnitt listet die unterstützten Annotationen zur Konfiguration verschiedener Verhaltensweisen im System auf. Wenn eine Annotation nicht explizit vom Benutzer festgelegt wird, verwendet das System den Standardwert.

Anmerkung	Typ	Beschreibung	Standardwert
protect.trident.netapp.io/data-mover-timeout-sec	Schnur	Die maximal zulässige Zeit (in Sekunden) für einen Stillstand des Datenübertragungsvorgangs.	"300"
protect.trident.netapp.io/kopia-content-cache-size-limit-mb	Schnur	Die maximale Größenbeschränkung (in Megabyte) für den Kopia-Inhaltscache.	"1000"