Wiederherstellen von Kubernetes-Anwendungen

NetApp Backup and Recovery

NetApp
June 15, 2026

Inhalt

Wiederherstellen von Kubernetes-Anwendungen

Kubernetes-Anwendungen mithilfe der Web-Benutzeroberfläche wiederherstellen

Mit NetApp Backup and Recovery können Sie Anwendungen wiederherstellen, die Sie mit einer Schutzrichtlinie geschützt haben. Zur Wiederherstellung einer Anwendung muss mindestens ein Wiederherstellungspunkt verfügbar sein. Ein Wiederherstellungspunkt besteht entweder aus dem lokalen Snapshot oder der Sicherung im Objektspeicher (oder beidem). Sie können eine Anwendung mithilfe des lokalen, sekundären oder Objektspeicherarchivs wiederherstellen.

Geschützte Ressourcen für einen Anwendungswiederherstellungspunkt anzeigen

Für jede Anwendung, die Sie mit Backup and Recovery schützen, können Sie die Ressourcen anzeigen, die für einen bestimmten Wiederherstellungspunkt gesichert wurden.

Erforderliche NetApp Console

Backup- und Wiederherstellungsanzeige. "Erfahren Sie mehr über die Zugriffsrollen für NetApp Backup and Recovery". "Erfahren Sie mehr über die Zugriffsrollen der NetApp Console für alle Dienste".

Schritte

1. In NetApp Backup and Recovery wählen Sie **Inventar** > **Anwendungen**.
2. Wählen Sie in der Anwendungsliste eine Anwendung aus und klicken Sie auf das Symbol „Aktionen“ ••• > **Anzeigen und Wiederherstellen**.
3. Wählen Sie in der Liste der Wiederherstellungspunkte einen Wiederherstellungspunkt aus und klicken Sie auf das Symbol „Aktionen“ ••• > **Ressourcen anzeigen**.

   Es wird eine Liste der Ressourcen und ihrer Details angezeigt. Sie können die Ressourcen nach Namensraum oder Clusterbereich anzeigen und die Liste als JSON-Datei für zukünftige Audits herunterladen.

4. Wenn Sie fertig sind, wählen Sie **Schließen**.

Wiederherstellen von Kubernetes-Anwendungen

Sie können Namespace-basierte oder VM-basierte Anwendungen von einem Wiederherstellungspunkt wiederherstellen, indem Sie entweder alle Ressourcen wiederherstellen oder eine Teilmenge der wiederherzustellenden Ressourcen auswählen.

Bevor Sie beginnen

Wenn Sie eine Anwendung wiederherstellen, die mit Trident Protect gesichert wurde, stellen Sie sicher, dass Trident Protect sowohl auf dem Quell-Cluster als auch auf dem Ziel-Cluster installiert ist.

Erforderliche NetApp Console

Superadministrator für Backup und Recovery oder Restore-Administrator für Backup und Recovery. "Erfahren Sie mehr über die Zugriffsrollen für NetApp Backup and Recovery". "Erfahren Sie mehr über die Zugriffsrollen der NetApp Console für alle Dienste".

Schritte

1. Wählen Sie im NetApp Backup und Recovery-Menü **Wiederherstellen** aus.
2. Wählen Sie eine Kubernetes-Anwendung aus der Liste und wählen Sie **Anzeigen und Wiederherstellen** für diese Anwendung.

   Die Liste der Wiederherstellungspunkte wird angezeigt.

3. Wählen Sie die Schaltfläche **Restore** für den Wiederherstellungspunkt aus, den Sie verwenden möchten.

   Der Assistent zum Wiederherstellen von Daten wird gestartet und die Seite *Allgemeine Einstellungen* wird angezeigt.

4. Wählen Sie den Quellspeicherort aus, von dem wiederhergestellt werden soll.
5. Wählen Sie den Zielcluster aus der Liste **Cluster** aus.
6. Wählen Sie, ob Sie in die ursprünglichen Namensräume oder in neue Namensräume wiederherstellen möchten.
7. Wenn Sie die Wiederherstellung in neuen Namensräumen gewählt haben, sind folgende Schritte erforderlich:
   a. Geben Sie den oder die Ziel-Namespaces ein, die verwendet werden.
   b. Den Namen der Zielanwendung eingeben.
   c. Optional kann die Option **Keine Anwendung für wiederhergestellte Ressourcen erstellen** ausgewählt werden, um Ressourcen wiederherzustellen, ohne ein benutzerdefiniertes Anwendungsressourcenobjekt zu erstellen. Dies reduziert unnötige Einträge im Anwendungsinventar.
8. Wählen Sie **Weiter**.

   Die Seite „Ressourcenauswahl“ wird angezeigt.

9. Wählen Sie aus, ob Sie alle mit der Anwendung verknüpften Ressourcen wiederherstellen möchten, oder verwenden Sie einen Filter, um bestimmte wiederherzustellende Ressourcen auszuwählen:

Alle Ressourcen

a. Wählen Sie **Alle Ressourcen wiederherstellen**.

   Bei der Wiederherstellung einer VM-basierten Anwendung listet Backup and Recovery alle virtuellen Maschinen im Wiederherstellungspunkt auf.

b. Wählen Sie **Weiter**.

Spezifische namensraumbasierte App-Ressourcen

a. Wählen Sie **Selektive Ressourcen** aus und entscheiden Sie, ob Sie die Ressourcen, die Sie auswählen, anhand von Regeln oder anhand des Namensraums filtern möchten.

Methode zur Ressourcenauswahl	Schritte
Ressourcen mithilfe von Regeln filtern	i. Wählen Sie die Registerkarte **Regeln** aus. ii. Wählen Sie das Verhalten des Ressourcenfilters. Wenn Sie **Einschließen** wählen, werden die von Ihnen ausgewählten Ressourcen wiederhergestellt. Wenn Sie **Ausschließen** wählen, werden die ausgewählten Ressourcen nicht wiederhergestellt. iii. Wählen Sie **Regeln hinzufügen** aus, um Regeln hinzuzufügen, die Filter für die Auswahl von Ressourcen definieren. Sie benötigen mindestens eine Regel zum Filtern von Ressourcen. Jede Regel kann nach Kriterien wie Ressourcennamespace, Bezeichnungen, Gruppe, Version und Art filtern. iv. Wählen Sie **Speichern**, um jede Regel zu speichern. v. Wenn Sie alle benötigten Regeln hinzugefügt haben, wählen Sie **Ressourcen anzeigen**, um die im Sicherungsarchiv verfügbaren Ressourcen anzuzeigen, die Ihren Filterkriterien entsprechen.

Methode zur Ressourcenauswahl	Schritte
Ressourcen manuell aus einer Liste auswählen	i. Wählen Sie die Registerkarte **Benutzerdefiniert** aus. ii. Wählen Sie **Namespace-bezogen** oder **Cluster-bezogen**, um die entsprechenden Ressourcen anzuzeigen. Backup and Recovery listet alle Ressourcen im Wiederherstellungspunkt auf. iii. Wählen Sie die Ressourcen aus, die in den Wiederherstellungsvorgang einbezogen werden sollen.

Bei den angezeigten Ressourcen handelt es sich um die Ressourcen, die derzeit im Cluster vorhanden sind.

b. Wenn Sie fertig sind, wählen Sie **Weiter**.

Spezifische VM-basierte Anwendungsressourcen

a. Wählen Sie **Selektive Ressourcen** aus.

b. Führen Sie einen der folgenden Schritte aus:

- Zur Wiederherstellung ganzer virtueller Maschinen die Registerkarte **Virtuelle Maschinen** auswählen.

  Backup and Recovery listet alle virtuellen Maschinen im Wiederherstellungspunkt auf. Sie können auswählen, welche VMs in den Wiederherstellungsvorgang einbezogen werden sollen.

- Zum Wiederherstellen einzelner persistenter Volume-Ansprüche die Registerkarte **Persistente Volume-Ansprüche** auswählen.

  Backup and Recovery listet alle Persistent Volume Claims im Wiederherstellungspunkt auf. Es kann ausgewählt werden, welche Persistent Volume Claims in die Wiederherstellungsoperation einbezogen werden.

c. Wenn Sie fertig sind, wählen Sie **Weiter**.

Die Seite „Zieleinstellungen“ wird angezeigt.

10. Erweitern Sie den Abschnitt **Destination settings** und wählen Sie aus, ob Sie entweder in der Standard-Speicherklasse, in einer anderen Speicherklasse wiederherstellen möchten oder, wenn Sie in einem anderen Cluster wiederherstellen, die Speicherklassen dem Ziel-Cluster zuordnen möchten.
11. Wenn Sie die Wiederherstellung in einer anderen Speicherklasse gewählt haben, wählen Sie eine Zielspeicherklasse aus, die zu jeder Quellspeicherklasse passt.
12. Optional können Sie, wenn Sie eine mit Trident Protect erstellte Sicherung oder einen Snapshot wiederherstellen, die Details des AppVault, der als Speicher-Bucket für die Wiederherstellungsoperation verwendet wurde, anzeigen. Wenn es eine Änderung in Ihrer Umgebung oder im AppVault-Status gibt,

wählen Sie **Sync App Vault**, um die Details zu aktualisieren.

> Wenn Sie einen AppVault auf einem Kubernetes-Cluster erstellen müssen, um die Wiederherstellung eines mit Trident Protect erstellten Backups oder Snapshots zu erleichtern, lesen Sie "Verwenden Sie Trident Protect AppVault-Objekte, um Buckets zu verwalten".

13. Optional können Sie den Abschnitt **Wiederherstellungsskripte** erweitern und die Option **Postscript** aktivieren, um eine Ausführungs-Hook-Vorlage auszuwählen, die nach Abschluss des Wiederherstellungsvorgangs ausgeführt wird. Geben Sie bei Bedarf alle Argumente ein, die das Skript benötigt, und fügen Sie Label-Selektoren hinzu, um Ressourcen anhand von Ressourcen-Labels zu filtern.
14. Optional können Sie den Abschnitt **Ressourcentransformationen** erweitern, um während des Wiederherstellungsprozesses Ressourcenattribute hinzuzufügen, zu entfernen oder zu ändern. Gehen Sie dann wie folgt vor:

    > Die Modifizierung von PersistentVolumeClaims und Namespaces wird derzeit nicht unterstützt.

    a. Aktivieren Sie die Option **Ressourcentransformation**, um Änderungen am Modifikator vorzunehmen.
    b. Wählen Sie eine Vorlage aus der **Vorlagen**-Liste, um häufig verwendete Modifikatoreinstellungen schnell anzuwenden. Diese Liste enthält vordefinierte Vorlagen für gängige Anwendungsfälle sowie von Ihnen erstellte benutzerdefinierte Vorlagen.

        > Erstellen Sie Ressourcentransformationsvorlagen im globalen "Einstellungen" Bereich.

    c. Geben Sie an, welche Ressource Sie ändern möchten, indem Sie die Ressourcengruppe, die Version, den Kind und den Namen eingeben.
    d. Geben Sie die Operation an, die Sie an der Ressource ausführen möchten, indem Sie eine Operation aus der Liste **Operation** auswählen.
    e. Geben Sie einen JSON-Pfad für den spezifischen Schlüssel ein, den Sie ändern möchten.
    f. Geben Sie gegebenenfalls einen neuen Wert ein. Das Feld **Wert** wird nur bei bestimmten Operationen angezeigt (z. B. **Hinzufügen** oder **Ersetzen**).
    g. Optional können bei Bedarf weitere Ressourcentransformationen hinzugefügt werden.
15. Wenn Sie fertig sind, wählen Sie **Wiederherstellen**.

Kubernetes-Anwendungen mithilfe einer benutzerdefinierten Ressource wiederherstellen

Sie können benutzerdefinierte Ressourcen verwenden, um Ihre Anwendungen aus einem Snapshot oder einem Backup wiederherzustellen. 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 Ausführungs-Hook nach der Wiederherstellung vorhanden ist, wird er automatisch als Teil des Wiederherstellungsvorgangs ausgeführt.
- Die Wiederherstellung aus einem Backup in einen anderen Namespace oder in den ursprünglichen Namespace wird für qtree Volumes unterstützt. Die Wiederherstellung aus einem Snapshot in einen anderen Namespace oder in den ursprünglichen Namespace wird für qtree Volumes jedoch nicht unterstützt.
- Sie können die Wiederherstellungsvorgänge mithilfe erweiterter Einstellungen anpassen. Weitere Informationen finden Sie unter "Erweiterte benutzerdefinierte Ressourcenwiederherstellungseinstellungen verwenden".

Eine Sicherung in einen anderen Namensraum wiederherstellen

Wenn Sie eine Sicherung mithilfe einer BackupRestore CR in einem anderen Namespace wiederherstellen, stellt Backup und Recovery die Anwendung in einem neuen Namespace wieder her und erstellt eine Anwendungs-CR für die wiederhergestellte Anwendung. Um die wiederhergestellte Anwendung zu schützen, erstellen Sie bedarfsgesteuerte Backups oder Snapshots oder legen Sie eine Datensicherungsstrategie fest.

- Die Wiederherstellung einer Sicherung in einem anderen Namensraum mit vorhandenen Ressourcen ändert keine Ressourcen, die denselben Namen wie die in der Sicherung haben. Um alle Ressourcen in der Sicherung wiederherzustellen, löschen und erstellen Sie entweder den Zielnamensraum neu oder stellen Sie die Sicherung in einem neuen Namensraum wieder her.
- Wenn Sie eine CR zur Wiederherstellung in einem neuen Namespace verwenden, müssen Sie den Ziel-Namespace manuell erstellen, bevor Sie die CR anwenden. Backup und Recovery erstellt Namespaces automatisch nur bei Verwendung der CLI.

Bevor Sie beginnen

Stellen Sie sicher, dass die Gültigkeitsdauer des AWS-Sitzungstokens für alle länger dauernden s3-Wiederherstellungsvorgänge ausreichend ist. Wenn das Token während des Wiederherstellungsvorgangs abläuft, kann der Vorgang fehlschlagen.

- Weitere Informationen zum Prüfen des Ablaufs des aktuellen Sitzungstokens finden Sie in der "AWS API-Dokumentation".
- Weitere Informationen zu Anmeldeinformationen für AWS-Ressourcen finden Sie im "AWS IAM-Dokumentation".

Wenn Sie Backups mit Kopia als Datenmover wiederherstellen, können Sie optional Anmerkungen in der CR angeben, um das Verhalten des von Kopia verwendeten temporären Speichers zu steuern. Weitere Informationen zu den Optionen, die Sie konfigurieren können, finden Sie in der "Kopia-Dokumentation".

Schritte

1. Erstellen Sie die benutzerdefinierte Ressourcendatei (CR) 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 sinnvollen Namen für Ihre Umgebung.

- **spec.appArchivePath**: Der Pfad innerhalb von AppVault, in dem die Sicherungsinhalte gespeichert sind. Sie können den 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 AppVault, in dem die Sicherungsinhalte gespeichert sind.
- **spec.namespaceMapping**: Die Zuordnung des Quell-Namespace des Wiederherstellungsvorgangs zum Ziel-Namespace. Ersetzen Sie `my-source-namespace` und `my-destination-namespace` durch 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 bestimmte Ressourcen automatisch aus, weil sie mit den von Ihnen ausgewählten Ressourcen in Beziehung stehen. Wenn Sie beispielsweise eine Ressource für einen persistenten Volume-Claim auswählen und diese einen zugehörigen Pod hat, wird Trident Protect auch den zugehörigen Pod wiederherstellen.

- **resourceFilter.resourceSelectionCriteria**: (Für die Filterung erforderlich) Verwenden Sie `Include` oder `Exclude`, um eine in resourceMatchers definierte Ressource ein- oder auszuschließen. Fügen Sie die folgenden resourceMatchers Parameter hinzu, um die Ressourcen zu definieren, die ein- oder auszuschließen sind:
  - **resourceFilter.resourceMatchers**: Ein Array von resourceMatcher-Objekten. Wenn Sie mehrere Elemente in diesem Array definieren, werden diese mit einer ODER-Verknüpfung verglichen und die Felder innerhalb jedes Elements (group, kind, version) werden mit einer UND-Verknüpfung verglichen.
    - **resourceMatchers[].group**: (*Optional*) Gruppe der zu filternden Ressource.
    - **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 Ressource, die gefiltert werden soll.
    - **resourceMatchers[].namespaces**: (*Optional*) Namespaces im Kubernetes metadata.name-

Feld der Ressource, die gefiltert werden soll.

- **resourceMatchers[].labelSelectors**: (*Optional*) Label-Selektorzeichenfolge im Kubernetes-Metadatenfeld name der Ressource, wie definiert in der "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 gefüllt haben, wenden Sie die CR an:

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

Stellen Sie ein Backup im ursprünglichen Namespace wieder her

Sie können ein Backup jederzeit im ursprünglichen Namespace wiederherstellen.

Bevor Sie beginnen

Stellen Sie sicher, dass die Gültigkeitsdauer des AWS-Sitzungstokens für alle länger dauernden s3-Wiederherstellungsvorgänge ausreichend ist. Wenn das Token während des Wiederherstellungsvorgangs abläuft, kann der Vorgang fehlschlagen.

- Weitere Informationen zum Prüfen des Ablaufs des aktuellen Sitzungstokens finden Sie in der "AWS API-Dokumentation".
- Weitere Informationen zu Anmeldeinformationen für AWS-Ressourcen finden Sie im "AWS IAM-Dokumentation".

Wenn Sie Backups mit Kopia als Datenmover wiederherstellen, können Sie optional Anmerkungen in der CR angeben, um das Verhalten des von Kopia verwendeten temporären Speichers zu steuern. Weitere Informationen zu den Optionen, die Sie konfigurieren können, finden Sie in der "Kopia-Dokumentation".

Schritte

1. Erstellen Sie die benutzerdefinierte Ressourcendatei (CR) 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 sinnvollen Namen für Ihre Umgebung.
   - **spec.appArchivePath**: Der Pfad innerhalb von AppVault, in dem die Sicherungsinhalte gespeichert sind. Sie können den 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 AppVault, 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 bestimmte Ressourcen automatisch aus, weil sie mit den von Ihnen ausgewählten Ressourcen in Beziehung stehen. Wenn Sie beispielsweise eine Ressource für einen persistenten Volume-Claim auswählen und diese einen zugehörigen Pod hat, wird Trident Protect auch den zugehörigen Pod wiederherstellen.

   - **resourceFilter.resourceSelectionCriteria**: (Für die Filterung erforderlich) Verwenden Sie `Include` oder `Exclude`, um eine in resourceMatchers definierte Ressource ein- oder auszuschließen. Fügen Sie die folgenden resourceMatchers Parameter hinzu, um die Ressourcen zu definieren, die ein- oder auszuschließen sind:
     - **resourceFilter.resourceMatchers**: Ein Array von resourceMatcher-Objekten. Wenn Sie mehrere Elemente in diesem Array definieren, werden diese mit einer ODER-Verknüpfung verglichen und die Felder innerhalb jedes Elements (group, kind, version) werden mit einer UND-Verknüpfung

verglichen.

- **resourceMatchers[].group**: (*Optional*) Gruppe der zu filternden Ressource.
- **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 Ressource, die gefiltert werden soll.
- **resourceMatchers[].namespaces**: (*Optional*) Namespaces im Kubernetes metadata.name-Feld der Ressource, die gefiltert werden soll.
- **resourceMatchers[].labelSelectors**: (*Optional*) Label-Selektorzeichenfolge im Kubernetes-Metadatenfeld name der Ressource, wie definiert in der "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 gefüllt haben, wenden Sie die CR an:

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

Stellen Sie ein Backup auf einem anderen Cluster wieder her

Sie können ein Backup auf einem anderen Cluster wiederherstellen, wenn es ein Problem mit dem ursprünglichen Cluster gibt.

- Wenn Sie Backups mit Kopia als Datenmover wiederherstellen, können Sie optional Anmerkungen in der CR angeben, um das Verhalten des von Kopia verwendeten temporären Speichers zu steuern. Weitere Informationen zu den Optionen, die Sie konfigurieren können, finden Sie in der "Kopia-Dokumentation".
- Wenn Sie eine CR verwenden, um in einem neuen Namespace wiederherzustellen, müssen Sie den Ziel-Namespace manuell erstellen, bevor Sie die CR anwenden.

Bevor Sie beginnen

Stellen Sie sicher, dass die folgenden Voraussetzungen erfüllt sind:

- Auf dem Ziel-Cluster ist Trident Protect installiert.
- Der Ziel-Cluster hat Zugriff auf den Bucket-Pfad desselben AppVault wie der Quell-Cluster, in dem die Sicherung gespeichert ist.
- Stellen Sie sicher, dass die Gültigkeitsdauer des AWS-Sitzungstokens für alle länger dauernden Wiederherstellungsvorgänge ausreichend ist. Wenn das Token während des Wiederherstellungsvorgangs abläuft, kann der Vorgang fehlschlagen.
  - Weitere Informationen zum Prüfen des Ablaufs des aktuellen Sitzungstokens finden Sie in der "AWS API-Dokumentation".
  - Weitere Informationen zu Anmeldeinformationen für AWS-Ressourcen finden Sie in der "AWS-Dokumentation".

Schritte

1. Überprüfen Sie die Verfügbarkeit des AppVault CR auf dem Ziel-Cluster 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 Ziel-Cluster vorhanden ist.

2. Zeigen Sie die Sicherungsinhalte des verfügbaren AppVault vom Ziel-Cluster 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, entsprechenden 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. Stellen Sie die Anwendung im Ziel-Cluster mithilfe des AppVault-Namens und des Archivpfads wieder her:
4. Erstellen Sie die benutzerdefinierte Ressourcendatei (CR) und benennen Sie sie `trident-protect-backup-restore-cr.yaml`.
5. Konfigurieren Sie in der von Ihnen erstellten Datei die folgenden Attribute:
   - **metadata.name**: (*Erforderlich*) Der Name dieser benutzerdefinierten Ressource; wählen Sie einen eindeutigen und sinnvollen Namen für Ihre Umgebung.
   - **spec.appVaultRef**: (*Erforderlich*) Der Name des AppVault, in dem die Sicherungsinhalte gespeichert sind.
   - **spec.appArchivePath**: Der Pfad innerhalb von AppVault, in dem die Sicherungsinhalte gespeichert sind. Sie können den 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 des Wiederherstellungsvorgangs zum Ziel-Namespace. Ersetzen Sie `my-source-namespace` und `my-destination-namespace` durch 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"}]
```

6. Nachdem Sie die `trident-protect-backup-restore-cr.yaml` Datei mit den korrekten Werten gefüllt haben, wenden Sie die CR an:

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

Einen Snapshot in einen anderen Namespace wiederherstellen

Sie können Daten aus einem Snapshot mithilfe einer benutzerdefinierten Ressourcendatei (CR) entweder in einem anderen Namespace oder im ursprünglichen Quell-Namespace wiederherstellen. Wenn Sie einen Snapshot mithilfe einer SnapshotRestore CR in einem anderen Namespace wiederherstellen, stellt Backup und Recovery die Anwendung in einem neuen Namespace wieder her und erstellt eine Anwendungs-CR für die wiederhergestellte Anwendung. Um die wiederhergestellte Anwendung zu schützen, erstellen Sie On-Demand-Backups oder Snapshots, oder legen Sie einen Datensicherungszeitplan fest.

- SnapshotRestore unterstützt das `spec.storageClassMapping` Attribut, jedoch nur, wenn die Quell- und Ziel-Speicherklassen dasselbe Speicher-Backend verwenden. Wenn Sie versuchen, auf eine `StorageClass` wiederherzustellen, die ein anderes Speicher-Backend verwendet, schlägt der Wiederherstellungsvorgang fehl.
- Wenn Sie eine CR verwenden, um in einem neuen Namespace wiederherzustellen, müssen Sie den Ziel-Namespace manuell erstellen, bevor Sie die CR anwenden.

Bevor Sie beginnen

Stellen Sie sicher, dass die Gültigkeitsdauer des AWS-Sitzungstokens für alle länger dauernden s3-Wiederherstellungsvorgänge ausreichend ist. Wenn das Token während des Wiederherstellungsvorgangs abläuft, kann der Vorgang fehlschlagen.

- Weitere Informationen zum Prüfen des Ablaufs des aktuellen Sitzungstokens finden Sie in der "AWS API-Dokumentation".
- Weitere Informationen zu Anmeldeinformationen für AWS-Ressourcen finden Sie im "AWS IAM-Dokumentation".

Schritte

1. Erstellen Sie die benutzerdefinierte Ressourcendatei (CR) 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 sinnvollen Namen für Ihre Umgebung.
   - **spec.appVaultRef**: (*Erforderlich*) Der Name des AppVault, in dem die Snapshot-Inhalte gespeichert sind.
   - **spec.appArchivePath**: Der Pfad innerhalb von AppVault, in dem die Snapshot-Inhalte gespeichert sind. Sie können den 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 des Wiederherstellungsvorgangs zum Ziel-Namespace. Ersetzen Sie `my-source-namespace` und `my-destination-namespace` durch 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 bestimmte Ressourcen automatisch aus, weil sie mit den von Ihnen ausgewählten Ressourcen in Beziehung stehen. Wenn Sie beispielsweise eine Ressource für einen persistenten Volume-Claim auswählen und diese einen zugehörigen Pod hat, wird Trident Protect auch den zugehörigen Pod wiederherstellen.

   - **resourceFilter.resourceSelectionCriteria**: (Für die Filterung erforderlich) Verwenden Sie `Include` oder `Exclude`, um eine in resourceMatchers definierte Ressource ein- oder auszuschließen. Fügen Sie die folgenden resourceMatchers Parameter hinzu, um die Ressourcen zu definieren, die ein- oder auszuschließen sind:
     - **resourceFilter.resourceMatchers**: Ein Array von resourceMatcher-Objekten. Wenn Sie mehrere Elemente in diesem Array definieren, werden diese mit einer ODER-Verknüpfung verglichen und die Felder innerhalb jedes Elements (group, kind, version) werden mit einer UND-Verknüpfung verglichen.
       - **resourceMatchers[].group**: (*Optional*) Gruppe der zu filternden Ressource.
       - **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 Ressource, die gefiltert werden soll.
- **resourceMatchers[].namespaces**: (*Optional*) Namespaces im Kubernetes metadata.name-Feld der Ressource, die gefiltert werden soll.
- **resourceMatchers[].labelSelectors**: (*Optional*) Label-Selektorzeichenfolge im Kubernetes-Metadatenfeld name der Ressource, wie definiert in der "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 gefüllt haben, wenden Sie die CR an:

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

Einen Snapshot im ursprünglichen Namensraum wiederherstellen

Sie können einen Snapshot jederzeit im ursprünglichen Namensraum wiederherstellen.

> Die In-Place-Wiederherstellung (Wiederherstellung im ursprünglichen Namespace und ursprünglichen Cluster) von VM-basierten Anwendungen aus lokalen Snapshots wird derzeit nicht unterstützt.

Bevor Sie beginnen

Stellen Sie sicher, dass die Gültigkeitsdauer des AWS-Sitzungstokens für alle länger dauernden s3-Wiederherstellungsvorgänge ausreichend ist. Wenn das Token während des Wiederherstellungsvorgangs abläuft, kann der Vorgang fehlschlagen.

- Weitere Informationen zum Prüfen des Ablaufs des aktuellen Sitzungstokens finden Sie in der "AWS API-Dokumentation".
- Weitere Informationen zu Anmeldeinformationen für AWS-Ressourcen finden Sie im "AWS IAM-Dokumentation".

Schritte

1. Erstellen Sie die benutzerdefinierte Ressourcendatei (CR) 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 sinnvollen Namen für Ihre Umgebung.
   - **spec.appVaultRef**: (*Erforderlich*) Der Name des AppVault, in dem die Snapshot-Inhalte gespeichert sind.
   - **spec.appArchivePath**: Der Pfad innerhalb von AppVault, in dem die Snapshot-Inhalte gespeichert sind. Sie können den 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 bestimmte Ressourcen automatisch aus, weil sie mit den von Ihnen ausgewählten Ressourcen in Beziehung stehen. Wenn Sie beispielsweise eine Ressource für einen persistenten Volume-Claim auswählen und diese einen zugehörigen Pod hat, wird Trident Protect auch den zugehörigen Pod wiederherstellen.

   - **resourceFilter.resourceSelectionCriteria**: (Für die Filterung erforderlich) Verwenden Sie `Include` oder `Exclude`, um eine in resourceMatchers definierte Ressource ein- oder auszuschließen. Fügen Sie die folgenden resourceMatchers Parameter hinzu, um die Ressourcen zu definieren, die ein- oder auszuschließen sind:
     - **resourceFilter.resourceMatchers**: Ein Array von resourceMatcher-Objekten. Wenn Sie mehrere Elemente in diesem Array definieren, werden diese mit einer ODER-Verknüpfung verglichen und die Felder innerhalb jedes Elements (group, kind, version) werden mit einer UND-Verknüpfung verglichen.
       - **resourceMatchers[].group**: (*Optional*) Gruppe der zu filternden Ressource.

- **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 Ressource, die gefiltert werden soll.
- **resourceMatchers[].namespaces**: (*Optional*) Namespaces im Kubernetes metadata.name-Feld der Ressource, die gefiltert werden soll.
- **resourceMatchers[].labelSelectors**: (*Optional*) Label-Selektorzeichenfolge im Kubernetes-Metadatenfeld name der Ressource, wie definiert in der "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 gefüllt haben, wenden Sie die CR an:

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

Erweiterte benutzerdefinierte Ressourcenwiederherstellungseinstellungen verwenden

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

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

Bei Wiederherstellung und Failover werden die Namespace-Bezeichnungen und -Annotationen des Ziels aktualisiert, um mit denen der Quelle übereinzustimmen: Schlüssel aus der Quelle werden zu den Zielschlüsseln hinzugefügt oder überschrieben, während Schlüssel, die nur im Ziel existieren, unverändert bleiben.

> In Red Hat OpenShift sind Namespace-Annotationen wichtig, da sie sicherstellen, dass wiederhergestellte Pods die korrekten Sicherheitskontextbeschränkungen und Berechtigungen erhalten, sodass sie auf Volumes zugreifen und ohne Berechtigungsfehler ausgeführt werden können. Weitere Informationen finden Sie unter "OpenShift security context constraints Dokumentation".

Setzen Sie die Kubernetes-Umgebungsvariable

```
RESTORE_SKIP_NAMESPACE_ANNOTATIONS
```

Vor der Wiederherstellung oder dem Failover sollte verhindert werden, dass bestimmte Ziel-Namespace-Annotationen überschrieben werden. Zum Beispiel:

```
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
```

> Während der Wiederherstellung oder des Failovers werden alle in `restoreSkipNamespaceAnnotations` und `restoreSkipNamespaceLabels` angegebenen Namespace-Annotationen und -Labels von der Wiederherstellungs- oder Failover-Operation ausgeschlossen. Stellen Sie sicher, dass diese Einstellungen während der initialen Helm-Installation konfiguriert sind. Weitere Informationen finden Sie unter "Konfigurieren Sie zusätzliche Trident Protect Helm-Chart-Einstellungen".

Wenn Sie Helm mit dem `--create-namespace` Flag zur Installation der Quellanwendung verwendet haben, kopiert Trident Protect die Namensbezeichnung in den Ziel-Namespace. Stimmt der Wert der Bezeichnung mit dem Namen des Quell-Namespace überein, wird er durch den Namen des Ziel-Namespace ersetzt; andernfalls bleibt er unverändert.

Beispiel

Das folgende Beispiel zeigt Quell- und Ziel-Namensräume mit unterschiedlichen Bezeichnungen und Annotationen und zeigt den Ziel-Namensraum vor und nach der Operation, um zu veranschaulichen, wie Schlüssel hinzugefügt, zusammengeführt 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	Etiketten
Namespace ns-1 (Quelle)	• annotation.one/key: "updatedvalue" • annotation.two/key: "true"	• environment=production • compliance=hipaa • name=ns-1
Namespace ns-2 (Ziel)	• annotation.one/key: "true" • annotation.three/key: "false"	• role=database

Nach dem Wiederherstellungsvorgang

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

Namensraum	Anmerkungen	Etiketten
Namespace ns-2 (Ziel)	• annotation.one/key: "updatedvalue" • annotation.two/key: "true" • annotation.three/key: "false"	• name=ns-2 • compliance=hipaa • environment=production • role=database

Unterstützte Felder

In diesem Abschnitt werden zusätzliche Felder beschrieben, die für Wiederherstellungsvorgänge zur Verfügung stehen.

Speicherklassenzuordnung

Das `spec.storageClassMapping` Attribut definiert eine Zuordnung von einer Speicherklasse in der Quellanwendung zu einer neuen Speicherklasse im Zielcluster. Sie können dies verwenden, wenn Sie Anwendungen zwischen Clustern mit unterschiedlichen Speicherklassen migrieren oder das Speicher-Backend für BackupRestore-Operationen 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	Zeichenkette	Die maximal zulässige Zeit (in Sekunden), in der der Datenübertragungsvorgang angehalten werden darf.	"300"
protect.trident.netapp.io/kopia-content-cache-size-limit-mb	Zeichenkette	Die maximale Größenbeschränkung (in Megabytes) für den Kopia-Inhaltscache.	"1000"
protect.trident.netapp.io/pvc-bind-timeout-sec	Zeichenkette	Maximale Zeit (in Sekunden), die auf neu erstellte PersistentVolumeClaims (PVCs) gewartet wird, um die `Bound` Phase zu erreichen, bevor der Vorgang fehlschlägt. Gilt für alle Restore-CR-Typen (BackupRestore, BackupInplaceRestore, SnapshotRestore, SnapshotInplaceRestore). Verwenden Sie einen höheren Wert, wenn Ihr Storage-Backend oder Cluster häufig mehr Zeit benötigt.	"1200" (20 Minuten)

Ändern Sie Ressourcen während der Wiederherstellung mithilfe benutzerdefinierter Ressourcen

Ressourcentransformationen ermöglichen es Ihnen, eine Ressource während der Wiederherstellung zu modifizieren. Dies ist nützlich, wenn die wiederhergestellte Version von der ursprünglichen Version abweichen soll – beispielsweise die Änderung der IP-Adresse einer virtuellen Maschine bei der Wiederherstellung in einem anderen Netzwerk. Sie können auch "Modifizieren Sie Ressourcen während der Wiederherstellung mithilfe der Web-Benutzeroberfläche".

Erforderliche NetApp Console-Rolle Backup and Recovery Superadministrator oder Backup and Recovery Wiederherstellungsadministrator. "Erfahren Sie mehr über die Zugriffsrollen für NetApp Backup and Recovery". "Erfahren Sie mehr über die Zugriffsrollen der NetApp Console für alle Dienste".

Wie Ressourcenmodifikation funktioniert

Das `transformations` Feld in `SnapshotRestore`, `BackupRestore`, `AppMirrorRelationship` und anderen Wiederherstellungsressourcen ermöglicht es Ihnen, Kubernetes-Ressourcen während des Wiederherstellungsprozesses zu ändern. Dies ist nützlich, um Anwendungen oder virtuelle Maschinen an einen neuen Cluster anzupassen, indem Sie Hostnamen, Registry-URLs, Ressourcenlimits oder Umgebungsvariablen ändern.

Ressourcentransformationen verwenden "RFC 6902" JSON-Patch-Operationen und "RFC 6901" JSON-Pointer-Pfade, um bestimmte Felder innerhalb von Kubernetes-Ressourcen anzusprechen und zu

modifizieren.

Hier ist die grundlegende Struktur eines Wiederherstellungsobjekts, das Ressourcentransformationen enthält:

```
apiVersion: protect.trident.netapp.io/v1
kind: SnapshotRestore
metadata:
  name: my-restore
  namespace: target-namespace
spec:
  appVaultRef: my-vault
  appArchivePath: /path/to/snapshot
  namespaceMapping:
    - source: source-ns
      destination: target-ns
  transformations:
    - resource:
        kind: ConfigMap       # Required: resource kind
        group: ""             # Optional: API group (empty for core
resources)
        version: ""           # Optional: API version
        name: ""              # Optional: specific resource name
      operations:
        - op: replace         # Operation type
          path: "/data/key"   # JSON Pointer path
          value: "new-value"  # New value (for add/replace/test)
```

Unterstützte Ressourcen

Sie können Ressourcentransformationen mit Ressourcen verwenden, die den folgenden Kriterien entsprechen:

- `kind` (erforderlich): Der Kubernetes-Ressourcentyp (zum Beispiel, `ConfigMap`, `Deployment`, `Pod`)
- `group` (optional): Die API-Gruppe (zum Beispiel, `apps`, `route.openshift.io`) - für Kernressourcen weglassen
- `version` (optional): Die API-Version (z. B. `v1`, `v1beta1`)
- `name` (optional): Nur auf eine bestimmte Ressource anhand des Namens anwenden

Die Modifizierung von PersistentVolumeClaims und Namespaces wird derzeit nicht unterstützt.

Unterstützte Operationen

Sie können die folgenden Operationen verwenden, um Ressourcen zu ändern:

- `add`: Füge einer Ressource einen Wert hinzu.
- `copy`: Einen Wert von einem Pfad in einen anderen kopieren.

- `move`: Einen Wert innerhalb einer Ressource verschieben.
- `remove`: Einen Wert aus einer Ressource entfernen.
- `replace`: Ersetzen Sie einen Wert innerhalb einer Ressource.
- `test`: Testen Sie eine Operation, bevor Sie sie ausführen.

Einer Ressource einen Wert hinzufügen

Verwenden Sie die `add`-Operation, um ein neues Feld oder einen neuen Wert am angegebenen Pfad hinzuzufügen. Sie können Daten zu Objekten oder Arrays hinzufügen. Das folgende Beispiel fügt einer Deployment-Ressource einen Knotenselektor hinzu:

```
transformations:
  - resource:
      kind: Deployment
    operations:
      - op: add
        path: "/spec/template/spec/nodeSelector"
        value:
          "topology.kubernetes.io/zone": "us-east-1a"
          disktype: "ssd"
```

Verwenden Sie den folgenden Befehl, um diese Transformation über die Befehlszeile auszuführen:

```
tridentctl-protect --transformation
'apps,v1,Deployment:add{"path":"/spec/template/spec/nodeSelector","value":
{"topology.kubernetes.io/zone":"us-east-1a","disktype":"ssd"}}'
```

Einen Wert innerhalb einer Ressource kopieren

Verwenden Sie die `copy`Operation, um einen Wert innerhalb derselben Ressource von einem Pfad in einen anderen zu kopieren. Die Quelle bleibt unverändert. Das folgende Beispiel dupliziert einen Datenschlüssel für ein ConfigMap-Objekt:

```
transformations:
  - resource:
      kind: ConfigMap
    operations:
      - op: copy
        from: "/data/source-key"
        path: "/data/backup-key"
```

Verwenden Sie den folgenden Befehl, um diese Transformation über die Befehlszeile auszuführen:

```
tridentctl-protect --transformation
',v1,ConfigMap:copy{"from":"/data/source-key","path":"/data/backup-key"}'
```

Verschieben Sie einen Wert innerhalb einer Ressource

Verwenden Sie die `move`-Operation, um einen Wert innerhalb derselben Ressource von einem Pfad zu einem anderen zu verschieben. Die Quelle wird entfernt und der Wert am Zielort eingefügt. Das folgende Beispiel benennt einen Datenschlüssel für ein ConfigMap-Objekt um:

```
transformations:
  - resource:
      kind: ConfigMap
    operations:
      - op: move
        from: "/data/OLD_KEY"
        path: "/data/NEW_KEY"
```

Verwenden Sie den folgenden Befehl, um diese Transformation über die Befehlszeile auszuführen:

```
tridentctl-protect --transformation
',v1,ConfigMap:move{"from":"/data/OLD_KEY","path":"/data/NEW_KEY"}'
```

Einen Wert aus einer Ressource entfernen

Verwenden Sie die `remove` Operation, um ein Feld oder einen Wert am angegebenen Pfad zu entfernen. Das folgende Beispiel entfernt eine Annotation aus einer ConfigMap-Ressource:

```
transformations:
  - resource:
      kind: ConfigMap
    operations:
      - op: remove
        path: "/metadata/annotations/kubectl.kubernetes.io~1last-applied-
configuration"
```

Im Pfad des obigen Beispiels `~1` ist die JSON Pointer Escape-Sequenz für `/`.

Verwenden Sie den folgenden Befehl, um diese Transformation über die Befehlszeile auszuführen:

```
tridentctl-protect --transformation
',v1,ConfigMap:remove{"path":"/metadata/annotations/kubectl.kubernetes.io~
1last-applied-configuration"}'
```

Ersetzen Sie einen Wert innerhalb einer Ressource

Verwenden Sie die `replace` Operation, um einen vorhandenen Wert innerhalb einer Ressource am angegebenen Pfad zu ersetzen. Der JSON-Pfad muss bereits existieren. Das folgende Beispiel ändert einen Hostnamen für ein Route-Objekt:

```
transformations:
  - resource:
      kind: Route
      group: route.openshift.io
    operations:
      - op: replace
        path: "/spec/host"
        value: "prod.example.com"
```

Verwenden Sie den folgenden Befehl, um diese Transformation über die Befehlszeile auszuführen:

```
tridentctl-protect --transformation
'route.openshift.io,v1,Route:replace{"path":"/spec/host","value":"prod.exa
mple.com"}'
```

Testen Sie die Ressourcenänderung

Verwenden Sie die `test`Operation, um zu testen, ob der Wert an einem Pfad dem erwarteten Wert entspricht. Wenn der Test fehlschlägt, wird die gesamte Änderung zurückgesetzt. Im folgenden Beispiel wird `database-host`nur aktualisiert, wenn `environment staging` entspricht:

```
transformations:
  - resource:
      kind: ConfigMap
    operations:
      - op: test
        path: "/data/environment"
        value: "staging"
      - op: replace
        path: "/data/database-host"
        value: "prod-db.example.com"
```

Verwenden Sie den folgenden Befehl, um diese Transformation über die Befehlszeile auszuführen:

```
tridentctl-protect --transformation
',v1,ConfigMap:test{"path":"/data/environment","value":"staging"},replace{
"path":"/data/database-host","value":"prod-db.example.com"}'
```