



Kubernetes アプリケーションを復元する NetApp Backup and Recovery

NetApp
February 10, 2026

目次

Kubernetesアプリケーションを復元する	1
Web UI を使用して Kubernetes アプリケーションをリストアする	1
カスタムリソースを使用した Kubernetes アプリケーションのリストア	3
バックアップを別の名前空間に復元する	3
バックアップを元の名前空間に復元する	5
バックアップを別のクラスタにリストアする	7
スナップショットを別のネームスペースにリストアする	9
スナップショットを元の名前空間に復元する	12
高度なカスタムリソースのリストア設定を使用する	14
リストアおよびフェイルオーバー処理中のネームスペースのアノテーションとラベル	14
サポートされているフィールド	15
サポートされているアノテーション	16

Kubernetesアプリケーションを復元する

Web UI を使用して Kubernetes アプリケーションをリストアする

NetApp Backup and Recovery を使用すると、保護ポリシーで保護したアプリケーションを復元できます。アプリケーションを復元するには、少なくとも1つの復元ポイントが必要です。復元ポイントは、ローカル スナップショットまたはオブジェクトストアへのバックアップ（あるいはその両方）で構成されます。アプリケーションは、ローカルアーカイブ、セカンダリ アーカイブ、またはオブジェクトストア アーカイブを使用して復元できます。

開始する前に

Trident Protect を使用してバックアップされたアプリケーションを復元する場合は、Trident Protect がソースクラスターとデスティネーションクラスターの両方にインストールされていることを確認してください。

必要なNetApp Consoleロール

組織管理者またはSnapCenter管理者。"[NetApp Backup and Recoveryのアクセス ロールについて学習します](#)"。
"[すべてのサービスに対するNetApp Consoleのアクセスロールについて学習します](#)"。

手順

1. NetApp Backup and Recoveryメニューで、*リストア*を選択します。
2. リストから Kubernetes アプリケーションを選択し、そのアプリケーションの *表示と復元* を選択します。

復元ポイントのリストが表示されます。

3. 使用する復元ポイントの*Restore*ボタンを選択します。

一般設定

1. 復元元のソースの場所を選択します。
2. クラスター リストから宛先クラスターを選択します。



Trident Protectで作成したローカルスナップショットを別のクラスターにリストアすることは、現時点ではサポートされていません。

3. 元のネームスペースに復元するか、新しいネームスペースに復元するかを選択します。
4. 新しいネームスペースにリストアすることを選択した場合は、使用するデスティネーションネームスペースを入力します。
5. *次へ*を選択します。

リソースの選択

1. アプリケーションに関連付けられているすべてのリソースを復元するか、フィルターを使用して復元する特定のリソースを選択するかを選択します。

すべてのリソースを復元する

1. *すべてのリソースを復元*を選択します。
2. *次へ*を選択します。

特定のリソースを復元する

1. *選択的リソース*を選択します。
2. リソース フィルターの動作を選択します。*含める*を選択すると、選択したリソースが復元されます。*除外*を選択した場合、選択したリソースは復元されません。
3. リソースを選択するためのフィルターを定義するルールを追加するには、[ルールの追加] を選択します。リソースをフィルタリングするには、少なくとも1つのルールが必要です。

各ルールは、リソースの名前空間、ラベル、グループ、バージョン、種類などの基準でフィルタリングできます。

4. 各ルールを保存するには、[保存] を選択します。
5. 必要なルールをすべて追加したら、「検索」を選択して、フィルター条件に一致するバックアップアーカイブ内の利用可能なリソースを表示します。



表示されるリソースは、クラスター上に現在存在するリソースです。

6. 結果に満足したら、[次へ] を選択します。

宛先設定

1. *Destination settings*セクションを展開し、デフォルトのストレージクラス、別のストレージクラスのいずれかに復元するか、別のクラスターに復元する場合は、ストレージクラスをデスティネーション クラスターにマッピングするかを選択します。
2. 別のストレージ クラスにリストアすることを選択した場合は、各ソース ストレージ クラスに一致するデスティネーション ストレージ クラスを選択します。
3. オプションで、Trident Protect を使用して作成されたバックアップまたは Snapshot をリストアする場合は、リストア処理のストレージバケットとして使用される AppVault の詳細を表示します。環境または AppVault のステータスに変更がある場合は、* App Vault を同期*を選択して詳細を更新します。



Trident Protectを使用して作成されたバックアップまたはスナップショットの復元を容易にするために、Kubernetesクラスター上にAppVaultを作成する必要がある場合は、"[Trident Protect AppVaultオブジェクトを使用してバケットを管理する](#)"を参照してください。

4. 必要に応じて、[復元スクリプト] セクションを展開し、[Postscript] オプションを有効にして、復元操作の完了後に実行される実行フック テンプレートを選択します。必要に応じて、スクリプトに必要な引数を入力し、ラベル セレクタを追加して、リソース ラベルに基づいてリソースをフィルタします。
5. *復元*を選択します。

カスタムリソースを使用した Kubernetes アプリケーションのリストア

カスタム リソースを使用して、スナップショットまたはバックアップからアプリケーションを復元できます。アプリケーションを同じクラスタに復元する場合、既存のスナップショットからの復元の方が高速になります。



- アプリケーションを復元すると、アプリケーションに対して設定されているすべての実行フックもアプリとともに復元されます。復元後の実行フックが存在する場合、復元操作の一部として自動的に実行されます。
- バックアップから別の名前空間または元の名前空間への復元は、qtree ボリュームでサポートされています。ただし、スナップショットから別の名前空間または元の名前空間への復元は、qtree ボリュームではサポートされていません。
- 詳細設定を使用して復元操作をカスタマイズできます。詳細については、"[高度なカスタムリソースのリストア設定を使用する](#)"を参照してください。

バックアップを別の名前空間に復元する

BackupRestore CR を使用して異なる名前空間にバックアップを復元すると、NetApp Backup and Recovery はアプリケーションを新しい名前空間に復元し、復元されたアプリケーションのアプリケーション CR を作成します。復元されたアプリケーションを保護するには、オンデマンドバックアップまたはスナップショットを作成するか、保護スケジュールを確立します。



- 既存のリソースを含む別の名前空間にバックアップをリストアしても、バックアップ内のリソースと名前を共有するリソースは変更されません。バックアップ内のすべてのリソースをリストアするには、ターゲット名前空間を削除して再作成するか、バックアップを新しい名前空間にリストアします。
- CR を使用して新しい名前空間に復元する場合は、CR を適用する前に、デスティネーション名前空間を手動で作成する必要があります。NetApp Backup and Recovery では、CLI を使用する場合にのみ名前空間が自動的に作成されます。

開始する前に

AWS セッショントークンの有効期限が、長時間実行される s3 復元処理に十分であることを確認します。復元処理中にトークンの有効期限が切れると、処理が失敗する可能性があります。

- 現在のセッショントークンの有効期限を確認する方法の詳細については、"[AWS API ドキュメント](#)"を参照してください。
- "[AWS IAM ドキュメント](#)"AWS リソースの認証情報の詳細については、こちらを参照してください。



Kopia をデータムーバーとして使用してバックアップを復元する場合、オプションで CR に注釈を指定して、Kopia が使用する一時ストレージの動作を制御できます。"[Kopiaのドキュメント](#)"設定できるオプションの詳細については、こちらを参照してください。

手順

1. カスタムリソース (CR) ファイルを作成し、名前を付けます `trident-protect-backup-restore-cr.yaml`。

2. 作成したファイルで、次の属性を設定します：

- **metadata.name**：（必須）このカスタム リソースの名前。環境に合わせて一意かつ適切な名前を選択してください。
- **spec.appArchivePath**：AppVault内でバックアップコンテンツが保存されるパス。このパスを見つけるには、次のコマンドを使用できます：

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

- **spec.appVaultRef**：（必須）バックアップコンテンツが保存されるAppVaultの名前。
- **spec.namespaceMapping**：リストア処理のソースネームスペースからデスティネーションネームスペースへのマッピング。`my-source-namespace`と`my-destination-namespace`を環境の情報に置き換えます。

```
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. （オプション）復元するアプリケーションの特定のリソースのみを選択する必要がある場合は、特定のラベルでマークされたリソースを含めるか除外するフィルタリングを追加します：



Trident Protect は、選択したリソースとの関係に基づいて、一部のリソースを自動的に選択します。たとえば、永続ボリュームクレームリソースを選択し、それに関連付けられたポッドがある場合、Trident Protect は関連付けられているポッドも復元します。

- **resourceFilter.resourceSelectionCriteria**：（フィルタリングに必須）`Include`または`Exclude`を使用して、resourceMatchersで定義されたリソースを含めるか除外します。含めるまたは除外するリソースを定義するには、以下のresourceMatchersパラメータを追加します：
 - **resourceFilter.resourceMatchers**：resourceMatcherオブジェクトの配列。この配列に複数の要素を定義すると、それらはOR演算として一致し、各要素内のフィールド（グループ、種類、バージョン）はAND演算として一致します。
 - **resourceMatchers[].group**：（オプション）フィルタリングするリソースのグループ。
 - **resourceMatchers[].kind**：（オプション）フィルタリングするリソースの種類。
 - **resourceMatchers[].version**：（オプション）フィルタリングするリソースのバージョン。
 - **resourceMatchers[].names**：（オプション）フィルタリングするリソースの Kubernetes metadata.name フィールド内の名前。

- `resourceMatchers[].namespaces` : (オプション) フィルタリングするリソースの Kubernetes metadata.name フィールド内の名前空間。
- `resourceMatchers[].labelSelectors` : (オプション) "[Kubernetesドキュメント](#)"で定義されているリソースのKubernetesメタデータ.nameフィールドのラベルセレクタ文字列。例：
"trident.netapp.io/os=linux"。

例：

```
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. `trident-protect-backup-restore-cr.yaml` ファイルに正しい値を入力したら、CRを適用します：

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

バックアップを元の名前空間に復元する

いつでもバックアップを元の名前空間に復元できます。

開始する前に

AWS セッショントークンの有効期限が、長時間実行される s3 復元処理に十分であることを確認します。復元処理中にトークンの有効期限が切れると、処理が失敗する可能性があります。

- 現在のセッショントークンの有効期限を確認する方法の詳細については、"[AWS API ドキュメント](#)"を参照してください。
- "[AWS IAM ドキュメント](#)"AWS リソースの認証情報の詳細については、こちらを参照してください。



Kopia をデータムーバーとして使用してバックアップを復元する場合、オプションで CR に注釈を指定して、Kopia が使用する一時ストレージの動作を制御できます。"[Kopiaのドキュメント](#)"設定できるオプションの詳細については、こちらを参照してください。

手順

1. カスタムリソース (CR) ファイルを作成し、名前を `trident-protect-backup-ipr-cr.yaml` とします。
2. 作成したファイルで、次の属性を設定します：
 - **metadata.name** : (必須) このカスタム リソースの名前。環境に合わせて一意かつ適切な名前を選択してください。
 - **spec.appArchivePath** : AppVault内でバックアップコンテンツが保存されるパス。このパスを見つけるには、次のコマンドを使用できます：

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

- **spec.appVaultRef** : (必須) バックアップコンテンツが保存されるAppVaultの名前。

例：

```
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. (オプション) 復元するアプリケーションの特定のリソースのみを選択する必要がある場合は、特定のラベルでマークされたリソースを含めるか除外するフィルタリングを追加します：



Trident Protect は、選択したリソースとの関係に基づいて、一部のリソースを自動的に選択します。たとえば、永続ボリュームクレームリソースを選択し、それに関連付けられたポッドがある場合、Trident Protect は関連付けられているポッドも復元します。

- **resourceFilter.resourceSelectionCriteria** : (フィルタリングに必須) `Include` または `Exclude` を使用して、resourceMatchersで定義されたリソースを含めるか除外します。含めるまたは除外するリソースを定義するには、以下のresourceMatchersパラメータを追加します：
 - **resourceFilter.resourceMatchers** : resourceMatcherオブジェクトの配列。この配列に複数の要素を定義すると、それらはOR演算として一致し、各要素内のフィールド (グループ、種類、バージョン) はAND演算として一致します。
 - **resourceMatchers[].group** : (オプション) フィルタリングするリソースのグループ。
 - **resourceMatchers[].kind** : (オプション) フィルタリングするリソースの種類。
 - **resourceMatchers[].version** : (オプション) フィルタリングするリソースのバージョン。
 - **resourceMatchers[].names** : (オプション) フィルタリングするリソースの Kubernetes metadata.name フィールド内の名前。

- **resourceMatchers[].namespaces** : (オプション) フィルタリングするリソースの Kubernetes metadata.name フィールド内の名前空間。
- **resourceMatchers[].labelSelectors** : (オプション) "Kubernetesドキュメント"で定義されているリソースのKubernetesメタデータ.nameフィールドのラベルセレクタ文字列。例：
"trident.netapp.io/os=linux"。

例：

```
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. `trident-protect-backup-ipr-cr.yaml` ファイルに正しい値を入力したら、CRを適用します：

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

バックアップを別のクラスタにリストアする

元のクラスタに問題がある場合は、バックアップを別のクラスタにリストアできます。



- Kopia をデータムーバーとして使用してバックアップを復元する場合、オプションで CR に注釈を指定して、Kopia が使用する一時ストレージの動作を制御できます。"Kopiaのドキュメント"設定できるオプションの詳細については、こちらを参照してください。
- CR を使用して新しいネームスペースにリストアする場合は、CR を適用する前に、デスティネーションネームスペースを手動で作成する必要があります。

開始する前に

次の前提条件が満たされていることを確認してください：

- デスティネーション クラスタに Trident Protect がインストールされている。
- デスティネーション クラスタは、バックアップが保存されているソース クラスタと同じAppVaultのバケ

ットパスにアクセスできます。

- AWS セッション トークンの有効期限が、長時間実行される復元操作に十分であることを確認します。復元操作中にトークンの有効期限が切れると、操作が失敗する可能性があります。
 - 現在のセッショントークンの有効期限を確認する方法の詳細については、"[AWS API ドキュメント](#)"を参照してください。
 - AWS リソースの認証情報の詳細については、"[AWSのドキュメント](#)"を参照してください。

手順

1. Trident Protect CLIプラグインを使用して、デスティネーション クラスタ上のAppVault CRの可用性を確認します：

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



アプリケーションのリストアに使用する名前空間がデスティネーション クラスタに存在することを確認します。

2. デスティネーション クラスタから利用可能なAppVaultのバックアップコンテンツを表示します：

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

このコマンドを実行すると、AppVault内の利用可能なバックアップが表示されます。これには、元のクラスタ、対応するアプリケーション名、タイムスタンプ、アーカイブパスが含まれます。

出力例：

```
+-----+-----+-----+-----+  
+-----+-----+  
| 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. AppVault名前とアーカイブパスを使用して、アプリケーションをデスティネーション クラスタに復元します：

4. カスタムリソース (CR) ファイルを作成し、名前を付けます `trident-protect-backup-restore-cr.yaml`。

5. 作成したファイルで、次の属性を設定します：

- **metadata.name**：(必須) このカスタム リソースの名前。環境に合わせて一意かつ適切な名前を選択してください。
- **spec.appVaultRef**：(必須) バックアップコンテンツが保存されるAppVaultの名前。
- **spec.appArchivePath**：AppVault内でバックアップコンテンツが保存されるパス。このパスを見つけるには、次のコマンドを使用できます：

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



BackupRestore CR が利用できない場合は、手順 2 に記載されているコマンドを使用してバックアップの内容を表示できます。

- **spec.namespaceMapping**：リストア処理のソース名前スペースからデスティネーション名前スペースへのマッピング。`my-source-namespace`と`my-destination-namespace`を環境の情報に置き換えます。

例：

```
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. `trident-protect-backup-restore-cr.yaml` ファイルに正しい値を入力したら、CRを適用します：

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

スナップショットを別の名前スペースにリストアする

カスタム リソース (CR) ファイルを使用して、スナップショットからデータを別の名前空間または元のソース名前空間に復元できます。SnapshotRestore CR を使用してスナップショットを別の名前空間に復元すると、Backup and Recovery はアプリケーションを新しい名前空間に復元し、復元されたアプリケーションのアプリケーション CR を作成します。復元されたアプリケーションを保護するには、オンデマンド バックアッ

またはスナップショットを作成するか、保護スケジュールを設定します。



- SnapshotRestoreは`spec.storageClassMapping`属性をサポートしていますが、ソースストレージクラスとデスティネーションストレージクラスが同じストレージバックエンドを使用する場合のみです。異なるストレージバックエンドを使用する`StorageClass`に復元しようとする、復元処理は失敗します。
- CRを使用して新しい名前空間にリストアする場合は、CRを適用する前に、デスティネーション名前空間を手動で作成する必要があります。

開始する前に

AWSセッショントークンの有効期限が、長時間実行されるs3復元処理に十分であることを確認します。復元処理中にトークンの有効期限が切れると、処理が失敗する可能性があります。

- 現在のセッショントークンの有効期限を確認する方法の詳細については、["AWS API ドキュメント"](#)を参照してください。
- ["AWS IAM ドキュメント"](#) AWS リソースの認証情報の詳細については、こちらを参照してください。

手順

1. カスタムリソース (CR) ファイルを作成し、名前を`trident-protect-snapshot-restore-cr.yaml`とします。
2. 作成したファイルで、次の属性を設定します：
 - **metadata.name** : (必須) このカスタムリソースの名前。環境に合わせて一意かつ適切な名前を選択してください。
 - **spec.appVaultRef** : (必須) スナップショットの内容が保存されるAppVaultの名前。
 - **spec.appArchivePath** : AppVault内のパスで、スナップショットの内容が保存される場所。このパスを見つけるには、次のコマンドを使用できます：

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

- **spec.namespaceMapping** : リストア処理のソース名前空間からデスティネーション名前空間へのマッピング。`my-source-namespace`と`my-destination-namespace`を環境の情報に置き換えます。

```
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. (オプション) 復元するアプリケーションの特定のリソースのみを選択する必要がある場合は、特定のラベルでマークされたリソースを含めるか除外するフィルタリングを追加します：



Trident Protect は、選択したリソースとの関係に基づいて、一部のリソースを自動的に選択します。たとえば、永続ボリュームクレームリソースを選択し、それに関連付けられたポッドがある場合、Trident Protect は関連付けられているポッドも復元します。

- **resourceFilter.resourceSelectionCriteria** : (フィルタリングに必須) `Include`または`Exclude`を使用して、resourceMatchersで定義されたリソースを含めるか除外します。含めるまたは除外するリソースを定義するには、以下のresourceMatchersパラメータを追加します：
 - **resourceFilter.resourceMatchers** : resourceMatcherオブジェクトの配列。この配列に複数の要素を定義すると、それらはOR演算として一致し、各要素内のフィールド（グループ、種類、バージョン）はAND演算として一致します。
 - **resourceMatchers[].group** : (オプション) フィルタリングするリソースのグループ。
 - **resourceMatchers[].kind** : (オプション) フィルタリングするリソースの種類。
 - **resourceMatchers[].version** : (オプション) フィルタリングするリソースのバージョン。
 - **resourceMatchers[].names** : (オプション) フィルタリングするリソースの Kubernetes metadata.name フィールド内の名前。
 - **resourceMatchers[].namespaces** : (オプション) フィルタリングするリソースの Kubernetes metadata.name フィールド内の名前空間。
 - **resourceMatchers[].labelSelectors** : (オプション) "[Kubernetesドキュメント](#)"で定義されているリソースのKubernetesメタデータ.nameフィールドのラベルセレクト文字列。例：
"trident.netapp.io/os=linux"。

例：

```
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. `trident-protect-snapshot-restore-cr.yaml` ファイルに正しい値を入力したら、CRを適用します：

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

スナップショットを元の名前空間に復元する

いつでもスナップショットを元の名前空間にリストアできます。

開始する前に

AWS セッショントークンの有効期限が、長時間実行される s3 復元処理に十分であることを確認します。復元処理中にトークンの有効期限が切れると、処理が失敗する可能性があります。

- 現在のセッショントークンの有効期限を確認する方法の詳細については、"[AWS API ドキュメント](#)"を参照してください。
- "[AWS IAM ドキュメント](#)"AWS リソースの認証情報の詳細については、こちらを参照してください。

手順

1. カスタムリソース (CR) ファイルを作成し、`trident-protect-snapshot-ipr-cr.yaml` という名前を付けます。
2. 作成したファイルで、次の属性を設定します：
 - **metadata.name** : (必須) このカスタム リソースの名前。環境に合わせて一意かつ適切な名前を選択してください。
 - **spec.appVaultRef** : (必須) スナップショットの内容が保存されるAppVaultの名前。
 - **spec.appArchivePath** : AppVault内のパスで、スナップショットの内容が保存される場所。このパスを見つけるには、次のコマンドを使用できます：

```
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. (オプション) 復元するアプリケーションの特定のリソースのみを選択する必要がある場合は、特定のラベルでマークされたリソースを含めるか除外するフィルタリングを追加します：



Trident Protect は、選択したリソースとの関係に基づいて、一部のリソースを自動的に選択します。たとえば、永続ボリュームクレームリソースを選択し、それに関連付けられたポッドがある場合、Trident Protect は関連付けられているポッドも復元します。

- **resourceFilter.resourceSelectionCriteria** : (フィルタリングに必須) `Include`または`Exclude`を使用して、resourceMatchersで定義されたリソースを含めるか除外します。含めるまたは除外するリソースを定義するには、以下のresourceMatchersパラメータを追加します：
 - **resourceFilter.resourceMatchers** : resourceMatcherオブジェクトの配列。この配列に複数の要素を定義すると、それらはOR演算として一致し、各要素内のフィールド（グループ、種類、バージョン）はAND演算として一致します。
 - **resourceMatchers[].group** : (オプション) フィルタリングするリソースのグループ。
 - **resourceMatchers[].kind** : (オプション) フィルタリングするリソースの種類。
 - **resourceMatchers[].version** : (オプション) フィルタリングするリソースのバージョン。
 - **resourceMatchers[].names** : (オプション) フィルタリングするリソースの Kubernetes metadata.name フィールド内の名前。
 - **resourceMatchers[].namespaces** : (オプション) フィルタリングするリソースの Kubernetes metadata.name フィールド内の名前空間。
 - **resourceMatchers[].labelSelectors** : (オプション) "[Kubernetesドキュメント](#)"で定義されているリソースのKubernetesメタデータ.nameフィールドのラベルセレクタ文字列。例：
"trident.netapp.io/os=linux"。

例：

```
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. `trident-protect-snapshot-ipr-cr.yaml` ファイルに正しい値を入力したら、CRを適用します：

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

高度なカスタムリソースのリストア設定を使用する

注釈、名前空間設定、ストレージ オプションなどの詳細設定を使用して、特定の要件を満たすように復元操作をカスタマイズできます。

リストアおよびフェイルオーバー処理中の名前スペースのアノテーションとラベル

復元およびフェイルオーバー処理中に、デスティネーション名前スペースのラベルとアノテーションは、ソース名前スペースのラベルとアノテーションと一致するように作成されます。デスティネーション名前スペースに存在しないソース名前スペースのラベルまたはアノテーションが追加され、既存のラベルまたはアノテーションはソース名前スペースの値と一致するように上書きされます。デスティネーション名前スペースにのみ存在するラベルまたはアノテーションは変更されません。



Red Hat OpenShiftを使用する場合は、OpenShift環境における名前空間アノテーションの重要な役割に注意することが重要です。名前空間アノテーションにより、リストアされたポッドがOpenShiftセキュリティコンテキスト制約（SCC）で定義された適切な権限とセキュリティ設定に準拠し、権限の問題なくボリュームにアクセスできるようになります。詳細については、"[OpenShiftセキュリティコンテキスト制約ドキュメント](#)"を参照してください。

リストアまたはフェイルオーバー処理を実行する前にKubernetes環境変数

`RESTORE_SKIP_NAMESPACE_ANNOTATIONS`を設定することで、デスティネーション名前スペース内の特定のアノテーションが上書きされるのを防ぐことができます。例：

```
helm upgrade trident-protect -n trident-protect netapp-trident-protect/trident-protect \
  --set-string
restoreSkipNamespaceAnnotations="{<annotation_key_to_skip_1>,<annotation_key_to_skip_2>}" \
  --reuse-values
```



復元またはフェイルオーバー操作を実行する場合、`restoreSkipNamespaceAnnotations`および`restoreSkipNamespaceLabels`で指定された名前スペースのアノテーションとラベルは、復元またはフェイルオーバー操作から除外されます。これらの設定がHelmの初期インストール時に構成されていることを確認してください。詳細については、"[追加のTrident Protectヘルムチャート設定を構成する](#)"を参照してください。

Helm を使用して `--create-namespace` フラグを指定してソースアプリケーションをインストールした場合、`name` ラベルキーには特別な処理が適用されます。リストアまたはフェイルオーバープロセス中に、Trident Protectはこのラベルをデスティネーション名前スペースにコピーしますが、ソースからの値がソース名前スペースと一致する場合は、値をデスティネーション名前スペースの値に更新します。この値がソース名前スペースと一致しない場合は、変更されずにデスティネーション名前スペースにコピーされます。

例

次の例は、それぞれ異なる注釈とラベルを持つソース名前空間と宛先名前空間を示しています。操作の前後の宛先名前空間の状態や、宛先名前空間で注釈とラベルがどのように結合または上書きされるかを確認できます。

リストアまたはフェイルオーバー処理の前に

次の表は、リストアまたはフェイルオーバー処理前のサンプルのソース名前スペースとデスティネーション名前スペースの状態を示しています：

名前スペース	アノテーション	ラベル
名前スペースns-1 (ソース)	<ul style="list-style-type: none">• annotation.one/key : "updatedvalue"• annotation.two/key : "true"	<ul style="list-style-type: none">• environment=production• コンプライアンス=hipaa• name=ns-1
名前スペースns-2 (宛先)	<ul style="list-style-type: none">• annotation.one/key : "true"• annotation.three/key : "false"	<ul style="list-style-type: none">• ロール=database

リストア処理後

次の表は、復元またはフェイルオーバー操作後の宛先名前空間の例の状態を示しています。いくつかのキーが追加され、いくつかは上書きされ、`name`ラベルは宛先名前空間と一致するように更新されました：

名前スペース	アノテーション	ラベル
名前スペースns-2 (宛先)	<ul style="list-style-type: none">• annotation.one/key : "updatedvalue"• annotation.two/key : "true"• annotation.three/key : "false"	<ul style="list-style-type: none">• name=ns-2• コンプライアンス=hipaa• environment=production• ロール=database

サポートされているフィールド

このセクションでは、復元操作に使用できる追加のフィールドについて説明します。

ストレージクラスのマッピング

``spec.storageClassMapping``属性は、ソース アプリケーションに存在するストレージクラスからターゲット クラスタ上の新しいストレージクラスへのマッピングを定義します。異なるストレージクラスを持つクラスタ間でアプリケーションを移行する場合や、BackupRestore操作のストレージバックエンドを変更する場合にこれを使用できます。

例：

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

サポートされているアノテーション

このセクションでは、システム内のさまざまな動作を構成するためにサポートされているアノテーションを一覧表示します。ユーザーがアノテーションを明示的に設定しない場合、システムはデフォルト値を使用します。

注釈	タイプ	説明	デフォルト値
protect.trident.netapp.io/data-mover-timeout-sec	string	データムーバー処理が停止する最大時間（秒単位）。	"300"
protect.trident.netapp.io/kopia-content-cache-size-limit-mb	string	Kopia コンテンツ キャッシュの最大サイズ制限（メガバイト単位）。	"1000"
protect.trident.netapp.io/pvc-bind-timeout-sec	string	新しく作成されたPersistentVolumeClaims（PVC）が`Bound`フェーズに到達するまで待機する最大時間（秒単位）。この時間を超えると処理は失敗します。環境：すべてのリストアCRタイプ（BackupRestore、BackupInplaceRestore、SnapshotRestore、SnapshotInplaceRestore）。ストレージバックエンドまたはクラスターで処理に時間がかかることが多い場合は、より大きい値を使用してください。	「1200」（20分）

著作権に関する情報

Copyright © 2026 NetApp, Inc. All Rights Reserved. Printed in the U.S.このドキュメントは著作権によって保護されています。著作権所有者の書面による事前承諾がある場合を除き、画像媒体、電子媒体、および写真複写、記録媒体、テープ媒体、電子検索システムへの組み込みを含む機械媒体など、いかなる形式および方法による複製も禁止します。

ネットアップの著作物から派生したソフトウェアは、次に示す使用許諾条項および免責条項の対象となります。

このソフトウェアは、ネットアップによって「現状のまま」提供されています。ネットアップは明示的な保証、または商品性および特定目的に対する適合性の暗示的保証を含み、かつこれに限定されないいかなる暗示的な保証も行いません。ネットアップは、代替品または代替サービスの調達、使用不能、データ損失、利益損失、業務中断を含み、かつこれに限定されない、このソフトウェアの使用により生じたすべての直接的損害、間接的損害、偶発的損害、特別損害、懲罰的損害、必然的損害の発生に対して、損失の発生の可能性が通知されていたとしても、その発生理由、根拠とする責任論、契約の有無、厳格責任、不法行為（過失またはそうでない場合を含む）にかかわらず、一切の責任を負いません。

ネットアップは、ここに記載されているすべての製品に対する変更を随時、予告なく行う権利を保有します。ネットアップによる明示的な書面による合意がある場合を除き、ここに記載されている製品の使用により生じる責任および義務に対して、ネットアップは責任を負いません。この製品の使用または購入は、ネットアップの特許権、商標権、または他の知的所有権に基づくライセンスの供与とはみなされません。

このマニュアルに記載されている製品は、1つ以上の米国特許、その他の国の特許、および出願中の特許によって保護されている場合があります。

権利の制限について：政府による使用、複製、開示は、DFARS 252.227-7013（2014年2月）およびFAR 5252.227-19（2007年12月）のRights in Technical Data -Noncommercial Items（技術データ - 非商用品目に関する諸権利）条項の(b)(3)項、に規定された制限が適用されます。

本書に含まれるデータは商用製品および / または商用サービス（FAR 2.101の定義に基づく）に関係し、データの所有権はNetApp, Inc.にあります。本契約に基づき提供されるすべてのネットアップの技術データおよびコンピュータソフトウェアは、商用目的であり、私費のみで開発されたものです。米国政府は本データに対し、非独占的かつ移転およびサブライセンス不可で、全世界を対象とする取り消し不能の制限付き使用权を有し、本データの提供の根拠となった米国政府契約に関連し、当該契約の裏付けとする場合にのみ本データを使用できます。前述の場合を除き、NetApp, Inc.の書面による許可を事前に得ることなく、本データを使用、開示、転載、改変するほか、上演または展示することはできません。国防総省にかかる米国政府のデータ使用权については、DFARS 252.227-7015(b)項（2014年2月）で定められた権利のみが認められます。

商標に関する情報

NetApp、NetAppのロゴ、<http://www.netapp.com/TM>に記載されているマークは、NetApp, Inc.の商標です。その他の会社名と製品名は、それを所有する各社の商標である場合があります。