응용 프로그램을 복원합니다

Trident

NetApp
September 26, 2025

목차

응용 프로그램을 복원합니다

Trident Protect를 사용하여 애플리케이션을 복원합니다

Trident Protect를 사용하여 스냅샷 또는 백업에서 애플리케이션을 복원할 수 있습니다. 애플리케이션을 동일한 클러스터로 복구할 경우 기존 스냅샷에서 복구하는 속도가 빨라집니다.

- Application Restore 시 해당 Application에 설정된 모든 execution hook이 App으로 복구된다. 복원 후 실행 후크가 있는 경우 복구 작업의 일부로 자동으로 실행됩니다.
- Qtree 볼륨의 경우 백업에서 다른 네임스페이스 또는 원래 네임스페이스로 복원하는 기능이 지원됩니다. 그러나 Qtree 볼륨의 경우 스냅샷에서 다른 네임스페이스 또는 원래 네임스페이스로 복원하는 기능은 지원되지 않습니다.
- 고급 설정을 사용하여 복원 작업을 사용자 정의할 수 있습니다. 자세한 내용은 다음을 참조하세요. "고급 Trident 보호 복원 설정 사용" .

백업에서 다른 네임스페이스로 복원합니다

BackupRestore CR을 사용하여 다른 네임스페이스로 백업을 복원하면 Trident Protect는 새 네임스페이스에서 응용 프로그램을 복원하고 복원된 응용 프로그램에 대한 응용 프로그램 CR을 만듭니다. 복원된 애플리케이션을 보호하려면 필요 시 백업 또는 스냅샷을 생성하거나 보호 일정을 설정합니다.

기존 리소스가 있는 다른 네임스페이스로 백업을 복원해도 백업의 리소스와 이름을 공유하는 리소스는 변경되지 않습니다. 백업의 모든 리소스를 복구하려면 타겟 네임스페이스를 삭제하고 다시 생성하거나 백업을 새 네임스페이스로 복원하십시오.

시작하기 전에

AWS 세션 토큰의 만료가 장시간 실행되는 S3 복원 작업에 충분한지 확인합니다. 복구 작업 중에 토큰이 만료되면 작업이 실패할 수 있습니다.

- 현재 세션 토큰 만료 확인에 대한 자세한 내용은 를 "AWS API 설명서" 참조하십시오.
- AWS 리소스의 자격 증명에 대한 자세한 내용은 를 "AWS IAM 설명서" 참조하십시오.

Kopia를 데이터 이동자로 사용하여 백업을 복원하는 경우 CR이나 CLI에서 주석을 지정하여 Kopia에서 사용하는 임시 저장소의 동작을 제어할 수 있습니다. 를 참조하세요 "코피아 문서" 구성할 수 있는 옵션에 대한 자세한 내용은 다음을 참조하세요. 사용하세요 `tridentctl-protect create --help` Trident Protect CLI로 주석을 지정하는 방법에 대한 자세한 내용은 명령을 참조하세요.

CR을 사용합니다

단계

1. CR(사용자 정의 리소스) 파일을 만들고 이름을 `trident-protect-backup-restore-cr.yaml`지정합니다.
2. 생성한 파일에서 다음 속성을 구성합니다.
   - **metadata.name**: (*required*) 이 사용자 정의 리소스의 이름입니다. 사용자 환경에 맞는 고유하고 합리적인 이름을 선택하십시오.
   - * spec.appArchivePath * : 백업 컨텐츠가 저장되는 AppVault 내부의 경로입니다. 다음 명령을 사용하여 이 경로를 찾을 수 있습니다.

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

   - * spec.appVaultRef *: (*required*) 백업 내용이 저장된 AppVault의 이름입니다.
   - **spec.namespaceMapping**: 복구 작업의 소스 네임스페이스를 대상 네임스페이스에 매핑하는 것입니다. `my-source-namespace`및 를 `my-destination-namespace` 사용자 환경의 정보로 바꿉니다.

     ```
     ---
     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. (선택 사항) 복원할 응용 프로그램의 특정 리소스만 선택해야 하는 경우 특정 레이블로 표시된 리소스를 포함하거나 제외하는 필터링을 추가합니다.

   > Trident Protect는 선택한 리소스와의 관계에 따라 일부 리소스를 자동으로 선택합니다. 예를 들어, 영구 볼륨 클레임 리소스를 선택했고 이 리소스에 연결된 Pod가 있는 경우 Trident Protect는 연결된 Pod도 복원합니다.

   - **resourceFilter.resourceSelectionCriteria**:(필터링에 필요) `Include` resourceMatchers에 정의된 리소스를 포함하거나 제외하려면 또는 을 `Exclude` 사용합니다. 다음 resourceMatchers 매개 변수를 추가하여 포함하거나 제외할 리소스를 정의합니다.
     - **resourceFilter.resourceMatchers**: resourceMatcher 개체의 배열입니다. 이 배열에서 여러 요소를 정의하는 경우 해당 요소는 OR 연산으로 일치하고 각 요소(그룹, 종류, 버전) 내의 필드는 AND 연산으로 일치합니다.

- **resourceMatchers[].group**:(*Optional*) 필터링할 리소스의 그룹입니다.
- * resourceMatchers [].kind *: (*Optional*) 필터링할 리소스의 종류입니다.
- **resourceMatchers [ ].version**: (*Optional*) 필터링할 리소스의 버전입니다.
- **resourceMatchers[].names**:(*Optional*) 필터링할 리소스의 Kubernetes metadata.name 필드에 있는 이름입니다.
- *resourceMatchers [].namespaces *: (*Optional*) 필터링할 리소스의 Kubernetes metadata.name 필드에 있는 네임스페이스입니다.
- **resourceMatchers [ ].labelSelectors**: (*Optional*) 에 정의된 대로 리소스의 Kubernetes metadata.name 필드에 있는 레이블 선택기 문자열입니다. "Kubernetes 문서" 예를 들면 다음과 `"trident.netapp.io/os=linux"`같습니다.

  예를 들면 다음과 같습니다.

```
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. 파일을 올바른 값으로 채운 후 `trident-protect-backup-restore-cr.yaml` CR:

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

CLI를 사용합니다

단계

1. 대괄호 안의 값을 환경의 정보로 대체하여 백업을 다른 네임스페이스로 복원합니다. 이 `namespace-mapping` 인수는 콜론으로 구분된 네임스페이스를 사용하여 소스 네임스페이스를 올바른 대상 네임스페이스에 형식 `source1:dest1,source2:dest2`으로 매핑합니다. 예를 들면 다음과 같습니다.

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

백업에서 원래 네임스페이스로 복구합니다

언제든지 원래 네임스페이스로 백업을 복원할 수 있습니다.

시작하기 전에

AWS 세션 토큰의 만료가 장시간 실행되는 S3 복원 작업에 충분한지 확인합니다. 복구 작업 중에 토큰이 만료되면 작업이 실패할 수 있습니다.

- 현재 세션 토큰 만료 확인에 대한 자세한 내용은 를 "AWS API 설명서" 참조하십시오.
- AWS 리소스의 자격 증명에 대한 자세한 내용은 를 "AWS IAM 설명서" 참조하십시오.

Kopia를 데이터 이동자로 사용하여 백업을 복원하는 경우 CR이나 CLI에서 주석을 지정하여 Kopia에서 사용하는 임시 저장소의 동작을 제어할 수 있습니다. 를 참조하세요 "코피아 문서" 구성할 수 있는 옵션에 대한 자세한 내용은 다음을 참조하세요. 사용하세요 `tridentctl-protect create --help` Trident Protect CLI로 주석을 지정하는 방법에 대한 자세한 내용은 명령을 참조하세요.

CR을 사용합니다

단계

1. CR(사용자 정의 리소스) 파일을 만들고 이름을 `trident-protect-backup-ipr-cr.yaml`지정합니다.
2. 생성한 파일에서 다음 속성을 구성합니다.
   - **metadata.name**: (*required*) 이 사용자 정의 리소스의 이름입니다. 사용자 환경에 맞는 고유하고 합리적인 이름을 선택하십시오.
   - * spec.appArchivePath * : 백업 컨텐츠가 저장되는 AppVault 내부의 경로입니다. 다음 명령을 사용하여 이 경로를 찾을 수 있습니다.

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

   - * spec.appVaultRef *: (*required*) 백업 내용이 저장된 AppVault의 이름입니다.

     예를 들면 다음과 같습니다.

```
---
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. (선택 사항) 복원할 응용 프로그램의 특정 리소스만 선택해야 하는 경우 특정 레이블로 표시된 리소스를 포함하거나 제외하는 필터링을 추가합니다.

   Trident Protect는 선택한 리소스와의 관계에 따라 일부 리소스를 자동으로 선택합니다. 예를 들어, 영구 볼륨 클레임 리소스를 선택했고 이 리소스에 연결된 Pod가 있는 경우 Trident Protect는 연결된 Pod도 복원합니다.

   - **resourceFilter.resourceSelectionCriteria**:(필터링에 필요) `Include` resourceMatchers에 정의된 리소스를 포함하거나 제외하려면 또는 을 `Exclude` 사용합니다. 다음 resourceMatchers 매개 변수를 추가하여 포함하거나 제외할 리소스를 정의합니다.
     - **resourceFilter.resourceMatchers**: resourceMatcher 개체의 배열입니다. 이 배열에서 여러 요소를 정의하는 경우 해당 요소는 OR 연산으로 일치하고 각 요소(그룹, 종류, 버전) 내의 필드는 AND 연산으로 일치합니다.
       - **resourceMatchers[].group**:(*Optional*) 필터링할 리소스의 그룹입니다.
       - * resourceMatchers [].kind *: (*Optional*) 필터링할 리소스의 종류입니다.
       - **resourceMatchers [ ].version**: (*Optional*) 필터링할 리소스의 버전입니다.

- **resourceMatchers[].names**:(*Optional*) 필터링할 리소스의 Kubernetes metadata.name 필드에 있는 이름입니다.
- *resourceMatchers [].namespaces *: (*Optional*) 필터링할 리소스의 Kubernetes metadata.name 필드에 있는 네임스페이스입니다.
- **resourceMatchers [ ].labelSelectors**: (*Optional*) 에 정의된 대로 리소스의 Kubernetes metadata.name 필드에 있는 레이블 선택기 문자열입니다. "Kubernetes 문서" 예를 들면 다음과 `"trident.netapp.io/os=linux"`같습니다.

  예를 들면 다음과 같습니다.

```
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. 파일을 올바른 값으로 채운 후 `trident-protect-backup-ipr-cr.yaml` CR:

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

CLI를 사용합니다

단계

1. 대괄호 안의 값을 환경의 정보로 대체하여 백업을 원래 네임스페이스로 복원합니다. `backup`인수에 네임스페이스 및 백업 이름이 형식으로 `<namespace>/<name>` 사용됩니다. 예를 들면 다음과 같습니다.

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

백업에서 다른 클러스터로 복원합니다

원래 클러스터에 문제가 있는 경우 백업을 다른 클러스터로 복구할 수 있습니다.

> Kopia를 데이터 이동자로 사용하여 백업을 복원하는 경우 CR이나 CLI에서 주석을 지정하여 Kopia에서 사용하는 임시 저장소의 동작을 제어할 수 있습니다. 를 참조하세요 "코피아 문서" 구성할 수 있는 옵션에 대한 자세한 내용은 다음을 참조하세요. 사용하세요 `tridentctl-protect create --help` Trident Protect CLI로 주석을 지정하는 방법에 대한 자세한 내용은 명령을 참조하세요.

시작하기 전에

다음 필수 구성 요소가 충족되는지 확인합니다.

- 대상 클러스터에 Trident Protect가 설치되어 있습니다.
- 대상 클러스터에는 백업이 저장되는 소스 클러스터와 동일한 AppVault의 버킷 경로에 대한 액세스 권한이 있습니다.
- AppVault CR에 정의된 개체 저장소 버킷에 로컬 환경이 연결할 수 있는지 확인하십시오. `tridentctl-protect get appvaultcontent` 명령. 네트워크 제한으로 인해 액세스할 수 없는 경우 대상 클러스터의 포드 내에서 Trident Protect CLI를 실행하세요.
- AWS 세션 토큰 만료가 장기 실행 중인 복구 작업에 충분한지 확인합니다. 복구 작업 중에 토큰이 만료되면 작업이 실패할 수 있습니다.
  - 현재 세션 토큰 만료 확인에 대한 자세한 내용은 를 "AWS API 설명서" 참조하십시오.
  - AWS 리소스의 자격 증명에 대한 자세한 내용은 를 "AWS 설명서" 참조하십시오.

단계

1. Trident Protect CLI 플러그인을 사용하여 대상 클러스터에서 AppVault CR의 가용성을 확인합니다.

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

> 응용 프로그램 복구용 네임스페이스가 대상 클러스터에 있는지 확인합니다.

2. 대상 클러스터에서 사용 가능한 AppVault의 백업 컨텐츠를 봅니다.

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

이 명령을 실행하면 원래 클러스터, 해당 응용 프로그램 이름, 타임스탬프 및 아카이브 경로를 비롯하여 AppVault에 사용 가능한 백업이 표시됩니다.

  - 출력 예: *

```
+-------------+-----------+--------+-----------------
+--------------------------+-------------+
|   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 이름 및 보관 경로를 사용하여 응용 프로그램을 대상 클러스터로 복원합니다.

CR을 사용합니다

1. CR(사용자 정의 리소스) 파일을 만들고 이름을 `trident-protect-backup-restore-cr.yaml`지정합니다.
2. 생성한 파일에서 다음 속성을 구성합니다.

   - **metadata.name**: (*required*) 이 사용자 정의 리소스의 이름입니다. 사용자 환경에 맞는 고유하고 합리적인 이름을 선택하십시오.
   - * spec.appVaultRef *: (*required*) 백업 내용이 저장된 AppVault의 이름입니다.
   - * spec.appArchivePath * : 백업 컨텐츠가 저장되는 AppVault 내부의 경로입니다. 다음 명령을 사용하여 이 경로를 찾을 수 있습니다.

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

     BackupRestore CR을 사용할 수 없는 경우 2단계에서 언급한 명령을 사용하여 백업 내용을 볼 수 있습니다.

   - **spec.namespaceMapping**: 복구 작업의 소스 네임스페이스를 대상 네임스페이스에 매핑하는 것입니다. `my-source-namespace`및 를 `my-destination-namespace` 사용자 환경의 정보로 바꿉니다.

     예를 들면 다음과 같습니다.

```
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. 파일을 올바른 값으로 채운 후 `trident-protect-backup-restore-cr.yaml` CR:

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

CLI를 사용합니다

1. 다음 명령을 사용하여 응용 프로그램을 복원하고 대괄호 안의 값을 사용자 환경의 정보로 바꿉니다. 네임스페이스 매핑 인수는 콜론으로 구분된 네임스페이스를 사용하여 소스 네임스페이스를 Source1:dest1, source2:dest2 형식으로 올바른 대상 네임스페이스에 매핑합니다. 예를 들면 다음과 같습니다.

```
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>
```

스냅샷에서 다른 네임스페이스로 복구합니다

사용자 지정 리소스(CR) 파일을 사용하여 스냅샷에서 데이터를 다른 네임스페이스 또는 원래 소스 네임스페이스로 복원할 수 있습니다. SnapshotRestore CR을 사용하여 스냅샷을 다른 네임스페이스로 복구할 경우 Trident Protect는 새 네임스페이스에서 애플리케이션을 복원하고 복원된 애플리케이션에 대한 애플리케이션 CR을 생성합니다. 복원된 애플리케이션을 보호하려면 필요 시 백업 또는 스냅샷을 생성하거나 보호 일정을 설정합니다.

SnapshotRestore는 다음을 지원합니다. `spec.storageClassMapping` 속성이지만 소스 및 대상 저장소 클래스가 동일한 저장소 백엔드를 사용하는 경우에만 해당됩니다. 복원을 시도하는 경우 `StorageClass` 다른 스토리지 백엔드를 사용하는 경우 복원 작업이 실패합니다.

시작하기 전에

AWS 세션 토큰의 만료가 장시간 실행되는 S3 복원 작업에 충분한지 확인합니다. 복구 작업 중에 토큰이 만료되면 작업이 실패할 수 있습니다.

- 현재 세션 토큰 만료 확인에 대한 자세한 내용은 를 "AWS API 설명서" 참조하십시오.
- AWS 리소스의 자격 증명에 대한 자세한 내용은 를 "AWS IAM 설명서" 참조하십시오.

CR을 사용합니다

단계

1. CR(사용자 정의 리소스) 파일을 만들고 이름을 `trident-protect-snapshot-restore-cr.yaml`지정합니다.
2. 생성한 파일에서 다음 속성을 구성합니다.
   - **metadata.name**: (*required*) 이 사용자 정의 리소스의 이름입니다. 사용자 환경에 맞는 고유하고 합리적인 이름을 선택하십시오.
   - * spec.appVaultRef *: (*required*) 스냅샷 내용이 저장된 AppVault의 이름입니다.
   - * spec.appArchivePath *: 스냅샷 내용이 저장되는 AppVault 내부 경로입니다. 다음 명령을 사용하여 이 경로를 찾을 수 있습니다.

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

   - **spec.namespaceMapping**: 복구 작업의 소스 네임스페이스를 대상 네임스페이스에 매핑하는 것입니다. `my-source-namespace`및 를 `my-destination-namespace` 사용자 환경의 정보로 바꿉니다.

     ```
     ---
     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. (선택 사항) 복원할 응용 프로그램의 특정 리소스만 선택해야 하는 경우 특정 레이블로 표시된 리소스를 포함하거나 제외하는 필터링을 추가합니다.

   > Trident Protect는 선택한 리소스와의 관계에 따라 일부 리소스를 자동으로 선택합니다. 예를 들어, 영구 볼륨 클레임 리소스를 선택했고 이 리소스에 연결된 Pod가 있는 경우 Trident Protect는 연결된 Pod도 복원합니다.

   - **resourceFilter.resourceSelectionCriteria**:(필터링에 필요) `Include` resourceMatchers에 정의된 리소스를 포함하거나 제외하려면 또는 을 `Exclude` 사용합니다. 다음 resourceMatchers 매개 변수를 추가하여 포함하거나 제외할 리소스를 정의합니다.
     - **resourceFilter.resourceMatchers**: resourceMatcher 개체의 배열입니다. 이 배열에서 여러 요소를 정의하는 경우 해당 요소는 OR 연산으로 일치하고 각 요소(그룹, 종류, 버전) 내의 필드는 AND 연산으로 일치합니다.

- **resourceMatchers[].group**:(*Optional*) 필터링할 리소스의 그룹입니다.
- * resourceMatchers [].kind *: (*Optional*) 필터링할 리소스의 종류입니다.
- **resourceMatchers [ ].version**: (*Optional*) 필터링할 리소스의 버전입니다.
- **resourceMatchers[].names**:(*Optional*) 필터링할 리소스의 Kubernetes metadata.name 필드에 있는 이름입니다.
- *resourceMatchers [].namespaces *: (*Optional*) 필터링할 리소스의 Kubernetes metadata.name 필드에 있는 네임스페이스입니다.
- **resourceMatchers [ ].labelSelectors**: (*Optional*) 에 정의된 대로 리소스의 Kubernetes metadata.name 필드에 있는 레이블 선택기 문자열입니다. "Kubernetes 문서" 예를 들면 다음과 `"trident.netapp.io/os=linux"`같습니다.

  예를 들면 다음과 같습니다.

```
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. 파일을 올바른 값으로 채운 후 `trident-protect-snapshot-restore-cr.yaml` CR:

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

CLI를 사용합니다

단계

1. 대괄호 안의 값을 사용자 환경의 정보로 대체하여 스냅샷을 다른 네임스페이스로 복원합니다.
   - `snapshot`인수에 네임스페이스 및 스냅숏 이름이 형식으로 `<namespace>/<name>` 사용됩니다.
   - 이 `namespace-mapping` 인수는 콜론으로 구분된 네임스페이스를 사용하여 소스 네임스페이스를 올바른 대상 네임스페이스에 형식 `source1:dest1,source2:dest2`으로 매핑합니다.

예를 들면 다음과 같습니다.

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

스냅샷에서 원래 네임스페이스로 복구합니다

언제든지 스냅샷을 원래 네임스페이스로 복구할 수 있습니다.

시작하기 전에

AWS 세션 토큰의 만료가 장시간 실행되는 S3 복원 작업에 충분한지 확인합니다. 복구 작업 중에 토큰이 만료되면 작업이 실패할 수 있습니다.

- 현재 세션 토큰 만료 확인에 대한 자세한 내용은 를 "AWS API 설명서" 참조하십시오.
- AWS 리소스의 자격 증명에 대한 자세한 내용은 를 "AWS IAM 설명서" 참조하십시오.

CR을 사용합니다

단계

1. CR(사용자 정의 리소스) 파일을 만들고 이름을 `trident-protect-snapshot-ipr-cr.yaml`지정합니다.
2. 생성한 파일에서 다음 속성을 구성합니다.
   - **metadata.name**: (*required*) 이 사용자 정의 리소스의 이름입니다. 사용자 환경에 맞는 고유하고 합리적인 이름을 선택하십시오.
   - * spec.appVaultRef *: (*required*) 스냅샷 내용이 저장된 AppVault의 이름입니다.
   - * spec.appArchivePath *: 스냅샷 내용이 저장되는 AppVault 내부 경로입니다. 다음 명령을 사용하여 이 경로를 찾을 수 있습니다.

```
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. (선택 사항) 복원할 응용 프로그램의 특정 리소스만 선택해야 하는 경우 특정 레이블로 표시된 리소스를 포함하거나 제외하는 필터링을 추가합니다.

> Trident Protect는 선택한 리소스와의 관계에 따라 일부 리소스를 자동으로 선택합니다. 예를 들어, 영구 볼륨 클레임 리소스를 선택했고 이 리소스에 연결된 Pod가 있는 경우 Trident Protect는 연결된 Pod도 복원합니다.

   - **resourceFilter.resourceSelectionCriteria**:(필터링에 필요) `Include` resourceMatchers에 정의된 리소스를 포함하거나 제외하려면 또는 을 `Exclude` 사용합니다. 다음 resourceMatchers 매개 변수를 추가하여 포함하거나 제외할 리소스를 정의합니다.
     - **resourceFilter.resourceMatchers**: resourceMatcher 개체의 배열입니다. 이 배열에서 여러 요소를 정의하는 경우 해당 요소는 OR 연산으로 일치하고 각 요소(그룹, 종류, 버전) 내의 필드는 AND 연산으로 일치합니다.
       - **resourceMatchers[].group**:(*Optional*) 필터링할 리소스의 그룹입니다.
       - * resourceMatchers [].kind *: (*Optional*) 필터링할 리소스의 종류입니다.
       - **resourceMatchers [ ].version**: (*Optional*) 필터링할 리소스의 버전입니다.
       - **resourceMatchers[].names**:(*Optional*) 필터링할 리소스의 Kubernetes metadata.name 필드에 있는 이름입니다.

- *resourceMatchers [].namespaces *: (*Optional*) 필터링할 리소스의 Kubernetes metadata.name 필드에 있는 네임스페이스입니다.
- **resourceMatchers [ ].labelSelectors**: (*Optional*) 에 정의된 대로 리소스의 Kubernetes metadata.name 필드에 있는 레이블 선택기 문자열입니다. "Kubernetes 문서" 예를 들면 다음과 `"trident.netapp.io/os=linux"`같습니다.

  예를 들면 다음과 같습니다.

```
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. 파일을 올바른 값으로 채운 후 `trident-protect-snapshot-ipr-cr.yaml` CR:

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

CLI를 사용합니다

단계

1. 대괄호 안의 값을 사용자 환경의 정보로 대체하여 스냅샷을 원래 네임스페이스로 복원합니다. 예를 들면 다음과 같습니다.

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

복구 작업의 상태를 확인합니다

명령줄을 사용하여 진행 중이거나, 완료되었거나, 실패한 복구 작업의 상태를 확인할 수 있습니다.

단계

1. 다음 명령을 사용하여 복원 작업의 상태를 검색하여 대괄호의 값을 사용자 환경의 정보로 바꿉니다.

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

고급 Trident 보호 복원 설정 사용

주석, 네임스페이스 설정, 스토리지 옵션 등의 고급 설정을 사용하여 복원 작업을 사용자 정의하여 특정 요구 사항을 충족할 수 있습니다.

복원 및 페일오버 작업 중 네임스페이스 주석 및 레이블

복원 및 페일오버 작업 중에 대상 네임스페이스의 레이블과 주석이 소스 네임스페이스의 레이블 및 주석과 일치하도록 만듭니다. 대상 네임스페이스에 없는 소스 네임스페이스의 레이블 또는 주석이 추가되고 이미 존재하는 모든 레이블 또는 주석이 소스 네임스페이스의 값과 일치하도록 덮어쓰여집니다. 대상 네임스페이스에만 있는 레이블이나 주석은 변경되지 않습니다.

Red Hat OpenShift를 사용하는 경우 OpenShift 환경에서 네임스페이스 주석의 중요한 역할을 알아두는 것이 중요합니다. 네임스페이스 주석은 복원된 포드가 OpenShift 보안 컨텍스트 제약(SCC)에 의해 정의된 적절한 권한과 보안 구성을 준수하고 권한 문제 없이 볼륨에 액세스할 수 있도록 보장합니다. 자세한 내용은 다음을 참조하세요. "OpenShift 보안 컨텍스트 제약 조건 문서" .

복원 또는 페일오버 작업을 수행하기 전에 Kubernetes 환경 변수를 설정하여 타겟 네임스페이스의 특정 주석을 덮어쓰지 않도록 할 수 있습니다 `RESTORE_SKIP_NAMESPACE_ANNOTATIONS`. 예를 들면 다음과 같습니다.

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

복원 또는 장애 조치 작업을 수행할 때 지정된 네임스페이스 주석 및 레이블 `restoreSkipNamespaceAnnotations` 그리고 `restoreSkipNamespaceLabels` 복구 또는 장애 조치 작업에서 제외됩니다. 이러한 설정은 Helm을 처음 설치할 때 구성되어야 합니다. 자세한 내용은 다음을 참조하세요."AutoSupport 및 네임스페이스 필터링 옵션 구성" .

플래그가 있는 Helm을 사용하여 소스 애플리케이션을 설치한 경우 `--create-namespace` 레이블 키에 특수 치료가 `name` 제공됩니다. 복구 또는 페일오버 프로세스 중에 Trident Protect는 이 레이블을 대상 네임스페이스에 복제하지만 소스의 값이 소스 네임스페이스와 일치하면 대상 네임스페이스 값으로 업데이트합니다. 이 값이 소스 네임스페이스와 일치하지 않으면 변경 없이 대상 네임스페이스로 복사됩니다.

예

다음 예제에서는 각각 다른 주석과 레이블이 있는 소스 및 대상 네임스페이스를 보여 줍니다. 작업 전후에 대상 네임스페이스의 상태와 대상 네임스페이스에서 주석과 레이블이 결합되거나 덮어써지는 방법을 확인할 수 있습니다.

복구 또는 페일오버 작업 전

다음 표에서는 복구 또는 페일오버 작업 이전의 예제 소스 및 대상 네임스페이스의 상태를 보여 줍니다.

네임스페이스	주석	라벨
네임스페이스 ns-1(소스)	• annotation.one/key:"updatedvalue"(주석 1개/키) • Annotation.two/key(주석.two/키):	• 환경 = 운영 • 규정 준수 = HIPAA • 이름 = ns-1
네임스페이스 ns-2(대상)	• Annotation.one/key(주석. 하나/키):"true" • Annotation.Three/key:"false"	• 역할 = 데이터베이스

복구 작업 후

다음 표에서는 복구 또는 페일오버 작업 후의 예제 대상 네임스페이스의 상태를 보여 줍니다. 일부 키가 추가되고, 일부 키가 덮어쓰여졌으며, `name` 대상 네임스페이스와 일치하도록 레이블이 업데이트되었습니다.

네임스페이스	주석	라벨
네임스페이스 ns-2(대상)	• annotation.one/key:"updatedvalue"(주석 1개/키) • Annotation.two/key(주석.two/키): • Annotation.Three/key:"false"	• 이름 = ns-2 • 규정 준수 = HIPAA • 환경 = 운영 • 역할 = 데이터베이스

지원되는 필드

이 섹션에서는 복원 작업에 사용할 수 있는 추가 필드에 대해 설명합니다.

스토리지 클래스 매핑

그만큼 `spec.storageClassMapping` 속성은 소스 애플리케이션에 있는 스토리지 클래스에서 대상 클러스터의 새 스토리지 클래스로의 매핑을 정의합니다. 서로 다른 스토리지 클래스를 사용하는 클러스터 간에 애플리케이션을 마이그레이션하거나 BackupRestore 작업에 대한 스토리지 백엔드를 변경할 때 이 기능을 사용할 수 있습니다.

- 예: *

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

지원되는 주석

이 섹션에서는 시스템의 다양한 동작을 구성하는 데 지원되는 애노테이션을 나열합니다. 사용자가 애노테이션을 명시적으로 설정하지 않으면 시스템은 기본값을 사용합니다.

주석	유형	설명	기본값
보호.트라이던트.넷앱.io/데이터-무버-타임아웃-초	문자열	데이터 이동 작업이 중단되는 데 허용되는 최대 시간(초)입니다.	"300"
보호.트라이던트.넷앱.io/코피아-콘텐츠-캐시-크기-제한-mb	문자열	Kopia 콘텐츠 캐시의 최대 크기 제한(메가바이트)입니다.	"1000"