애플리케이션 복원

Trident

NetApp
July 01, 2026

목차

애플리케이션 복원

Trident Protect를 사용하여 애플리케이션을 복원하세요

Trident Protect를 사용하면 스냅샷 또는 백업에서 애플리케이션을 복원할 수 있습니다. 기존 스냅샷에서 복원하면 동일한 클러스터에 애플리케이션을 복원할 때 더 빠릅니다.

- 애플리케이션을 복원하면 해당 애플리케이션에 대해 구성된 모든 실행 후크가 애플리케이션과 함께 복원됩니다. 복원 후 실행 후크가 있는 경우 복원 작업의 일부로 자동으로 실행됩니다.
- qtree 볼륨의 경우 백업에서 다른 네임스페이스 또는 원래 네임스페이스로 복원하는 것이 지원됩니다. 그러나 스냅샷에서 다른 네임스페이스 또는 원래 네임스페이스로 복원하는 것은 qtree 볼륨에서 지원되지 않습니다.
- 고급 설정을 사용하여 복원 작업을 사용자 지정할 수 있습니다. 자세한 내용은 "고급 Trident Protect 복원 설정을 사용하십시오"를 참조하십시오.

백업에서 다른 네임스페이스로 복원

BackupRestore CR을 사용하여 다른 네임스페이스로 백업을 복원하면 Trident Protect는 새 네임스페이스에 애플리케이션을 복원하고 복원된 애플리케이션에 대한 애플리케이션 CR을 생성합니다. 복원된 애플리케이션을 보호하려면 온디맨드 백업 또는 스냅샷을 생성하거나 보호 일정을 설정하십시오.

- 기존 리소스가 있는 다른 네임스페이스로 백업을 복원해도 백업에 포함된 리소스와 이름이 같은 리소스는 변경되지 않습니다. 백업의 모든 리소스를 복원하려면 대상 네임스페이스를 삭제하고 다시 생성하거나 새 네임스페이스로 백업을 복원해야 합니다.
- CR을 사용하여 새 네임스페이스로 복원할 경우, CR을 적용하기 전에 타겟 네임스페이스를 수동으로 생성해야 합니다. Trident Protect는 CLI를 사용할 때만 네임스페이스를 자동으로 생성합니다.

시작하기 전에

장시간 소요되는 s3 복원 작업의 경우 AWS 세션 토큰 만료 기간이 충분한지 확인하십시오. 복원 작업 중에 토큰이 만료되면 작업이 실패할 수 있습니다.

- 현재 세션 토큰 만료 확인에 대한 자세한 내용은 "AWS API 문서"을 참조하십시오.
- AWS 리소스에 대한 자격 증명 관련 자세한 내용은 "AWS IAM 문서"를 참조하십시오.

Kopia를 데이터 이동 도구로 사용하여 백업을 복원할 때 CR 또는 CLI를 사용하여 Kopia에서 사용하는 임시 스토리지의 동작을 제어하는 주석을 선택적으로 지정할 수 있습니다. 구성할 수 있는 옵션에 대한 자세한 내용은 "Kopia 문서"을 참조하십시오. Trident Protect CLI를 사용하여 주석을 지정하는 방법에 대한 자세한 내용은 `tridentctl-protect create --help` 명령을 사용하십시오.

CR 사용

단계

1. 사용자 정의 리소스(CR) 파일을 생성하고 이름을 지정합니다 `trident-protect-backup-restore-cr.yaml`.
2. 생성한 파일에서 다음 속성을 구성하십시오.
   - **metadata.name**: (필수) 이 사용자 지정 리소스의 이름입니다. 환경에 맞는 고유하고 적절한 이름을 선택하세요.
   - **spec.appArchivePath**: 백업 콘텐츠가 저장되는 AppVault 내부 경로입니다. 다음 명령을 사용하여 이 경로를 찾을 수 있습니다.

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

   - **spec.appVaultRef**: (필수) 백업 콘텐츠가 저장되는 AppVault의 이름입니다.
   - **spec.destinationApplicationName**: (선택 사항) 복원된 애플리케이션의 이름입니다. 제공된 경우 복원된 애플리케이션은 이 이름을 사용합니다. 제공되지 않은 경우 복원된 애플리케이션은 원본 애플리케이션 이름을 사용합니다.
   - **spec.namespaceMapping**: 복원 작업의 소스 네임스페이스를 타겟 네임스페이스로 매핑합니다. `my-source-namespace` 및 `my-destination-namespace`를 사용자 환경의 정보로 교체하십시오.

     ```
     ---
     apiVersion: protect.trident.netapp.io/v1
     kind: BackupRestore
     metadata:
       name: my-cr-name
       namespace: my-destination-namespace
     spec:
       appArchivePath: my-backup-path
       appVaultRef: appvault-name
       destinationApplicationName: my-new-app-name
       namespaceMapping: [{"source": "my-source-namespace",
     "destination": "my-destination-namespace"}]
     ```

3. (선택 사항) 애플리케이션의 특정 리소스만 복원해야 하는 경우, 특정 레이블로 표시된 리소스를 포함하거나 제외하는 필터링을 추가하십시오.

   > Trident Protect는 사용자가 선택한 리소스와의 연관성 때문에 일부 리소스를 자동으로 선택합니다. 예를 들어, 영구 볼륨 클레임 리소스를 선택하고 해당 리소스에 연결된 Pod가 있는 경우 Trident Protect는 연결된 Pod도 복원합니다.

   - **resourceFilter.resourceSelectionCriteria**: (필터링에 필수) `Include` 또는 `Exclude`를 사용하여 resourceMatchers에 정의된 리소스를 포함하거나 제외합니다. 포함 또는 제외할 리소스를 정의하려면 다음 resourceMatchers 매개변수를 추가하십시오.

- **resourceFilter.resourceMatchers**: resourceMatcher 객체의 배열입니다. 이 배열에 여러 요소를 정의하는 경우, 요소들은 OR 연산으로 일치하며, 각 요소 내부의 필드(그룹, 종류, 버전)는 AND 연산으로 일치합니다.
  - **resourceMatchers[].group**: (선택 사항) 필터링할 리소스의 그룹입니다.
  - **resourceMatchers[].kind**: (선택 사항) 필터링할 리소스의 종류입니다.
  - **resourceMatchers[].version**: (선택 사항) 필터링할 리소스의 버전입니다.
  - **resourceMatchers[].names**: (선택 사항) 필터링할 리소스의 Kubernetes metadata.name 필드에 있는 이름입니다.
  - **resourceMatchers[].namespaces**: (선택 사항) 필터링할 리소스의 Kubernetes metadata.name 필드에 있는 네임스페이스입니다.
  - **resourceMatchers[].labelSelectors**: (선택 사항) "Kubernetes 문서"에 정의된 리소스의 Kubernetes metadata.name 필드에 있는 레이블 선택기 문자열입니다. 예를 들면 다음과 같습니다: `"trident.netapp.io/os=linux"`.

    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. `trident-protect-backup-restore-cr.yaml` 파일에 올바른 값을 입력한 후 CR을 적용하십시오.

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

CLI 사용

단계

1. 백업을 다른 네임스페이스로 복원하고, 괄호 안의 값을 사용자 환경 정보로 바꿉니다. `namespace-mapping` 인수는 콜론으로 구분된 네임스페이스를 사용하여 소스 네임스페이스를 `source1:dest1,source2:dest2` 형식으로 올바른 타겟 네임스페이스에 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> \
--destination-app-name<custom_app_name>\
-n <application_namespace>
```

원래 네임스페이스로 백업에서 복원합니다

언제든지 백업을 원래 네임스페이스로 복원할 수 있습니다. 제자리 복원을 수행하면 Trident Protect는 잘못된 복구 지점이 생성되는 것을 방지하기 위해 보호 일정과 진행 중인 작업을 자동으로 관리합니다.

- 복원 작업이 시작되기 전에 애플리케이션에 대해 활성화된 모든 보호 일정이 비활성화됩니다. 이렇게 하면 애플리케이션 리소스가 복원되는 동안 예약된 백업이나 스냅샷이 실행되는 것을 방지할 수 있습니다.
- 복원 작업이 성공적으로 완료되면 복원 전에 활성화되어 있던 일정만 다시 활성화됩니다. 이미 비활성화된 일정은 계속 비활성화된 상태로 유지됩니다.
- 복원이 시작되기 전에 진행 중인 모든 백업 또는 스냅샷 작업이 취소됩니다. 5분 이내에 작업이 취소되지 않으면 복원이 진행되고 복원 CR 상태에 경고가 기록됩니다.

시작하기 전에

장시간 소요되는 s3 복원 작업의 경우 AWS 세션 토큰 만료 기간이 충분한지 확인하십시오. 복원 작업 중에 토큰이 만료되면 작업이 실패할 수 있습니다.

- 현재 세션 토큰 만료 확인에 대한 자세한 내용은 "AWS API 문서"을 참조하십시오.
- AWS 리소스에 대한 자격 증명 관련 자세한 내용은 "AWS IAM 문서"를 참조하십시오.

Kopia를 데이터 이동 도구로 사용하여 백업을 복원할 때 CR 또는 CLI를 사용하여 Kopia에서 사용하는 임시 스토리지의 동작을 제어하는 주석을 선택적으로 지정할 수 있습니다. 구성할 수 있는 옵션에 대한 자세한 내용은 "Kopia 문서"을 참조하십시오. Trident Protect CLI를 사용하여 주석을 지정하는 방법에 대한 자세한 내용은 `tridentctl-protect create --help` 명령을 사용하십시오.

CR 사용

단계

1. 사용자 정의 리소스(CR) 파일을 생성하고 이름을 지정합니다 `trident-protect-backup-ipr-cr.yaml`.
2. 생성한 파일에서 다음 속성을 구성하십시오.
   - **metadata.name**: (필수) 이 사용자 지정 리소스의 이름입니다. 환경에 맞는 고유하고 적절한 이름을 선택하세요.
   - **spec.appArchivePath**: 백업 콘텐츠가 저장되는 AppVault 내부 경로입니다. 다음 명령을 사용하여 이 경로를 찾을 수 있습니다.

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

   - **spec.appVaultRef**: (필수) 백업 콘텐츠가 저장되는 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) 애플리케이션의 특정 리소스만 복원해야 하는 경우, 특정 레이블로 표시된 리소스를 포함하거나 제외하는 필터링을 추가하십시오.

   Trident Protect는 사용자가 선택한 리소스와의 연관성 때문에 일부 리소스를 자동으로 선택합니다. 예를 들어, 영구 볼륨 클레임 리소스를 선택하고 해당 리소스에 연결된 Pod가 있는 경우 Trident Protect는 연결된 Pod도 복원합니다.

   - **resourceFilter.resourceSelectionCriteria**: (필터링에 필수) `Include` 또는 `Exclude`를 사용하여 resourceMatchers에 정의된 리소스를 포함하거나 제외합니다. 포함 또는 제외할 리소스를 정의하려면 다음 resourceMatchers 매개변수를 추가하십시오.
     - **resourceFilter.resourceMatchers**: resourceMatcher 객체의 배열입니다. 이 배열에 여러 요소를 정의하는 경우, 요소들은 OR 연산으로 일치하며, 각 요소 내부의 필드(그룹, 종류, 버전)는 AND 연산으로 일치합니다.
       - **resourceMatchers[].group**: (선택 사항) 필터링할 리소스의 그룹입니다.
       - **resourceMatchers[].kind**: (선택 사항) 필터링할 리소스의 종류입니다.

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

  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. `trident-protect-backup-ipr-cr.yaml` 파일에 올바른 값을 입력한 후 CR을 적용하십시오.

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

CLI 사용

단계

1. 원래 네임스페이스에 백업을 복원하고, 괄호 안의 값을 사용자 환경 정보로 바꿉니다. `backup` 인수는 `<namespace>/<name>` 형식으로 네임스페이스와 백업 이름을 사용합니다. 예를 들면 다음과 같습니다.

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

다른 클러스터로 백업에서 복원

원래 클러스터에 문제가 발생한 경우 백업을 다른 클러스터로 복원할 수 있습니다.

- Kopia를 데이터 이동 도구로 사용하여 백업을 복원할 때 CR 또는 CLI를 사용하여 Kopia에서 사용하는 임시 스토리지의 동작을 제어하는 주석을 선택적으로 지정할 수 있습니다. 구성할 수 있는 옵션에 대한 자세한 내용은 "Kopia 문서"을 참조하십시오. Trident Protect CLI를 사용하여 주석을 지정하는 방법에 대한 자세한 내용은 `tridentctl-protect create --help` 명령을 사용하십시오.
- CR을 사용하여 새 네임스페이스로 복원할 경우, CR을 적용하기 전에 타겟 네임스페이스를 수동으로 생성해야 합니다. Trident Protect는 CLI를 사용할 때만 네임스페이스를 자동으로 생성합니다.

시작하기 전에

다음 사전 요구 사항이 충족되는지 확인하십시오.

- 타겟 클러스터에 Trident Protect가 설치되어 있습니다.
- 타겟 클러스터는 백업이 저장된 소스 클러스터와 동일한 AppVault의 버킷 경로에 액세스할 수 있습니다.
- `tridentctl-protect get appvaultcontent` 명령을 실행할 때 로컬 환경에서 AppVault CR에 정의된 오브젝트 스토리지 버킷에 연결할 수 있는지 확인하십시오. 네트워크 제한으로 인해 액세스할 수 없는 경우 타겟 클러스터의 Pod 내에서 Trident Protect CLI를 실행하십시오.
- 장시간 소요되는 복원 작업의 경우 AWS 세션 토큰 만료 기간이 충분한지 확인하십시오. 복원 작업 중에 토큰이 만료되면 작업이 실패할 수 있습니다.
  - 현재 세션 토큰 만료 확인에 대한 자세한 내용은 "AWS API 문서"을 참조하십시오.
  - AWS 리소스에 대한 자격 증명 관련 자세한 내용은 "AWS 문서"를 참조하십시오.

단계

1. Trident Protect CLI 플러그인을 사용하여 타겟 클러스터에 AppVault CR이 있는지 확인합니다.

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

타겟 클러스터에 AppVault CR이 없는 경우 "Trident Protect AppVault 객체를 사용하여 버킷을 관리하세요."의 단계에 따라 생성하십시오.

2. 타겟 클러스터에서 사용 가능한 AppVault의 백업 콘텐츠를 확인하고 복원할 백업의 `appArchivePath`을 기록해 두십시오.

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

이 명령을 실행하면 AppVault에서 사용 가능한 백업이 표시되며, 여기에는 백업이 생성된 클러스터, 해당 애플리케이션 이름, 타임스탬프 및 아카이브 경로가 포함됩니다.

예시 출력:

```
+-------------+-----------+--------+-----------------
+--------------------------+-------------+
|   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 이름과 아카이브 경로를 사용하여 타겟 클러스터에 애플리케이션을 복원합니다.

CR을 사용할 때는 애플리케이션 복원에 사용할 네임스페이스가 타겟 클러스터에 있는지 확인하십시오.

CR 사용

1. 사용자 정의 리소스(CR) 파일을 생성하고 이름을 지정합니다 `trident-protect-backup-restore-cr.yaml`.
2. 생성한 파일에서 다음 속성을 구성하십시오.
   - **metadata.name**: (필수) 이 사용자 지정 리소스의 이름입니다. 환경에 맞는 고유하고 적절한 이름을 선택하세요.
   - **spec.appVaultRef**: (필수) 백업 콘텐츠가 저장되는 AppVault의 이름입니다.
   - **spec.appArchivePath**: (필수) AppVault 내에서 백업 콘텐츠가 저장되는 경로입니다. 2단계의 명령을 사용하여 백업 콘텐츠를 확인하고 `appArchivePath`복원하려는 백업을 찾으십시오.
   - **spec.destinationApplicationName**: (선택 사항) 복원된 애플리케이션의 이름입니다. 제공된 경우 복원된 애플리케이션은 이 이름을 사용합니다. 제공되지 않은 경우 복원된 애플리케이션은 원본 애플리케이션 이름을 사용합니다.
   - **spec.namespaceMapping**: 복원 작업의 소스 네임스페이스를 타겟 네임스페이스로 매핑합니다. `my-source-namespace` 및 `my-destination-namespace`를 사용자 환경의 정보로 교체하십시오.

     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
  destinationApplicationName: my-new-app-name
  namespaceMapping: [{"source": "my-source-namespace", "
destination": "my-destination-namespace"}]
```

3. `trident-protect-backup-restore-cr.yaml` 파일에 올바른 값을 입력한 후 CR을 적용하십시오.

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

CLI 사용

1. 다음 명령을 사용하여 애플리케이션을 복원하고 대괄호 안의 값을 사용자 환경의 정보로 바꿉니다. namespace-mapping 인수는 콜론으로 구분된 네임스페이스를 사용하여 소스 네임스페이스를 source1:dest1,source2:dest2 형식으로 올바른 타겟 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> \
--destination-app-name <custom_app_name> \
--context <destination_cluster_name> \
-n <application_namespace>
```

스냅샷에서 다른 네임스페이스로 복원

스냅샷에서 사용자 지정 리소스(CR) 파일을 사용하여 다른 네임스페이스 또는 원래 소스 네임스페이스로 데이터를 복원할 수 있습니다. SnapshotRestore CR을 사용하여 스냅샷을 다른 네임스페이스로 복원하면 Trident Protect는 새 네임스페이스에 애플리케이션을 복원하고 복원된 애플리케이션에 대한 애플리케이션 CR을 생성합니다. 복원된 애플리케이션을 보호하려면 온디맨드 백업 또는 스냅샷을 생성하거나 보호 일정을 설정하십시오.

- SnapshotRestore는 `spec.storageClassMapping` 속성을 지원하지만, 소스 및 타겟 스토리지 클래스가 동일한 스토리지 백엔드를 사용하는 경우에만 가능합니다. 다른 스토리지 백엔드를 사용하는 `StorageClass`로 복원을 시도하면 복원 작업이 실패합니다.
- CR을 사용하여 새 네임스페이스로 복원할 경우, CR을 적용하기 전에 타겟 네임스페이스를 수동으로 생성해야 합니다. Trident Protect는 CLI를 사용할 때만 네임스페이스를 자동으로 생성합니다.

시작하기 전에

장시간 소요되는 s3 복원 작업의 경우 AWS 세션 토큰 만료 기간이 충분한지 확인하십시오. 복원 작업 중에 토큰이 만료되면 작업이 실패할 수 있습니다.

- 현재 세션 토큰 만료 확인에 대한 자세한 내용은 "AWS API 문서"을 참조하십시오.
- AWS 리소스에 대한 자격 증명 관련 자세한 내용은 "AWS IAM 문서"를 참조하십시오.

CR 사용

단계

1. 사용자 정의 리소스(CR) 파일을 생성하고 이름을 지정합니다 `trident-protect-snapshot-restore-cr.yaml`.
2. 생성한 파일에서 다음 속성을 구성하십시오.
   - **metadata.name**: (필수) 이 사용자 지정 리소스의 이름입니다. 환경에 맞는 고유하고 적절한 이름을 선택하세요.
   - **spec.appVaultRef**: (필수) 스냅샷 콘텐츠가 저장된 AppVault의 이름입니다.
   - **spec.appArchivePath**: 스냅샷 콘텐츠가 저장되는 AppVault 내부의 경로입니다. 다음 명령을 사용하여 이 경로를 찾을 수 있습니다.

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

   - **spec.destinationApplicationName**: (선택 사항) 복원된 애플리케이션의 이름입니다. 제공된 경우 복원된 애플리케이션은 이 이름을 사용합니다. 제공되지 않은 경우 복원된 애플리케이션은 원본 애플리케이션 이름을 사용합니다.
   - **spec.namespaceMapping**: 복원 작업의 소스 네임스페이스를 타겟 네임스페이스로 매핑합니다. `my-source-namespace` 및 `my-destination-namespace`를 사용자 환경의 정보로 교체하십시오.

     ```
     ---
     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) 애플리케이션의 특정 리소스만 복원해야 하는 경우, 특정 레이블로 표시된 리소스를 포함하거나 제외하는 필터링을 추가하십시오.

   Trident Protect는 사용자가 선택한 리소스와의 연관성 때문에 일부 리소스를 자동으로 선택합니다. 예를 들어, 영구 볼륨 클레임 리소스를 선택하고 해당 리소스에 연결된 Pod가 있는 경우 Trident Protect는 연결된 Pod도 복원합니다.

   - **resourceFilter.resourceSelectionCriteria**: (필터링에 필수) `Include` 또는 `Exclude`를 사용하여 resourceMatchers에 정의된 리소스를 포함하거나 제외합니다. 포함 또는 제외할 리소스를 정의하려면 다음 resourceMatchers 매개변수를 추가하십시오.

- **resourceFilter.resourceMatchers**: resourceMatcher 객체의 배열입니다. 이 배열에 여러 요소를 정의하는 경우, 요소들은 OR 연산으로 일치하며, 각 요소 내부의 필드(그룹, 종류, 버전)는 AND 연산으로 일치합니다.
  - **resourceMatchers[].group**: (선택 사항) 필터링할 리소스의 그룹입니다.
  - **resourceMatchers[].kind**: (선택 사항) 필터링할 리소스의 종류입니다.
  - **resourceMatchers[].version**: (선택 사항) 필터링할 리소스의 버전입니다.
  - **resourceMatchers[].names**: (선택 사항) 필터링할 리소스의 Kubernetes metadata.name 필드에 있는 이름입니다.
  - **resourceMatchers[].namespaces**: (선택 사항) 필터링할 리소스의 Kubernetes metadata.name 필드에 있는 네임스페이스입니다.
  - **resourceMatchers[].labelSelectors**: (선택 사항) "Kubernetes 문서"에 정의된 리소스의 Kubernetes metadata.name 필드에 있는 레이블 선택기 문자열입니다. 예를 들면 다음과 같습니다: `"trident.netapp.io/os=linux"`.

    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. `trident-protect-snapshot-restore-cr.yaml` 파일에 올바른 값을 입력한 후 CR을 적용하십시오.

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

CLI 사용

단계

1. 스냅샷을 다른 네임스페이스로 복원하고 괄호 안의 값을 사용자 환경의 정보로 바꾸십시오.

   ◦ `snapshot` 인수는 `<namespace>/<name>` 형식의 네임스페이스와 스냅샷 이름을 사용합니다.

◦ `namespace-mapping` 인수는 콜론으로 구분된 네임스페이스를 사용하여 소스 네임스페이스를 `source1:dest1,source2:dest2` 형식으로 올바른 타겟 네임스페이스에 매핑합니다.

예를 들면 다음과 같습니다.

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

스냅샷에서 원래 네임스페이스로 복원합니다

언제든지 스냅샷을 원래 네임스페이스로 복원할 수 있습니다. 제자리 복원을 수행하면 Trident Protect는 잘못된 복구 지점이 생성되는 것을 방지하기 위해 보호 일정과 진행 중인 작업을 자동으로 관리합니다.

- 복원 작업이 시작되기 전에 애플리케이션에 대해 활성화된 모든 보호 일정이 비활성화됩니다. 이렇게 하면 애플리케이션 리소스가 복원되는 동안 예약된 백업이나 스냅샷이 실행되는 것을 방지할 수 있습니다.
- 복원 작업이 성공적으로 완료되면 복원 전에 활성화되어 있던 일정만 다시 활성화됩니다. 이미 비활성화된 일정은 계속 비활성화된 상태로 유지됩니다.
- 복원이 시작되기 전에 진행 중인 모든 백업 또는 스냅샷 작업이 취소됩니다. 5분 이내에 작업이 취소되지 않으면 복원이 진행되고 복원 CR 상태에 경고가 기록됩니다.

시작하기 전에

장시간 소요되는 s3 복원 작업의 경우 AWS 세션 토큰 만료 기간이 충분한지 확인하십시오. 복원 작업 중에 토큰이 만료되면 작업이 실패할 수 있습니다.

- 현재 세션 토큰 만료 확인에 대한 자세한 내용은 "AWS API 문서"을 참조하십시오.
- AWS 리소스에 대한 자격 증명 관련 자세한 내용은 "AWS IAM 문서"를 참조하십시오.

CR 사용

단계

1. 사용자 정의 리소스(CR) 파일을 생성하고 이름을 지정합니다 `trident-protect-snapshot-ipr-cr.yaml`.
2. 생성한 파일에서 다음 속성을 구성하십시오.
   - **metadata.name**: (필수) 이 사용자 지정 리소스의 이름입니다. 환경에 맞는 고유하고 적절한 이름을 선택하세요.
   - **spec.appVaultRef**: (필수) 스냅샷 콘텐츠가 저장된 AppVault의 이름입니다.
   - **spec.appArchivePath**: 스냅샷 콘텐츠가 저장되는 AppVault 내부의 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) 애플리케이션의 특정 리소스만 복원해야 하는 경우, 특정 레이블로 표시된 리소스를 포함하거나 제외하는 필터링을 추가하십시오.

> Trident Protect는 사용자가 선택한 리소스와의 연관성 때문에 일부 리소스를 자동으로 선택합니다. 예를 들어, 영구 볼륨 클레임 리소스를 선택하고 해당 리소스에 연결된 Pod가 있는 경우 Trident Protect는 연결된 Pod도 복원합니다.

   - **resourceFilter.resourceSelectionCriteria**: (필터링에 필수) `Include` 또는 `Exclude`를 사용하여 resourceMatchers에 정의된 리소스를 포함하거나 제외합니다. 포함 또는 제외할 리소스를 정의하려면 다음 resourceMatchers 매개변수를 추가하십시오.
     - **resourceFilter.resourceMatchers**: resourceMatcher 객체의 배열입니다. 이 배열에 여러 요소를 정의하는 경우, 요소들은 OR 연산으로 일치하며, 각 요소 내부의 필드(그룹, 종류, 버전)는 AND 연산으로 일치합니다.
       - **resourceMatchers[].group**: (선택 사항) 필터링할 리소스의 그룹입니다.
       - **resourceMatchers[].kind**: (선택 사항) 필터링할 리소스의 종류입니다.
       - **resourceMatchers[].version**: (선택 사항) 필터링할 리소스의 버전입니다.
       - **resourceMatchers[].names**: (선택 사항) 필터링할 리소스의 Kubernetes metadata.name

필드에 있는 이름입니다.

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

  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. `trident-protect-snapshot-ipr-cr.yaml` 파일에 올바른 값을 입력한 후 CR을 적용하십시오.

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

CLI 사용

단계

1. 스냅샷을 원래 네임스페이스로 복원하고 괄호 안의 값을 사용자 환경 정보로 바꾸십시오. 예를 들면 다음과 같습니다.

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

복원 작업의 상태를 확인합니다

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

단계

1. 다음 명령을 사용하여 복원 작업의 상태를 검색하고 대괄호 안의 값을 사용자 환경의 정보로 바꾸십시오.

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

고급 Trident Protect 복원 설정을 사용하십시오

주석, 네임스페이스 설정, 스토리지 옵션과 같은 고급 설정을 사용하여 특정 요구 사항에 맞게 복원 작업을 사용자 지정할 수 있습니다.

복원 및 페일오버 작업 중 네임스페이스 주석 및 레이블

복원 및 페일오버 작업 중에 대상 네임스페이스의 레이블과 주석은 소스 네임스페이스의 레이블과 주석과 일치하도록 조정됩니다. 대상 네임스페이스에 존재하지 않는 소스 네임스페이스의 레이블 또는 주석은 추가되며, 이미 존재하는 레이블 또는 주석은 소스 네임스페이스의 값과 일치하도록 덮어쓰여집니다. 대상 네임스페이스에만 존재하는 레이블 또는 주석은 변경되지 않습니다.

Red Hat OpenShift를 사용하는 경우 OpenShift 환경에서 네임스페이스 어노테이션의 중요한 역할을 알아두는 것이 중요합니다. 네임스페이스 어노테이션은 복원된 Pod가 OpenShift 보안 컨텍스트 제약 조건(SCC)에 정의된 적절한 권한 및 보안 구성을 준수하고 권한 문제 없이 볼륨에 액세스할 수 있도록 보장합니다. 자세한 내용은 "OpenShift 보안 컨텍스트 제약 조건 문서"를 참조하십시오.

복원 또는 장애 조치 작업을 수행하기 전에 Kubernetes 환경 변수 `RESTORE_SKIP_NAMESPACE_ANNOTATIONS`를 설정하여 대상 네임스페이스의 특정 어노테이션이 덮어쓰여지는 것을 방지할 수 있습니다. 예를 들면 다음과 같습니다.

```
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
```

복원 또는 페일오버 작업을 수행할 때 `restoreSkipNamespaceAnnotations` 및 `restoreSkipNamespaceLabels`에 지정된 네임스페이스 주석 및 레이블은 복원 또는 페일오버 작업에서 제외됩니다. 초기 Helm 설치 시 이러한 설정이 구성되어 있는지 확인하십시오. 자세한 내용은 "Trident Protect 헬름 차트의 추가 설정을 구성합니다"를 참조하십시오.

`--create-namespace` 플래그와 함께 `Helm`을 사용하여 소스 애플리케이션을 설치한 경우 `name` 레이블 키에 특별한 처리가 적용됩니다. 복원 또는 페일오버 프로세스 중에 `Trident Protect`는 이 레이블을 대상 네임스페이스로 복사하지만 소스의 값이 소스 네임스페이스와 일치하는 경우 값을 대상 네임스페이스 값으로 업데이트합니다. 이 값이 소스 네임스페이스와 일치하지 않으면 변경 없이 대상 네임스페이스로 복사됩니다.

예

다음 예제는 서로 다른 어노테이션과 레이블을 가진 소스 및 대상 네임스페이스를 보여줍니다. 작업 전후의 대상 네임스페이스 상태와 어노테이션 및 레이블이 대상 네임스페이스에서 어떻게 결합되거나 덮어쓰여지는지 확인할 수 있습니다.

복원 또는 페일오버 작업 전에

다음 표는 복원 또는 페일오버 작업 전 예시 소스 및 대상 네임스페이스의 상태를 보여줍니다.

네임스페이스	주석	라벨
네임스페이스 ns-1(소스)	• annotation.one/key: "updatedvalue" • annotation.two/key: "true"	• environment=production • 컴플라이언스=hipaa • name=ns-1
네임스페이스 ns-2(대상)	• annotation.one/key: "true" • annotation.three/key: "false"	• role=database

복원 작업 후

다음 표는 복원 또는 장애 조치 작업 후 예시 대상 네임스페이스의 상태를 보여줍니다. 일부 키가 추가되었고, 일부는 덮어쓰여졌으며, `name` 레이블은 대상 네임스페이스와 일치하도록 업데이트되었습니다.

네임스페이스	주석	라벨
네임스페이스 ns-2(대상)	• annotation.one/key: "updatedvalue" • annotation.two/key: "true" • annotation.three/key: "false"	• name=ns-2 • 컴플라이언스=hipaa • environment=production • role=database

지원되는 필드

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

스토리지 클래스 매핑

`spec.storageClassMapping` 속성은 소스 애플리케이션에 있는 스토리지 클래스를 대상 클러스터의 새 스토리지 클래스로 매핑하는 방법을 정의합니다. 이 속성은 스토리지 클래스가 다른 클러스터 간에 애플리케이션을 마이그레이션하거나 `BackupRestore` 작업의 스토리지 백엔드를 변경할 때 사용할 수 있습니다.

예:

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

지원되는 주석

이 섹션에서는 시스템의 다양한 동작을 구성하는 데 지원되는 어노테이션 목록을 제공합니다. 사용자가 어노테이션을 명시적으로 설정하지 않으면 시스템은 기본값을 사용합니다.

주석	유형	설명	기본값
protect.trident.netapp.io/data-mover-timeout-sec	문자열	데이터 이동기 작동이 정지될 수 있는 최대 허용 시간(초)입니다.	"300"
protect.trident.netapp.io/kopia-content-cache-size-limit-mb	문자열	Kopia 콘텐츠 캐시의 최대 크기 제한(메가바이트)입니다.	"1000"
protect.trident.netapp.io/pvc-bind-timeout-sec	문자열	새로 생성된 PersistentVolumeClaims(PVC)가 `Bound` 단계에 도달하기까지 기다리는 최대 시간(초)입니다. 이 시간이 지나면 작업이 실패합니다. 모든 복원 CR 유형(BackupRestore, BackupInplaceRestore, SnapshotRestore, SnapshotInplaceRestore)에 적용됩니다. 스토리지 백엔드 또는 클러스터에서 더 많은 시간이 필요한 경우 더 높은 값을 사용하십시오.	"1200" (20분)