



Trident Protect를 사용하여 데이터 보호

NetApp public and hybrid cloud solutions

NetApp
August 18, 2025

목차

Trident Protect를 사용하여 데이터 보호	1
Trident Protect를 사용한 OpenShift Container Platform의 컨테이너 앱에 대한 데이터 보호	1
Trident Protect를 사용하여 Red Hat OpenShift Virtualization에서 VM 보호	14
개체 저장소용 App Vault 만들기	15
OpenShift Virtualization에서 VM 만들기	17
앱 만들기	21
백업을 만들어 앱을 보호하세요	22
백업에서 복원	25
스냅샷을 사용하여 앱 보호	29
스냅샷에서 복원	32
특정 VM 복원	36
비디오 데모	40

Trident Protect를 사용하여 데이터 보호

Trident Protect를 사용한 OpenShift Container Platform의 컨테이너 앱에 대한 데이터 보호

이 참조 문서 섹션에서는 Trident Protect를 사용하여 컨테이너 앱의 스냅샷과 백업을 만드는 방법에 대한 세부 정보를 제공합니다. NetApp Trident Protect는 NetApp ONTAP 스토리지 시스템과 NetApp Trident CSI 스토리지 프로비저너가 지원하는 상태 저장 Kubernetes 애플리케이션의 기능과 가용성을 향상시키는 고급 애플리케이션 데이터 관리 기능을 제공합니다. Trident Protect는 애플리케이션 스냅샷과 백업을 생성합니다. 즉, 영구 볼륨에 있는 애플리케이션 데이터의 스냅샷과 백업이 생성될 뿐만 아니라 애플리케이션 메타데이터의 스냅샷과 백업도 생성됩니다. Trident Protect에서 생성된 스냅샷과 백업은 다음 개체 스토리지에 저장하여 나중에 복원할 수 있습니다.

- AWS S3
- Azure Blob 저장소
- 구글 클라우드 스토리지
- 온택 S3
- 스토리지그리드
- 다른 S3 호환 스토리지

Trident Protect는 역할 기반 액세스 제어(RBAC)의 Kubernetes 모델을 사용합니다. 기본적으로 Trident Protect는 trident-protect라는 단일 시스템 네임스페이스와 관련 기본 서비스 계정을 제공합니다. 사용자나 특정 보안 요구 사항이 많은 조직이 있는 경우 Trident Protect의 RBAC 기능을 사용하여 리소스와 네임스페이스에 대한 액세스를 보다 세부적으로 제어할 수 있습니다.

Trident Protect의 RBAC에 대한 추가 정보는 다음에서 찾을 수 있습니다. "[Trident 프로텍트 문서](#)"



클러스터 관리자는 기본 trident-protect 네임스페이스의 리소스에 액세스할 수 있으며 다른 모든 네임스페이스의 리소스에도 액세스할 수 있습니다. 사용자는 trident-protect 네임스페이스에서 스냅샷 및 백업 CR과 같은 애플리케이션 데이터 관리 사용자 정의 리소스(CR)를 생성할 수 없습니다. 가장 좋은 방법은 사용자가 애플리케이션 네임스페이스에서 해당 CR을 만드는 것입니다.

Trident Protect는 설명서에 제공된 지침을 사용하여 설치할 수 있습니다. "[여기](#)" 이 섹션에서는 Trident Protect를 사용하여 컨테이너 애플리케이션의 데이터 보호 및 애플리케이션 복원 워크플로를 보여줍니다. 1. 스냅샷 생성(예약된 시간에 따라) 2. 스냅샷에서 복원(동일한 네임스페이스 및 다른 네임스페이스로 복원) 3. 백업 생성 4. 백업에서 복원

필수 조건

애플리케이션의 스냅샷과 백업을 생성하기 전에 Trident Protect에서 스냅샷과 백업을 저장할 개체 스토리지를 구성해야 합니다. 이 작업은 버킷 CR을 사용하여 수행됩니다. 관리자만 버킷 CR을 만들고 구성할 수 있습니다. Trident Protect에서는 버킷 CR을 AppVault라고 합니다. AppVault 객체는 스토리지 버킷의 선언적 Kubernetes 워크플로 표현입니다. AppVault CR에는 백업, 스냅샷, 복원 작업, SnapMirror 복제와 같은 보호 작업에 버킷을 사용하는 데 필요한 구성이 포함되어 있습니다.

이 예에서는 ONTAP S3를 개체 스토리지로 사용하는 방법을 보여드리겠습니다. ONTAP S3용 AppVault CR을 만드는 워크플로는 다음과 같습니다. 1. ONTAP 클러스터의 SVM에 S3 객체 저장 서버를 생성합니다. 2. Object Store Server에 버킷을 만듭니다. 3. SVM에서 S3 사용자를 생성합니다. 액세스 키와 비밀 키를 안전한 곳에 보관하세요. 4. OpenShift에서 ONTAP S3 자격 증명을 저장할 비밀을 만듭니다. 5. ONTAP S3에 대한 AppVault 개체 만들기

- ONTAP S3에 대한 Trident Protect AppVault 구성**

AppVault로 ONTAP S3를 사용하여 Trident Protect를 구성하기 위한 샘플 yaml 파일

```
# alias tp='tridentctl-protect'

appvault-secret.yaml

apiVersion: v1
stringData:
  accessKeyID: "<access key id created for a user to access ONTAP S3
bucket>"
  secretAccessKey: "corresponding Secret Access Key"
#data:
# base 64 encoded values
#  accessKeyID: <base64 access key id created for a user to access
ONTAP S3 bucket>
#  secretAccessKey: <base 64 Secret Access Key>
kind: Secret
metadata:
  name: appvault-secret
  namespace: trident-protect
type: Opaque

appvault.yaml

apiVersion: protect.trident.netapp.io/v1
kind: AppVault
metadata:
  name: ontap-s3-appvault
  namespace: trident-protect
spec:
  providerConfig:
    azure:
```

```

    accountName: ""
    bucketName: ""
    endpoint: ""
  gcp:
    bucketName: ""
    projectID: ""
  s3:
    bucketName: <bucket-name for storing the snapshots and backups>
    endpoint: <endpoint IP for S3>
    secure: "false"
    skipCertValidation: "true"
  providerCredentials:
    accessKeyID:
      valueFromSecret:
        key: accessKeyID
        name: appvault-secret
    secretAccessKey:
      valueFromSecret:
        key: secretAccessKey
        name: appvault-secret
  providerType: OntapS3

# oc create -f appvault-secret.yaml -n trident-protect
# oc create -f appvault.yaml -n trident-protect

```

```

[root@localhost openshift-benchmark]#
[root@localhost openshift-benchmark]# tp get appvault -n trident-protect
+-----+-----+-----+-----+-----+
|      NAME      | PROVIDER |  STATE  | AGE  | ERROR |
+-----+-----+-----+-----+-----+
| ontap-s3-appvault | OntapS3  | Available | 2d2h |      |
+-----+-----+-----+-----+-----+
[root@localhost openshift-benchmark]# █

```

postgresql 앱 설치를 위한 샘플 yaml 파일

```

postgres.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: postgres
spec:
  replicas: 1
  selector:
    matchLabels:
      app: postgres

```

```

template:
  metadata:
    labels:
      app: postgres
  spec:
    containers:
      - name: postgres
        image: postgres:14
        env:
          - name: POSTGRES_USER
            #value: "myuser"
            value: "admin"
          - name: POSTGRES_PASSWORD
            #value: "mypassword"
            value: "adminpass"
          - name: POSTGRES_DB
            value: "mydb"
          - name: PGDATA
            value: "/var/lib/postgresql/data/pgdata"
        ports:
          - containerPort: 5432
        volumeMounts:
          - name: postgres-storage
            mountPath: /var/lib/postgresql/data
    volumes:
      - name: postgres-storage
        persistentVolumeClaim:
          claimName: postgres-pvc
---
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: postgres-pvc
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 5Gi
---
apiVersion: v1
kind: Service
metadata:
  name: postgres
spec:
  selector:

```

```
app: postgres
ports:
- protocol: TCP
  port: 5432
  targetPort: 5432
type: ClusterIP
```

Now create the Trident protect application CR for the postgres app. Include the objects in the namespace postgres and create it in the postgres namespace.

```
# tp create app postgres-app --namespaces postgres -n postgres
```

```
[root@localhost RedHat]# tp get app -n postgres
+-----+-----+-----+-----+
| NAME           | NAMESPACE | STATE | AGE |
+-----+-----+-----+-----+
| postgres-app  | postgres  | Ready | 24s |
+-----+-----+-----+-----+
[root@localhost RedHat]# █
```

스냅샷 만들기

주문형 스냅샷 만들기

```
# tp create snapshot postgres-snap1 --app postgres-app --appvault ontap-s3-appvault -n postgres
Snapshot "postgres-snap1" created.
```

```
[root@localhost RedHat]# tp get snapshot -n postgres
+-----+-----+-----+-----+-----+
| NAME          | APP REF      | STATE    | AGE   | ERROR |
+-----+-----+-----+-----+-----+
| postgres-snap1 | postgres-app | Completed | 19s   |       |
+-----+-----+-----+-----+-----+
[root@localhost RedHat]#
```

```
[root@localhost DataProtection]# oc get all,pvc,volumesnapshot -n postgres
apps.openshift.io/v1 DeploymentConfig is deprecated in v4.14+, unavailable in v4.10000+
kubevirt.io/v1 VirtualMachineInstancePresets is now deprecated and will be removed in v2.
NAME                                READY  STATUS   RESTARTS  AGE
pod/postgres-cd9d6ccb-jfx49         1/1    Running  0          3h47m

NAME                                TYPE          CLUSTER-IP    EXTERNAL-IP  PORT(S)    AGE
service/postgres                    ClusterIP     172.30.132.112 <none>       5432/TCP   3h47m

NAME                                READY  UP-TO-DATE  AVAILABLE   AGE
deployment.apps/postgres            1/1    1           1           3h47m

NAME                                DESIRED  CURRENT  READY   AGE
replicaset.apps/postgres-cd9d6ccb   1        1        1       3h47m

NAME                                STATUS   VOLUME          CAPACITY   ACCESS MODES   STORAGECLASS
ECLASS VOLUMEATTRIBUTESCLASS  AGE
persistentvolumeclaim/data-postgres-postgresql-0  Bound   pvc-9f89514e-3f2c-41ad-b7a3-792cea503f00  8Gi      RWO           sc-zon
ea-nas <unset>                4h40m
persistentvolumeclaim/postgres-pvc              Bound   pvc-951a9910-9edb-48ae-898a-1aed9aa25dc3  5Gi      RWO           sc-zon
ea-nas <unset>                3h47m

NAME                                READYTO
USE  SOURCEPVC                SOURCESNAPSHOTCONTENT  RESTORESIZE  SNAPSHOTCLASS  SNAPSHOTCONTENT
CREATIONTIME  AGE
volumesnapshot.snapshot.storage.k8s.io/snapshot-2e94d04c-c8ea-446a-8d47-64b0bee67107-pvc-951a9910-9edb-48ae-898a-1aed9aa25dc3  true
postgres-pvc  53676Ki  trident-snapshotclass  snapcontent-796ea7f8-59a0-493e-bbd8-3a
e76fe9036c  13m     13m
volumesnapshot.snapshot.storage.k8s.io/snapshot-2e94d04c-c8ea-446a-8d47-64b0bee67107-pvc-9f89514e-3f2c-41ad-b7a3-792cea503f00  true
data-postgres-postgresql-0  368Ki  trident-snapshotclass  snapcontent-86a464d4-ffd8-4279-9cf7-88
88a097c001  13m     13m
```

일정 만들기 다음 명령을 사용하면 매일 15:33에 스냅샷이 생성되고 두 개의 스냅샷과 백업이 보관됩니다.

```
# tp create schedule schedule1 --app postgres-app --appvault ontap-s3-appvault --backup-retention 2 --snapshot-retention 2 --granularity Daily --hour 15 --minute 33 --data-mover Restic -n postgres
Schedule "schedule1" created.
```

```
[root@localhost DataProtection]# tp get schedule -n postgres
```

NAME	APP	SCHEDULE	ENABLED	STATE	AGE	ERROR
schedule1	postgres-app	Daily:hour=15,min=33	true		17s	

```
[root@localhost DataProtection]#
```

yaml을 사용하여 일정 만들기

```
# tp create schedule schedule2 --app postgres-app --appvault ontap-s3-
appvault --backup-retention 2 --snapshot-retention 2 --granularity
Daily --hour 15 --minute 33 --data-mover Restic -n postgres --dry-run >
hourly-snapshotschedule.yaml
```

```
cat hourly-snapshotschedule.yaml
```

```
apiVersion: protect.trident.netapp.io/v1
kind: Schedule
metadata:
  creationTimestamp: null
  name: schedule2
  namespace: postgres
spec:
  appVaultRef: ontap-s3-appvault
  applicationRef: postgres-app
  backupRetention: "2"
  dataMover: Restic
  dayOfMonth: ""
  dayOfWeek: ""
  enabled: true
  granularity: Hourly
  #hour: "15"
  minute: "33"
  recurrenceRule: ""
  snapshotRetention: "2"
status: {}
```

```
[root@localhost DataProtection]# tp get schedule -n postgres
```

NAME	APP	SCHEDULE	ENABLED	STATE	AGE	ERROR
schedule1	postgres-app	Daily:hour=15,min=33	true		8d7h	
schedule2	postgres-app	Hourly:min=33	true		8d7h	

```
[root@localhost DataProtection]#
```

이 일정에 따라 생성된 스냅샷을 볼 수 있습니다.

```
[root@localhost DataProtection]# tp get snap -n postgres
```

NAME	APP REF	STATE	AGE	ERROR
hourly-3f1ee-20250214183300	postgres-app	Completed	19s	
postgres-snap1	postgres-app	Completed	1h25m	

```
[root@localhost DataProtection]#
```

볼륨 스냅샷도 생성됩니다.

```
[root@localhost DataProtection]# oc get volumesnapshots -n postgres
```

NAME	CREATIONTIME	AGE	READYTOUSE	SOURCEPVC
snapshot-2e94d04c-c8ea-446a-8d47-64b0bee67107-pvc-951a9910-9edb-48ae-898a-1aed9aa25dc3-59a0-493e-bbd8-3ae76fe9036c	114m	114m	true	postgres-pvc
snapshot-2e94d04c-c8ea-446a-8d47-64b0bee67107-pvc-9f89514e-3f2c-41ad-b7a3-792cea503f00-ffd8-4279-9cf7-8888a097c001	114m	114m	true	data-postgres-postgresql-0
snapshot-ce75a274-ecb2-48c9-a0a5-94c10f8e6cb1-pvc-951a9910-9edb-48ae-898a-1aed9aa25dc3-7adc-4042-a8c9-7606d1103ead	30m	30m	true	postgres-pvc
snapshot-ce75a274-ecb2-48c9-a0a5-94c10f8e6cb1-pvc-9f89514e-3f2c-41ad-b7a3-792cea503f00-3a82-43f6-9868-dcadd2cc04e2	30m	30m	true	data-postgres-postgresql-0

애플리케이션 손실을 시뮬레이션하기 위해 애플리케이션을 삭제합니다.

```
# oc delete deployment/postgres -n postgres
# oc get pod,pvc -n postgres
No resources found in postgres namespace.
```

스냅샷에서 동일한 네임스페이스로 복원

```
# tp create sir postgres-sir --snapshot postgres/hourly-3flee-20250214183300 -n postgres
SnapshotInplaceRestore "postgres-sir" created.
```

```
[root@localhost DataProtection]# tp get sir -n postgres
```

NAME	APPVAULT	STATE	AGE	ERROR
postgres-sir	ontap-s3-appvault	Completed	2m39s	

애플리케이션과 해당 PVC가 동일한 네임스페이스로 복원됩니다.

```
[root@localhost DataProtection]# oc get pods,pvc -n postgres
```

NAME	READY	STATUS	RESTARTS	AGE
pod/postgres-cd9d6ccb-x85tg	1/1	Running	0	98s

NAME	STATUS	VOLUME	CAPACITY	ACCESS MODES	STORAGECLASS	VOLUMEATTRIBUTESCLASS	AGE
persistentvolumeclaim/data-postgres-postgresql-0	Bound	pvc-b2cd67fd-fe4d-49b1-9e06-a53bf7be575e	8Gi	RWO	sc-zonea-nas	<unset>	105s
persistentvolumeclaim/postgres-pvc	Bound	pvc-2d549395-0cc6-4529-b2b9-7361bfb14fa8	5Gi	RWO	sc-zonea-nas	<unset>	105s

스냅샷에서 다른 네임스페이스로 복원

```
# tp create snapshotrestore postgres-restore --snapshot postgres/hourly-3flee-20250214183300 --namespace-mapping postgres:postgres-restore -n postgres-restore
SnapshotRestore "postgres-restore" created.
```

```
[root@localhost DataProtection]# tp get snapshotrestore -n postgres-restore
```

NAME	APPVAULT	STATE	AGE	ERROR
postgres-restore	ontap-s3-appvault	Completed	1m15s	

애플리케이션이 새로운 네임스페이스로 복원된 것을 볼 수 있습니다.

```
[root@localhost DataProtection]# [root@localhost DataProtection]# oc get pods,pvc -n postgres-restore
```

NAME	READY	STATUS	RESTARTS	AGE
pod/postgres-cd9d6ccb-pfxw9	1/1	Running	0	8d

NAME	STATUS	VOLUME	CAPACITY	ACCESS MODES	STORAGECLASS
persistentvolumeclaim/data-postgres-postgresql-0	Bound	pvc-3dff4e42-828d-49f9-84e9-4daf75479292	8Gi	RWO	sc-zonea-nas
persistentvolumeclaim/postgres-pvc	Bound	pvc-f57321b2-f21f-4eb7-8f86-4a17f915318b	5Gi	RWO	sc-zonea-nas

백업 만들기

주문형 백업 생성

```
# tp create backup postgres-backup1 --app postgres-app --appvault
ontap-s3-appvault -n postgres
Backup "postgres-backup1" created.
```

```
[root@localhost DataProtection]# tp get backup -n postgres
```

NAME	APP REF	STATE	AGE	ERROR
backup1	postgres-app	Completed	5d12h	
daily-feac1-20250222153300	postgres-app	Completed	1d10h	
daily-feac1-20250223153300	postgres-app	Completed	10h36m	
hourly-3f1ee-20250224003300	postgres-app	Completed	1h36m	
hourly-3f1ee-20250224013300	postgres-app	Completed	36m27s	
postgres-backup1	postgres-app	Completed	6m19s	

백업 일정 생성

위 목록의 일일 백업과 매시간 백업은 이전에 설정한 일정에 따라 생성됩니다.

```
# tp create schedule schedule1 --app postgres-app --appvault ontap-s3-
appvault --backup-retention 2 --snapshot-retention 2 --granularity
Daily --hour 15 --minute 33 --data-mover Restic -n postgres
Schedule "schedule1" created.
```

```
[root@localhost DataProtection]# tp get schedule -n postgres
```

NAME	APP	SCHEDULE	ENABLED	STATE	AGE	ERROR
schedule1	postgres-app	Daily:hour=15,min=33	true		9d8h	
schedule2	postgres-app	Hourly:min=33	true		9d8h	

데이터 손실을 시뮬레이션하려면 애플리케이션과 **PVC**를 삭제하세요.

```
[root@localhost DataProtection]# oc get pods -n postgres
NAME                READY   STATUS    RESTARTS   AGE
postgres-cd9d6ccb-dftkt 1/1     Running   0           11s
[root@localhost DataProtection]# oc get deployment -n postgres
NAME                READY   UP-TO-DATE   AVAILABLE   AGE
postgres            1/1     1             1           20s
[root@localhost DataProtection]# oc delete deployment/postgres -n postgres
deployment.apps "postgres" deleted
[root@localhost DataProtection]# oc get pods -n postgres
No resources found in postgres namespace.
[root@localhost DataProtection]# oc get pvc -n postgres
NAME                STATUS    VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS   VOLUMEATTRIBUTE
SCLASS   AGE
data-postgres-postgresql-0  Bound    pvc-b2cd67fd-fe4d-49b1-9e06-a53bf7be575e  8Gi        RWO             sc-zonea-nas  <unset>
5d13h
postgres-pvc              Bound    pvc-2d549395-0cc6-4529-b2b9-7361bf14fa8  5Gi        RWO             sc-zonea-nas  <unset>
5d13h
[root@localhost DataProtection]# oc delete pvc/data-postgres-postgresql-0 -n postgres
persistentvolumeclaim "data-postgres-postgresql-0" deleted
[root@localhost DataProtection]# oc delete pvc/postgres-pvc -n postgres
persistentvolumeclaim "postgres-pvc" deleted
[root@localhost DataProtection]# oc get pvc -n postgres
No resources found in postgres namespace.
[root@localhost DataProtection]#
```

동일한 네임스페이스로 복원 #tp create bir postgres-bir --backup postgres/hourly-3f1ee-20250224023300 -n postgres BackupInplaceRestore "postgres-bir"이 생성되었습니다.

```
[root@localhost DataProtection]# tp get bir -n postgres
+-----+-----+-----+-----+-----+
| NAME          | APPVAULT | STATE   | AGE   | ERROR |
+-----+-----+-----+-----+-----+
| postgres-bir  | ontap-s3-appvault | Completed | 2m19s |      |
+-----+-----+-----+-----+-----+
```

애플리케이션과 PVC는 동일한 네임스페이스에 복원됩니다.

```
[root@localhost DataProtection]# oc get pods -n postgres
NAME                READY   STATUS    RESTARTS   AGE
postgres-cd9d6ccb-t857w 1/1     Running   0           10m
[root@localhost DataProtection]# oc get pvc -n postgres
NAME                STATUS    VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS   VOLUMEATTRIBUTE
ESCLASS   AGE
data-postgres-postgresql-0  Bound    pvc-0a849c19-16fe-466f-9733-85e82a8b1677  8Gi        RWO             sc-zonea-nas  <unset>
10m
postgres-pvc              Bound    pvc-ded304ea-02d4-4225-b606-63007666ad66  5Gi        RWO             sc-zonea-nas  <unset>
10m
```

다른 네임스페이스로 복원 새로운 네임스페이스를 만듭니다. 백업에서 새 네임스페이스로 복원합니다.

```

[root@localhost DataProtection]# oc create ns postgres-restore-from-backup
namespace/postgres-restore-from-backup created
[root@localhost DataProtection]# tp create backuprestore postgres-restore-from-backup --backup postgres/postgres-backup1 --namespace-map
ping postgres:postgres-restore-from-backup -n postgres-restore-from-backup
BackupRestore "postgres-restore-from-backup" created.
[root@localhost DataProtection]# oc get backuprestore -n postgres-restore-from-backup
NAME                               STATE      ERROR      AGE
postgres-restore-from-backup       Running      

[root@localhost DataProtection]# oc get backuprestore -n postgres-restore-from-backup
NAME                               STATE      ERROR      AGE
postgres-restore-from-backup       Running      

[root@localhost DataProtection]# oc get backuprestore -n postgres-restore-from-backup
NAME                               STATE      ERROR      AGE
postgres-restore-from-backup       Completed    

[root@localhost DataProtection]# oc get pods -n postgres-restore-from-backup
NAME                                READY      STATUS      RESTARTS      AGE
postgres-cd9d6ccb-p659p             1/1       Running    0              2m9s
[root@localhost DataProtection]# oc get pvc -n postgres-restore-from-backup
NAME                                STATUS      VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS   VOLUMEATTRIBUT
ESCLASS   AGE
data-postgres-postgresql-0          Bound      pvc-36df7399-95da-4c67-a621-af9434015bdb   8Gi        RWO              sc-zonea-nas   <unset>
2m18s
postgres-pvc                        Bound      pvc-633de3aa-a4f9-4f3b-93cc-e91afbd4fe02   5Gi        RWO              sc-zonea-nas   <unset>
2m18s
[root@localhost DataProtection]#

```

Activate Windows
 Go to Settings to activate Windows.

다른 클러스터로 애플리케이션을 복제하거나 마이그레이션하려면(클러스터 간 복제 수행) 소스 클러스터에서 백업을 만든 다음, 해당 백업을 다른 클러스터로 복원합니다. 대상 클러스터에 Trident Protect가 설치되어 있는지 확인하세요.

소스 클러스터에서 아래 이미지에 표시된 단계를 수행합니다.

```
[root@localhost DataProtection]# tp create backup postgres-backup-cluster1 --app postgres-app --appvault ontap-s3-appvault -n postgres
Backup "postgres-backup-cluster1" created.
[root@localhost DataProtection]# tp get backup -n postgres
```

NAME	APP REF	STATE	AGE	ERROR
backup1	postgres-app	Completed	5d14h	
daily-feac1-20250222153300	postgres-app	Completed	1d12h	
daily-feac1-20250223153300	postgres-app	Completed	12h18m	
hourly-3f1ee-20250224023300	postgres-app	Completed	1h18m	
hourly-3f1ee-20250224033300	postgres-app	Completed	18m26s	
postgres-backup-cluster1	postgres-app	Running	35s	
postgres-backup1	postgres-app	Completed	1h48m	

```
[root@localhost DataProtection]# tp get backup -n postgres
```

NAME	APP REF	STATE	AGE	ERROR
backup1	postgres-app	Completed	5d14h	
daily-feac1-20250222153300	postgres-app	Completed	1d12h	
daily-feac1-20250223153300	postgres-app	Completed	12h19m	
hourly-3f1ee-20250224023300	postgres-app	Completed	1h19m	
hourly-3f1ee-20250224033300	postgres-app	Completed	19m41s	
postgres-backup-cluster1	postgres-app	Completed	1m50s	
postgres-backup1	postgres-app	Completed	1h49m	

Activate Windows
Go to Settings to activate Windows.

소스 클러스터에서 대상 클러스터로 컨텍스트를 전환합니다. 그런 다음 대상 클러스터 컨텍스트에서 AppVault에 액세스할 수 있는지 확인하고 대상 클러스터에서 AppVault 콘텐츠를 가져옵니다.

```
[root@localhost DataProtection]# kubectl config use-context default/api-bm-cluster5-min-ocpv-sddc-netapp-com:6443/kube:admin
Switched to context "default/api-bm-cluster5-min-ocpv-sddc-netapp-com:6443/kube:admin".
[root@localhost DataProtection]# tp get appvault -n trident-protect
```

NAME	PROVIDER	STATE	AGE	ERROR
ontap-s3-appvault	OntapS3	Available	3d6h	

```
[root@localhost DataProtection]# tp get appvaultcontent ontap-s3-appvault --show-resources backup --show-paths
```

CLUSTER	APP PATH	TYPE	NAME	TIMESTAMP
ocp-cluster11	bbox	backup	bboxbackup1	2025-01-17 15:57:49 (UTC)
ocp-cluster11	postgres-app	backup	backup1	2025-02-18 13:31:50 (UTC)
ocp-cluster11	postgres-app	backup	daily-feac1-20250222153300	2025-02-22 15:34:44 (UTC)
ocp-cluster11	postgres-app	backup	daily-feac1-20250223153300	2025-02-23 15:34:42 (UTC)
ocp-cluster11	postgres-app	backup	hourly-3f1ee-20250224033300	2025-02-24 03:34:44 (UTC)
ocp-cluster11	postgres-app	backup	hourly-3f1ee-20250224043300	2025-02-24 04:34:47 (UTC)
ocp-cluster11	postgres-app	backup	postgres-backup-cluster1	2025-02-24 03:52:36 (UTC)

목록에서 백업 경로를 사용하고 아래 명령에 표시된 대로 backuprestore CR 객체를 만듭니다.

```
# tp create backuprestore backup-restore-cluster2 --namespace-mapping
postgres:postgres --appvault ontap-s3-appvault --path postgres-
app_4d798ed5-cfa8-49ff-a5b6-c5e2d89aeb89/backups/postgres-backup-
cluster1_ec0ed3f3-5500-4e72-afa8-117a04a0b1c3 -n postgres
BackupRestore "backup-restore-cluster2" created.
```

```
[root@localhost DataProtection]# tp get backuprestore -n postgres
```

NAME	APPVAULT	STATE	AGE	ERROR
backup-restore-cluster2	ontap-s3-appvault	Completed	12m41s	

이제 대상 클러스터에 애플리케이션 포드와 PVC가 생성된 것을 볼 수 있습니다.

```
[root@localhost DataProtection]# oc get pods -n postgres
NAME READY STATUS RESTARTS AGE
postgres-cd9d6ccb-2lvcq 1/1 Running 0 13m
[root@localhost DataProtection]# oc get pvc -n postgres
NAME STATUS VOLUME CAPACITY ACCESS MODES STORAGECLASS VOLUMEATTRIBUT
ESCLASS AGE
data-postgres-postgresql-0 Bound pvc-872a5182-601b-4848-b410-fef368337d07 8Gi RWO sc-zoneb-san <unset>
13m
postgres-pvc Bound pvc-caf9fa71-76a8-4645-9bb5-2ed72e72948b 5Gi RWO sc-zoneb-san <unset>
13m
[root@localhost DataProtection]#
```

Trident Protect를 사용하여 Red Hat OpenShift Virtualization에서 VM 보호

스냅샷과 백업을 사용하여 OpenShift Virtualization에서 VM을 보호합니다. 이 절차에는 ONTAP S3 개체 스토리지를 사용하여 AppVault를 만들고, Kubernetes 리소스 개체, 영구 볼륨, 내부 이미지를 비롯한 VM 데이터를 캡처하도록 Trident Protect를 구성하고, 필요한 경우 데이터를 복원하는 작업이 포함됩니다.

OpenShift 가상화 환경의 가상 머신은 OpenShift 컨테이너 플랫폼의 워커 노드에서 실행되는 컨테이너화된 애플리케이션입니다. VM 메타데이터와 VM의 영구 디스크를 보호하는 것은 손실되거나 손상될 경우 복구할 수 있도록 하는 데 중요합니다.

OpenShift Virtualization VM의 영구 디스크는 OpenShift 클러스터에 통합된 ONTAP 스토리지로 백업할 수 있습니다. "Trident CSI". 이 섹션에서는 다음을 사용합니다. "Trident 프로텍트" 데이터 볼륨을 포함한 VM의 스냅샷과 백업을 ONTAP Object Storage에 생성합니다.

필요한 경우 스냅샷이나 백업에서 복원합니다.

Trident Protect를 사용하면 OpenShift 클러스터에서 애플리케이션과 VM의 스냅샷, 백업, 복원 및 재해 복구가 가능합니다. OpenShift Virtualization VM의 경우 Trident Protect로 보호할 수 있는 데이터에는 VM과 연결된 Kubernetes 리소스 개체, 영구 볼륨, 내부 이미지가 포함됩니다.

다음은 이 섹션의 예제에 사용된 다양한 구성 요소의 버전입니다.

- "오픈시프트 클러스터 4.17"
- "Red Hat에서 제공하는 OpenShift Virtualization Operator를 통해 설치된 OpenShift Virtualization"
- "Trident 25.02"
- "Trident 프로젝트 25.02"
- "ONTAP 9.16"

개체 저장소용 **App Vault** 만들기

AppVault 만들기

애플리케이션이나 VM에 대한 스냅샷과 백업을 생성하기 전에 Trident Protect에서 스냅샷과 백업을 저장할 개체 스토리지를 구성해야 합니다. 이 작업은 버킷 CR을 사용하여 수행됩니다. 관리자만 버킷 CR을 만들고 구성할 수 있습니다. Trident Protect에서는 버킷 CR을 AppVault라고 합니다. AppVault 객체는 스토리지 버킷의 선언적 Kubernetes 워크플로 표현입니다. AppVault CR에는 백업, 스냅샷, 복원 작업, SnapMirror 복제와 같은 보호 작업에 버킷을 사용하는 데 필요한 구성이 포함되어 있습니다.

이 예에서는 ONTAP S3를 개체 스토리지로 사용하는 방법을 보여드리겠습니다. ONTAP S3용 AppVault CR을 만드는 워크플로는 다음과 같습니다. 1. ONTAP 클러스터의 SVM에 S3 객체 저장 서버를 생성합니다. 2. Object Store Server에 버킷을 만듭니다. 3. SVM에서 S3 사용자를 생성합니다. 액세스 키와 비밀 키를 안전한 곳에 보관하세요. 4. OpenShift에서 ONTAP S3 자격 증명을 저장할 비밀을 만듭니다. 5. ONTAP S3에 대한 AppVault 개체 만들기

- ONTAP S3에 대한 Trident Protect AppVault 구성**

```
# alias tp='tridentctl-protect'

# cat appvault-secret.yaml
apiVersion: v1
stringData:
  accessKeyID: "<access key of S3>"
  secretAccessKey: "<secret access key of S3>"
# you can also provide base 64 encoded values instead of string values
#data:
# base 64 encoded values
#  accessKeyID: < base 64 encoded access key>
#  secretAccessKey: <base 64 encoded secretAccess key>
kind: Secret
metadata:
  name: appvault-secret
  namespace: trident-protect
type: Opaque

# cat appvault.yaml
apiVersion: protect.trident.netapp.io/v1
kind: AppVault
metadata:
  name: ontap-s3-appvault
  namespace: trident-protect
spec:
  providerConfig:
    azure:
      accountName: ""
      bucketName: ""
      endpoint: ""
    gcp:
```

```

bucketName: ""
projectID: ""
s3:
  bucketName: trident-protect
  endpoint: <lif for S3 access>
  secure: "false"
  skipCertValidation: "true"
providerCredentials:
  accessKeyID:
    valueFromSecret:
      key: accessKeyID
      name: appvault-secret
  secretAccessKey:
    valueFromSecret:
      key: secretAccessKey
      name: appvault-secret
providerType: OntapS3

# oc create -f appvault-secret.yaml -n trident-protect
# oc create -f appvault.yaml -n trident-protect

```

```

[root@localhost VM-DataProtection]# tp get appvault
+-----+-----+-----+-----+-----+-----+
| NAME          | PROVIDER | STATE   | ERROR | MESSAGE | AGE   |
+-----+-----+-----+-----+-----+-----+
| ontap-s3-appvault | OntapS3  | Available |      |          | 8d17h |
+-----+-----+-----+-----+-----+-----+
[root@localhost VM-DataProtection]#

```

OpenShift Virtualization에서 VM 만들기

OpenShift Virtualization에서 VM 만들기

다음 스크린샷은 템플릿을 사용하여 콘솔에서 VM(네임스페이스 demo의 demo-fedora)을 만드는 방법을 보여줍니다. 루트 디스크는 기본 저장소 클래스를 자동으로 선택하므로 기본 저장소 클래스가 적절하게 설정되었는지 확인하세요. 이 설정에서 기본 저장소 클래스는 **sc-zonea-san**입니다. 추가 디스크를 생성할 때 스토리지 클래스 sc-zonea-san을 선택하고 "최적화된 스토리지 설정 적용" 확인란을 선택하세요. 이렇게 하면 액세스 모드가 RWX로 설정되고 볼륨 모드가 차단으로 설정됩니다.



Trident SAN(iSCSI, NVMe/TCP 및 FC)의 블록 볼륨 모드에서 RWX 액세스 모드를 지원합니다. (NAS의 기본 접속 모드입니다.) 나중에 VM의 라이브 마이그레이션을 수행해야 하는 경우 RWX 액세스 모드가 필요합니다.

Name	Provisioner
SC sc-zonea-san - Default	csi.trident.netapp.io
SC thin-csi	csi.vsphere.vmware.com

Project: demo

VirtualMachines

No VirtualMachines found

Click Create VirtualMachine to create your first VirtualMachine or view the catalog tab to

Create VirtualMachine

- From InstanceType
- From template
- With YAML



Fedora VM

fedora-server-small

✕

▼ **Template info**

Operating system
Fedora VM

Workload type
Server (default)

Description
Template for Fedora Linux 39 VM or newer. A PVC with the Fedora disk image must be

▼ **Storage** ⓘ

Boot from CD ⓘ

Disk source * ⓘ

Template default ▼

Disk size *

−

30

+

GiB ▼

Quick create VirtualMachine

VirtualMachine name *

demo-fedora

Start this VirtualMachine after creation

Project

demo

Public SSH key

Not configured [✎](#)

Quick create VirtualMachine

Customize VirtualMachine

Activate Windows

Go to Settings to activate Windows.

Catalog

Customize and create VirtualMachine

Template: Fedora VM

YAML

Overview
YAML
Scheduling
Environment
Network interfaces
Disks
Scripts
Metadata

Add disk ▼

Empty disk (blank)

Create a disk with no contents.

Use existing

Ephemeral disk (Container image)

Any changes are lost upon reboot

Volume

Add a volume already available on the cluster.

Volume snapshot

Add a snapshot available on the cluster to the VirtualMachine.

Clone volume

Clone a volume available on the cluster and add it to the VirtualMachine.

Create VirtualMachine

Cancel

Mount Windows drivers disk

Drive	Interface	Storage class	
Disk	virtio	-	⋮
Disk	virtio	-	⋮

Activate Windows

Go to Settings to activate Windows.

Add disk



disk1

Disk size *



30



GiB



Type

Disk



Hot plug is enabled only for "Disk" and "Lun" types

Interface *

VirtIO



Hot plug is enabled only for "SCSI" interface

StorageClass

 sc-zonea-san



Apply optimized StorageProfile settings

Optimized values Access mode: ReadWriteMany, Volume mode: Block.

Save

Cancel

Project: demo

Overview | YAML | Scheduling | Environment | Network interfaces | **Disks** | Scripts | Metadata

Add disk

Filter Search by name... Mount Windows drivers disk

Name ↑	Source ↓	Size ↓	Drive ↓	Interface ↓	Storage class ↓
cloudinitdisk	Other	-	Disk	virtio	-
disk1	Other	30 GiB	Disk	virtio	sc-zonea-san
rootdisk	PVC (auto import)	30 GiB	Disk	virtio	-

Start this VirtualMachine after creation

Create VirtualMachine Cancel

Activate Windows
Go to Settings to activate V

```
[root@localhost VM-DataProtection]# oc get vm,pods,pvc -n demo
NAME                                     AGE   STATUS   READY
virtualmachine.kubevirt.io/demo-fedora  2m54s Running True

NAME                READY   STATUS   RESTARTS   AGE
pod/virt-launcher-demo-fedora-69cts  1/1    Running  0          110s

NAME                                     STATUS   VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS
persistentvolumeclaim/demo-fedora       Bound    pvc-406d9d57-4a5e-4123-93c1-13070b7b0140  30Gi      RWX             sc-zonea-san
persistentvolumeclaim/dv-demo-fedora-fuchsia-shrew-87  Bound    pvc-311f2f81-1d25-4a9a-b0cb-836b4e702f04  30Gi      RWX             sc-zonea-san
[root@localhost VM-DataProtection]#
```

앱 만들기

VM에 대한 Trident Protect 앱 만들기

이 예에서 데모 네임스페이스에는 하나의 VM이 있으며, 앱을 생성할 때 네임스페이스의 모든 리소스가 포함됩니다.

```
# alias tp='tridentctl-protect'
# tp create app demo-vm --namespaces demo -n demo --dry-run > app.yaml

# cat app.yaml
apiVersion: protect.trident.netapp.io/v1
kind: Application
metadata:
  creationTimestamp: null
  name: demo-vm
  namespace: demo
spec:
  includedNamespaces:
  - namespace: demo
# oc create -f app.yaml -n demo
```

```
[root@localhost VM-DataProtection]# tp get app -n demo
+-----+-----+-----+-----+
| NAME   | NAMESPACES | STATE | AGE |
+-----+-----+-----+-----+
| demo-vm | demo       | Ready | 45s |
+-----+-----+-----+-----+
[root@localhost VM-DataProtection]# █
```

백업을 만들어 앱을 보호하세요

백업 만들기

주문형 백업 만들기

이전에 만든 앱(demo-vm)에 대한 백업을 만들고, 이 백업에는 demo 네임스페이스의 모든 리소스가 포함됩니다. 백업이 저장될 AppVault 이름을 입력하세요.

```
# tp create backup demo-vm-backup-on-demand --app demo-vm --appvault
ontap-s3-appvault -n demo
Backup "demo-vm-backup-on-demand" created.
```

```
[root@localhost VM-DataProtection]# tp get backup -n demo
+-----+-----+-----+-----+-----+
|          NAME          | APP | RECLAIM POLICY | STATE | ERROR | AGE |
+-----+-----+-----+-----+-----+
| demo-vm-backup-on-demand | demo-vm | Retain          | Completed |      | 12m53s |
+-----+-----+-----+-----+-----+
[root@localhost VM-DataProtection]#
```

일정에 따라 백업 만들기

보존할 백업의 세분성과 개수를 지정하여 백업 일정을 만듭니다.

```

# tp create schedule backup-schedule1 --app demo-vm --appvault ontap-
s3-appvault --granularity Hourly --minute 45 --backup-retention 1 -n
demo --dry-run>backup-schedule-demo-vm.yaml
schedule.protect.trident.netapp.io/backup-schedule1 created

#cat backup-schedule-demo-vm.yaml
apiVersion: protect.trident.netapp.io/v1
kind: Schedule
metadata:
  creationTimestamp: null
  name: backup-schedule1
  namespace: demo
spec:
  appVaultRef: ontap-s3-appvault
  applicationRef: demo-vm
  backupRetention: "1"
  dayOfMonth: ""
  dayOfWeek: ""
  enabled: true
  granularity: Hourly
  hour: ""
  minute: "45"
  recurrenceRule: ""
  snapshotRetention: "0"
status: {}
# oc create -f backup-schedule-demo-vm.yaml -n demo

```

```

[root@localhost VM-DataProtection]# tp get schedule -n demo

```

NAME	APP	SCHEDULE	ENABLED	STATE	ERROR	AGE
backup-schedule1	demo-vm	Hourly:min=45	true			9s

```

[root@localhost VM-DataProtection]# tp get backups -n demo

```

NAME	APP	RECLAIM POLICY	STATE	ERROR	AGE
demo-vm-backup-on-demand	demo-vm	Retain	Completed		44m4s
hourly-4c094-20250312144500	demo-vm	Retain	Completed		20m34s

```

[root@localhost VM-DataProtection]#

```

백업에서 복원

VM을 동일한 네임스페이스로 복원

예시에서 백업 demo-vm-backup-on-demand에는 Fedora VM용 demo-app을 사용한 백업이 포함되어 있습니다.

먼저 VM을 삭제하고 PVC, Pod 및 VM 개체가 "demo" 네임스페이스에서 삭제되었는지 확인하세요.

```
[root@localhost VM-DataProtection]# oc get vm,pods,pvc -n demo
NAME                                     AGE   STATUS   READY
virtualmachine.kubevirt.io/demo-fedora  59m   Running  True

NAME                                     READY   STATUS   RESTARTS   AGE
pod/virt-launcher-demo-fedora-69cts     1/1     Running  0           58m

NAME                                     STATUS   VOLUME                                     CAPACITY
CLASS  VOLUMEATTRIBUTESCLASS  AGE
persistentvolumeclaim/demo-fedora      Bound   pvc-406d9d57-4a5e-4123-93c1-13070b7b0140  30Gi
a-san  <unset>                59m
persistentvolumeclaim/dv-demo-fedora-fuchsia-shrew-87  Bound   pvc-311f2f81-1d25-4a9a-b0cb-836b4e702f04  30Gi
a-san  <unset>                59m
[root@localhost VM-DataProtection]#
[root@localhost VM-DataProtection]# oc delete vm demo-fedora -n demo
virtualmachine.kubevirt.io "demo-fedora" deleted
[root@localhost VM-DataProtection]# oc get vm,pods,pvc -n demo
No resources found in demo namespace.
[root@localhost VM-DataProtection]#
```

Activate
Go to Sett

이제 백업-인-플레이스 복원 개체를 만듭니다.

```
# tp create bir demo-fedora-restore --backup demo/demo-vm-backup-on-
demand -n demo --dry-run>vm-demo-bir.yaml

# cat vm-demo-bir.yaml
apiVersion: protect.trident.netapp.io/v1
kind: BackupInplaceRestore
metadata:
  annotations:
    protect.trident.netapp.io/max-parallel-restore-jobs: "25"
  creationTimestamp: null
  name: demo-fedora-restore
  namespace: demo
spec:
  appArchivePath: demo-vm_cc8adc7a-0c28-460b-a32f-
0a7b3d353e13/backups/demo-vm-backup-on-demand_f6af3513-9739-480e-88c7-
4cca45808a80
  appVaultRef: ontap-s3-appvault
  resourceFilter: {}
status:
  postRestoreExecHooksRunResults: null
  state: ""

# oc create -f vm-demo-bir.yaml -n demo
backupinplacerestore.protect.trident.netapp.io/demo-fedora-restore
created
```

```
[root@localhost VM-DataProtection]# tp get bir -n demo
+-----+-----+-----+-----+
| NAME | APPVAULT | STATE | ERROR | AGE |
+-----+-----+-----+-----+
| demo-fedora-restore | ontap-s3-appvault | Completed | | 28m17s |
+-----+-----+-----+-----+
[root@localhost VM-DataProtection]#
```

VM, Pod 및 PVC가 복구되었는지 확인하세요.

```
[root@localhost VM-DataProtection]# oc get vm,pods,pvc -n demo
NAME AGE STATUS READY
virtualmachine.kubevirt.io/demo-fedora 116s Running True

NAME READY STATUS RESTARTS AGE
pod/virt-launcher-demo-fedora-9kfxh 1/1 Running 0 116s

NAME STATUS VOLUME CAPACITY ACCESS MODES STORAGECLASS
persistentvolumeclaim/demo-fedora Bound pvc-6f69a62c-285c-4980-b0dd-6c85baccf346 30Gi RWX sc-zonea-san
persistentvolumeclaim/dv-demo-fedora-fuchsia-shrew-87 Bound pvc-81d82d82-7aca-40fc-8f8f-6e99246e63f8 30Gi RWX sc-zonea-san
[root@localhost VM-DataProtection]#
```

VM을 다른 네임스페이스로 복원

먼저 앱을 복원할 새 네임스페이스를 만듭니다. 이 예에서는 demo2입니다. 그런 다음 백업 복원 개체를 만듭니다.

```
# tp create br demo2-fedora-restore --backup demo/hourly-4c094-20250312154500 --namespace-mapping demo:demo2 -n demo2 --dry-run>vm-demo2-br.yaml

# cat vm-demo2-br.yaml
apiVersion: protect.trident.netapp.io/v1
kind: BackupRestore
metadata:
  annotations:
    protect.trident.netapp.io/max-parallel-restore-jobs: "25"
  creationTimestamp: null
  name: demo2-fedora-restore
  namespace: demo2
spec:
  appArchivePath: demo-vm_cc8adc7a-0c28-460b-a32f-0a7b3d353e13/backups/hourly-4c094-20250312154500_aaa14543-a3fa-41f1-a04c-44b1664d0f81
  appVaultRef: ontap-s3-appvault
  namespaceMapping:
    - destination: demo2
      source: demo
  resourceFilter: {}
status:
  conditions: null
  postRestoreExecHooksRunResults: null
  state: ""

# oc create -f vm-demo2-br.yaml -n demo2
```

```
[root@localhost VM-DataProtection]# tp get br -n demo2
```

NAME	APPVAULT	STATE	ERROR	AGE
demo2-fedora-restore	ontap-s3-appvault	Completed		38m52s

VM, Pod, PVC가 새 네임스페이스 demo2에 생성되었는지 확인합니다.

```
[root@localhost VM-DataProtection]#
[root@localhost VM-DataProtection]# oc get vm,pods,pvc -n demo2
NAME                                     AGE   STATUS   READY
virtualmachine.kubevirt.io/demo-fedora  5m8s  Running  True

NAME                                     READY  STATUS  RESTARTS  AGE
pod/virt-launcher-demo-fedora-c7xc6     1/1    Running  0          5m7s

NAME                                     STATUS  VOLUME                                     CAPACITY  ACCESS MODES  STORAGECLASS
persistentvolumeclaim/demo-fedora       Bound   pvc-4d278ae2-76cc-46f1-bbf8-071ae75e4a82  30Gi      RWX            sc-zonea-san
persistentvolumeclaim/dv-demo-fedora-fuchsia-shrew-87  Bound   pvc-9b96d09c-7226-4ffc-829b-2cee88e7a117  30Gi      RWX            sc-zonea-san
[root@localhost VM-DataProtection]#
```

스냅샷을 사용하여 앱 보호

스냅샷 만들기

주문형 스냅샷 만들기 앱에 대한 스냅샷을 만들고 저장해야 하는 AppVault를 지정합니다.

```
# tp create snapshot demo-vm-snapshot-ondemand --app demo-vm --appvault
ontap-s3-appvault -n demo --dry-run
# cat demo-vm-snapshot-on-demand.yaml
apiVersion: protect.trident.netapp.io/v1
kind: Snapshot
metadata:
  creationTimestamp: null
  name: demo-vm-snapshot-ondemand
  namespace: demo
spec:
  appVaultRef: ontap-s3-appvault
  applicationRef: demo-vm
  completionTimeout: 0s
  volumeSnapshotsCreatedTimeout: 0s
  volumeSnapshotsReadyToUseTimeout: 0s
status:
  conditions: null
  postSnapshotExecHooksRunResults: null
  preSnapshotExecHooksRunResults: null
  state: ""

# oc create -f demo-vm-snapshot-on-demand.yaml
snapshot.protect.trident.netapp.io/demo-vm-snapshot-ondemand created
```

```
[root@localhost VM-DataProtection]#
[root@localhost VM-DataProtection]# oc get vm,pods,pvc -n demo2
NAME          AGE   STATUS  READY
virtualmachine.kubevirt.io/demo-fedora  5m8s  Running True

NAME          READY  STATUS  RESTARTS  AGE
pod/virt-launcher-demo-fedora-c7xc6  1/1    Running  0          5m7s

NAME          STATUS  VOLUME          CAPACITY  ACCESS MODES  STORAGECLASS
persistentvolumeclaim/demo-fedora    Bound  pvc-4d278ae2-76cc-46f1-bbf8-071ae75e4a82  30Gi        RWX          sc-zonea-san
persistentvolumeclaim/dv-demo-fedora-fuchsia-shrew-87  Bound  pvc-9b96d09c-7226-4ffc-829b-2cee88e7a117  30Gi        RWX          sc-zonea-san
[root@localhost VM-DataProtection]#
```

스냅샷 일정 만들기 스냅샷 일정을 만듭니다. 보존할 스냅샷의 세분성과 개수를 지정합니다.

```
# tp create Schedule snapshot-schedule1 --app demo-vm --appvault ontap-
s3-appvault --granularity Hourly --minute 50 --snapshot-retention 1 -n
demo --dry-run>snapshot-schedule-demo-vm.yaml

# cat snapshot-schedule-demo-vm.yaml
apiVersion: protect.trident.netapp.io/v1
kind: Schedule
metadata:
  creationTimestamp: null
  name: snapshot-schedule1
  namespace: demo
spec:
  appVaultRef: ontap-s3-appvault
  applicationRef: demo-vm
  backupRetention: "0"
  dayOfMonth: ""
  dayOfWeek: ""
  enabled: true
  granularity: Hourly
  hour: ""
  minute: "50"
  recurrenceRule: ""
  snapshotRetention: "1"
status: {}

# oc create -f snapshot-schedule-demo-vm.yaml
schedule.protect.trident.netapp.io/snapshot-schedule1 created
```

```
[root@localhost VM-DataProtection]# tp get schedule -n demo
```

NAME	APP	SCHEDULE	ENABLED	STATE	ERROR	AGE
backup-schedule1	demo-vm	Hourly:min=45	true			5d23h
snapshot-schedule1	demo-vm	Hourly:min=50	true			12s

```
[root@localhost VM-DataProtection]#
```

```
[root@localhost VM-DataProtection]# tp get snapshots -n demo
```

NAME	APP	RECLAIM POLICY	STATE	ERROR	AGE
backup-39b67e1c-f875-4045-93df-78634bae9dfb	demo-vm	Delete	Completed		6m29s
demo-vm-snapshot-ondemand	demo-vm	Delete	Completed		21m30s
hourly-51839-20250318135000	demo-vm	Delete	Completed		1m29s

```
[root@localhost VM-DataProtection]#
```

스냅샷에서 복원

스냅샷에서 동일한 네임스페이스로 VM 복원 demo2 네임스페이스에서 VM demo-fedora를 삭제합니다.

```
[root@localhost RedHat]# oc get vm,pvc -n demo
NAME                                     AGE   STATUS   READY
virtualmachine.kubevirt.io/demo-fedora  28h   Running  True

NAME                                     STATUS   VOLUME
DRAGECLASS  VOLUMEATTRIBUTESCLASS  AGE
persistentvolumeclaim/demo-fedora          Bound    pvc-e0d5f79d-dff9-450d-be0e-90ab6880b7af
-zonea-san  <unset>                28h
persistentvolumeclaim/dv-demo-fedora-fuchsia-shrew-87  Bound    pvc-e6f7890a-70c7-4538-9035-5e2e9379511d
-zonea-san  <unset>                28h
[root@localhost RedHat]# oc delete virtualmachine.kubevirt.io/demo-fedora -n demo
virtualmachine.kubevirt.io "demo-fedora" deleted
[root@localhost RedHat]#
```

VM의 스냅샷에서 스냅샷-인-플레이스-복원 개체를 만듭니다.

```
# tp create sir demo-fedora-restore-from-snapshot --snapshot demo/demo-vm-snapshot-ondemand -n demo --dry-run>vm-demo-sir.yaml

# cat vm-demo-sir.yaml
apiVersion: protect.trident.netapp.io/v1
kind: SnapshotInplaceRestore
metadata:
  creationTimestamp: null
  name: demo-fedora-restore-from-snapshot
  namespace: demo
spec:
  appArchivePath: demo-vm_cc8adc7a-0c28-460b-a32f-0a7b3d353e13/snapshots/20250318132959_demo-vm-snapshot-ondemand_e3025972-30c0-4940-828a-47c276d7b034
  appVaultRef: ontap-s3-appvault
  resourceFilter: {}
status:
  conditions: null
  postRestoreExecHooksRunResults: null
  state: ""

# oc create -f vm-demo-sir.yaml
snapshotinplacerestore.protect.trident.netapp.io/demo-fedora-restore-from-snapshot created
```

```
[root@localhost VM-DataProtection]# tp get sir -n demo
```

NAME	APPVault	STATE	ERROR	AGE
demo-fedora-restore-from-snapshot	ontap-s3-appvault	Completed		58m17s

```
[root@localhost VM-DataProtection]#
```

VM과 해당 PVC가 데모 네임스페이스에 생성되었는지 확인합니다.

```
[root@localhost RedHat]# oc get vm,pvc -n demo
```

NAME	AGE	STATUS	READY
virtualmachine.kubevirt.io/demo-fedora	5m17s	Running	True

NAME	STATUS	VOLUME
persistentvolumeclaim/demo-fedora	Bound	pvc-e2f418b0-1b97-40fc-9cb8-943b370d85bc
persistentvolumeclaim/dv-demo-fedora-fuchsia-shrew-87	Bound	pvc-db085154-079f-45ad-9e62-9656e913d01c

```
[root@localhost RedHat]#
```

스냅샷에서 다른 네임스페이스로 VM 복원

이전에 백업에서 복원한 demo2 네임스페이스의 VM을 삭제합니다.

```
[root@localhost VM-DataProtection]# oc get vm,pods,pvc -n demo2
```

NAME	AGE	STATUS	READY
virtualmachine.kubevirt.io/demo-centos	3m12s	Running	True
virtualmachine.kubevirt.io/demo-fedora	3m11s	Running	True

NAME	READY	STATUS	RESTARTS	AGE
pod/virt-launcher-demo-centos-w77rr	1/1	Running	0	3m11s
pod/virt-launcher-demo-fedora-wwdtc	1/1	Running	0	3m11s

NAME	STORAGECLASS	VOLUMEATTRIBUTESCLASS	AGE	STATUS	VOLUME	CAPACITY	ACCESS MODES
persistentvolumeclaim/demo-centos				Bound	pvc-7aafd5e2-d4cf-4af6-a259-c68e016ec6cd	30Gi	RWX
sc-zonea-san	<unset>		3m26s				
persistentvolumeclaim/demo-fedora				Bound	pvc-c1f8145a-56ea-42c5-abb4-6457b6853e1c	30Gi	RWX
sc-zonea-san	<unset>		3m25s				
persistentvolumeclaim/dv-demo-centos-lavender-tortoise-34				Bound	pvc-e82b6d98-6762-4473-9bb8-7e98efcb7987	30Gi	RWX
sc-zonea-san	<unset>		3m20s				
persistentvolumeclaim/dv-demo-fedora-fuchsia-shrew-87				Bound	pvc-ee13cfdc-2921-4129-a214-a0f6b10335cd	30Gi	RWX
sc-zonea-san	<unset>		3m16s				

```
[root@localhost VM-DataProtection]# oc delete virtualmachine.kubevirt.io/demo-fedora -n demo2
```

```
virtualmachine.kubevirt.io "demo-fedora" deleted
```

스냅샷에서 스냅샷 복원 개체를 만들고 네임스페이스 매핑을 제공합니다.

```

# tp create sr demo2-fedora-restore-from-snapshot --snapshot demo/demo-
vm-snapshot-ondemand --namespace-mapping demo:demo2 -n demo2 --dry
-run>vm-demo2-sr.yaml

# cat vm-demo2-sr.yaml
apiVersion: protect.trident.netapp.io/v1
kind: SnapshotRestore
metadata:
  creationTimestamp: null
  name: demo2-fedora-restore-from-snapshot
  namespace: demo2
spec:
  appArchivePath: demo-vm_cc8adc7a-0c28-460b-a32f-
0a7b3d353e13/snapshots/20250318132959_demo-vm-snapshot-
ondemand_e3025972-30c0-4940-828a-47c276d7b034
  appVaultRef: ontap-s3-appvault
  namespaceMapping:
  - destination: demo2
    source: demo
  resourceFilter: {}
status:
  postRestoreExecHooksRunResults: null
  state: ""

# oc create -f vm-demo2-sr.yaml
snapshotrestore.protect.trident.netapp.io/demo2-fedora-restore-from-
snapshot created

```

```

[root@localhost VM-DataProtection]# tp get sr -n demo2

```

NAME	APPVAULT	STATE	ERROR	AGE
demo2-fedora-restore-from-snapshot	ontap-s3-appvault	Completed		15m22s

VM과 해당 PVC가 새 네임스페이스 demo2에 복원되었는지 확인합니다.

```

[root@localhost RedHat]# oc get vm,pvc -n demo2

```

NAME	AGE	STATUS	READY
virtualmachine.kubevirt.io/demo-fedora	29h	Running	True

NAME	DRAGECLASS	VOLUMEATTRIBUTESCLASS	AGE	STATUS	VOLUME
persistentvolumeclaim/demo-fedora				Bound	pvc-35dcd9b2-4fca-486c-af9e-596bc5b4dc15
-zonea-san	<unset>		29h		
persistentvolumeclaim/dv-demo-fedora-fuchsia-shrew-87				Bound	pvc-575a3111-382f-4933-a778-0089fa1ea2af
-zonea-san	<unset>		29h		

```

[root@localhost RedHat]#

```

특정 VM 복원

네임스페이스에서 특정 VM을 선택하여 스냅샷/백업을 생성하고 복원합니다.

이전 예에서는 네임스페이스 내에 단일 VM이 있었습니다. 백업에 전체 네임스페이스를 포함함으로써 해당 VM과 연관된 모든 리소스가 캡처되었습니다. 다음 예제에서는 동일한 네임스페이스에 다른 VM을 추가하고 레이블 선택기를 사용하여 이 새로운 VM에 대한 앱을 만듭니다.

demo 네임스페이스에 새 VM(demo-centos vm)을 만듭니다.

```
[root@localhost VM-DataProtection]# oc get vm,pod,pvc -n demo
NAME                                     AGE      STATUS   READY
virtualmachine.kubevirt.io/demo-centos  2m47s   Running  True
virtualmachine.kubevirt.io/demo-fedora   81m     Running  True

NAME                                     READY   STATUS   RESTARTS   AGE
pod/virt-launcher-demo-centos-2nq6g     1/1     Running  0           113s
pod/virt-launcher-demo-fedora-9kfxh     1/1     Running  0           81m

NAME                                     STATUS   VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS
persistentvolumeclaim/demo-centos       Bound   pvc-ed0f492b-0109-471d-b395-9077ae5f1fa7  30Gi       RWX             sc-zonea-san
persistentvolumeclaim/demo-fedora       Bound   pvc-6f69a62c-285c-4980-b0dd-6c85baccf346  30Gi       RWX             sc-zonea-san
persistentvolumeclaim/dv-demo-centos-lavender-tortoise-34  Bound   pvc-3c01142a-4344-4293-ae67-7d3925c56211  30Gi       RWX             sc-zonea-san
persistentvolumeclaim/dv-demo-fedora-fuchsia-shrew-87      Bound   pvc-81d82d82-7aca-40fc-8f8f-6e99246e63f8   30Gi       RWX             sc-zonea-san
[root@localhost VM-DataProtection]#
```

demo-centos vm 및 관련 리소스에 레이블을 지정합니다.

```
[root@localhost VM-DataProtection]# oc label vm demo-centos category=protect-demo-centos -n demo
virtualmachine.kubevirt.io/demo-centos labeled
[root@localhost VM-DataProtection]# oc label pvc demo-centos category=protect-demo-centos -n demo
persistentvolumeclaim/demo-centos labeled
[root@localhost VM-DataProtection]# oc label pvc dv-demo-centos-lavender-tortoise-34 category=protect-demo-centos -n demo
persistentvolumeclaim/dv-demo-centos-lavender-tortoise-34 labeled
[root@localhost VM-DataProtection]#
```

demo-centos vm 및 pvcs에 레이블이 지정되었는지 확인하세요.

```
[root@localhost VM-DataProtection]# oc get vm --show-labels -n demo
NAME                                     AGE      STATUS   READY   LABELS
demo-centos 6m31s Running True   app-demo-centos,category=protect-demo-centos,kubevirt.io/dynamic-credentials-support=true,vm.kubevirt.io/template.namespace=openshift,vm.kubevirt.io/template.revision=1,vm.kubevirt.io/template.version=v0.31.1,vm.kubevirt.io/template=centos-stream9-server-small
demo-fedora 85m Running True   app-demo-fedora,vm.kubevirt.io/template.namespace=openshift,vm.kubevirt.io/template.revision=1,vm.kubevirt.io/template.version=v0.31.1,vm.kubevirt.io/template=fedora-server-8m11
[root@localhost VM-DataProtection]#
```

```
[root@localhost VM-DataProtection]# oc get pvc --show-labels -n demo
NAME                                     STATUS   VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS   VOLUMEATTRIBUTESCLASS   AGE      LABELS
demo-centos Bound   pvc-ed0f492b-0109-471d-b395-9077ae5f1fa7  30Gi       RWX             sc-zonea-san   <unset>                  7m41s   app.kubernetes.io/component=storage,app.kubernetes.io/managed-by=cdi-controller,app.kubernetes.io/part-of=hyperconverged-cluster,app.kubernetes.io/version=4.17.5,app-containerized-data-importer,category=protect-demo-centos,instancetype.kubevirt.io/default-instancetype=ul.medium,instancetype.kubevirt.io/default-instancetype=ul.medium,instancetype.kubevirt.io/default-preference=centos.stream9,kubevirt.io/created-by=a6a7b49c-669a-4e21-aa78-20743671b284
demo-fedora Bound   pvc-6f69a62c-285c-4980-b0dd-6c85baccf346  30Gi       RWX             sc-zonea-san   <unset>                  86m     app.kubernetes.io/component=storage,app.kubernetes.io/managed-by=cdi-controller,app.kubernetes.io/part-of=hyperconverged-cluster,app.kubernetes.io/version=4.17.5,app-containerized-data-importer,instancetype.kubevirt.io/default-instancetype=ul.medium,instancetype.kubevirt.io/default-preference=fedora,kubevirt.io/created-by=7d5184e9-22f8-4456-9afe-3d1904c430f9
dv-demo-centos-lavender-tortoise-34 Bound   pvc-3c01142a-4344-4293-ae67-7d3925c56211  30Gi       RWX             sc-zonea-san   <unset>                  7m41s   app.kubernetes.io/component=storage,app.kubernetes.io/managed-by=cdi-controller,app.kubernetes.io/part-of=hyperconverged-cluster,app.kubernetes.io/version=4.17.5,app-containerized-data-importer,category=protect-demo-centos,kubevirt.io/created-by=a6a7b49c-669a-4e21-aa78-20743671b284
dv-demo-fedora-fuchsia-shrew-87 Bound   pvc-81d82d82-7aca-40fc-8f8f-6e99246e63f8   30Gi       RWX             sc-zonea-san   <unset>                  86m     app.kubernetes.io/component=storage,app.kubernetes.io/managed-by=cdi-controller,app.kubernetes.io/part-of=hyperconverged-cluster,app.kubernetes.io/version=4.17.5,app-containerized-data-importer,kubevirt.io/created-by=7d5184e9-22f8-4456-9afe-3d1904c430f9
[root@localhost VM-DataProtection]#
```

레이블 선택기를 사용하여 특정 VM(demo-centos)에 대한 앱만 만들기

```
# tp create app demo-centos-app --namespaces 'demo(category=protect-
demo-centos)' -n demo --dry-run>demo-centos-app.yaml

# cat demo-centos-app.yaml

apiVersion: protect.trident.netapp.io/v1
kind: Application
metadata:
  creationTimestamp: null
  name: demo-centos-app
  namespace: demo
spec:
  includedNamespaces:
  - labelSelector:
    matchLabels:
      category: protect-demo-centos
      namespace: demo
status:
  conditions: null

# oc create -f demo-centos-app.yaml -n demo
application.protect.trident.netapp.io/demo-centos-app created
```

```
[root@localhost VM-DataProtection]# tp get app -n demo
```

NAME	NAMESPACES	STATE	AGE
demo-centos-app	demo	Ready	56s
demo-vm	demo	Ready	4h6m

필요에 따라 일정에 따라 백업과 스냅샷을 만드는 방법은 이전에 보여준 것과 동일합니다. 스냅샷이나 백업을 생성하는 데 사용되는 trident-protect 앱에는 네임스페이스의 특정 VM만 포함되어 있으므로, 이를 복원하면 특정 VM만 복원됩니다. 아래는 백업/복원 작업의 예를 보여줍니다.

해당 앱을 사용하여 네임스페이스의 특정 **VM** 백업을 만듭니다.

이전 단계에서는 레이블 선택기를 사용하여 demo 네임스페이스에 centos vm만 포함하도록 앱을 만들었습니다. 이 앱에 대한 백업(이 예에서는 주문형 백업)을 만듭니다.

```
# tp create backup demo-centos-backup-on-demand --app demo-centos-app
--appvault ontap-s3-appvault -n demo
Backup "demo-centos-backup-on-demand" created.
```

NAME	APP	RECLAIM POLICY	STATE	ERROR	AGE
demo-centos-backup-on-demand	demo-centos-app	Retain	Completed		13m22s
demo-vm-backup-on-demand	demo-vm	Retain	Completed		4h19m
hourly-4c094-20250312174500	demo-vm	Retain	Completed		56m17s

특정 VM을 동일한 네임스페이스로 복원 특정 VM(centos)의 백업은 해당 앱을 사용하여 생성되었습니다. 여기에서 백업-인-플레이스-복원이나 백업-복원이 생성되면 해당 VM만 복원됩니다. Centos VM을 삭제합니다.

```
[root@localhost RedHat]# oc get vm,pvc -n demo
NAME                                     AGE      STATUS   READY
virtualmachine.kubevirt.io/demo-centos  4m27s   Running  True
virtualmachine.kubevirt.io/demo-fedora   4m27s   Running  True

NAME                                     STATUS   VOLUME
persistentvolumeclaim/demo-centos       Bound    pvc-e8faeaf8-fc0c-4d92-96de-c83a335a7a17
  sc-zonea-san <unset>                    4m33s
persistentvolumeclaim/demo-fedora       Bound    pvc-e2f418b0-1b97-40fc-9cb8-943b370d85bc
  sc-zonea-san <unset>                    4m33s
persistentvolumeclaim/dv-demo-centos-lavender-tortoise-34 Bound    pvc-66eb7996-1420-4513-a67c-2824f08534da
  sc-zonea-san <unset>                    4m33s
persistentvolumeclaim/dv-demo-fedora-fuchsia-shrew-87   Bound    pvc-db085154-079f-45ad-9e62-9656e913d01c
  sc-zonea-san <unset>                    4m32s
```

```
[root@localhost RedHat]# oc delete virtualmachine.kubevirt.io/demo-centos -n demo
virtualmachine.kubevirt.io "demo-centos" deleted
[root@localhost RedHat]# oc get vm,pvc -n demo
NAME                                     AGE      STATUS   READY
virtualmachine.kubevirt.io/demo-fedora   5m17s   Running  True

NAME                                     STATUS   VOLUME
persistentvolumeclaim/demo-fedora       Bound    pvc-e2f418b0-1b97-40fc-9cb8-943b370d85bc
persistentvolumeclaim/dv-demo-fedora-fuchsia-shrew-87   Bound    pvc-db085154-079f-45ad-9e62-9656e913d01c
[root@localhost RedHat]#
```

demo-centos-backup-on-demand에서 백업 인플레이스 복원을 생성하고 CentOS VM이 다시 생성되었는지 확인합니다.

```
#tp create bir demo-centos-restore --backup demo/demo-centos-backup-on-demand -n demo
BackupInplaceRestore "demo-centos-restore" created.
```

```
[root@localhost RedHat]# tp get bir -n demo
+-----+-----+-----+-----+-----+
| NAME | APPVAULT | STATE | ERROR | AGE |
+-----+-----+-----+-----+-----+
| demo-centos-restore | ontap-s3-appvault | Completed | | 57m9s |
| demo-fedora-restore | ontap-s3-appvault | Completed | | 7d5h |
+-----+-----+-----+-----+-----+
```

```
[root@localhost RedHat]# oc get vm,pvc -n demo
NAME                                     AGE   STATUS   READY
virtualmachine.kubevirt.io/demo-centos  29m   Running  True
virtualmachine.kubevirt.io/demo-fedora   85m   Running  True

NAME                                     STATUS   VOLUME
STORAGECLASS   VOLUMEATTRIBUTESCLASS   AGE
persistentvolumeclaim/demo-centos      Bound   pvc-82954bf7-4a7e-4e0c-9a04-4fa152e1b0ef
  sc-zonea-san   <unset>                 29m
persistentvolumeclaim/demo-fedora      Bound   pvc-e2f418b0-1b97-40fc-9cb8-943b370d85bc
  sc-zonea-san   <unset>                 85m
persistentvolumeclaim/dv-demo-centos-lavender-tortoise-34 Bound   pvc-2a8d4eb5-ed6d-4408-b85d-e218e9a5d4b0
  sc-zonea-san   <unset>                 29m
persistentvolumeclaim/dv-demo-fedora-fuchsia-shrew-87   Bound   pvc-db085154-079f-45ad-9e62-9656e913d01c
  sc-zonea-san   <unset>                 85m
[root@localhost RedHat]#
```

특정 VM을 다른 네임스페이스로 복원 demo-centos-backup-on-demand에서 다른 네임스페이스(demo3)로 백업 복원을 만들고 centos VM이 다시 생성되었는지 확인합니다.

```
# tp create br demo2-centos-restore --backup demo/demo-centos-backup-on-demand --namespace-mapping demo:demo3 -n demo3
BackupRestore "demo2-centos-restore" created.
```

```
[root@localhost RedHat]#
[root@localhost RedHat]# tp get br -n demo3
+-----+-----+-----+-----+-----+
| NAME | APPVAULT | STATE | ERROR | AGE |
+-----+-----+-----+-----+-----+
| demo2-centos-restore | ontap-s3-appvault | Completed | | 52m57s |
+-----+-----+-----+-----+-----+
[root@localhost RedHat]#
```

```
[root@localhost RedHat]#
[root@localhost RedHat]# oc get vm,pvc -n demo3
NAME                                     AGE   STATUS   READY
virtualmachine.kubevirt.io/demo-centos  19m   Running  True

NAME                                     STATUS   VOLUME
STORAGECLASS   VOLUMEATTRIBUTESCLASS   AGE
persistentvolumeclaim/demo-centos      Bound   pvc-0a14e38f-07de-4e09-8f88-14a9a8bb45c2
  sc-zonea-san   <unset>                 19m
persistentvolumeclaim/dv-demo-centos-lavender-tortoise-34 Bound   pvc-d4f9cf2f-264c-4d02-94bf-0db28b427acc
  sc-zonea-san   <unset>                 19m
[root@localhost RedHat]#
```

비디오 데모

다음 비디오는 스냅샷을 사용하여 VM을 보호하는 방법을 보여줍니다.

VM 보호

저작권 정보

Copyright © 2025 NetApp, Inc. All Rights Reserved. 미국에서 인쇄된 본 문서의 어떠한 부분도 저작권 소유자의 사전 서면 승인 없이는 어떠한 형식이나 수단(복사, 녹음, 녹화 또는 전자 검색 시스템에 저장하는 것을 비롯한 그래픽, 전자적 또는 기계적 방법)으로도 복제될 수 없습니다.

NetApp이 저작권을 가진 자료에 있는 소프트웨어에는 아래의 라이선스와 고지사항이 적용됩니다.

본 소프트웨어는 NetApp에 의해 '있는 그대로' 제공되며 상품성 및 특정 목적에의 적합성에 대한 명시적 또는 묵시적 보증을 포함하여(이에 제한되지 않음) 어떠한 보증도 하지 않습니다. NetApp은 대체품 또는 대체 서비스의 조달, 사용 불능, 데이터 손실, 이익 손실, 영업 중단을 포함하여(이에 국한되지 않음), 이 소프트웨어의 사용으로 인해 발생하는 모든 직접 및 간접 손해, 우발적 손해, 특별 손해, 징벌적 손해, 결과적 손해의 발생에 대하여 그 발생 이유, 책임론, 계약 여부, 엄격한 책임, 불법 행위(과실 또는 그렇지 않은 경우)와 관계없이 어떠한 책임도 지지 않으며, 이와 같은 손실의 발생 가능성이 통지되었다 하더라도 마찬가지입니다.

NetApp은 본 문서에 설명된 제품을 언제든지 예고 없이 변경할 권리를 보유합니다. NetApp은 NetApp의 명시적인 서면 동의를 받은 경우를 제외하고 본 문서에 설명된 제품을 사용하여 발생하는 어떠한 문제에도 책임을 지지 않습니다. 본 제품의 사용 또는 구매의 경우 NetApp에서는 어떠한 특허권, 상표권 또는 기타 지적 재산권이 적용되는 라이선스도 제공하지 않습니다.

본 설명서에 설명된 제품은 하나 이상의 미국 특허, 해외 특허 또는 출원 중인 특허로 보호됩니다.

제한적 권리 표시: 정부에 의한 사용, 복제 또는 공개에는 DFARS 252.227-7013(2014년 2월) 및 FAR 52.227-19(2007년 12월)의 기술 데이터-비상업적 품목에 대한 권리(Rights in Technical Data -Noncommercial Items) 조항의 하위 조항 (b)(3)에 설명된 제한사항이 적용됩니다.

여기에 포함된 데이터는 상업용 제품 및/또는 상업용 서비스(FAR 2.101에 정의)에 해당하며 NetApp, Inc.의 독점 자산입니다. 본 계약에 따라 제공되는 모든 NetApp 기술 데이터 및 컴퓨터 소프트웨어는 본질적으로 상업용이며 개인 비용만으로 개발되었습니다. 미국 정부는 데이터가 제공된 미국 계약과 관련하여 해당 계약을 지원하는 데에만 데이터에 대한 전 세계적으로 비독점적이고 양도할 수 없으며 재사용이 불가능하며 취소 불가능한 라이선스를 제한적으로 가집니다. 여기에 제공된 경우를 제외하고 NetApp, Inc.의 사전 서면 승인 없이는 이 데이터를 사용, 공개, 재생산, 수정, 수행 또는 표시할 수 없습니다. 미국 국방부에 대한 정부 라이선스는 DFARS 조항 252.227-7015(b)(2014년 2월)에 명시된 권한으로 제한됩니다.

상표 정보

NETAPP, NETAPP 로고 및 <http://www.netapp.com/TM>에 나열된 마크는 NetApp, Inc.의 상표입니다. 기타 회사 및 제품 이름은 해당 소유자의 상표일 수 있습니다.