



**Trident**  
**Protect**を使用したコンテナアプリケーション  
のデータ保護  
NetApp Solutions

NetApp  
March 04, 2025

# 目次

Trident Protectを使用したコンテナアプリケーションのデータ保護.....	1
Trident PROTECTを使用したOpenShift Container Platformのコンテナアプリケーションのデータ保護 ...	1

# Trident Protectを使用したコンテナアプリケーションのデータ保護

## Trident PROTECTを使用したOpenShift Container Platformのコンテナアプリケーションのデータ保護

作成者：Banu Sundhar、NetApp

リファレンスドキュメントのこのセクションでは、Trident保護を使用してコンテナアプリケーションのスナップショットとバックアップを作成する方法について詳しく説明します。NetApp Trident Protectは、NetApp ONTAPストレージシステムとNetApp Trident CSIストレージプロビジョニングツールを基盤とするステートフルなKubernetesアプリケーションの機能と可用性を強化する、高度なアプリケーションデータ管理機能を提供します。Trident保護では、アプリケーションのSnapshotとバックアップが作成されます。つまり、永続ボリューム内のアプリケーションデータのSnapshotとバックアップが作成されるだけでなく、アプリケーションメタデータのSnapshotとバックアップも作成されます。Trident保護で作成されたSnapshotとバックアップは、次のいずれかのオブジェクトストレージに格納して、あとでリストアできます。

- AWS S3
- Azure BLOBストレージ
- Google Cloud Storage
- ONTAP S3
- StorageGRID
- その他のS3互換ストレージ

Trident保護では、KubernetesモデルのRole-Based Access Control (RBAC；ロールベースアクセス制御)が使用されます。デフォルトでは、Trident protectは、Trident protectと呼ばれる単一のシステムネームスペースとそれに関連付けられたデフォルトのサービスアカウントを提供します。多数のユーザがいる組織や、特定のセキュリティニーズがある組織では、Trident保護のRBAC機能を使用して、リソースやネームスペースへのアクセスをより細かく制御できます。

Trident保護のRBACの詳細については、[を参照してください。](#) "[Trident保護に関するドキュメント](#)"



クラスタ管理者は、デフォルトのTrident保護ネームスペース内のリソースにアクセスできます。また、他のすべてのネームスペース内のリソースにもアクセスできます。SnapshotやバックアップCRSなどのアプリケーションデータ管理カスタムリソース (CRS) をTrident保護ネームスペースに作成することはできません。ベストプラクティスとして、アプリケーションネームスペースにCRSを作成する必要があります。

Trident protectは、ドキュメントに記載されている手順に従ってインストールでき["ここをクリック"](#)ます。このセクションでは、コンテナアプリケーションのデータ保護と、Trident protectを使用したアプリケーションのリストアのワークフローを示します。1.スナップショットの作成(スケジュールに応じてオンデマンドで) 2.Snapshotからのリストア (同じ別のネームスペースへのリストア) 3.バックアップの作成4.バックアップからリストア

アプリケーションのSnapshotとバックアップを作成する前に、Trident保護でオブジェクトストレージを設定してSnapshotとバックアップを格納する必要があります。これにはバケットCRを使用します。バケットCRを作成して設定できるのは管理者だけです。バケットCRは、Trident保護ではAppVaultと呼ばれています。AppVaultオブジェクトは、ストレージバケットの宣言型Kubernetesワークフロー表現です。AppVault CRには、バックアップ、Snapshot、リストア処理、SnapMirrorレプリケーションなど、保護処理でバケットを使用するために必要な設定が含まれています。

この例では、ONTAP S3をオブジェクトストレージとして使用しています。ONTAP S3用のAppVault CRを作成するためのワークフローは次のとおりです。1.ONTAPクラスタのSVMにS3オブジェクトストアサーバを作成します。2.オブジェクトストアサーバにバケットを作成します。3.SVMにS3ユーザを作成します。アクセスキーとシークレットキーは安全な場所に保管してください。4.OpenShiftで、ONTAP S3クレデンシャルを格納するシークレットを作成します。5.ONTAP S3用のAppVaultオブジェクトの作成

- ONTAP S3用のTrident保護AppVaultの構成\*\*
  - ONTAP S3をAppVaultとしてTrident保護を設定するためのサンプルYAMLファイル\*\*\*

```
# 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アプリケーションをインストールするためのサンプルYAMLファイル\*\*\*

```

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]#
```

## Snapshotの作成

### オンデマンドスナップショットの作成

```
# 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  AGE
persistentvolumeclaim/data-postgres-postgresql-0  Bound  pvc-9f89514e-3f2c-41ad-b7a3-792cea503f00  8Gi          RWO            sc-zon          4h40m
ea-nas <unset>
persistentvolumeclaim/postgres-pvc  Bound  pvc-951a9910-9edb-48ae-898a-1aed9aa25dc3  5Gi          RWO            sc-zon          3h47m
ea-nas <unset>

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

スケジュールの作成次のコマンドを使用すると、毎日15時33分にスナップショットが作成され、2つのスナップショットとバックアップが保持されます。

```
# 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]#
```

- YAMLを使用したスケジュールの作成\*\*

```
# 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]#
```

このスケジュールで作成されたSnapshotを表示できます。

```
[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]#
```

ボリュームSnapshotも作成されます。

```
[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

アプリケーションを削除してアプリケーションの損失をシミュレートする

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

## Snapshotから同じ名前スペースへのリストア

```
# 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 | |
+-----+-----+-----+-----+
```

アプリケーションとそのPVCが同じ名前スペースにリストアされます。

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

## Snapshotから別の名前スペースへのリストア

```
# 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 | |
+-----+-----+-----+-----+
```

アプリケーションが新しい名前スペースにリストアされたことがわかります。

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

## バックアップを作成します

### オンデマンドバックアップの作成

```
# 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	

### バックアップスケジュールの作成

上記のリストに含まれる日単位および時間単位のバックアップは、以前に設定したスケジュールに基づいて作成されます。

```
# 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	



データ損失をシミュレートするには、アプリケーションとPVCを削除します。

```
[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-b2b9-7361bfb14fa8  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]#
```

同じネームスペースにリストア#tp create bir postgres-bir --backup postgres/hourly-3f1ee-20250224023300 -n postgres BackupInplaceRestore "postgres-bir"が作成されました。

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

アプリケーションとPVCが同じネームスペースにリストアされます。

```
[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
```

別の名前空間に復元新しい名前空間を作成します。バックアップから新しいネームスペースにリストアします。

```

[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.

## アプリケーションの移行

アプリケーションを別のクラスタにクローニングまたは移行する（クラスタ間クローニングを実行する）には、ソースクラスタでバックアップを作成し、そのバックアップを別のクラスタにリストアします。デスティネーションクラスタにTrident protectがインストールされていることを確認します。

ソースクラスタで、次の図に示す手順を実行します。

```
[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.

ソースクラスタから、コンテキストをデスティネーションクラスタに切り替えます。次に、AppVaultがデスティネーションクラスタコンテキストからアクセス可能であることを確認し、AppVaultのコンテンツをデスティネーションクラスタから取得します。

```
[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)

リストからバックアップパスを指定し、以下のコマンドのようにBackupRestore CRオブジェクトを作成します。

```
# 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	

これで、アプリケーションポッドとPVCがデスティネーションクラスタに作成されたことがわかります。

```
[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	VOLUMEATTRIBUTION
data-postgres-postgresql-0	Bound	pvc-872a5182-601b-4848-b410-fef368337d07	8Gi	RWO	sc-zoneb-san	<unset>
postgres-pvc	Bound	pvc-caf9fa71-76a8-4645-9bb5-2ed72e72948b	5Gi	RWO	sc-zoneb-san	<unset>



## 著作権に関する情報

Copyright © 2025 NetApp, Inc. All Rights Reserved. Printed in the U.S.このドキュメントは著作権によって保護されています。著作権所有者の書面による事前承諾がある場合を除き、画像媒体、電子媒体、および写真複写、記録媒体、テープ媒体、電子検索システムへの組み込みを含む機械媒体など、いかなる形式および方法による複製も禁止します。

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

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

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

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

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

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

## 商標に関する情報

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