



ディザスタ リカバリ

NetApp virtualization solutions

NetApp
July 21, 2026

目次

ディザスタ リカバリ	1
Trident Protect/AppMirror の OpenShift Virtualization ディザスタリカバリの前提条件	1
前提条件	1
NetApp Trident Protect を OpenShift Virtualization 用にインストールする	2
Red Hat OpenShift 仮想化ディザスタリカバリのソース クラスタ設定	2
OpenShift Virtualization のディザスタリカバリ ソース クラスタのセットアップ	2
ONTAP S3を使用してAppVaultを作成する	3
ソース VM の Trident Protect アプリを作成する	5
Trident Protect を使用して VM のスナップショットを作成する	5
Trident Protect を使用して VM のスケジュールされたスナップショットを作成する	6
Red Hat OpenShift 仮想化ディザスタリカバリのデスティネーション クラスタ セットアップ	7
OpenShift Virtualization Disaster Recovery における VM のフェイルオーバー	9
サイト停止なしのOpenShift Virtualization のフェイルオーバー	10
OpenShift Virtualization Disaster Recovery における VM のフェイルバック	12
OpenShift Virtualization ディザスタリカバリのフェイルバック	12

ディザスタ リカバリ

Trident Protect/AppMirror の OpenShift Virtualization ディザスタリカバリの前提条件

このセクションでは、OpenShift VirtualizationワークロードのディザスタリカバリにNetApp Trident ProtectおよびAppMirrorを使用するための前提条件について説明します。これには、OpenShift Virtualization上で実行されているVMの効果的なディザスタリカバリのために、Trident ProtectおよびAppMirrorをセットアップするために必要なクラスタ構成、ストレージ要件、およびインストール手順が含まれます。

前提条件

1. Red Hat OpenShift Virtualization がインストールおよび設定された OpenShift クラスター。インストール手順については、["ドキュメントはこちら"](#)を参照してください。
2. 2つの NetApp ONTAP ストレージシステムにそれぞれ SVM が構成されており、アクティブおよびディザスタリカバリ用の OpenShift クラスターにストレージを提供します。各 SVM には、少なくとも 1 つの管理 LIF と 1 つのデータ LIF が設定され、対応する OpenShift クラスターからアクセス可能である必要があります。
3. 2つの ONTAP システム間でクラスタピアリングと SVM ピアリングが構成されており、ソースクラスターの SVM からターゲットクラスターの SVM へのデータレプリケーションが可能です。クラスタピアリングと SVM ピアリングの設定方法については、["ONTAP クラスタピアリングに関するドキュメントはこちら"](#)を参照してください。
4. NetApp Trident が OpenShift クラスターにインストールおよび構成され、ONTAP SVM から永続ストレージをプロビジョニングします。インストール手順については、["Tridentドキュメントはこちら"](#)を参照してください。
5. ONTAP SVM用に作成されたTridentバックエンド構成、およびそのTridentバックエンドをプロビジョナーとして使用するようにOpenShiftクラスター上で構成されたStorageClass。Tridentバックエンド構成とStorageClassを作成する手順については、["Tridentバックエンド設定に関するドキュメントはこちら"](#)を参照してください。
6. OpenShift クラスターへのクラスター管理者アクセスと、ONTAP ストレージシステムへの管理者アクセスにより、必要な設定を実行します。
7. `tridentctl` および `oc` ツールがインストールされ、\$PATH環境変数に追加された管理ワークステーション。OpenShiftクラスターおよび Trident と連携します。
8. OpenShift クラスター上に、OpenShift Virtualization オペレーターとプロビジョニングされる VM をサポートするための十分なハードウェアリソースが必要です。リソース要件の詳細については、["ドキュメントはこちら"](#)を参照してください。
9. オプションで、OpenShift Virtualization のノード配置ルールを設定して、OpenShift Virtualization のオペレーター、コントローラー、および VM をホストするクラスターノードのサブセットを指定します。ノード配置ルールの設定方法については、["ドキュメントはこちら"](#)を参照してください。
10. OpenShift Virtualization のストレージバックングには、特定の Trident バックエンドからストレージを要求する専用のStorageClassを使用することを推奨します。この Trident バックエンドは、専用の SVM によってサポートされています。これにより、OpenShiftクラスター上の VM ベースのワークロードに提供されるデータに関して、マルチテナンシーのレベルが維持されます。

11. VM は OpenShift Virtualization で利用可能である必要があります。新しい VM をデプロイする方法、または既存の VM を OpenShift Virtualization に移行する方法の詳細については、ドキュメントの適切なセクションを参照してください。

NetApp Trident Protect を OpenShift Virtualization 用にインストールする

NetApp Trident Protectを使用してOpenShift Virtualizationワークロードのディザスタリカバリ機能を有効にするには、OpenShiftクラスタにTrident Protectがインストールされ、構成されている必要があります。Trident Protectは、Helmチャートを使用してインストールできます。



Trident Protect のインストールは、Trident のインストール後に行う必要があります。

インストール手順の詳細については、"[Trident Protect のドキュメントはこちら](#)"を参照してください。

例を表示

```
[root@00-50-56-b5-2b-9b ~]# helm repo add netapp-trident-protect https://netapp.github.io/trident-protect-helm-chart  
[root@00-50-56-b5-2b-9b ~]# helm repo update  
[root@00-50-56-b5-2b-9b ~]# helm install trident-protect-crds netapp-trident-protect/trident-protect-crds --version 100.2602.0 --create-namespace --namespace trident-protect  
[root@00-50-56-b5-2b-9b ~]# helm install trident-protect netapp-trident-protect/trident-protect --set clusterName=ubr-plugin-dr --version 100.2602.0 --create-namespace --namespace trident-protect
```

```
[root@localhost SnapMirror]#  
[root@localhost SnapMirror]# oc get pods -n trident-protect  
NAME                                                    READY   STATUS    RESTARTS   AGE  
autosupportbundle-e9252a48-34a9-4b40-99c2-c00876d962ee-bk2vx  1/1     Running   0           16h  
trident-protect-controller-manager-7b76c8b59f-2rmh2        2/2     Running   0           22h  
[root@localhost SnapMirror]#
```

Red Hat OpenShift 仮想化ディザスタリカバリのソース クラスタ設定

このセクションでは、NetApp Trident ProtectおよびAppMirrorを使用してOpenShift Virtualizationワークロードの災害復旧用にソース クラスタをセットアップする方法について説明します。これには、ONTAP S3を使用したAppVaultの作成、ソースVM用のTrident Protect Applicationの作成、およびTrident Protectを使用したVMのオンデマンドおよびスケジュールされたスナップショットの作成が含まれます。これらの手順は、ソース クラスタが災害復旧ターゲット クラスタにデータをレプリケートするように適切に構成され、災害時にVMの復旧を可能にするために不可欠です。

OpenShift Virtualization のディザスタリカバリ ソース クラスタのセットアップ

以下の手順はすべて、保護対象の仮想マシンが現在実行されているソースクラスタ上で実行されます。

ONTAP S3を使用してAppVaultを作成する

このセクションでは、ONTAP S3 オブジェクトストレージを使用して Trident Protect で AppVault オブジェクトを設定する方法について説明します。このステップで作成した AppVault は、ソース クラスタからレプリケートされたデータを格納するために使用され、そのデータはディザスタリカバリのデスティネーション クラスタでの復旧に使用されます。

oc コマンドと以下に示す yaml ファイルを使用して、ONTAP s3 のシークレットと AppVault カスタムリソースを作成します。必ず trident-protect 名前空間内に作成してください。

例を表示

```
apiVersion: v1
# You can provide the keys either as stringData or base 64 encoded data
stringData:
  accessKeyID: "<access key id as obtained from ONTAP>"
  secretAccessKey: "<secret access key as obtained from ONTAP>"
#data:
  #accessKeyID: <base 64 encoded value of access key>
  #secretAccessKey: <base 64 encoded value of secret access key>
kind: Secret
metadata:
  name: appvault-secret
  namespace: trident-protect
type: Opaque
```

```
apiVersion: protect.trident.netapp.io/v1
kind: AppVault
metadata:
  name: ontap-s3-appvault
  namespace: trident-protect
spec:
  providerConfig:
    s3:
      bucketName: oc-dr
      endpoint: <data lif to use to access S3>
      secure: "false" # Setting this to "false" will disable TLS, which
is not recommended for production environments.
      skipCertValidation: "true" # When TLS is enabled, setting
skipCertValidation to "true" will skip TLS certificate validation,
which is not recommended for production environments.
      providerCredentials:
        accessKeyID:
          valueFromSecret:
            key: accessKeyID
            name: appvault-secret
        secretAccessKey:
          valueFromSecret:
            key: secretAccessKey
            name: appvault-secret
      providerType: OntapS3
```

ONTAP S3ボールドが作成され、利用可能な状態になっていることを確認してください

```
oc create -f app-vault-secret.yaml -n trident-protect
oc create -f app-vault.yaml -n trident-protect
```

```
[root@00-50-56-b5-2b-9b dr]# oc get appvault -n trident-protect
NAME                STATE      ERROR  MESSAGE  AGE
ontap-s3-appvault   Available  2d23h
```

ソース VM の Trident Protect アプリを作成する

ソースVMが配置されている名前空間に、アプリケーションカスタムリソースを作成します。

例を表示

```
# cat app.yaml
apiVersion: protect.trident.netapp.io/v1
kind: Application
metadata:
  creationTimestamp: null
  name: test-source-app
  namespace: test-dr-ns
spec:
  includedNamespaces:
  - namespace: test-dr-ns
```

```
tridentctl-protect create app <source-vm> -n <source-ns> --namespaces
<source-ns>
```

```
[root@00-50-56-b5-2b-9b dr]# oc get applications.protect.trident.netapp.io -n test-dr-ns
NAME                PROTECTION STATE  AGE
test-source-app     Partial  25h
```

アプリのカスタムリソースに含まれる名前空間内のVMとそのPVCを以下に示します。

```
[root@00-50-56-b5-2b-9b dr]# oc get vm,pods,pvc -n test-dr-ns
NAME                AGE  STATUS  READY
virtualmachine.kubevirt.io/test-dr-vm1  25h  Running  True

NAME                READY  STATUS  RESTARTS  AGE
pod/virt-launcher-test-dr-vm1-ns9qq  1/1    Running  0          25h

NAME                STATUS  VOLUME                                     CAPACITY  ACCESS MODES  STORAGECLASS  VOLUMEATTRIBUTESCLASS  AGE
persistentvolumeclaim/test-dr-vm1  Bound  pvc-43b51d12-ce76-4063-bd59-6e74588ee266  34144990004  RWX          sc-nas        <unset>                25h
```

Trident Protect を使用して VM のスナップショットを作成する

Trident Protect CLI または API を使用して、VM のオンデマンド スナップショットを作成できます。このスナップショットは、前の手順で作成した AppVault に保存され、ディザスタリカバリのデスティネーションクラ

スタでの復旧に使用できます。

例を表示

```
# snapshot.yaml
apiVersion: protect.trident.netapp.io/v1
kind: Snapshot
metadata:
  name: test-source-dr-init-snapshot
  namespace: test-dr-ns
spec:
  applicationRef: test-source-app
  appVaultRef: ontap-s3-appvault
  reclaimPolicy: Delete
```

```
[root@00-50-56-b5-2b-9b dr]# oc get snapshot -n test-dr-ns
NAME                                APP                RECLAIM POLICY    STATE        ERROR    AGE
test-source-dr-init-snapshot        test-source-app    Delete             Completed    25h
```

Trident Protect を使用して VM のスケジュールされたスナップショットを作成する

アプリケーションカスタムリソースと同じ名前空間にスケジュールカスタムリソースを作成することで、スケジュールされたスナップショットを作成することもできます。スケジュールは、指定されたスケジュールで VM のスナップショットを自動的に作成し、AppVault に保存します。

例を表示

```
apiVersion: protect.trident.netapp.io/v1
kind: Schedule
metadata:
  name: appmirror-sched1
spec:
  appVaultRef: ontap-s3-appvault
  applicationRef: test-source-app
  backupRetention: "0"
  enabled: true
  granularity: Custom
  recurrenceRule: |-
    DTSTART:20240901T000200Z
    RRULE:FREQ=MINUTELY;INTERVAL=5
  snapshotRetention: "2"
```

Red Hat OpenShift 仮想化ディザスタリカバリのデスティネーション クラスタ セットアップ

このセクションでは、NetApp Trident ProtectおよびAppMirrorを使用して、OpenShift Virtualizationワークロードのディザスタリカバリ用にターゲットクラスタをセットアップする方法について説明します。これには、ONTAP S3を使用したAppVaultの作成、およびデスティネーションVM用の新しいネームスペースの作成が含まれます。これらの手順は、ソースクラスタからレプリケートされたデータを受信するようにデスティネーションクラスタを適切に構成し、災害時にVMをリカバリできるようにするために不可欠です。

1. デスティネーション クラスタに AppVault オブジェクトを作成します。これはソースと同じ S3 を指すようにする必要があります。そうすることで、ソース クラスタがスナップショットを配置した S3 からスナップショットを取得できます。

```

apiVersion: v1
# You can provide the keys either as stringData or base 64 encoded
data
stringData:
  accessKeyID: "<access key id as obtained from ONTAP>"
  secretAccessKey: "<secret access key as obtained from ONTAP>"
#data:
  #accessKeyID: <base 64 encoded value of access key>
  #secretAccessKey: <base 64 encoded value of secret access key>
kind: Secret
metadata:
  name: appvault-secret
  namespace: trident-protect
type: Opaque

```

```

apiVersion: protect.trident.netapp.io/v1
kind: AppVault
metadata:
  name: ontap-s3-appvault
  namespace: trident-protect
spec:
  providerConfig:
    s3:
      bucketName: oc-dr
      endpoint: <data lif to use to access S3>
      secure: "false" # Setting this to "false" will disable TLS,
which is not recommended for production
      skipCertValidation: "true" # When TLS is enabled, setting
skipCertValidation to "true" will skip TLS certificate validation,
which is not recommended for production environments.
  providerCredentials:
    accessKeyID:
      valueFromSecret:
        key: accessKeyID
        name: appvault-secret
    secretAccessKey:
      valueFromSecret:
        key: secretAccessKey
        name: appvault-secret
  providerType: OntapS3

```

```
oc create -f app-vault-secret.yaml -n trident-protect
oc create -f app-vault.yaml -n trident-protect
```

```
[root@00-50-56-b5-01-5b dr]# oc get appvault -A
```

NAMESPACE	NAME	STATE	ERROR	MESSAGE	AGE
trident-protect	ontap-s3-appvault	Available			29h

2. フェイルオーバーした仮想マシンをホストするための新しい名前空間をデスティネーション クラスタに作成します。ステップ1で作成したAppVaultと新しい名前空間を参照するアプリをデスティネーション クラスタに作成します。このアプリは、ソースとデスティネーション クラスタ間のAppMirror関係を確立し、ソース クラスタからデスティネーション クラスタへのデータのレプリケーションを管理するために使用されます。

例を表示

```
# cat app.yaml
apiVersion: protect.trident.netapp.io/v1
kind: Application
metadata:
  creationTimestamp: null
  name: test-dest-app
  namespace: test-dr-ns-dest
spec:
  includedNamespaces:
  - namespace: test-dr-ns-dest
```

```
tridentctl-protect create app <dest-app-name> -n <dest-ns>
--namespaces <dest-ns>
```

OpenShift Virtualization Disaster Recovery における VM のフェイルオーバー

このセクションでは、NetApp Trident Protect と AppMirror を使用して、ソース クラスターから災害復旧ターゲット クラスターへの VM のフェイルオーバーを実行する方法について説明します。これは、フェイルオーバープロセスのテストや、ソースクラスタに対する計画的なメンテナンスの実施に役立ちます。これらの手順に従うことで、災害復旧イベント後も、VMがすべての変更を保持したまま災害復旧ターゲットクラスタ上で実行されていることを確認できます。

サイト停止なしのOpenShift Virtualization のフェイルオーバー

このセクションでは、サイト停止なしでソースクラスタから VM のフェイルオーバーを実行する方法について説明します。これは、フェイルオーバープロセスのテストや、ソースクラスタに対する計画的なメンテナンスの実施に役立ちます。+

サイト停止なしでフェイルオーバーを実行するには、まず前のセクションで説明したように、ソース クラスタとターゲット クラスタがディザスタリカバリ用に設定されていることを確認してください。次に、以下の手順を実行してください。

1. ソースクラスターとターゲットクラスター間の AppMirror 関係を確立します。AppMirror 関係が確立されると、最新のスナップショットがデスティネーション ネームスペースに転送されます。デスティネーション ネームスペース内の VM に対して PVC が作成されますが、VM ポッドはまだデスティネーション ネームスペース内に作成されていません。

例を表示

```
# application-mirror-relationship.yaml
apiVersion: protect.trident.netapp.io/v1
kind: AppMirrorRelationship
metadata:
  name: amr1
spec:
  desiredState: Established
  destinationAppVaultRef: ontap-s3-appvault
  destinationApplicationRef: test-dest-app
  namespaceMapping:
  - destination: test-dr-ns-dest
    source: test-dr-ns
  recurrenceRule: |-
    DTSTART:20240901T000200Z
    RRULE:FREQ=MINUTELY;INTERVAL=5
  sourceAppVaultRef: ontap-s3-appvault
  sourceApplicationName: test-source-app
  sourceApplicationUID: "<application-uid-of-the-source-app>"
  storageClassName: "sc-nas"
```

```
[root@00-50-56-b5-2b-9b dr]# oc get amr -n test-dr-ns
```

NAME	DESIRED STATE	STATE	ERROR	AGE
amr1	Established	Establishing		115s

```
[root@00-50-56-b5-2b-9b dr]# oc get amr -A
```

NAMESPACE	NAME	DESIRED STATE	STATE	ERROR	AGE
test-dr-ns	amr1	Established	Established		94s



AppMirror関係は、デスティネーション クラスタの CLI から作成する必要があります。作成時には、希望する状態を「確立済み」に設定する必要があります。これにより、最新のスナップショットデータがデスティネーション クラスタにレプリケートされることが保証されます。

2. AppMirror 関係をプロモートします。関係の目的の状態を「昇格済み」に変更して、デスティネーション ネームスペースに VM を作成します。VM は引き続きソース ネームスペースで実行されています。最新のスナップショット データはデスティネーション クラスタにレプリケートされ、VM はデスティネーション クラスタで起動されます。この方法を使用すると、ソース クラスタ全体を停止させることなく、フェイルオーバーのテストと VM の可用性の維持が可能になります。

例を表示

```
oc patch amr amr1 -n test-dr-ns-dest --type=merge -p
'{"spec":{"desiredState":"Promoted"}}'
```

```
[root@00-50-56-b5-01-5b dr]# oc get amr -n test-dr-ns-dest
```

NAME	DESIRED STATE	STATE	ERROR	AGE
amr2	Promoted	Promoted		25h

```
oc get pod, pvc,vm -n test-dr-ns-dest
oc get pod, pvc,vm -n test-dr-ns
```

OpenShift Virtualization Disaster Recovery における VM のフェイルバック

このセクションでは、NetApp Trident ProtectおよびAppMirrorを使用したOpenShift Virtualization災害復旧における、VMのフェイルバック プロセスに関する情報を提供します。災害復旧先となるデスティネーション クラスタに切り替えるフェイルオーバー処理を実行したあと、フェイルバック プロセスを使用してレプリケーション方向を反転し、元のソース クラスタをVMのプライマリ クラスタとして復元できます。これには、フェイルオーバー期間中にVMに加えられた変更を元のソース クラスタに再同期したあと、レプリケーション関係を逆方向にプロモートする処理が含まれます。これらの手順に従うことで、災害復旧イベントの発生後に、すべての変更を維持した状態で元のソース クラスタ上でVMを実行できます。

OpenShift Virtualization ディザスタリカバリのフェイルバック

Trident Protect を使用すると、次の一連の操作を使用して、フェイルオーバー操作後に「フェイルバック」を実現できます。このワークフローでは、元のレプリケーション方向を復元するために、Trident Protect は、レプリケーションの方向を反転する前に、アプリケーションの変更を元のソース アプリケーションに複製（再同期）します。

このプロセスは、デスティネーションへのフェイルオーバーを完了した関係から開始され、次の手順が含まれます：

1. フェイルオーバー状態から開始します。
2. レプリケーション関係を逆方向に再同期する

フェールオーバーされたレプリケーション関係を逆再同期すると、デスティネーション アプリケーションがソース アプリケーションになり、ソースがデスティネーションになります。フェイルオーバー中にデスティネーション アプリケーションに加えられた変更は保持されます。

- a. 元のデスティネーション クラスターで、AppMirrorRelationship CRを削除します。これにより、デスティネーションがソースになります。新しいデスティネーション クラスターに保護スケジュールが残っている場合は、それらを削除します。

例を表示

```
[root@00-50-56-b5-01-5b dr]# oc delete -f app-mirror-relationship.yaml -n test-dr-ns-dest
appmirrorrelationship.protect.trident.netapp.io "amr2" deleted from test-dr-ns-dest namespace
```

- b. フェイルオーバー期間中に加えられた変更を記録するために、新しいソースクラスター（元のデスティネーション クラスター）にSnapshotを作成します。これにより、再同期処理中にすべての変更が元のソースクラスターに確実にレプリケートされます。

例を表示

```
# snapshot.yaml
apiVersion: protect.trident.netapp.io/v1
kind: Snapshot
metadata:
  name: test-source-dr-init-snapshot-failback
  namespace: test-dr-ns
spec:
  applicationRef: test-dest-app
  appVaultRef: ontap-s3-appvault
  reclaimPolicy: Delete
```

```
[root@00-50-56-b5-01-5b dr]# cat failback-snapshot.yaml
apiVersion: protect.trident.netapp.io/v1
kind: Snapshot
metadata:
  name: test-source-dr-init-snapshot-failback
  namespace: test-dr-ns-dest
spec:
  applicationRef: test-dest-app
  appVaultRef: ontap-s3-appvault
  reclaimPolicy: Delete
```

```
[root@00-50-56-b5-01-5b dr]# oc get snapshot -n test-dr-ns-dest
```

NAME	APP	RECLAIM POLICY	STATE	ERROR	AGE
test-source-dr-init-snapshot-failback	test-dest-app	Delete	Running		10s

```
[root@00-50-56-b5-01-5b dr]# oc get snapshot -n test-dr-ns-dest
```

NAME	APP	RECLAIM POLICY	STATE	ERROR	AGE
test-source-dr-init-snapshot-failback	test-dest-app	Delete	Completed		8m50s
test-source-dr-init-snapshot-failback-2	test-dest-app	Delete	Completed		32s

- c. 新しいソースクラスター（元のデスティネーション クラスター）から元のソースクラスターへのレプリケーション関係を設定し、逆方向の値を構成します。

例を表示

application-mirror-relationship.yaml

```
apiVersion: protect.trident.netapp.io/v1
kind: AppMirrorRelationship
metadata:
  name: amr1
spec:
  desiredState: Established
  destinationAppVaultRef: ontap-s3-appvault
  destinationApplicationRef: dr-source-vm-failback
  namespaceMapping:
    - destination: test-dr-ns
      source: test-dr-ns-dest
  recurrenceRule: |-
    DTSTART:20240901T000200Z
    RRULE:FREQ=MINUTELY;INTERVAL=5
  sourceAppVaultRef: ontap-s3-appvault
  sourceApplicationName: test-dest-app
  sourceApplicationUID: "uid of the app in the original source
cluster"
  storageClassName: "sc-nas"
```

```
[root@00-50-56-b5-2b-9b dr]# oc get amr -n test-dr-ns
NAME      DESIRED STATE    STATE              ERROR    AGE
amr1      Established      Establishing       115s

[root@00-50-56-b5-2b-9b dr]# oc get amr -A
NAMESPACE   NAME    DESIRED STATE    STATE              ERROR    AGE
test-dr-ns  amr1    Established      Established        94s
```

- d. AppMirrorRelationship CRのdesiredStateフィールドを「Promoted」に更新して、新しいソース クラスタ（元のデスティネーション）から元のソース クラスタへのレプリケーション関係を昇格させます。

例を表示

```
[root@00-50-56-b5-2b-9b dr]# oc get vm -n test-dr-ns
No resources found in test-dr-ns namespace.
[root@00-50-56-b5-2b-9b dr]# oc patch amr amr1 -n test-dr-ns --type=merge -p '{"spec":{"desiredState":"Promoted"}}'
appmirrorrelationship.protect.trident.netapp.io/amr1 patched
```

これで、VMが元のソースクラスタで実行されていることが確認できます。フェイルバック処理が完了しました。これで、元のデスティネーションクラスタのリソースをクリーンアップできます。フェイルバック処理後もディザスタリカバリ機能を維持する場合は、新しいレプリケーション関係を設定することもできます。



通常の再同期操作は実行しないでください。フェイルオーバー手順中にデスティネーションクラスタに書き込まれたデータが破棄されます。

著作権に関する情報

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

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

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

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

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

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

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

商標に関する情報

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