Google Cloud NetApp Volumes

Trident

NetApp
October 29, 2026

This PDF was generated from https://docs.netapp.com/ja-jp/trident-2606/trident-use/gcnv.html on October 29, 2026. Always check docs.netapp.com for the latest.

目次

Google Cloud NetApp Volumes . . . 1
- Google Cloud NetApp Volumes を設定 . . . 1
  - 概要 . . . 1
  - 設定の準備 . . . 1
  - NASストレージの設定 . . . 2
- Google Cloud NetApp Volumes の SAN ワークロード向け設定 . . . 5
  - 概要 . . . 5
  - Flex Unified ストレージプール . . . 5
  - Trident SANバックエンドを設定する . . . 6
  - StorageClass を作成する . . . 6
  - ブロックボリュームのプロビジョニング . . . 7
  - ブロックボリュームの動作 . . . 8
  - アクセスモード . . . 8
  - volumeMode の動作 . . . 9
  - サポートされている操作 . . . 9
  - 余分な GiB のオーバープロビジョニング動作 . . . 9
  - Podの例 . . . 10
  - アタッチおよびマウント動作 . . . 11
- Google Cloud NetApp Volumes バックエンドを構成する準備をする . . . 11
  - NFSまたはSMBボリュームの前提条件 . . . 11
- Google Cloud NetApp Volumes バックエンドの設定オプションと例 . . . 12
  - バックエンド構成オプション . . . 12
  - ボリュームのプロビジョニング オプション . . . 13
  - 構成例 . . . 13
  - 次の手順 . . . 19
  - ストレージクラスの定義 . . . 20
- Google Cloud NetApp Volumes の自動階層化を設定する . . . 23
  - 概要 . . . 23
  - 概念 . . . 23
  - 構成モデル . . . 23
  - Trident 26.06 でサポートされている機能 . . . 24
  - Trident 26.06 でサポートされていない機能 . . . 24
  - バックエンド構成パラメータ . . . 24
  - PersistentVolumeClaim アノテーションを使用したボリュームレベルのオーバーライド . . . 24
  - 動作と制限事項 . . . 25
- Google Cloud NetApp Volumes ONTAP モードストレージプールの設定 . . . 25
  - 仕組み . . . 26
  - サポート対象範囲とサポート対象外範囲 . . . 26
  - 前提条件 . . . 26

バックエンドの設定 27

Google Cloud NetApp Volumes

Google Cloud NetApp Volumes を設定

Google Cloud NetApp Volumes を Trident のバックエンドとして設定し、Kubernetes ワークロード用のストレージをプロビジョニングできます。

概要

Trident は、NAS（NFS および SMB）とブロック（iSCSI）ワークロードの両方で Google Cloud NetApp Volumes をサポートしています。

- NASワークロードは `google-cloud-netapp-volumes`バックエンドを使用します
- ブロック（iSCSI）ワークロードは `google-cloud-netapp-volumes-san`バックエンドを使用します

NASボリュームはファイルベースのストレージを提供し、NFSまたはSMBプロトコルを使用してアクセスされます。これらのボリュームは、複数のポッドまたはノード間での共有アクセスをサポートします。

ブロックボリュームは生のブロックストレージを提供し、Kubernetesノードに接続されたiSCSIデバイスとしてアクセスされます。これらのボリュームは、アプリケーションがブロックレベルのアクセスを必要とする場合に使用されます。

これは以下の環境に適用されます：

- Trident 26.02以降
- Google Kubernetes Engine（GKE）または Red Hat OpenShift
- Google Cloud NetApp Volumes ストレージプール

ブロック（iSCSI）ストレージを構成するには、"ブロックストレージ（iSCSI）の設定"を参照してください。

設定の準備

Cloud IDを使用すると、Kubernetesワークロードは静的な認証情報を使用する代わりに、ワークロードIDとして認証することでGoogle Cloudリソースにアクセスできるようになります。

Google Cloud NetApp Volumes でクラウド ID を使用するには、以下が必要です：

- Google Kubernetes Engine（GKE）を使用してデプロイされたKubernetesクラスター
- GKEクラスタでワークロードIDが有効になっており、ノードプールでメタデータサーバーが有効になっています。
- Google Cloud NetApp Volumes 管理者ロール（`roles/netapp.admin`）または同等のカスタムロールを持つ Google Cloud サービスアカウント
- クラウド プロバイダが `GCP`に設定され、クラウドID注釈が構成された状態でTridentがインストールされています

Trident オペレータ

Trident オペレータを使用して Trident をインストールするには、 `tridentorchestrator_cr.yaml`を編集します：

```
apiVersion: trident.netapp.io/v1
kind: TridentOrchestrator
metadata:
  name: trident
spec:
  namespace: trident
  cloudProvider: "GCP"
  cloudIdentity: "iam.gke.io/gcp-service-account: cloudvolumes-admin-
sa@mygcpproject.iam.gserviceaccount.com"
```

Helm

Helmを使用してTridentをインストールする際に、クラウド プロバイダとクラウドIDを設定します：

```
helm install trident trident-operator-100.6.0.tgz \
  --set cloudProvider=GCP \
  --set cloudIdentity="iam.gke.io/gcp-service-account: cloudvolumes-
admin-sa@mygcpproject.iam.gserviceaccount.com"
```

tridentctl

クラウド プロバイダとクラウドIDを指定してTridentをインストールします：

```
tridentctl install \
  --cloud-provider=GCP \
  --cloud-identity="iam.gke.io/gcp-service-account: cloudvolumes-admin-
sa@mygcpproject.iam.gserviceaccount.com" \
  -n trident
```

NASストレージの設定

Google Cloud NetApp Volumes UNIFIEDストレージプールの場合、Tridentはボリューム操作中にUNIFIED固有のネーミングおよび検証ルールを適用します。

ボリュームを検索する際、Tridentは複数の互換性のあるボリューム名のバリアント（ハイフン形式やアンダースコア形式など）を評価することで、インポートと検出の信頼性を向上させることができます。

ドライバの詳細

Tridentは、Google Cloud NetApp VolumesからNASストレージをプロビジョニングするための `google-cloud-

netapp-volumes`ドライバーを提供します。

このドライバは、以下のアクセスモードをサポートしています。

- ReadWriteOnce（RWO）
- ReadOnlyMany（ROX）
- ReadWriteMany（RWX）
- ReadWriteOncePod（RWOP）

Driver	プロトコル	volumeMode	サポートされているアクセスモード	サポートされているファイルシステム
google-cloud-netapp-volumes	NFS SMB	Filesystem	RWO、ROX、RWX、RWOP	nfs, smb

Trident NASバックエンドを設定する

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: gcnv-nas
  namespace: trident
spec:
  version: 1
  storageDriverName: google-cloud-netapp-volumes
  projectNumber: "<project-number>"
  location: "<region>"
  sdkTimeout: "600"
  storage:
  - labels:
      cloud: gcp
    network: "<vpc-network>"
```

NAS ボリュームのプロビジョニング

NASボリュームは、 `google-cloud-netapp-volumes`バックエンドを使用してプロビジョニングされ、NFSおよびSMBプロトコルをサポートします。

StorageClass（**NFS**ボリューム用）

NFSボリュームをプロビジョニングするには、 `nasType`を `nfs`に設定します。

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: gcnv-nfs
```

```
provisioner: csi.trident.netapp.io
parameters:
  backendType: "google-cloud-netapp-volumes"
  trident.netapp.io/nasType: "nfs"
allowVolumeExpansion: true
```

StorageClass（SMBボリューム用）

SMB ボリュームをプロビジョニングするには、 `nasType`を `smb`に設定してクレデンシャルを指定します。

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: gcnv-smb
provisioner: csi.trident.netapp.io
parameters:
  backendType: "google-cloud-netapp-volumes"
  trident.netapp.io/nasType: "smb"
  csi.storage.k8s.io/node-stage-secret-name: "smbcreds"
  csi.storage.k8s.io/node-stage-secret-namespace: "default"
allowVolumeExpansion: true
```

PersistentVolumeClaim の例（RWX）

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: gcnv-nas-rwx
spec:
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 100Gi
  storageClassName: gcnv-nfs
```

PersistentVolumeClaim の例（RWO）

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: gcnv-nas-rwo
spec:
```

```
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 100Gi
  storageClassName: gcnv-nfs
```

NASボリュームは `volumeMode: Filesystem`を使用します。

Google Cloud NetApp Volumes の SAN ワークロード向け設定

iSCSI プロトコルを使用して Google Cloud NetApp Volumes からブロックストレージボリュームをプロビジョニングするように Trident を設定できます。SAN ボリュームは、`google-cloud-netapp-volumes-san`ストレージドライバを使用して Flex Unified ストレージプールからプロビジョニングされます。

このドライバはブロックワークロード専用であり、NASプロトコルはサポートしていません。

`google-cloud-netapp-volumes-san`バックエンドは、iSCSIブロックボリュームのプロビジョニングに必要です。 `google-cloud-netapp-volumes`バックエンドはNASプロトコルのみをサポートしており、SANワークロードには使用できません。

概要

Trident は、 `google-cloud-netapp-volumes-san`ドライバを使用して、Google Cloud NetApp Volumes SAN（iSCSI）ワークロードをサポートしています。

SANボリュームはFlex Unifiedストレージプールからプロビジョニングされ、iSCSIブロックデバイスとしてKubernetesノードに提示されます。

これは以下の環境に適用されます：

- Trident 26.02以降
- Google Kubernetes Engine（GKE）または Red Hat OpenShift
- Google Cloud NetApp Volumes Flex統合ストレージプール
- iSCSIベースのワークロード

Flex Unified ストレージプール

Flex Unifiedストレージプールは、iSCSIプロトコルを使用してブロックストレージを提供し、SANプロビジョニングに必要です：

- Flex Unified REGIONAL プールがサポートされています。

- Flex Unified ZONAL プールは、Trident 26.02.1 以降でサポートされています。
- SANワークロードでは、*Flex*サービスレベルのみがサポートされます。

Trident SANバックエンドを設定する

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: gcnv-san
  namespace: trident
spec:
  version: 1
  storageDriverName: google-cloud-netapp-volumes-san
  projectNumber: "<project-number>"
  location: "<region>"
  sdkTimeout: "600"
  storage:
  - labels:
      cloud: gcp
      performance: flex
    network: "<vpc-network>"
    serviceLevel: Flex
```

StorageClass を作成する

SANバックエンドを設定したら、 `google-cloud-netapp-volumes-san`ドライバを参照するStorageClassを作成します。

ファイルシステムタイプは、バックエンドではなく StorageClass で定義されます。

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: gcnv-san
provisioner: csi.trident.netapp.io
parameters:
  backendType: "google-cloud-netapp-volumes-san"
  fsType: "ext4"
allowVolumeExpansion: true
```

サポートされているファイルシステムの種類：

- `ext4`（デフォルト）
- `ext3`

- `xfs`

> SAN ドライバーは Flex サービス レベルのみをサポートし、 `exportRule`、 `unixPermissions`、 `nasType`、 `snapshotDir`、 `nfsMountOptions`、または階層化関連の設定などの NAS 固有のバックエンド パラメーターは使用しません。

ブロックボリュームのプロビジョニング

ReadWriteOnce（RWO）

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: gcnv-san-rwo
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 100Gi
  storageClassName: gcnv-san
```

ReadWriteOncePod（RWOP）

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: gcnv-san-rwop
spec:
  accessModes:
    - ReadWriteOncePod
  resources:
    requests:
      storage: 100Gi
  storageClassName: gcnv-san
```

ReadOnlyMany（ROX）

ROXの一般的なパターンは、既存のReadWriteOnceボリュームをクローンし、そのクローンを読み取り専用としてマウントすることです。

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
```

```
  name: gcnv-san-rox
spec:
  accessModes:
    - ReadOnlyMany
  resources:
    requests:
      storage: 100Gi
  storageClassName: gcnv-san
  dataSource:
    kind: PersistentVolumeClaim
    name: gcnv-san-rwo
```

ReadWriteMany（RWX）— 生ブロックのみ

ReadWriteMany は、 `volumeMode: Block`の場合にのみサポートされます。

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: gcnv-san-raw-rwx
spec:
  accessModes:
    - ReadWriteMany
  volumeMode: Block
  resources:
    requests:
      storage: 100Gi
  storageClassName: gcnv-san
```

ブロックボリュームの動作

ブロックボリュームはiSCSI LUNとしてプロビジョニングされ、ブロックデバイスとしてKubernetesノードに提供されます。

ブロックボリューム：

- iSCSIプロトコルを使用する
- ファイルシステムと生ブロックの表示をサポート
- Tridentによってアタッチおよび管理される
- 複数のKubernetesアクセスモードをサポート

アクセスモード

Trident によってプロビジョニングされたブロックボリュームは、以下のアクセスモードをサポートします：

- `ReadWriteOnce`（RWO）
- `ReadOnlyMany`（ROX）
- `ReadWriteOncePod`（RWOP）
- `ReadWriteMany`（RWX）は、以下の場合にのみサポートされます。 `volumeMode: Block`

volumeMode の動作

`volumeMode`フィールドは、ブロックボリュームの公開方法を制御します。

- `Filesystem` Trident はボリュームをフォーマットしてマウントします。
- `Block` Trident はデバイスを接続し、rawブロックデバイスとして公開します。

サポートされている操作

`google-cloud-netapp-volumes-san`ドライバーを使用してプロビジョニングされたブロックボリュームがサポートする機能：

- 作成
- 削除
- クローン
- Snapshot
- サイズ変更
- インポート

余分な GiB のオーバープロビジョニング動作

Google Cloud NetApp Volumes ブロックボリュームには、内部メタデータのオーバーヘッドが含まれます。このオーバーヘッドにより、カーネルから見えるデバイスサイズは、プロビジョニングされた容量と比較して小さくなります。

テスト結果：

- 初回作成時に約300 KiBのオーバーヘッドが発生します。
- サイズ変更後、最大約 107 MiB のオーバーヘッドが発生します。

Google Cloud NetApp Volumes はGiB単位の割り当てのみを受け入れるため、Trident は以下の方法により、使用可能なデバイスサイズが常にPVCの要求を満たすか、それを超えることを保証します：

- 要求されたサイズを次の整数GiBに切り上げます
- 1 GiB のバッファを追加する

例：

- PVCリクエスト：100 GiB
- Google Cloud NetApp Volumes でプロビジョニングされたサイズ：101 GiB
- アプリケーションから見える使用可能な容量：少なくとも100 GiB

Podの例

ファイルシステムマウントされたブロックボリューム（**RWO**）

```
apiVersion: v1
kind: Pod
metadata:
  name: app-rwo
spec:
  containers:
  - name: app
    image: ubuntu:22.04
    command: ["sleep", "infinity"]
    volumeMounts:
    - name: data
      mountPath: /mnt/data
  volumes:
  - name: data
    persistentVolumeClaim:
      claimName: gcnv-san-rwo
```

生ブロックデバイス（**RWX**）

```
apiVersion: v1
kind: Pod
metadata:
  name: app-raw-rwx
spec:
  containers:
  - name: app
    image: ubuntu:22.04
    command: ["sleep", "infinity"]
    volumeDevices:
    - name: data
      devicePath: /dev/xda
  volumes:
  - name: data
    persistentVolumeClaim:
      claimName: gcnv-san-raw-rwx
```

アタッチおよびマウント動作

Google Cloud NetApp Volumes からプロビジョニングされた SAN ボリュームの場合：

- Tridentは、Flex Unifiedストレージプール内に論理ユニット番号（LUN）を作成します。
- 公開中、Trident は LUN をノードごとのホストグループにマッピングします。
- ノードステージング中、Trident：
  - iSCSIターゲットにログインします
  - LUNを検出します
  - マルチパスを設定します
- `volumeMode: Filesystem`の場合、Tridentは必要に応じてデバイスをフォーマットし、マウントします。
- `volumeMode: Block`の場合、Tridentはデバイスを接続し、フォーマットやマウントを行わずに直接Podに公開します。

> SANブロックボリュームは、分散ロックや書き込み調整機能を提供しません。ブロックボリュームが複数のノードからアクセスされる場合（ReadWriteManyと `volumeMode: Block`）、アプリケーションまたはファイルシステムは、並行処理を管理する必要があります。

Google Cloud NetApp Volumes バックエンドを構成する準備をする

Google Cloud NetApp Volumes バックエンドを設定する前に、次の要件が満たされていることを確認する必要があります。

NFSまたはSMBボリュームの前提条件

Google Cloud NetApp Volumes を初めて使用する場合、または新しい場所で使用する場合は、Google Cloud NetApp Volumes をセットアップして NFS または SMB ボリュームを作成するために、初期設定が必要です。"開始する前に"を参照してください。

Google Cloud NetApp Volumes バックエンドを構成する前に、次のものを用意してください。

- Google Cloud NetApp Volumes サービスが設定された Google Cloud アカウント。"Google Cloud NetApp Volumes"を参照してください。
- Google Cloud アカウントのプロジェクト番号。"プロジェクトの特定"を参照してください。
- NetApp Volumes Admin (`roles/netapp.admin`）ロールを持つGoogle Cloudサービスアカウント。"Identity and Access Managementのロールと権限"を参照してください。
- GCNV アカウントの API キー ファイル。参照先："サービスアカウントキーを作成する"
- ストレージプール。"ストレージプールの概要"を参照してください。

Google Cloud NetApp Volumes へのアクセスを設定する方法の詳細については、"Google Cloud NetApp Volumes へのアクセスを設定する"を参照してください。

Google Cloud NetApp Volumes バックエンドの設定オプションと例

Google Cloud NetApp Volumes のバックエンド構成オプションについて学習し、構成例を確認します。

バックエンド構成オプション

各バックエンドは、単一の Google Cloud リージョンにボリュームをプロビジョニングします。他のリージョンにボリュームを作成するには、追加のバックエンドを定義できます。

パラメータ	概要	デフォルト
`version`		常に1
`storageDriverName`	ストレージドライバーの名前	`storageDriverName`の値は「google-cloud-netapp-volumes」として指定する必要があります。
`backendName`	（オプション）ストレージバックエンドのカスタム名	ドライバー名 + "_" + API キーの一部
`storagePools`	ボリューム作成用のストレージプールを指定するために使用されるオプションのパラメータ。	
`projectNumber`	Google Cloud アカウントのプロジェクト番号。値は Google Cloud ポータルのホームページにあります。	
`location`	Google Cloud の所在地。Trident が GCNV ボリュームを作成します。リージョンをまたがる Kubernetes クラスタを作成する場合、 `location`で作成されたボリュームは、複数の Google Cloud リージョンにまたがるノードでスケジュールされたワークロードで使用できます。リージョン間のトラフィックには追加コストが発生します。	
`apiKey`	`netapp.admin`ロールを持つGoogle Cloudサービスアカウントの`API`キー。これには、`Google Cloud`サービスアカウントの秘密鍵ファイルの`JSON`形式の内容（バックエンド構成ファイルにそのままコピーされます）が含まれます。 `apiKey`には、次のキーのキーと値のペアを含める必要があります：`type`、`project_id`、`client_email`、`client_id`、`auth_uri`、`token_uri`、`auth_provider_x509_cert_url`、および`client_x509_cert_url`。	
`nfsMountOptions`	NFSマウントオプションをきめ細かく制御できます。	"nfsvers=3"
`limitVolumeSize`	要求されたボリュームサイズがこの値を超える場合、プロビジョニングは失敗します。	""（デフォルトでは強制されません）

パラメータ	概要	デフォルト
`serviceLevel`	ストレージ プールとそのボリュームのサービス レベル。値は `flex`、 `standard`、 `premium`、または `extreme`です。	
`labels`	ボリュームに適用する任意の JSON 形式のラベルのセット	""
`network`	GCNV ボリュームに使用される Google Cloud ネットワーク。	
`debugTraceFlags`	トラブルシューティング時に使用するデバッグ フラグ。例： `{"api":false, "method":true}`。トラブルシューティングを行っており、詳細なログダンプが必要な場合を除き、これを使用しないでください。	null
`nasType`	NFS または SMB ボリュームの作成を設定します。オプションは `nfs`、 `smb`、またはnullです。null に設定すると、デフォルトで NFS ボリュームになります。	`nfs`
`supportedTopologies`	このバックエンドでサポートされているリージョンとゾーンのリストを表します。詳細については、"CSI トポロジを使用する"を参照してください。例： `supportedTopologies: - topology.kubernetes.io/region: asia-east1 topology.kubernetes.io/zone: asia-east1-a`	

ボリュームのプロビジョニング オプション

デフォルトのボリュームプロビジョニングは、構成ファイルの `defaults`セクションで制御できます。

パラメータ	概要	デフォルト
`exportRule`	新しいボリュームのエクスポートルール。任意の IPv4 アドレスの組み合わせをコンマで区切ったリストにする必要があります。	"0.0.0.0/0"
`snapshotDir`	`.snapshot`ディレクトリへのアクセス	`true`、 `false` （デフォルトの動作は異なる場合があります。明示的に設定）NFSv3の場合は「false」
`snapshotReserve`	Snapshot 用に予約されているボリュームの割合	""（デフォルトの0を受け入れます）
`unixPermissions`	新規ボリュームのUnixパーミッション（8進数4桁）。	""

構成例

次の例は、ほとんどのパラメータをデフォルトのままにする基本構成を示しています。これはバックエンドを

定義する最も簡単な方法です。

▼ 最小限の構成

これは絶対的に最小限のバックエンド構成です。この構成では、Tridentは構成された場所でGoogle Cloud NetApp Volumesに委任されたすべてのストレージプールを検出し、新しいボリュームをそれらのプールの1つにランダムに配置します。 `nasType`が省略されているため、 `nfs`デフォルトが適用され、バックエンドはNFSボリュームをプロビジョニングします。

この構成は、Google Cloud NetApp Volumes を使い始めたばかりでいろいろ試している場合に最適ですが、実際にはプロビジョニングするボリュームに対して追加のスコープを設定する必要があるかもしれません。

`<id_value>`と `<key_value>`をサービスアカウントの認証情報に置き換えます。

```
---
apiVersion: v1
kind: Secret
metadata:
  name: backend-tbc-gcnv-secret
type: Opaque
stringData:
  private_key_id: "<id_value>"
  private_key: |
    -----BEGIN PRIVATE KEY-----
    <key_value>
    -----END PRIVATE KEY-----
---
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-gcnv
spec:
  version: 1
  storageDriverName: google-cloud-netapp-volumes
  projectNumber: "123455380079"
  location: europe-west6
  serviceLevel: premium
  apiKey:
    type: service_account
    project_id: my-gcnv-project
    client_email: myproject-prod@my-gcnv-project.iam.gserviceaccount.com
    client_id: "103346282737811234567"
    auth_uri: https://accounts.google.com/o/oauth2/auth
    token_uri: https://oauth2.googleapis.com/token
    auth_provider_x509_cert_url:
https://www.googleapis.com/oauth2/v1/certs
    client_x509_cert_url:
```

```
https://www.googleapis.com/robot/v1/metadata/x509/myproject-prod%40my-
gcnv-project.iam.gserviceaccount.com
  credentials:
    name: backend-tbc-gcnv-secret
```

▼ **SMB** ボリュームの設定

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-gcnv1
  namespace: trident
spec:
  version: 1
  storageDriverName: google-cloud-netapp-volumes
  projectNumber: "123456789"
  location: asia-east1
  serviceLevel: flex
  nasType: smb
  apiKey:
    type: service_account
    project_id: cloud-native-data
    client_email: trident-sample@cloud-native-
data.iam.gserviceaccount.com
    client_id: "123456789737813416734"
    auth_uri: https://accounts.google.com/o/oauth2/auth
    token_uri: https://oauth2.googleapis.com/token
    auth_provider_x509_cert_url:
https://www.googleapis.com/oauth2/v1/certs
    client_x509_cert_url:
https://www.googleapis.com/robot/v1/metadata/x509/trident-
sample%40cloud-native-data.iam.gserviceaccount.com
  credentials:
    name: backend-tbc-gcnv-secret
```

▼ **StoragePools**フィルターを使用した構成

```
---
apiVersion: v1
kind: Secret
metadata:
  name: backend-tbc-gcnv-secret
type: Opaque
stringData:
  private_key_id: "<id_value>"
```

```
  private_key: |
    -----BEGIN PRIVATE KEY-----
    <key_value>
    -----END PRIVATE KEY-----
---

---
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-gcnv
spec:
  version: 1
  storageDriverName: google-cloud-netapp-volumes
  projectNumber: "123455380079"
  location: europe-west6
  serviceLevel: premium
  storagePools:
    - premium-pool1-europe-west6
    - premium-pool2-europe-west6
  apiKey:
    type: service_account
    project_id: my-gcnv-project
    client_email: myproject-prod@my-gcnv-project.iam.gserviceaccount.com
    client_id: "103346282737811234567"
    auth_uri: https://accounts.google.com/o/oauth2/auth
    token_uri: https://oauth2.googleapis.com/token
    auth_provider_x509_cert_url:
https://www.googleapis.com/oauth2/v1/certs
    client_x509_cert_url:
https://www.googleapis.com/robot/v1/metadata/x509/myproject-prod%40my-
gcnv-project.iam.gserviceaccount.com
  credentials:
    name: backend-tbc-gcnv-secret
```

▼ 仮想プールの構成

このバックエンド構成では、単一のファイルで複数の仮想プールを定義します。仮想プールは `storage`セクションで定義されます。異なるサービスレベルをサポートする複数のストレージプールがあり、Kubernetesでそれらを表すストレージクラスを作成する場合に役立ちます。仮想プールラベルは、プールを区別するために使用されます。たとえば、以下の例では `performance`ラベルと `serviceLevel`タイプが仮想プールを区別するために使用されます。

一部のデフォルト値をすべての仮想プールに適用できるように設定し、個々の仮想プールのデフォルト値を上書きすることもできます。次の例では、 `snapshotReserve`と `exportRule`がすべての仮想プールのデフォルトとして機能します。

詳細については、"仮想プール"を参照してください。

```
---
apiVersion: v1
kind: Secret
metadata:
  name: backend-tbc-gcnv-secret
type: Opaque
stringData:
  private_key_id: "<id_value>"
  private_key: |
    -----BEGIN PRIVATE KEY-----
    <key_value>
    -----END PRIVATE KEY-----

---
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-gcnv
spec:
  version: 1
  storageDriverName: google-cloud-netapp-volumes
  projectNumber: "123455380079"
  location: europe-west6
  apiKey:
    type: service_account
    project_id: my-gcnv-project
    client_email: myproject-prod@my-gcnv-project.iam.gserviceaccount.com
    client_id: "103346282737811234567"
    auth_uri: https://accounts.google.com/o/oauth2/auth
    token_uri: https://oauth2.googleapis.com/token
    auth_provider_x509_cert_url:
https://www.googleapis.com/oauth2/v1/certs
    client_x509_cert_url:
https://www.googleapis.com/robot/v1/metadata/x509/myproject-prod%40my-
gcnv-project.iam.gserviceaccount.com
  credentials:
    name: backend-tbc-gcnv-secret
  defaults:
    snapshotReserve: "10"
    exportRule: 10.0.0.0/24
  storage:
    - labels:
        performance: extreme
      serviceLevel: extreme
      defaults:
```

```
        snapshotReserve: "5"
        exportRule: 0.0.0.0/0
    - labels:
        performance: premium
      serviceLevel: premium
    - labels:
        performance: standard
      serviceLevel: standard
```

▼ **GKE のクラウド ID**

GKEクラウドIDを使用してTridentをインストールすると、バックエンドはワークロードIDとして認証され、 `apiKey`ブロックは省略されます。クラウドIDは、両方のGoogle Cloud NetApp Volumesドライバーで同じように設定されます。唯一の違いは `storageDriverName`の値です。NAS（NFSまたはSMB）ボリュームには `google-cloud-netapp-volumes`を使用し、SAN（iSCSIブロック）ボリュームには `google-cloud-netapp-volumes-san`を使用します。

> `google-cloud-netapp-volumes-san`ドライバーのクラウドID サポートには、Trident 26.06以降が必要です。SANボリュームはFlex Unifiedストレージプールからプロビジョニングされるため、 `serviceLevel`を `flex`に設定してください。

NASドライバー（<code>google-cloud-netapp-volumes</code>）

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-gcp-gcnv
spec:
  version: 1
  storageDriverName: google-cloud-netapp-volumes
  projectNumber: '012345678901'
  network: gcnv-network
  location: us-west2
  serviceLevel: Premium
  storagePool: pool-premium1
```

SANドライバ（<code>google-cloud-netapp-volumes-san</code>）

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-gcp-gcnv-san
spec:
  version: 1
  storageDriverName: google-cloud-netapp-volumes-san
  projectNumber: '012345678901'
  network: gcnv-network
```

```
  location: us-west2
  serviceLevel: flex
  storagePool: pool-flex1
```

▼ サポートされているトポロジ構成

Tridentは、リージョンとアベイラビリティゾーンに基づいてワークロードのボリュームのプロビジョニングを容易にします。このバックエンド構成の `supportedTopologies`ブロックは、バックエンドごとのリージョンとゾーンのリストを提供するために使用されます。ここで指定するリージョンとゾーンの値は、各Kubernetesクラスターノードのラベルのリージョンとゾーンの値と一致する必要があります。これらのリージョンとゾーンは、ストレージクラスで提供できる許容値のリストを表します。バックエンドで提供されるリージョンとゾーンのサブセットを含むストレージクラスの場合、Tridentは指定されたリージョンとゾーンにボリュームを作成します。詳細については、"CSI トポロジを使用する"を参照してください。

```
---
version: 1
storageDriverName: google-cloud-netapp-volumes
subscriptionID: 9f87c765-4774-fake-ae98-a721add45451
tenantID: 68e4f836-edc1-fake-bff9-b2d865ee56cf
clientID: dd043f63-bf8e-fake-8076-8de91e5713aa
clientSecret: SECRET
location: asia-east1
serviceLevel: flex
supportedTopologies:
  - topology.kubernetes.io/region: asia-east1
    topology.kubernetes.io/zone: asia-east1-a
  - topology.kubernetes.io/region: asia-east1
    topology.kubernetes.io/zone: asia-east1-b
```

次の手順

バックエンド構成ファイルを作成したら、次のコマンドを実行します：

```
kubectl create -f <backend-file>
```

バックエンドが正常に作成されたことを確認するには、次のコマンドを実行します：

```
kubectl get tridentbackendconfig

NAME                BACKEND NAME        BACKEND UUID
PHASE   STATUS
backend-tbc-gcnv    backend-tbc-gcnv    b2fd1ff9-b234-477e-88fd-713913294f65
Bound   Success
```

バックエンドの作成に失敗した場合は、バックエンドの構成に問題があります。 `kubectl get tridentbackendconfig <backend-name>`コマンドを使用してバックエンドを記述するか、次のコマンドを実行してログを表示し、原因を特定できます：

```
tridentctl logs
```

構成ファイルの問題を特定して修正したら、バックエンドを削除して、create コマンドを再度実行できます。

ストレージクラスの定義

以下は、上記のバックエンドを参照する基本的な `StorageClass`定義です。

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: gcnv-nfs-sc
provisioner: csi.trident.netapp.io
parameters:
  backendType: "google-cloud-netapp-volumes"
```

- `parameter.selector`フィールドを使用した定義例：*

  `parameter.selector`を使用すると、各`StorageClass`に対して、ボリュームをホストするために使用されるlink:../trident-concepts/virtual-storage-pool.html["仮想プール"]を指定できます。ボリュームには、選択したプールで定義された側面が含まれます。

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: extreme-sc
provisioner: csi.trident.netapp.io
parameters:
  selector: performance=extreme
  backendType: google-cloud-netapp-volumes

---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: premium-sc
provisioner: csi.trident.netapp.io
parameters:
```

```
  selector: performance=premium
  backendType: google-cloud-netapp-volumes

---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: standard-sc
provisioner: csi.trident.netapp.io
parameters:
  selector: performance=standard
  backendType: google-cloud-netapp-volumes
```

ストレージクラスの詳細については、"ストレージクラスを作成する"を参照してください。

SMB ボリュームの定義例

`nasType`、 `node-stage-secret-name`、および `node-stage-secret-namespace`を使用して、SMBボリュームを指定し、必要なActive Directory資格情報を提供できます。任意の権限または権限のないActive Directoryユーザー/パスワードをノードステージシークレットに使用できます。

▼ デフォルトの名前空間での基本設定

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: gcnv-sc-smb
provisioner: csi.trident.netapp.io
parameters:
  backendType: "google-cloud-netapp-volumes"
  trident.netapp.io/nasType: "smb"
  csi.storage.k8s.io/node-stage-secret-name: "smbcreds"
  csi.storage.k8s.io/node-stage-secret-namespace: "default"
```

▼ 名前空間ごとに異なるシークレットを使用する

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: gcnv-sc-smb
provisioner: csi.trident.netapp.io
parameters:
  backendType: "google-cloud-netapp-volumes"
```

```
    trident.netapp.io/nasType: "smb"
    csi.storage.k8s.io/node-stage-secret-name: "smbcreds"
    csi.storage.k8s.io/node-stage-secret-namespace: ${pvc.namespace}
```

▼ ボリュームごとに異なるシークレットを使用する

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: gcnv-sc-smb
provisioner: csi.trident.netapp.io
parameters:
  backendType: "google-cloud-netapp-volumes"
  trident.netapp.io/nasType: "smb"
  csi.storage.k8s.io/node-stage-secret-name: ${pvc.name}
  csi.storage.k8s.io/node-stage-secret-namespace: ${pvc.namespace}
```

`nasType: smb` SMB ボリュームをサポートするプールのフィルタ。 `nasType: nfs`または `nasType: null` NFS プールのフィルタ。

PVC定義の例

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: gcnv-nfs-pvc
spec:
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 100Gi
  storageClassName: gcnv-nfs-sc
```

PVC がバインドされているかどうかを確認するには、次のコマンドを実行します：

```
kubectl get pvc gcnv-nfs-pvc

NAME          STATUS   VOLUME                                     CAPACITY
ACCESS MODES   STORAGECLASS AGE
gcnv-nfs-pvc  Bound    pvc-b00f2414-e229-40e6-9b16-ee03eb79a213  100Gi
RWX            gcnv-nfs-sc  1m
```

Google Cloud NetApp Volumes の自動階層化を設定する

自動階層化は、TridentバックエンドパラメータとPersistentVolumeClaimアノテーションを使用して、ボリュームのプロビジョニング時に設定されます。Tridentを使用して、Google Cloud NetApp Volumesの自動階層化を設定できます。

概要

自動階層化により、Trident は非アクティブなデータをパフォーマンス層から容量層に自動的に移動するボリュームをプロビジョニングできます。これにより、頻繁にアクセスされるデータのパフォーマンスを維持しながら、ストレージコストを削減できます。

Trident は、ボリューム作成時にのみ自動階層化設定を適用します。プロビジョニング後の変更は Trident 26.06 ではサポートされていません。

概念

自動階層化

自動階層化機能は、アクセスパターンに基づいて、アクセス頻度の低いデータをパフォーマンス階層から容量階層に移動します。データ転送は非同期で行われるため、即時ではありません。

階層化ポリシー

階層化ポリシーは、ボリュームに対して自動階層化を有効にするかどうかを決定します。

以下のポリシーがサポートされています：* `auto`：アクセスパターンに基づいて自動階層化を有効にします * `none`：自動階層化を無効にします

冷却日数

冷却日数とは、データ ブロックが階層化の対象となるために非アクティブ状態を維持する必要がある最小日数を指定します。冷却日数は、階層化ポリシーが `auto`に設定されている場合にのみ適用されます。

構成モデル

構成スコープ

自動階層化は複数のスコープで設定できます：

- ストレージプールの範囲 環境：プールからプロビジョニングされたすべてのボリュームに適用されます。
- ボリューム スコープ 単一ボリュームに適用されます（PersistentVolumeClaim アノテーション経由）。

Trident は、各設定が定義されている場所に基づいて、有効な構成を決定します。

設定の優先順位

同じ設定が複数のスコープで定義されている場合、Trident は以下の優先順位を適用します：

1. PersistentVolumeClaim アノテーション
2. Tridentバックエンド構成
3. ストレージプールのデフォルト設定

優先順位の高い設定は、優先順位の低い設定を上書きします。

Trident 26.06 でサポートされている機能

Trident 26.06は、Google Cloud NetApp Volumesの次の自動階層化機能をサポートしています：

- ボリュームプロビジョニング時の自動階層化の有効化または無効化
- Trident バックエンド構成での階層化ポリシーの定義
- PVC アノテーションを使用して、階層化ポリシーとボリュームごとの冷却日数を上書きする
- 自動階層化が有効になっているボリュームの冷却日を設定する

Trident 26.06 でサポートされていない機能

次の処理はサポートされていません。

- ボリューム作成後に自動階層化設定を変更する
- Kubernetesアップデートを使用して既存ボリュームの階層化ポリシーを変更する
- Tridentが管理するプロビジョニングワークフロー以外で自動階層化設定を適用する

バックエンド構成パラメータ

以下のパラメータは、Trident バックエンド設定で定義された場合の自動階層化の動作を制御します：

パラメータ	必須	概要
tieringPolicy	いいえ	ボリュームごとの階層化ポリシー (`auto` または `none`)
tieringMinimumCoolingDays	いいえ	データが階層化されるまでの非アクティブ日数（範囲：2〜183、デフォルト：31）

PersistentVolumeClaim アノテーションを使用したボリュームレベルのオーバーライド

サポートされているアノテーション

PersistentVolumeClaimアノテーションを使用すると、ボリュームごとに自動階層化設定を上書きできます。

注釈	概要
`trident.netapp.io/tieringPolicy`	ボリュームの階層化ポリシーを上書きします
`trident.netapp.io/tieringMinimumCoolingDays`	ボリュームの冷却日数の値を上書きします

例：自動階層化オーバーライド付き **PersistentVolumeClaim**

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: auto-tiering-pvc
  annotations:
    trident.netapp.io/tieringPolicy: auto
    trident.netapp.io/tieringMinimumCoolingDays: "45"
spec:
  accessModes:
    - ReadWriteOnce
  storageClassName: google-cloud-netapp-volumes-auto-tiering
  resources:
    requests:
      storage: 500Gi
```

動作と制限事項

プロビジョニング動作

- 自動階層化設定は、ボリューム作成時にのみ評価および適用されます。
- Tridentは、プロビジョニング後に階層化の設定を調整しません。
- 階層化ポリシーが `none`に設定されている場合、冷却日数は無視されます。

プラットフォームの制限

- 自動階層化は、NASボリューム（NFSおよびSMB）でのみサポートされています。
- ブロックボリューム（iSCSI）は自動階層化をサポートしていません。
- Google Cloud NetApp Volumes のストレージプールでは、Google Cloud で自動階層化が有効になっている必要があります。

サポートされている値

- `tieringMinimumCoolingDays`の有効範囲：2～183
- デフォルト値：31

Google Cloud NetApp Volumes ONTAP モードストレージプールの設定

Trident 26.06以降、既存の `ontap-san`および `ontap-nas`バックエンドを設定して、Google Cloud NetApp Volumes（GCNV）のONTAPモードストレージプールにストレージをプロビジョニングできます。この機能はテクニカルプレビューです。

メモ	この機能は Trident 26.06 のテクニカルプレビューです。テクニカルプレビュー機能は、本番環境での使用はサポートされていません。機能、設定項目、およびサポートされるマトリックスは、一般提供開始（GA）前に変更される場合があります。

仕組み

Trident は既存の `ontap-san`および `ontap-nas`ドライバーロジックを再利用し、GCNV プロキシエンドポイントを介して ONTAP REST 呼び出しをルーティングします。このアプローチにより、直接 ONTAP バックエンドに使用するのと同じ運用ワークフローを維持しながら、GCNV ホスト上の ONTAP クラスターのストレージをプロビジョニングおよび管理できます。ONTAP モードでは、新しいストレージドライバーは導入されません。バックエンドごとに有効にするには、 `gcnv`構成ブロックを `ontap-san`または `ontap-nas`バックエンドに追加します。

プロビジョニングのパスは次のとおりです：

`PersistentVolumeClaim` → Trident (`ontap-san` または `ontap-nas`) → ONTAP REST クライアント → GCNV プロキシ → GCNV プール内の ONTAP クラスター

サポート対象範囲とサポート対象外範囲

このテクニカルプレビューでは、以下をサポートしています。

- ドライバー： `ontap-san` （iSCSI）および `ontap-nas` （NFSまたはSMB）。
- プロキシを介したライフサイクル操作：バックエンドの初期化、ボリュームの作成と削除、および関連する ONTAP REST フロー。
- 認証モデル：Workload Identity Pool （WIP）、サービスアカウントキー、およびApplication Default Credentials （ADC）フォールバック。

このテクニカルプレビューでは、以下の項目は対象外となります。

- `ontap-san-economy`、 `ontap-nas-economy`、および `ontap-nas-flexgroup`ドライバー。
- ASA r2 パーソナリティパス。
- ZAPIのフォールバック。ONTAPモードはONTAP RESTのみを使用します。

前提条件

ONTAPモードのバックエンドを設定する前に、以下のものを用意してください：

- Trident 26.06以降。
- 対象プロジェクトおよび対象場所に、GCNV ONTAP モードのストレージプールが必要です。
- プロキシ操作に必要な GCP IAM 権限を、お客様の環境に合わせて適切にスコープ設定します。
- `proxyURL`、 `projectNumber`、 `location`、および `poolID`を含むバックエンド構成。
- iSCSI SANワークフローの場合、プラットフォームのノード側iSCSIおよびマルチパスの前提条件が必要です。"ワーカーノードを準備する"を参照してください。

バックエンドの設定

A `TridentBackendConfig`には `credentials`が必要です。`ONTAP` モードの場合、そのシークレットには `GCP` プロキシ認証情報が含まれています。`ONTAP` `managementLIF`/ `username`/ `password`は含まれません。 `svm`はオプションです。各 GCNV ONTAP モードプールには現在 1 つの SVM があり、省略された場合、Trident はプールから取得します。

ネイティブ GCNV と同じサービスアカウントパターンを使用します。非機密フィールドは `gcnv.apiKey`、`private_key`、シークレット内は `private_key_id`に記述します。"Google Cloud NetApp Volumes の構成例"を参照してください。

ontap-san バックエンド

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: gcnv-ontap-san
  namespace: trident
spec:
  version: 1
  storageDriverName: ontap-san
  backendName: gcnv-ontap-san
  credentials:
    name: gcnv-sa-secret
    type: secret
  gcnv:
    proxyURL: "https://netapp.googleapis.com"
    projectNumber: "<project-number>"
    location: "<region-or-zone>"
    poolID: "<pool-id>"
  svm: "<svm-name>"
```

ontap-nas backend

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: gcnv-ontap-nas
  namespace: trident
spec:
  version: 1
  storageDriverName: ontap-nas
  backendName: gcnv-ontap-nas
  credentials:
    name: gcnv-sa-secret
    type: secret
  gcnv:
```

```
    proxyURL: "https://netapp.googleapis.com"
    projectNumber: "<project-number>"
    location: "<region-or-zone>"
    poolID: "<pool-id>"
    apiKey:
      type: service_account
      project_id: "<project-id>"
      client_email: "<service-account-email>"
      client_id: "<client-id>"
      auth_uri: "https://accounts.google.com/o/oauth2/auth"
      token_uri: "https://oauth2.googleapis.com/token"
      auth_provider_x509_cert_url:
"https://www.googleapis.com/oauth2/v1/certs"
      client_x509_cert_url: "<client-x509-cert-url>"
```

シークレット

```
apiVersion: v1
kind: Secret
metadata:
  name: gcnv-sa-secret
  namespace: trident
type: Opaque
stringData:
  private_key_id: "<private-key-id>"
  private_key: |
    -----BEGIN PRIVATE KEY-----
    <private-key>
    -----END PRIVATE KEY-----
```

メモ

`TridentBackendConfig`仕様には `private_key`や`private_key_id`を指定しないでください。CRDによって拒否されます。ブロックストレージには、同じ `gcnv`とクレデンシャル パターンを持つ `ontap-san`を使用してください。

```
svm: "<svm-name>"
```

```
== Authentication

Trident resolves credentials for proxy access in the following order:
```

```
. Workload Identity Pool (WIP)
. Service account key
. Application Default Credentials (ADC)

For this Tech Preview, use the following practices:

* Prefer Workload Identity Pool where it is available.
* Do not embed raw private keys in version-controlled backend files. Store
credentials in a Kubernetes secret.
* Apply least-privilege IAM scoping to the service account.

== Validation and initialization behavior

Trident validates an ONTAP-mode backend during initialization and fails
fast rather than partially initializing.

Backend initialization fails if a required `gcnv` field is missing, the
driver is unsupported for ONTAP-mode, or credential resolution fails.

== Known limitations

* This feature is a Tech Preview. Behavior and the supported matrix can
change before GA.
* ZAPI fallback is not used in ONTAP-mode.
* Protocol and node-readiness requirements still apply. For example, iSCSI
workflows require the node-side prerequisites described in
xref:{relative_path}../trident-use/worker-node-prep.html[Prepare the
worker node].
* Existing ONTAP and GCNV environment constraints still apply.

== Upgrade and compatibility

ONTAP-mode is opt-in per backend through the `gcnv` configuration block:

* Existing non-GCNV ONTAP backends are unaffected.
* Mixed deployments that combine direct ONTAP backends and ONTAP-mode
proxy backends are supported through backend-level configuration. Each
backend is configured independently.
* To stop using ONTAP-mode, remove or replace the affected backends. No
global switch is required.

== What's next?

Apply secret before TBC; use `kubectl apply` and `-n trident` to match
examples.
```

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権を有し、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権については、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。