



Google Cloud NetApp Volumes

Trident

NetApp
January 15, 2026

This PDF was generated from <https://docs.netapp.com/ja-jp/trident-2506/trident-use/gcnv.html> on January 15, 2026. Always check docs.netapp.com for the latest.

目次

Google Cloud NetApp Volumes	1
Google Cloud NetApp Volumesバックエンドを構成する	1
Google Cloud NetApp Volumesドライバの詳細	1
GKE のクラウド ID	1
Google Cloud NetApp Volumesバックエンドを構成する準備をする	3
NFSボリュームの前提条件	3
Google Cloud NetApp Volumes のバックエンド構成オプションと例	3
バックエンド構成オプション	3
ボリュームのプロビジョニング オプション	5
構成例	5
次の手順	13
ストレージクラスの定義	14

Google Cloud NetApp Volumes

Google Cloud NetApp Volumesバックエンドを構成する

Google Cloud NetApp Volumes をTridentのバックエンドとして構成できるようになりました。 Google Cloud NetApp Volumesバックエンドを使用して、NFS ボリュームと SMB ボリュームを接続できます。

Google Cloud NetApp Volumesドライバの詳細

Tridentは`google-cloud-netapp-volumes`クラスターと通信するためのドライバー。サポートされているアクセス モードは、*ReadWriteOnce* (RWO)、*ReadOnlyMany* (ROX)、*ReadWriteMany* (RWX)、*ReadWriteOncePod* (RWOP) です。

ドライバ	プロトコル	ボリュームモード	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

GKE のクラウド ID

Cloud Identity を使用すると、Kubernetes ポッドは、明示的な Google Cloud 認証情報を提供する代わりに、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 でクラウド ID を活用するには、次のものがが必要です。

- GKE を使用してデプロイされた Kubernetes クラスター。
- GKE クラスターに構成されたワークロード ID と、ノードプールに構成された GKE メタデータ サーバー。
- Google Cloud NetApp Volumes管理者 (roles/netapp.admin) ロールまたはカスタムロールを持つ GCP サービス アカウント。
- 「GCP」を指定する cloudProvider と、新しい GCP サービス アカウントを指定する cloudIdentity を含むTrident がインストールされています。以下に例を示します。

Tridentオペレーター

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に`設定する` `cloudProvider`に`"GCP"`そして設定
`cloudIdentity`に`iam.gke.io/gcp-service-account: cloudvolumes-admin-
sa@mygcpproject.iam.gserviceaccount.com`

例えば：

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

舵

次の環境変数を使用して、**cloud-provider (CP)** および **cloud-identity (CI)** フラグの値を設定しま
す。

```
export CP="GCP"
export ANNOTATION="'iam.gke.io/gcp-service-account: cloudvolumes-admin-
sa@mygcpproject.iam.gserviceaccount.com'"
```

次の例では、Tridentをインストールし、cloudProvider`環境変数を使用してGCPへ`\$CP`そし
て、`cloudIdentity`環境変数を使用する`\$ANNOTATION`:

```
helm install trident trident-operator-100.6.0.tgz --set
cloudProvider=$CP --set cloudIdentity="$ANNOTATION"
```

<code>トライデントctl</code>

次の環境変数を使用して、クラウド プロバイダー および クラウド ID フラグの値を設定します。

```
export CP="GCP"
export ANNOTATION="'iam.gke.io/gcp-service-account: cloudvolumes-admin-
sa@mygcpproject.iam.gserviceaccount.com'"
```

次の例では、Tridentをインストールし、cloud-provider`フラグを`\$CP、そして cloud-
identity`に`\$ANNOTATION`:

```
tridentctl install --cloud-provider=$CP --cloud
-identity="$ANNOTATION" -n trident
```

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

Google Cloud NetApp Volumesバックエンドを構成する前に、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ボリュームの前提条件

Google Cloud NetApp Volumes を初めて使用するか、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 ボリュームを作成するための初期設定が必要です。参照["開始する前に"](#)。

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サービスで構成された Google Cloud アカウント。参照["Google Cloud NetApp Volumes"](#)。
- Google Cloud アカウントのプロジェクト番号。参照["プロジェクトの特定"](#)。
- NetApp Volumes 管理者権限を持つ Google Cloud サービス アカウント(roles/netapp.admin) 役割。参照["アイデンティティとアクセス管理のロールと権限"](#)。
- GCNV アカウントの API キー ファイル。参照["サービスアカウントキーを作成する"](#)
- ストレージ プール。参照["ストレージプールの概要"](#)。

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、以下を参照してください。"[Google Cloud NetApp Volumesへのアクセスを設定する](#)"。

Google Cloud NetApp Volumes のバックエンド構成オプションと例

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	説明	デフォルト
version		常に1

パラメータ	説明	デフォルト
storageDriverName	ストレージ ドライバーの名前	の値 storageDriverName 「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ー。	
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	Trident が GCNV ボリュームを作成する Google Cloud のロケーション。クロスリージョン Kubernetes クラスターを作成する場合、`location` 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	Google Cloud サービスアカウントの API キーと netapp.admin`役割。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。その `apiKey` 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`、そして `client_x509_cert_url`。	
nfsMountOptions	NFS マウント オプションのきめ細かな制御。	「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)
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。	
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。	ヌル
nasType	NFS または SMB ボリュームの作成を構成します。オプションは nfs、`smb` または null。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

パラメータ	説明	デフォルト
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 "。例えば： 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、`defaults`構成ファイルのセクション。

パラメータ	説明	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	アクセス `.snapshot`ディレクトリ	NFSv4の場合は「true」、NFSv3の場合は「false」
snapshotReserve	スナップショット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	新しいボリュームの UNIX 権限 (4桁の 8 進数)。	""

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 は構成された場所にある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、それらのプールの1 つに新しいボリュームをランダムに配置します。なぜなら ``nasType`` 省略された場合、``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を使い始めて試してみ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のスコープをさらに指定する必要がある可能性が高くなります。

```

---
apiVersion: v1
kind: Secret
metadata:
  name: backend-tbc-gcnv-secret
type: Opaque
stringData:
  private_key_id: f2cb6ed6d7cc10c453f7d3406fc700c5df0ab9ec
  private_key: |
    -----BEGIN PRIVATE KEY-----\n
    znHczZsrtrHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m\n
    znHczZsrtrHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m\n
    znHczZsrtrHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m\n
    XsYg6gyxy4zq7OlwWgLwGa==\n
    -----END PRIVATE KEY-----\n

---
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

```

```
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: f2cb6ed6d7cc10c453f7d3406fc700c5df0ab9ec
  private_key: |
    -----BEGIN PRIVATE KEY-----
    znHczZsrtrHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrtrHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrtrHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m
    XsYg6gyxy4zq7OlwWgLwGa==
    -----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。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。例えば、以下の例では `performance` ラベルと `serviceLevel` タイプは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`そして`exportRule`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](#)"。

```
---
apiVersion: v1
kind: Secret
metadata:
  name: backend-tbc-gcnv-secret
type: Opaque
stringData:
  private_key_id: f2cb6ed6d7cc10c453f7d3406fc700c5df0ab9ec
  private_key: |
    -----BEGIN PRIVATE KEY-----
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m
    znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m
    XsYg6gyxy4zq7OlwWgLwGa==
    -----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

```

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

```

サポートされているトポロジ構成

Trident は、リージョンと可用性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。その `supportedTopologies` このバックエンド構成のブロックは、バックエンドごとに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"](#)。

```
---
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
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、バックエンドの構成に問題があります。バックエンドを記述するには、`kubectl get tridentbackendconfig <backend-name>` 次のコマンドを実行して、コマンドを実行するか、ログを表示して原因を特定します。

```
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

以下は基本的な `StorageClass` 上記のバックエンドを参照する定義。

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

定義例 `parameter.selector` 分野：

使用 `parameter.selector` それぞれ指定できます `StorageClass` その"[仮想プール](#)"ボリュームをホストするために使用されます。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、以下を参照してください。"[ストレージクラスを作成する](#)"。

SMBボリュームの定義例

使用 `nasType`、`node-stage-secret-name`、そして `node-stage-secret-namespace`、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任意の 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秘密を使用する

```
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ー。`nasType: nfs`または`nasType: null`NFS プールのフィルター。

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

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。