



Trident Protectをインストールする

Trident

NetApp
January 14, 2026

目次

Trident Protectをインストールする	1
Tridentプロテクトの要件	1
Trident Protect Kubernetes クラスターの互換性	1
Trident Protect ストレージバックエンドの互換性	1
NASエコノミーボリュームの要件	2
KubeVirt VMによるデータ保護	2
SnapMirrorレプリケーションの要件	3
Trident Protectのインストールと設定	4
Trident Protectをインストールする	4
Trident Protect CLIプラグインをインストールする	7
Trident Protect CLIプラグインをインストールする	7
Trident CLIプラグインのヘルプを表示	9
コマンドの自動補完を有効にする	9
Trident Protectのインストールをカスタマイズする	11
Trident Protectコンテナのリソース制限を指定する	11
セキュリティコンテキスト制約のカスタマイズ	12
Trident Protect のNetApp AutoSupport接続を構成する	13
Trident Protectポッドを特定のノードに制限する	14
毎日のTrident Protect AutoSupportバンドルのアップロードを無効にする	15

Trident Protectをインストールする

Tridentプロジェクトの要件

まず、運用環境、アプリケーションクラスター、アプリケーション、ライセンスの準備状況を確認します。Trident Protect を展開および運用するには、環境がこれらの要件を満たしていることを確認してください。

Trident Protect Kubernetes クラスターの互換性

Trident Protect は、以下を含む幅広いフルマネージドおよびセルフマネージド Kubernetes 製品と互換性があります。

- Amazon Elastic Kubernetes Service (EKS)
- Google Kubernetes Engine (GKE)
- Microsoft Azure Kubernetes Service (AKS)
- Red Hat OpenShift のサービスです
- SUSE Rancher
- VMware Tanzuポートフォリオ
- アップストリームKubernetes



Trident Protect をインストールするクラスターに、実行中のスナップショットコントローラーと関連する CRD が設定されていることを確認します。スナップショットコントローラをインストールするには、["以下の手順を参照して"](#)。

Trident Protect ストレージバックエンドの互換性

Trident Protect は次のストレージ バックエンドをサポートしています。

- Amazon FSx for NetApp ONTAP
- Cloud Volumes ONTAP
- ONTAPストレエシアレイ
- Google Cloud NetAppボリューム
- Azure NetApp Files

ストレージバックエンドが次の要件を満たしていることを確認します。

- クラスタに接続されているNetAppストレージがAstra Trident 24.02以降を使用していることを確認します (Trident 24.10を推奨)。
 - Astra Tridentが24.06.1より前のバージョンで、NetApp SnapMirrorディザスタリカバリ機能を使用する場合は、Astra Control Provisionerを手動で有効にする必要があります。
- 最新のAstra Control Provisionerがインストールされていることを確認します (Astra Trident 24.06.1以降ではデフォルトでインストールおよび有効化されています)。

- NetApp ONTAPストレージバックエンドがあることを確認します。
- バックアップを格納するオブジェクトストレージバケットを設定しておきます。
- アプリケーションまたはアプリケーション データ管理操作に使用する予定のアプリケーション名前空間を作成します。Trident Protect はこれらの名前空間を作成しません。カスタム リソースに存在しない名前空間を指定すると、操作は失敗します。

NASエコノミー ボリュームの要件

Trident Protect は、NAS エコノミー ボリュームへのバックアップおよび復元操作をサポートします。スナップショット、クローン、および NAS エコノミー ボリュームへの SnapMirror レプリケーションは現在サポートされていません。Trident Protect で使用する予定の各 nas-economy ボリュームに対してスナップショット ディレクトリを有効にする必要があります。

一部のアプリケーションは、Snapshotディレクトリを使用するボリュームと互換性があります。これらのアプリケーションでは、ONTAPストレージシステムで次のコマンドを実行して、snapshotディレクトリを非表示にする必要があります。



```
nfs modify -vserver <svm> -v3-hide-snapshot enabled
```

snapshotディレクトリを有効にするには、NASエコノミーボリュームごとに次のコマンドを実行し、を変更するボリュームのUUIDに置き換え`<volume-UUID>`ます。

```
tridentctl update volume <volume-UUID> --snapshot-dir=true --pool-level=true -n trident
```



新しいボリュームに対してSnapshotディレクトリをデフォルトで有効にするには、Tridentバックエンド構成オプションをに`true`設定し`snapshotDir`ます。既存のボリュームには影響しません。

KubeVirt VMによるデータ保護

Trident Protect 24.10 と 24.10.1 以降では、KubeVirt VM 上で実行されているアプリケーションを保護する場合の動作が異なります。どちらのバージョンでも、データ保護操作中にファイルシステムのフリーズとフリーズ解除を有効または無効にすることができます。

Tridentプロジェクト 24.10

Trident Protect 24.10 は、データ保護操作中に KubeVirt VM ファイルシステムの一貫した状態を自動的に保証しません。Trident Protect 24.10 を使用して KubeVirt VM データを保護する場合は、データ保護操作の前に、ファイルシステムのフリーズ/アンフリーズ機能を手動で有効にする必要があります。これにより、ファイルシステムが一貫した状態になることが保証されます。

Trident Protect 24.10を設定して、データ保護操作中にVMファイルシステムの凍結と解凍を管理することができます。["仮想化の設定"](#)そして次のコマンドを使用します。

```
kubectl set env deployment/trident-protect-controller-manager  
NEPTUNE_VM_FREEZE=true -n trident-protect
```

Trident Protect 24.10.1以降

Trident Protect 24.10.1 以降、Trident Protect はデータ保護操作中に KubeVirt ファイルシステムを自動的にフリーズおよびアンフリーズします。オプションで、次のコマンドを使用してこの自動動作を無効にすることができます。

```
kubectl set env deployment/trident-protect-controller-manager  
NEPTUNE_VM_FREEZE=false -n trident-protect
```

SnapMirrorレプリケーションの要件

NetApp SnapMirrorレプリケーションは、次のONTAPソリューションのTrident Protect で使用できます。

- ・ オンプレミスのNetApp FAS、AFF、ASAクラスタ
- ・ NetApp ONTAP Select の略
- ・ NetApp Cloud Volumes ONTAP の略
- ・ Amazon FSx for NetApp ONTAP

SnapMirrorレプリケーション用のONTAPクラスタの要件

SnapMirrorレプリケーションを使用する場合は、ONTAPクラスタが次の要件を満たしていることを確認します。

- ・ * Astra Control Provisioner またはTrident *: ONTAP をバックエンドとして使用するソース Kubernetes クラスターと宛先 Kubernetes クラスターの両方にAstra Control Provisioner またはTrident が存在する必要があります。Trident Protect は、次のドライバーによってサポートされるストレージ クラスを使用して、NetApp SnapMirrorテクノロジによるレプリケーションをサポートします。
 - ontap-nas
 - ontap-san
- ・ ライセンス：Data Protection Bundleを使用するONTAP SnapMirror非同期ライセンスが、ソースとデステイネーションの両方のONTAPクラスタで有効になっている必要があります。詳細については、を参照してください "[ONTAP のSnapMirrorライセンスの概要](#)"。

SnapMirrorレプリケーションのピアリングに関する考慮事項

ストレージバックエンドピアリングを使用する場合は、環境が次の要件を満たしていることを確認してください。

- ・ *クラスタとSVM * : ONTAPストレージバックエンドにピア関係が設定されている必要があります。詳細については、を参照してください "[クラスタと SVM のピアリングの概要](#)"。



2つのONTAPクラスタ間のレプリケーション関係で使用されるSVM名が一意であることを確認してください。

- * Astra Control ProvisionerまたはTridentとSVM *：ピア関係にあるリモートSVMは、デスティネーションクラスタのAstra Control ProvisionerまたはTridentで使用できる必要があります。
- 管理対象バックエンド: レプリケーション関係を作成するには、 Trident Protect でONTAPストレージ バックエンドを追加および管理する必要があります。
- **NVMe over TCP**: Trident Protect は、 NVMe over TCP プロトコルを使用しているストレージ バックエンドのNetApp SnapMirrorレプリケーションをサポートしていません。

SnapMirrorレプリケーション用のTrident / ONTAPの設定

Trident Protect では、ソース クラスターと宛先クラスターの両方のレプリケーションをサポートするストレージ バックエンドを少なくとも 1 つ構成する必要があります。ソース クラスターと宛先クラスターが同じ場合、復元力を最大限に高めるには、宛先アプリケーションでソース アプリケーションとは異なるストレージ バックエンドを使用する必要があります。

Trident Protectのインストールと設定

環境がTrident Protect の要件を満たしている場合は、次の手順に従ってクラスターにTrident Protect をインストールできます。Trident Protect はNetAppから入手するか、独自のプライベートレジストリからインストールすることができます。クラスターがインターネットにアクセスできない場合は、プライベート レジストリからインストールすると便利です。

Trident Protectをインストールする

NetAppからTrident Protectをインストールする

手順

1. Trident Helmリポジトリを追加します。

```
helm repo add netapp-trident-protect  
https://netapp.github.io/trident-protect-helm-chart
```

2. Trident Protect CRD をインストールします。

```
helm install trident-protect-crds netapp-trident-protect/trident-  
protect-crds --version 100.2502.0 --create-namespace --namespace  
trident-protect
```

3. Helm を使用してTrident Protect をインストールします。交換する`<name-of-cluster>`クラスター名。このクラスター名はクラスターに割り当てられ、クラスターのバックアップとスナップショットを識別するために使用されます。

```
helm install trident-protect netapp-trident-protect/trident-protect  
--set clusterName=<name-of-cluster> --version 100.2502.0 --create  
-namespace --namespace trident-protect
```

プライベートレジストリからTrident Protectをインストールする

Kubernetes クラスターがインターネットにアクセスできない場合は、プライベートイメージレジストリからTrident Protect をインストールできます。これらの例では、括弧内の値を環境の情報に置き換えます。

手順

1. 次のイメージをローカルマシンにプルし、タグを更新して、プライベートレジストリにプッシュします。

```
netapp/controller:25.02.0  
netapp/restic:25.02.0  
netapp/kopia:25.02.0  
netapp/trident-autosupport:25.02.0  
netapp/exechook:25.02.0  
netapp/resourcebackup:25.02.0  
netapp/resourcerestore:25.02.0  
netapp/resourcedelete:25.02.0  
bitnami/kubectl:1.30.2  
kubebuilder/kube-rbac-proxy:v0.16.0
```

例えば：

```
docker pull netapp/controller:25.02.0
```

```
docker tag netapp/controller:25.02.0 <private-registry-url>/controller:25.02.0
```

```
docker push <private-registry-url>/controller:25.02.0
```

2. Trident Protect システム名前空間を作成します。

```
kubectl create ns trident-protect
```

3. レジストリにログインします。

```
helm registry login <private-registry-url> -u <account-id> -p <api-token>
```

4. プライベートレジストリ認証に使用するプルシークレットを作成します。

```
kubectl create secret docker-registry regcred --docker  
-username=<registry-username> --docker-password=<api-token> -n  
trident-protect --docker-server=<private-registry-url>
```

5. Trident Helmリポジトリを追加します。

```
helm repo add netapp-trident-protect  
https://netapp.github.io/trident-protect-helm-chart
```

6. という名前のファイルを作成します protectValues.yaml。次のTrident Protect 設定が含まれていることを確認します。

```
---
image:
  registry: <private-registry-url>
imagePullSecrets:
- name: regcred
controller:
  image:
    registry: <private-registry-url>
rbacProxy:
  image:
    registry: <private-registry-url>
crCleanup:
  imagePullSecrets:
- name: regcred
webhooksCleanup:
  imagePullSecrets:
- name: regcred
```

7. Trident Protect CRD をインストールします。

```
helm install trident-protect-crds netapp-trident-protect/trident-
protect-crds --version 100.2502.0 --create-namespace --namespace
trident-protect
```

8. Helm を使用してTrident Protect をインストールします。交換する`<name_of_cluster>`クラスター名。このクラスターネームはクラスターに割り当てられ、クラスターのバックアップとスナップショットを識別するために使用されます。

```
helm install trident-protect netapp-trident-protect/trident-protect
--set clusterName=<name_of_cluster> --version 100.2502.0 --create
--namespace --namespace trident-protect -f protectValues.yaml
```

Trident Protect CLIプラグインをインストールする

Trident Protectコマンドラインプラグインを使用できます。これはTridentの拡張機能です。 tridentctl Trident Protect カスタム リソース (CR) を作成し、操作するためのユーティリティです。

Trident Protect CLIプラグインをインストールする

コマンドラインユーティリティを使用する前に、クラスタへのアクセスに使用するマシンにインストールする必要があります。マシンがx64またはARM CPUを使用しているかどうかに応じて、次の手順を実行します。

Linux AMD64 CPU用プラグインのダウンロード

手順

1. Trident Protect CLI プラグインをダウンロードします。

```
curl -L -o tridentctl-protect https://github.com/NetApp/tridentctl-protect/releases/download/25.02.0/tridentctl-protect-linux-amd64
```

Linux ARM64 CPU用プラグインのダウンロード

手順

1. Trident Protect CLI プラグインをダウンロードします。

```
curl -L -o tridentctl-protect https://github.com/NetApp/tridentctl-protect/releases/download/25.02.0/tridentctl-protect-linux-arm64
```

Mac AMD64 CPU用プラグインのダウンロード

手順

1. Trident Protect CLI プラグインをダウンロードします。

```
curl -L -o tridentctl-protect https://github.com/NetApp/tridentctl-protect/releases/download/25.02.0/tridentctl-protect-macos-amd64
```

Mac ARM64 CPU用プラグインのダウンロード

手順

1. Trident Protect CLI プラグインをダウンロードします。

```
curl -L -o tridentctl-protect https://github.com/NetApp/tridentctl-protect/releases/download/25.02.0/tridentctl-protect-macos-arm64
```

1. プラグインバイナリの実行権限を有効にします。

```
chmod +x tridentctl-protect
```

2. プラグインバイナリをPATH変数で定義されている場所にコピーします。たとえば、 /usr/bin、または `/usr/local/bin (昇格されたPrivilegesが必要な場合があります)。

```
cp ./tridentctl-protect /usr/local/bin/
```

- 必要に応じて、プラグインバイナリをホームディレクトリ内の場所にコピーできます。この場合、locationがPATH変数の一部であることを確認することをお勧めします。

```
cp ./tridentctl-protect ~/bin/
```



プラグインをPATH変数の場所にコピーすると、任意の場所からまたはを`tridentctl protect`入力してプラグインを使用でき`tridentctl-protect`ます。

Trident CLI プラグインのヘルプを表示

組み込みプラグインヘルプ機能を使用して、プラグインの機能に関する詳細なヘルプを表示できます。

手順

- ヘルプ機能を使用して、使用方法に関するガイダンスを表示します。

```
tridentctl-protect help
```

コマンドの自動補完を有効にする

Trident Protect CLI プラグインをインストールした後、特定のコマンドの自動補完を有効にすることができます。

Bashシェルの自動補完を有効にする

手順

- 完了スクリプトをダウンロードします。

```
curl -L -O https://github.com/NetApp/tridentctl-  
protect/releases/download/25.02.0/tridentctl-completion.bash
```

- ホームディレクトリにスクリプトを格納する新しいディレクトリを作成します。

```
mkdir -p ~/.bash/completions
```

- ダウンロードしたスクリプトをディレクトリに移動し `~/.bash/completions` ます。

```
mv tridentctl-completion.bash ~/.bash/completions/
```

- ホームディレクトリ内のファイルに次の行を追加し `~/.bashrc` ます。

```
source ~/.bash/completions/tridentctl-completion.bash
```

Zシェルの自動補完を有効にする

手順

- 完了スクリプトをダウンロードします。

```
curl -L -O https://github.com/NetApp/tridentctl-  
protect/releases/download/25.02.0/tridentctl-completion.zsh
```

- ホームディレクトリにスクリプトを格納する新しいディレクトリを作成します。

```
mkdir -p ~/.zsh/completions
```

- ダウンロードしたスクリプトをディレクトリに移動し `~/.zsh/completions` ます。

```
mv tridentctl-completion.zsh ~/.zsh/completions/
```

- ホームディレクトリ内のファイルに次の行を追加し `~/.zprofile` ます。

```
source ~/.zsh/completions/tridentctl-completion.zsh
```

結果

次のシェルログイン時に、tridentctl-protectプラグインで自動補完コマンドを使用できます。

Trident Protectのインストールをカスタマイズする

環境の特定の要件を満たすように、Trident Protect のデフォルト構成をカスタマイズできます。

Trident Protectコンテナのリソース制限を指定する

Trident Protect をインストールした後、構成ファイルを使用してTrident Protect コンテナーのリソース制限を指定できます。リソース制限を設定すると、Trident Protect 操作によって消費されるクラスターのリソースの量を制御できます。

手順

1. という名前のファイルを作成します resourceLimits.yaml。
2. 環境のニーズに応じて、Trident Protect コンテナのリソース制限オプションをファイルに入力します。

次の構成ファイルの例は、使用可能な設定を示しています。このファイルには、各リソース制限のデフォルト値が含まれています。

```
---
jobResources:
  defaults:
    limits:
      cpu: 8000m
      memory: 10000Mi
      ephemeralStorage: ""
    requests:
      cpu: 100m
      memory: 100Mi
      ephemeralStorage: ""
resticVolumeBackup:
  limits:
    cpu: ""
    memory: ""
    ephemeralStorage: ""
  requests:
    cpu: ""
    memory: ""
    ephemeralStorage: ""
```

```

resticVolumeRestore:
  limits:
    cpu: ""
    memory: ""
    ephemeralStorage: ""
  requests:
    cpu: ""
    memory: ""
    ephemeralStorage: ""
kopiaVolumeBackup:
  limits:
    cpu: ""
    memory: ""
    ephemeralStorage: ""
  requests:
    cpu: ""
    memory: ""
    ephemeralStorage: ""
kopiaVolumeRestore:
  limits:
    cpu: ""
    memory: ""
    ephemeralStorage: ""
  requests:
    cpu: ""
    memory: ""
    ephemeralStorage: ""

```

3. ファイルから値を適用し `resourceLimits.yaml` ます。

```

helm upgrade trident-protect -n trident-protect netapp-trident-
protect/trident-protect -f resourceLimits.yaml --reuse-values

```

セキュリティコンテキスト制約のカスタマイズ

Trident Protect をインストールした後、構成ファイルを使用して、Trident Protect コンテナーの OpenShift セキュリティ コンテキスト制約 (SCC) を変更できます。これらの制約は、Red Hat OpenShift クラスター内のポッドのセキュリティ制限を定義します。

手順

1. という名前のファイルを作成します `sccconfig.yaml`。
2. SCCオプションをファイルに追加し、環境のニーズに応じてパラメータを変更します。

次に、SCCオプションのパラメータのデフォルト値の例を示します。

```

scc:
  create: true
  name: trident-protect-job
  priority: 1

```

次の表では、SCCオプションのパラメータについて説明します。

パラメータ	製品説明	デフォルト
作成	SCCリソースを作成できるかどうかを決定します。SCCリソースは、がに設定され `true`、HelmのインストールプロセスでOpenShift環境が指定されている場合にのみ作成され `scc.create` ます。OpenShiftで動作していない場合、またはがに設定されている `false` 場合 `scc.create`、SCCリソースは作成されません。	正しい
名前	SCCの名前を指定します。	Trident - protect-job
優先度	SCCのプライオリティを定義します。優先度の高いSCCSは、低い値のSCCSよりも先に評価されます。	1

3. ファイルから値を適用し `sccconfig.yaml` ます。

```

helm upgrade trident-protect netapp-trident-protect/trident-protect -f
sccconfig.yaml --reuse-values

```

これにより、デフォルト値がファイルで指定された値に置き換えられ `sccconfig.yaml` ます。

Trident Protect のNetApp AutoSupport接続を構成する

接続用のプロキシを設定することで、Trident Protect がNetAppサポートに接続してサポート バンドルをアップロードする方法を変更できます。ニーズに応じて、安全な接続または安全でない接続のいずれかを使用するようにプロキシを設定できます。

セキュアプロキシ接続の設定

手順

1. Trident Protect サポート バンドルのアップロード用に安全なプロキシ接続を構成します。

```
helm upgrade trident-protect -n trident-protect netapp-trident-  
protect/trident-protect --set autoSupport.proxy=http://my.proxy.url  
--reuse-values
```

セキュアでないプロキシ接続を設定する

手順

1. TLS 検証をスキップする、Trident Protect サポート バンドルのアップロード用の安全でないプロキシ接続を構成します。

```
helm upgrade trident-protect -n trident-protect netapp-trident-  
protect/trident-protect --set autoSupport.proxy=http://my.proxy.url  
--set autoSupport.insecure=true --reuse-values
```

Trident Protectポッドを特定のノードに制限する

Kubernetes nodeSelector ノード選択制約を使用すると、ノード ラベルに基づいて、どのノードがTrident Protect ポッドを実行できるかを制御できます。デフォルトでは、Trident Protect は Linux を実行しているノードに制限されています。ニーズに応じてこれらの制約をさらにカスタマイズできます。

手順

1. という名前のファイルを作成します `nodeSelectorConfig.yaml`。
2. `nodeSelector`オプションをファイルに追加し、ファイルを変更してノードラベルを追加または変更して、環境のニーズに応じて制限します。たとえば、次のファイルにはデフォルトのOS制限が含まれていますが、特定の地域とアプリ名も対象としています。

```
nodeSelector:  
  kubernetes.io/os: linux  
  region: us-west  
  app.kubernetes.io/name: mysql
```

3. ファイルから値を適用し `nodeSelectorConfig.yaml` ます。

```
helm upgrade trident-protect -n trident-protect netapp-trident-  
protect/trident-protect -f nodeSelectorConfig.yaml --reuse-values
```

これにより、デフォルトの制限がファイルで指定した制限に置き換えられます

nodeSelectorConfig.yaml。

毎日のTrident Protect AutoSupportバンドルのアップロードを無効にする

オプションで、毎日スケジュールされているTrident Protect AutoSupportサポート バンドルのアップロードを無効にすることができます。



デフォルトでは、Trident Protect は、クラスターと管理対象アプリケーションに関するログ、メトリック、トポロジ情報など、開く可能性のあるNetAppサポート ケースに役立つサポート情報を収集します。Trident Protect は、これらのサポート バンドルを毎日スケジュールに従ってNetAppに送信します。手動で["サポートバンドルの生成"](#)いつでも。

手順

1. という名前のファイルを作成します autosupportconfig.yaml。
2. AutoSupportオプションをファイルに追加し、環境のニーズに応じてパラメータを変更します。

次の例は、AutoSupportオプションのパラメータのデフォルト値を示しています。

```
autoSupport:  
  enabled: true
```

`autoSupport.enabled`をに設定する `false`と、
AutoSupportサポートバンドルの日次アップロードが無効になります。

3. ファイルから値を適用し `autosupportconfig.yaml` ます。

```
helm upgrade trident-protect netapp-trident-protect/trident-protect -f  
autosupportconfig.yaml --reuse-values
```

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隨時、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 5225.227-19（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。