



# セットアップ

## NetApp Solutions

NetApp  
March 04, 2025

# 目次

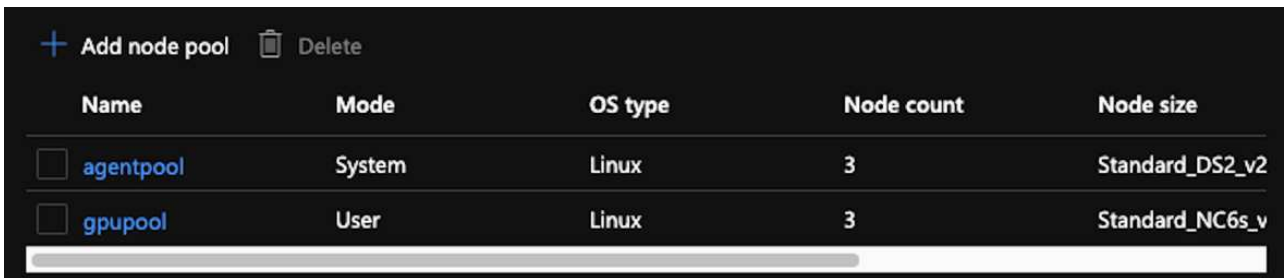
セットアップ .....	1
AKS クラスターをインストールしてセットアップします .....	1
Azure NetApp Files の委譲されたサブネットを作成します .....	2
ピア AKS の VNet と Azure NetApp Files VNet .....	4
Trident をインストール .....	4
Helm を使用して、AKS で Rapids デプロイメントを使用して Dask をセットアップします .....	7
Azure NetApp Files のパフォーマンス階層 .....	9
ボリュームのサービスレベルを動的に変更する .....	9
パフォーマンス階層の変更を自動化 .....	10

# セットアップ

## AKS クラスタをインストールしてセットアップします

AKS クラスタをインストールしてセットアップするには、Web ページを参照して ["AKS クラスタを作成します"](#) から、次の手順を実行します。

1. ノードのタイプ（system [CPU] ノードまたは worker [GPU] ノード）を選択するときは、次のいずれかを選択します。
  - a. プライマリシステムノードは、標準 DS2v2 デフォルトの 3 ノードである必要があります（`agentpool` ます）。
  - b. 次に、という名前のユーザグループ（GPU ノードの場合）に、ワーカーノード Standard\_NC6s\_v3 プール（最小 3 ノード）を追加します gpupool。



Name	Mode	OS type	Node count	Node size
<input type="checkbox"/> agentpool	System	Linux	3	Standard_DS2_v2
<input type="checkbox"/> gpupool	User	Linux	3	Standard_NC6s_v3

2. 導入には 5 ~ 10 分かかります。完了したら、Connect to Cluster（クラスタへの接続）をクリックします。
3. 新しく作成した AKS クラスタに接続するには、ローカル環境（ラップトップ / PC）から次のものをインストールします。
  - a. Kubernetes コマンドラインツールで ["使用している OS に応じた手順が表示されます"](#)
  - b. Azure CLI（のドキュメントを参照） ["Azure CLI をインストールします"](#)
4. ターミナルから AKS クラスタにアクセスするには、`az login` クレデンシャルを入力して入力します。
5. 次の 2 つのコマンドを実行します。

```
az account set --subscription xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxxxxxxxx
aks get-credentials --resource-group resourcegroup --name aksclustername
```

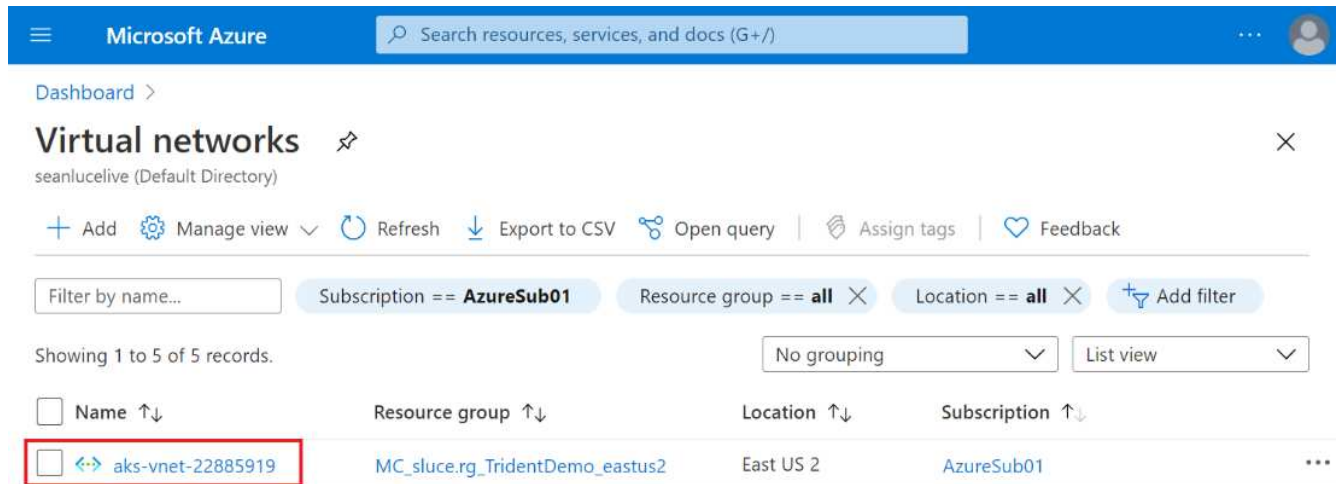
6. と入力し `Azure CLI: kubectl get nodes` ます。
7. 次の例に示すように、6 つのノードがすべて稼働していれば、AKS クラスタをローカル環境に接続することができます

```
verronmartina@verron-mac-0 ~ % kubectl get nodes
NAME                                STATUS    ROLES    AGE   VERSION
aks-agentpool-34613062-vmss000000  Ready    agent    22m   v1.18.14
aks-agentpool-34613062-vmss000001  Ready    agent    22m   v1.18.14
aks-agentpool-34613062-vmss000002  Ready    agent    22m   v1.18.14
aks-gpupool-34613062-vmss000000    Ready    agent    20m   v1.18.14
aks-gpupool-34613062-vmss000001    Ready    agent    20m   v1.18.14
aks-gpupool-34613062-vmss000002    Ready    agent    20m   v1.18.14
verronmartina@verron-mac-0 ~ %
```

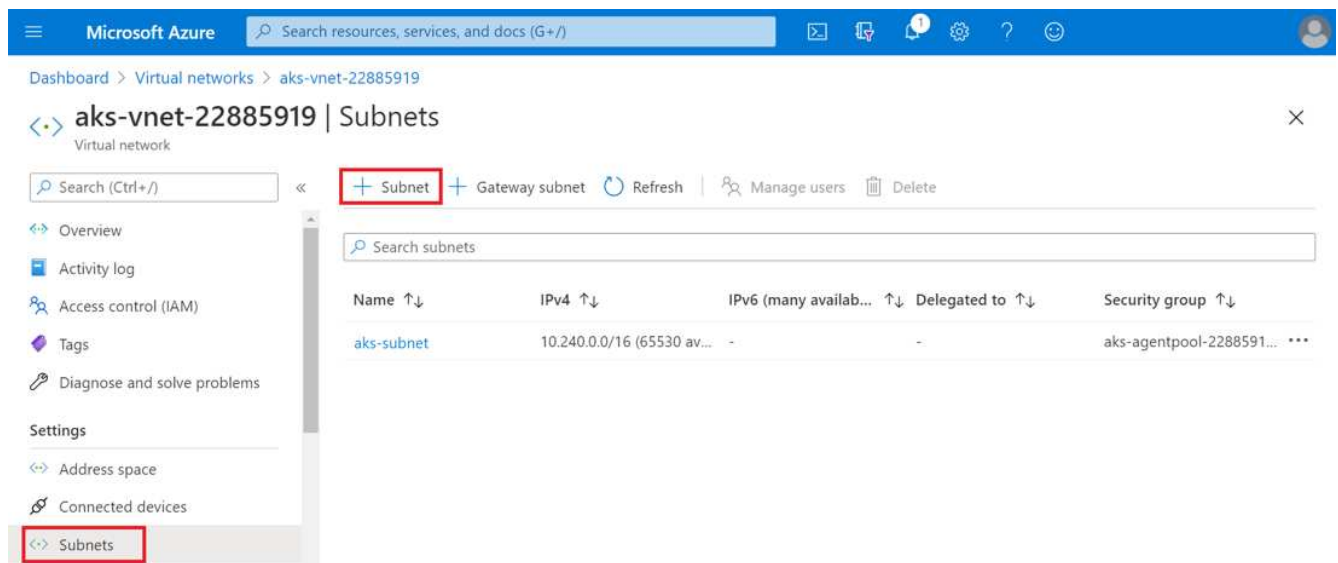
## Azure NetApp Files の委任されたサブネットを作成します

Azure NetApp Files の委任されたサブネットを作成するには、次の手順を実行します。

1. Azure ポータル内の仮想ネットワークに移動します。新しく作成した仮想ネットワークを検索します。プレフィックスは、などにする必要があります aks-vnet。
2. VNet の名前をクリックします。



3. [サブネット] をクリックし、上部のツールバーの [サブネット] をクリックします。



4. サブネットに、やなどの名前を指定します。 ANF.sn[Subnet Delegation]見出しでを選択します Microsoft.Netapp/volumes。他のものは変更しないでください。[OK]をクリックします。

## Add subnet ×

Name \*  ✓

Subnet address range \* ⓘ  10.0.0.0 - 10.0.0.255 (251 + 5 Azure reserved addresses)

Add IPv6 address space ⓘ

NAT gateway ⓘ  ▼

Network security group  ▼

Route table  ▼

### SERVICE ENDPOINTS

Create service endpoint policies to allow traffic to specific azure resources from your virtual network over service endpoints. [Learn more](#)

Services ⓘ  ▼

### SUBNET DELEGATION

Delegate subnet to a service ⓘ  ▼

Azure NetApp Files ボリュームはアプリケーションクラスタに割り当てられ、Kubernetes で永続ボリューム要求（PVC）として使用されます。その結果、Jupyter ノートブック、サーバーレス関数などのさまざまなサービスに柔軟にマップできます。

サービスのユーザは、プラットフォームのストレージをさまざまな方法で消費できます。このテクニカルレポートでは NFS について説明しているため、Azure NetApp Files の主なメリットは次のとおりです。

- ユーザに Snapshot コピーを使用できるようにする。
- ユーザが Azure NetApp Files ボリュームに大量のデータを格納できるようにする。
- 大容量のファイルセットでモデルを実行する場合、Azure NetApp Files のパフォーマンスが向上します。

# ピア AKS の VNet と Azure NetApp Files VNet

AKS VNet を Azure NetApp Files VNet にピアリングするには、次の手順を実行します。

1. 検索フィールドに Virtual Networks と入力します。
2. `vnet aks-vnet-name.` クリックして検索フィールドに peerings と入力します。
3. + Add をクリックします。
4. 次の記述子を入力します。
  - a. ピアリングリンク名はです aks-vnet-name\_to\_anf。
  - b. VNet ピアリングパートナーとしての SubscriptionID および Azure NetApp Files VNet
  - c. アスタリスク以外のすべてのセクションは、デフォルト値のままにします。
5. 追加をクリックします。

詳細については、を参照してください ["仮想ネットワークピアリングを作成、変更、削除します"](#)。

## Trident をインストール

Helm を使用して Trident をインストールするには、次の手順を実行します。

1. Helm をインストールします（インストール手順については、を参照してください ["ソース"](#)）。
2. Trident 20.01.1 インストーラをダウンロードして展開します。

```
$wget  
$star -xf trident-installer-21.01.1.tar.gz
```

3. ディレクトリをに変更します trident-installer。

```
$cd trident-installer
```

4. システムのディレクトリに ` \$PATH ` コピーし `tridentctl` ます。

```
$sudo cp ./tridentctl /usr/local/bin
```

5. Helm ( ) を使用して Kubernetes (K8s) クラスタに Trident をインストールし ["ソース"](#) ます。
  - a. ディレクトリをディレクトリに変更します helm。

```
$cd helm
```

- b. Trident をインストール

```
$helm install trident trident-operator-21.01.1.tgz --namespace
trident --create-namespace
```

c. Trident ポッドのステータスを確認

```
$kubectl -n trident get pods
```

すべてのポッドが稼働中の場合は、Trident がインストールされてから次のポッドに移動できます。

6. AKS の Azure NetApp Files バックエンドとストレージクラスをセットアップします。

a. Azure サービスプリンシパルを作成します。

サービスプリンシパルは、Trident が Azure と通信して Azure NetApp Files リソースを操作する方法を示します。

```
$az ad sp create-for-rbac --name ""
```

出力は次の例のようになります。

```
{
  "appId": "xxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
  "displayName": "netapptrident",
  "name": "",
  "password": "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
  "tenant": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx"
}
```

7. TridentバックエンドJSONファイル（例：name）を作成します anf-backend.json。

8. 任意のテキストエディタを使用して、ファイル内の次のフィールドに値を入力し `anf-backend.json` ます。

```

{
  "version": 1,
  "storageDriverName": "azure-netapp-files",
  "subscriptionID": "fakec765-4774-fake-ae98-a721add4fake",
  "tenantID": "fakef836-edc1-fake-bff9-b2d865eefake",
  "clientID": "fake0f63-bf8e-fake-8076-8de91e57fake",
  "clientSecret": "SECRET",
  "location": "westeurope",
  "serviceLevel": "Standard",
  "virtualNetwork": "anf-vnet",
  "subnet": "default",
  "nfsMountOptions": "vers=3,proto=tcp",
  "limitVolumeSize": "500Gi",
  "defaults": {
    "exportRule": "0.0.0.0/0",
    "size": "200Gi"
  }
}

```

9. 次のフィールドを置き換えます。

- `subscriptionID` です。お客様の Azure サブスクリプション ID
- `tenantID` です。前の手順の出力に表示された Azure テナント ID `az ad sp`。
- `clientID` です。前の手順での出力から取得した appID `az ad sp`。
- `clientSecret` です。前の手順の出力で確認したパスワード `az ad sp`。

10. を構成ファイルとして使用して、ネームスペースに `anf-backend.json` Azure NetApp Files バックエンドを作成するように Trident に指示し `trident` ます。

```
$tridentctl create backend -f anf-backend.json -n trident
```

NAME	STORAGE DRIVER	UUID	STATE	VOLUMES
azurenetappfiles_86181	azure-netapp-files	2ca85462-59ac-4946-be05-c03f5575a2ad	online	0

11. ストレージクラスを作成する。Kubernetes ユーザは、名前ストレージクラスを指定する PVC を使用してボリュームをプロビジョニングします。Kubernetes で、前の手順で作成した Trident バックエンドを参照するストレージクラスを作成するように指示します `azurenetappfiles`。

12. (`anf-storage-class.yaml` ストレージクラス用の YAML) ファイルを作成し、コピーします。



```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: azurenetappfiles
provisioner: netapp.io/trident
parameters:
  backendType: "azure-netapp-files"
$kubectl create -f anf-storage-class.yaml
```

13. ストレージクラスが作成されたことを確認します。

```
kubectl get sc azurenetappfiles
```

NAME	PROVISIONER	RECLAIMPOLICY	VOLUMEBINDINGMODE	ALLOWVOLUMEEXPANSION	AGE
azurenetappfiles	csi.trident.netapp.io	Delete	Immediate	false	98s

## Helm を使用して、AKS で Rapids デプロイメントを使用して Dask をセットアップします

Helm を使用して AKS で Rapids を使用して Dask をセットアップするには、次の手順を実行します。

1. Dask with Rapids をインストールするための名前空間を作成します。

```
kubectl create namespace rapids-dask
```

2. クリックスルーレートデータセットを保存する PVC を作成します。
  - a. 次の YAML コンテンツをファイルに保存して PVC を作成します。

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: pvc-criteo-data
spec:
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 1000Gi
  storageClassName: azurenetappfiles
```

- b. YAML ファイルを Kubernetes クラスタに適用します。

```
kubectl -n rapids-dask apply -f <your yaml file>
```

3. リポジトリのクローンを作成し `rapidsai git` ます "<https://github.com/rapidsai/helm-chart>" () 。

```
git clone https://github.com/rapidsai/helm-chart helm-chart
```

4. 作業者とJupyterワークスペース用に以前に作成したPVCを修正し `values.yaml` で含めます。

- a. リポジトリのディレクトリに移動し `rapidsai` ます。

```
cd helm-chart/rapidsai
```

- b. ファイルを更新し `values.yaml`、PVCを使用してボリュームをマウントします。

```
dask:
  ...
  worker:
    name: worker
  ...
  mounts:
    volumes:
      - name: data
        persistentVolumeClaim:
          claimName: pvc-criteo-data
    volumeMounts:
      - name: data
        mountPath: /data
  ...
  jupyter:
    name: jupyter
  ...
  mounts:
    volumes:
      - name: data
        persistentVolumeClaim:
          claimName: pvc-criteo-data
    volumeMounts:
      - name: data
        mountPath: /data
  ...
```

- リポジトリのホーム・ディレクトリに移動し 'Helm' を使用して AKS 上に 3 つのワーカー・ノードを持つ Dask を展開します

```
cd ..
helm dep update rapidsai
helm install rapids-dask --namespace rapids-dask rapidsai
```

## Azure NetApp Files のパフォーマンス階層

既存のボリュームのサービスレベルを変更するには、そのボリュームに必要なサービスレベルを使用する別の容量プールにボリュームを移動します。この解決策を使用することで、お客様は、まず小規模なデータセットと少数の GPU を標準階層に配置し、データ量と GPU の増加に合わせてスケールアウトまたは Premium Tier へのスケールアップを行うことができます。Premium Tier は、Standard 階層のテラバイトあたりスループットの 4 倍を提供し、ボリュームのサービスレベルを変更するためにデータを移動することなくスケールアップを実行できます。

### ボリュームのサービスレベルを動的に変更する

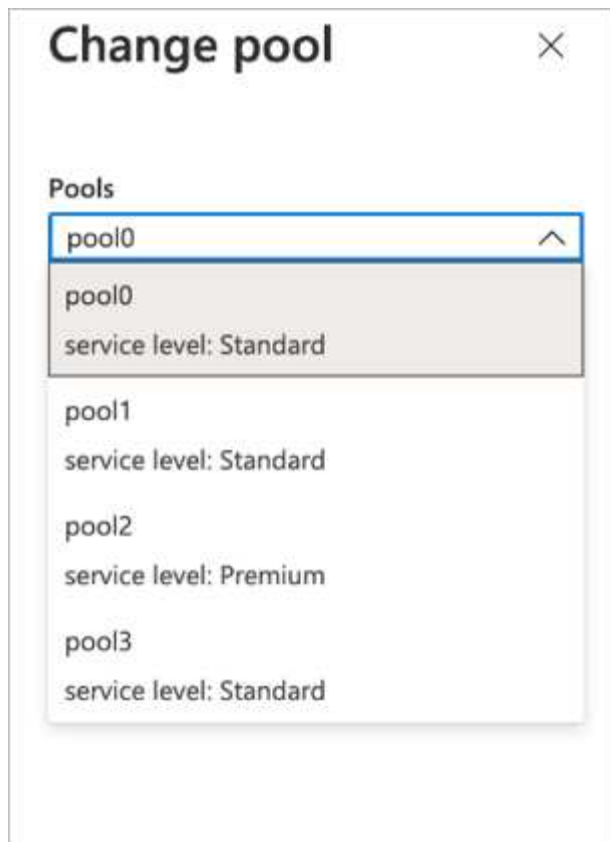
ボリュームのサービスレベルを動的に変更するには、次の手順を実行します。

- Volumes (ボリューム) ページで、サービスレベルを変更するボリュームを右クリックします。[ プールの変更 ] を選択します

NFSv3	10.28.254.4:/norootfor	Standard	pool0	...
NFSv4.1	NAS-735a.docs.lab:/fox	Premium		...
NFSv4.1	NAS-735a.docs.lab:/krt	Premium		...
NFSv3	10.28.254.4:/moveme0	Premium		...
NFSv3	10.28.254.4:/placeholder	Premium		...



- プールの変更ウィンドウで、ボリュームの移動先となる容量プールを選択します。



3. [OK]をクリックします。

## パフォーマンス階層の変更を自動化

パフォーマンス階層の変更を自動化するには、次のオプションを使用できます。

- 現在も動的サービスレベルの変更はパブリックプレビューで有効になっており、デフォルトでは有効になっていません。Azureサブスクリプションでこの機能を有効にするには、[の方法に関するドキュメント](#)を参照してください "[ボリュームのサービスレベルを動的に変更する](#)"。
- 次の例では、Azure CLIのボリュームプール変更コマンドをおよびに "[ボリュームプールの変更に関するドキュメント](#)"示します。

```
az netappfiles volume pool-change -g mygroup --account-name myaccname  
--pool-name mypoolname --name myvolname --new-pool-resource-id  
mynewresourceid
```

- PowerShell：では、"[set-AzNetAppFilesVolumePool コマンドレット](#)"Azure NetApp Filesボリュームのプールが変更されます。次の例を参照してください。

```
Set-AzNetAppFilesVolumePool
-ResourceGroupName "MyRG"
-AccountName "MyAnfAccount"
-PoolName "MyAnfPool"
-Name "MyAnfVolume"
-NewPoolResourceId 7d6e4069-6c78-6c61-7bf6-c60968e45fbf
```

## 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。