



解決策の自動化

NetApp Solutions

NetApp
April 10, 2024

This PDF was generated from https://docs.netapp.com/ja-jp/netapp-solutions/automation/automation_introduction.html on April 10, 2024. Always check docs.netapp.com for the latest.

目次

解決策の自動化	1
NetApp 解決策の自動化	1
NetApp解決策 自動化の導入	1
NetApp 解決策の自動化	6
Terraform による Cloud Volume オートメーション	8
AWS Lambda関数を使用したFSx for ONTAP の監視と自動サイズ変更	51

解決策の自動化

NetApp 解決策の自動化

はじめに

ネットアップは、今日のビジネス上の課題に対処するソリューションを提供するにあたり、次の目標を達成したソリューションを提供しています。

- 検証済みの導入と設定手順を提供し、
- 簡単に利用できるソリューションを提供する
- 予測可能な成果をもたらす解決策 環境を提供することは、お客様の企業全体で同じことを繰り返して容易に拡張することができます。

これらの目標を達成するためには、当社のソリューションを通じて提供されるインフラやアプリケーションの導入と構成を自動化によって簡易化することが何よりも重要です。ネットアップは、自動化を通じて解決策消費を簡易化することに取り組んでいます。

ネットアップのソリューションには、Red Hat Ansible、H橋本、Microsoft PowerShellなどのオープンソースの自動化ツールが採用されており、アプリケーションの導入、クラウドのプロビジョニング、設定管理など、一般的なITタスクを自動化できます。ネットアップのソリューションは、一般に公開されている自動化アーティファクトを活用し、ネットアップが独自に開発した自動化機能を提供することで、解決策 の全体的な導入を簡易化します。

自動化機能が利用可能な場合は、特定の自動化ツールを使用して解決策 や解決策 の手順を自動化するプロセスを、解決策 の資料で順を追って説明します。

NetApp解決策 自動化の導入

ネットアップのソリューションで使用される一般的なタスクの多くに、NetApp解決策 の自動化機能でシンプルさと再現性が提供されます。

解決策 の自動化を実行する前に、自動化の実行方法に合わせて環境を設定する必要があります。コマンドラインから、またはAWXやタワーなどのツールを使用して自動化を実行するオプションがあります。

以降のセクションでは、それぞれの環境について、環境を設定するために必要な手順について説明します。

RHEL に CLI を導入するための Ansible Control Node をセットアップします /CentOS

1. Ansible コントロールノードの要件：

a. 次のパッケージがインストールされた RHEL / CentOS マシン：

- i. Python3
- ii. Pip3
- iii. Ansible （バージョン 2.10.0 より前）
- iv. Git

上記の要件がインストールされていない新しい RHEL / CentOS マシンがある場合は、次の手順に従ってそのマシンを Ansible の制御ノードとしてセットアップします。

1. RHEL-8/RHEL-7 の Ansible リポジトリを有効にします

a. RHEL-8 （root として次のコマンドを実行）

```
subscription-manager repos --enable ansible-2.9-for-rhel-8-  
x86_64-rpms
```

b. RHEL-7 （root として次のコマンドを実行）

```
subscription-manager repos --enable rhel-7-server-ansible-2.9-  
rpms
```

2. 以下の内容をターミナルに貼り付けます

```
sudo yum -y install python3 >> install.log  
sudo yum -y install python3-pip >> install.log  
python3 -W ignore -m pip --disable-pip-version-check install ansible  
>> install.log  
sudo yum -y install git >> install.log
```

Ubuntu で CLI 環境に Ansible Control Node をセットアップします /Debian

1. Ansible コントロールノードの要件：

a. 次のパッケージがインストールされた Ubuntu / Debian マシン：

- i. Python3
- ii. Pip3
- iii. Ansible （バージョン 2.10.0 より前）
- iv. Git

上記の要件がインストールされていない新しい Ubuntu / Debian マシンを使用している場合は、次の手順に従ってそのマシンを Ansible の制御ノードとしてセットアップします。

1. 以下の内容を端末に貼り付けます

```
sudo apt-get -y install python3 >> outputlog.txt
sudo apt-get -y install python3-pip >> outputlog.txt
python3 -W ignore -m pip --disable-pip-version-check install ansible
>> outputlog.txt
sudo apt-get -y install git >> outputlog.txt
```

このセクションでは、ネットアップの自動化ソリューションを使用する環境を準備する AWX/Ansible タワーのパラメータを設定するために必要な手順について説明します。

1. インベントリを設定します。
 - a. リソース→インベントリ→追加と進み、インベントリの追加をクリックします。
 - b. 名前と組織の詳細を入力し、[保存] をクリックします。
 - c. インベントリページで、作成したインベントリリソースをクリックします。
 - d. インベントリ変数がある場合は、その変数を変数フィールドに貼り付けます。
 - e. [グループ] サブメニューに移動し、[追加] をクリックします。
 - f. グループの名前を入力し、必要に応じてグループ変数にコピーして、[保存] をクリックします。
 - g. 作成したグループをクリックし、Hosts サブメニューに移動して、Add New Host をクリックします。
 - h. ホストのホスト名と IP アドレスを入力し、必要に応じてホスト変数に貼り付けて、Save をクリックします。
2. クレデンシャルタイプを作成する。ONTAP、Element、VMware、またはその他の HTTPS ベースの転送接続を使用するソリューションの場合は、ユーザ名とパスワードのエントリに一致するクレデンシャルタイプを設定する必要があります。
 - a. [管理] → [資格情報の種類] に移動して、[追加] をクリックし
 - b. 名前と概要を指定します。
 - c. 次の内容を入力構成に貼り付けます。

```
fields:
- id: username
  type: string
  label: Username
- id: password
  type: string
  label: Password
  secret: true
- id: vsadmin_password
  type: string
  label: vsadmin_password
  secret: true
```

1. インジェクタの設定に次の内容を貼り付けます。

```
extra_vars:
password: '{{ password }}'
username: '{{ username }}'
vsadmin_password: '{{ vsadmin_password }}'
```

1. クレデンシャルの設定

- a. リソース→資格情報に移動して、追加をクリックします。
- b. 名前と組織の詳細を入力します
- c. 正しいクレデンシャルタイプを選択します。標準 SSH ログインを使用する場合は、「Machine」タイプを選択するか、作成したカスタムクレデンシャルタイプを選択します。
- d. 対応するその他の詳細情報を入力し、[保存]をクリックします。

2. プロジェクトを設定します。

- a. リソース→プロジェクトに移動し、追加をクリックします
- b. 名前と組織の詳細を入力します
- c. ソース管理資格情報タイプとして Git を選択します。
- d. 特定の解決策に対応するソース制御 URL（または git クローン URL）を貼り付けます。
- e. Git URL がアクセス制御されている場合は、必要に応じて、Source Control Credential で対応するクレデンシャルを作成して添付します。
- f. [保存]をクリックします。

3. ジョブテンプレートを設定します。

- a. [リソース]→[テンプレート]→[追加]に移動し、[ジョブテンプレートの追加]をクリックします。
- b. 名前と概要を入力します
- c. ジョブタイプを選択します。Run は、プレイブックに基づいてシステムを設定し、Check は実際にシステムを設定することなく、プレイブックの事前チェックを実行します。
- d. このプレイブックに対応するインベントリ、プロジェクト、クレデンシャルを選択します
- e. ジョブテンプレートの一部として実行するプレイブックを選択します。
- f. 通常、変数は実行時に貼り付けられます。そのため、実行時に変数を入力するように求めるプロンプトを表示するには、必ず [変数] フィールドに対応する [起動時にプロンプトを表示] チェックボックスをオンにしてください。
- g. 必要に応じてその他の詳細情報を入力し、[保存]をクリックします。

4. ジョブテンプレートを起動します。

- a. [リソース]→[テンプレート]に移動します。
- b. 目的のテンプレートをクリックし、[起動]をクリックします。
- c. 起動時にプロンプトが表示されたら変数を入力し、[再起動]をクリックします。

NetApp 解決策の自動化

NetApp Cloud Manager を使用した CVO と Connector の AWS 認証の要件

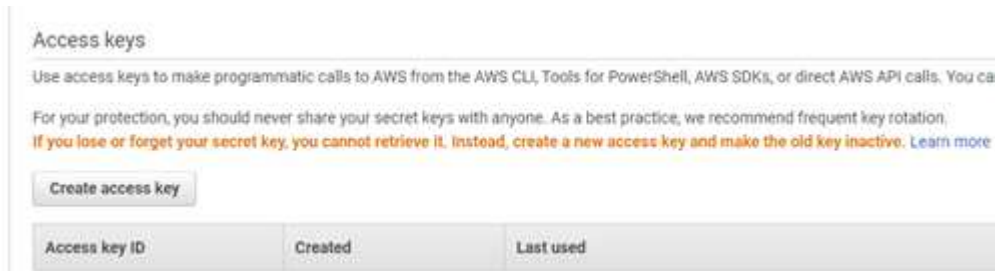
AWX/Ansible タワーを介した Ansible プレイブックを使用して、CVO とコネクタの自動導入を設定するには、次の情報が必要です。

AWS からアクセスキーとシークレットキーを取得する

1. Cloud Manager に CVO と Connector を導入するには、AWS Access/Secret Key が必要です。IAM → ユーザー → ユーザー名 → セキュリティ資格情報 → アクセスキーの作成を起動して、AWS コンソールでキーを取得します。
2. アクセスキーをコピーし、Connector および CVO の導入で使用するためのセキュリティを確保しておきます。



キーが紛失した場合は、別のアクセスキーを作成し、失われたアクセスキーを削除できます



NetApp Cloud Central から Refresh Token を取得しています

1. のアカウントクレデンシャルを使用して、Cloud Central アカウントにログインします
<https://services.cloud.netapp.com/refresh-token>
2. 更新トークンを生成し、展開用に保存します。

Refresh Token Generator

You can use this refresh token to obtain an access tokens for users. Store this refresh token securely. If necessary, you can revoke the token at a later time by navigating to the [Refresh Token Generator](#).

Note that this token is displayed on this page only—it is not stored on our servers. The token will no longer be displayed if you refresh or leave this page.

REFRESH TOKEN:

Copy to clipboard

EAAfPTMCuu4QJl9hR2PTRT75Lswr0fHp4BheEjT2XFst

クライアント ID を取得しています

1. API ページにアクセスして、クライアント ID をにコピーします <https://services.cloud.netapp.com/developer-hub>。
2. 右上にある [Learn how to Authenticate] をクリックします。

3. ユーザ名とパスワードを入力してログインする必要がある場合は、ポップアップ表示される [Authentication] ウィンドウから通常のアクセスからクライアント ID をコピーします。SSO を使用するフェデレーテッドユーザは、[トークンの更新] タブからクライアント ID をコピーする必要があります。

Authentication Information

×

NetApp Cloud Central Services use OAuth 2.0, an industry-standard protocol, for authorization.

Communicating with an authenticated endpoint is a two step-process.

1. Acquire a JWT access token from the OAuth token endpoint.
2. Call an API endpoint with the JWT access token.

Non-federated users can use regular access or refresh token access, federated users must use refresh token access.

[Regular Access](#) Refresh Token Access (Required for federated users)

How to Acquire a JWT Access Token via regular token access

1. Make an HTTP POST request to the endpoint

`https://netapp-cloud-account.auth0.com/oauth/token`

Include the header Content-Type: application/json

Include the body:

```
{
  "grant_type": "password",
  "username": "YOUR_EMAIL_ADDRESS",
  "password": "YOUR_PASSWORD",
  "audience": "https://api.cloud.netapp.com",
  "client_id": 
}
```

Copy to clipboard

AWS からキーペアを取得しています

1. AWS コンソールで、「キーペア」を検索し、「PEM」とのキーペアを作成します。ここでは、key_pair の名前を覚えておいてください。この名前を使用してコネクタを配置します。

EC2 > Key pairs > Create key pair

Create key pair

Key pair
A key pair, consisting of a private key and a public key, is a set of security credentials that you use to prove your identity when connecting to an instance.

Name:

The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

Private key file format
☒ pem
For use with OpenSSH
☐ ppk
For use with PuTTY

Tags (Optional)
No tags associated with the resource.

You can add 50 more tags.

アカウント ID を取得しています

1. Cloud Manager で、Account → Manage Accounts の順にクリックし、AWX の変数で使用するアカウント ID をコピーします。

account-LwOlqR76

Account ID

Terraform による Cloud Volume オートメーション

この解決策では、Terraform モジュールを使用して、AWS（CVO シングルノード、CVO HA、FSX ONTAP）および Azure（CVO シングルノード、CVO HA、ANF）への Cloud Volume の自動導入を文書化しています。コードは、から入手できます
https://github.com/NetApp/na_cloud_volumes_automation

前提条件

1. Terraform ≥ 0.13
2. Cloud Manager アカウント
3. クラウドプロバイダアカウント—AWS、Azure
4. ホストマシン（Terraform がサポートするすべての OS）

プロバイダのドキュメント

Cloud Manager の Terraform プロバイダのドキュメントは、次の URL から入手できます。
["https://registry.terraform.io/providers/NetApp/netapp-cloudmanager/latest/docs"](https://registry.terraform.io/providers/NetApp/netapp-cloudmanager/latest/docs)

プロバイダバージョンの制御

プロバイダのバージョンを制御することもできます。これは、Terraform 設定の `required_providers` ブロックによって制御されます。

構文は次のとおりです。

```
terraform {  
  required_providers {  
    netapp-cloudmanager = {  
      source = "NetApp/netapp-cloudmanager"  
      version = ">= 23.0.0"  
    }  
  }  
}
```

プロバイダバージョン管理の詳細については、こちらをご覧ください。

実行中の特定のモジュール



CVO シングルノードの導入

NetApp CVO（シングルノードインスタンス）を**AWS**に導入するための構成ファイルが**Terraform**にあります

このセクションでは、AWS（Amazon Web Services）上でシングルノードの NetApp CVO（Cloud Volumes ONTAP）を導入 / 構成するための、さまざまな Terraform 構成ファイルを提供しています。

Terraform ドキュメント：<https://registry.terraform.io/providers/NetApp/netapp-cloudmanager/latest/docs>

手順

テンプレートを実行するには、次の手順を実行します。

1. リポジトリをクローニングします。

```
git clone
https://github.com/NetApp/na_cloud_volumes_automation.git
```

2. 目的のフォルダに移動します

```
cd na_cloud_volumes_automation/
```

3. CLI から AWS クレデンシャルを設定する。

```
aws configure
```

- AWS アクセスキー ID [None] : アクセスキー
- AWS Secret Access Key [None] : secretkey
- デフォルトのリージョン名 [None] : us-west-2
- デフォルトの出力形式は [None] : json です

4. 変数 /AWS_CVO-SILE_Node_deployment.tfvar' の変数値を更新します



変数「AWS_Connector_deploy_bool」の値を true または false に設定することで、コネクタを導入することができます。

5. Terraform リポジトリを初期化して、すべての前提条件をインストールし、導入の準備をします。

```
terraform init
```

6. terraform validate コマンドを使用して、terraform ファイルを確認します。

```
terraform validate
```

7. 設定を事前に実行して、導入で想定されるすべての変更をプレビューします。

```
terraform plan -target="module.aws_sn" -var  
-file="vars/aws_cvo_single_node_deployment.tfvars"
```

8. 導入を実行

```
terraform apply -target="module.aws_sn" -var  
-file="vars/aws_cvo_single_node_deployment.tfvars"
```

展開を削除します

```
terraform destroy
```

降水量：

「コネクタ」

CVO 導入用の NetApp AWS Connector インスタンスの Terraform 変数。

* 名前 *	* タイプ *	* 概要 *
* AWS_Connector_ deploy_bool *	ブール 値	(必須) コネクタの配置を確認します。
* AWS_Connector_ name*	文字列	(必須) Cloud Manager Connector の名前。
* AWS_connector_r egion *	文字列	(必須) Cloud Manager Connector を作成するリージョン。
* AWS_Connector_ key_name*	文字列	(必須) コネクタインスタンスに使用するキーペアの名前。
* AWS_connector_c ompany *	文字列	(必須) ユーザの会社名。
* AWS_Connector_i nstance_type *	文字列	(必須) インスタンスのタイプ (t3.xlarge など) 。少なくとも 4 つ の CPU と 16 GB のメモリが必要です。

* 名前 *	* タイプ *	* 概要 *
* AWS_connector_subnet_id *	文字列	(必須) インスタンスのサブネットの ID。
* AWS_Connector_security_group_id *	文字列	(必須) インスタンスのセキュリティグループの ID。複数のセキュリティグループを区切って指定できます。
* AWS_Connector_iAM_instance_profile_name *	文字列	(必須) コネクタのインスタンスプロファイルの名前。
* AWS_Connector_account_id *	文字列	(オプション) コネクタに関連付けるネットアップアカウントの ID。指定しない場合、Cloud Manager は最初のアカウントを使用します。アカウントが存在しない場合、Cloud Manager は新しいアカウントを作成します。アカウント ID は、Cloud Manager のアカウントタブにあります https://cloudmanager.netapp.com 。
* AWS_connector_public_ip_bool *	ブール値	(任意) インスタンスにパブリック IP アドレスを関連付けるかどうかを指定します。指定しない場合は、サブネットの設定に基づいて関連付けが行われます。

「シングルノードインスタンス」

単一の NetApp CVO インスタンスの Terraform 変数。

* 名前 *	* タイプ *	* 概要 *
* CVO-NAME *	文字列	(必須) Cloud Volumes ONTAP 作業環境の名前。
* CVF_REGION *	文字列	(必須) 作業環境を作成するリージョン。
* CVO-subnet_id *	文字列	(必須) 作業環境を作成するサブネット ID。
* CVO-vPC_id *	文字列	(オプション) 作業環境を作成する VPC ID。この引数を指定しない場合は、指定したサブネット ID を使用して VPC が計算されます。
* CVO-svm_password* をクリックします	文字列	(必須) Cloud Volumes ONTAP の管理パスワード。
* CVF_Writing_speed_state *	文字列	(オプション) Cloud Volumes ONTAP の書き込み速度設定：[「normal」、「high」]。デフォルトは「normal」です。

CVO HA の導入

NetApp CVO (HAペア) を**AWS**に導入するための構成ファイルが**Terraform**にあります

このセクションでは、AWS (Amazon Web Services) のハイアベイラビリティペアに NetApp CVO (Cloud Volumes ONTAP) を導入 / 構成するための、さまざまな Terraform 構成ファイルを提供しています。

Terraform ドキュメント : <https://registry.terraform.io/providers/NetApp/netapp-cloudmanager/latest/docs>

手順

テンプレートを実行するには、次の手順を実行します。

1. リポジトリをクローニングします。

```
git clone
https://github.com/NetApp/na_cloud_volumes_automation.git
```

2. 目的のフォルダに移動します

```
cd na_cloud_volumes_automation/
```

3. CLI から AWS クレデンシャルを設定する。

```
aws configure
```

- AWS アクセスキー ID [None] : アクセスキー
- AWS Secret Access Key [None] : secretkey
- デフォルトのリージョン名 [None] : us-west-2
- デフォルトの出力形式は [None] : json です

4. 変数 /AWS_CVO-HA_DEVELOP.tfvars の変数値を更新します。



変数「AWS_Connector_deploy_bool」の値を true または false に設定することで、コネクタを導入することができます。

5. Terraform リポジトリを初期化して、すべての前提条件をインストールし、導入の準備をします。

```
terraform init
```

6. terraform validate コマンドを使用して、terraform ファイルを確認します。

```
terraform validate
```

7. 設定を事前に実行して、導入で想定されるすべての変更をプレビューします。


```
terraform plan -target="module.aws_ha" -var  
-file="vars/aws_cvo_ha_deployment.tfvars"
```

8. 導入を実行

```
terraform apply -target="module.aws_ha" -var  
-file="vars/aws_cvo_ha_deployment.tfvars"
```

展開を削除します

```
terraform destroy
```

降水量：

「コネクタ」

CVO 導入用の NetApp AWS Connector インスタンスの Terraform 変数。

* 名前 *	* タイプ *	* 概要 *
* AWS_Connector_ deploy_bool *	ブール 値	(必須) コネクタの配置を確認します。
* AWS_Connector_ name*	文字列	(必須) Cloud Manager Connector の名前。
* AWS_connector_r egion *	文字列	(必須) Cloud Manager Connector を作成するリージョン。
* AWS_Connector_ key_name*	文字列	(必須) コネクタインスタンスに使用するキーペアの名前。
* AWS_connector_c ompany *	文字列	(必須) ユーザの会社名。
* AWS_Connector_i nstance_type *	文字列	(必須) インスタンスのタイプ (t3.xlarge など) 。少なくとも 4 つ の CPU と 16 GB のメモリが必要です。
* AWS_connector_s ubnet_id *	文字列	(必須) インスタンスのサブネットの ID 。

* 名前 *	* タイプ *	* 概要 *
* AWS_Connector_security_group_id *	文字列	(必須) インスタンスのセキュリティグループの ID。複数のセキュリティグループを区切って指定できます。
* AWS_Connector_iAM_instance_profile_name *	文字列	(必須) コネクタのインスタンスプロファイルの名前。
* AWS_Connector_account_id *	文字列	(オプション) コネクタに関連付けるネットアップアカウントの ID。指定しない場合、Cloud Manager は最初アカウントを使用します。アカウントが存在しない場合、Cloud Manager は新しいアカウントを作成します。アカウント ID は、Cloud Manager のアカウントタブにあります https://cloudmanager.netapp.com 。
* AWS_connector_public_ip_bool *	ブール値	(任意) インスタンスにパブリック IP アドレスに関連付けるかどうかを指定します。指定しない場合は、サブネットの設定に基づいて関連付けが行われます。

HA ペア

HA ペアの NetApp CVO インスタンスの変数はテラフォームされます。

* 名前 *	* タイプ *	* 概要 *
* CVO-is_HA *	ブール値	(オプション) 作業環境が HA ペアであるかどうかを示します ([true、false])。デフォルトは false です。
* CVO-NAME *	文字列	(必須) Cloud Volumes ONTAP 作業環境の名前。
* CVF_REGION *	文字列	(必須) 作業環境を作成するリージョン。
* CVO-node1_subnet_id *	文字列	(必須) 最初のノードを作成するサブネット ID。
* CVO-node2_subnet_id *	文字列	(必須) 2 つ目のノードを作成するサブネット ID。
* CVO-vPC_id *	文字列	(オプション) 作業環境を作成する VPC ID。この引数を指定しない場合は、指定したサブネット ID を使用して VPC が計算されます。
* CVO-svm_password* をクリックします	文字列	(必須) Cloud Volumes ONTAP の管理パスワード。
* CVF_Failover_mode *	文字列	(任意) HA の場合、HA ペアのフェイルオーバーモード: [PrivateIP]、[FloatingIP]。「PrivateIP」は 1 つのアベイラビリティゾーン用で、「FloatingIP」は複数のアベイラビリティゾーン用です。
* CVO-mediator_subnet_id *	文字列	(オプション) HA の場合は、メディエーターのサブネット ID。

* 名前 *	* タイプ *	* 概要 *
* CVO-mediator_key_pair_name *	文字列	(オプション) HA の場合は、メディアエーターインスタンスのキーペアの名前。
* CVO-cluster_floating_IP *	文字列	(任意) HA FloatingIP の場合、クラスタ管理のフローティング IP アドレス。
* CVO-data_floating_IP *	文字列	(任意) HA FloatingIP の場合は、データフローティング IP アドレス。
* CVO-data_floating_ip2 *	文字列	(任意) HA FloatingIP の場合は、データフローティング IP アドレス。
* CVO-SVM_floating_IP *	文字列	(オプション) HA FloatingIP の場合、SVM 管理のフローティング IP アドレス。
* CVO-ROT_ROTLE_IDS *	リスト	(任意) HA FloatingIP の場合、フローティング IP で更新されるルートテーブル ID のリスト。

FSX の導入

NetApp ONTAP FSXをAWSに導入するためのTerraform構成ファイル

このセクションには、AWS (Amazon Web Services) 上で NetApp ONTAP FSX を導入 / 設定するための、さまざまな Terraform 構成ファイルが含まれています。

Terraform ドキュメント : <https://registry.terraform.io/providers/NetApp/netapp-cloudmanager/latest/docs>

手順

テンプレートを実行するには、次の手順を実行します。

1. リポジトリをクローニングします。

```
git clone
https://github.com/NetApp/na_cloud_volumes_automation.git
```

2. 目的のフォルダに移動します

```
cd na_cloud_volumes_automation/
```

3. CLI から AWS クレデンシャルを設定する。

```
aws configure
```

◦ AWS アクセスキー ID [None] : アクセスキー

- AWS Secret Access Key [None] : secretkey
- デフォルトのリージョン名 [None] : us-west-2
- デフォルトの出力形式 [None] :

4. 変数 /AWS_FSX_deployment.tfvars の変数値を更新します



変数「AWS_Connector_deploy_bool」の値を true または false に設定することで、コネクタを導入することができます。

5. Terraform リポジトリを初期化して、すべての前提条件をインストールし、導入の準備をします。

```
terraform init
```

6. terraform validate コマンドを使用して、terraform ファイルを確認します。

```
terraform validate
```

7. 設定を事前に実行して、導入で想定されるすべての変更をプレビューします。

```
terraform plan -target="module.aws_fsx" -var  
-file="vars/aws_fsx_deployment.tfvars"
```

8. 導入を実行

```
terraform apply -target="module.aws_fsx" -var  
-file="vars/aws_fsx_deployment.tfvars"
```

展開を削除します

```
terraform destroy
```

レシピ:

「コネクタ」

NetApp AWS Connector インスタンスの Terraform 変数。

* 名前 *	* タイプ *	* 概要 *
* AWS_Connector_deploy_bool *	ブール値	(必須) コネクタの配置を確認します。
* AWS_Connector_name*	文字列	(必須) Cloud Manager Connector の名前。
* AWS_connector_region *	文字列	(必須) Cloud Manager Connector を作成するリージョン。
* AWS_Connector_key_name*	文字列	(必須) コネクタインスタンスに使用するキーペアの名前。
* AWS_connector_company *	文字列	(必須) ユーザの会社名。
* AWS_Connector_instance_type *	文字列	(必須) インスタンスのタイプ (t3.xlarge など) 。少なくとも 4 つの CPU と 16 GB のメモリが必要です。
* AWS_connector_subnet_id *	文字列	(必須) インスタンスのサブネットの ID 。
* AWS_Connector_security_group_id *	文字列	(必須) インスタンスのセキュリティグループの ID 。複数のセキュリティグループをで区切って指定できます。
* AWS_Connector_iam_instance_profile_name *	文字列	(必須) コネクタのインスタンスプロファイルの名前。
* AWS_Connector_account_id *	文字列	(オプション) コネクタに関連付けるネットアップアカウントの ID 。指定しない場合、Cloud Manager は最初アカウントを使用します。アカウントが存在しない場合、Cloud Manager は新しいアカウントを作成します。アカウント ID は、Cloud Manager のアカウントタブにあります https://cloudmanager.netapp.com 。
* AWS_connector_public_ip_bool *	ブール値	(任意) インスタンスにパブリック IP アドレスに関連付けるかどうかを指定します。指定しない場合は、サブネットの設定に基づいて関連付けが行われます。

「FSX インスタンス」

NetApp ONTAP FSX インスタンスの Terraform 変数。

* 名前 *	* タイプ *	* 概要 *
* FSX_NAME*	文字列	(必須) Cloud Volumes ONTAP 作業環境の名前。
* FSX_REGION *	文字列	(必須) 作業環境を作成するリージョン。
* FSX_primary_subnet_id *	文字列	(必須) 作業環境を作成するプライマリサブネット ID。
* fsx_secondary_subnet_id *	文字列	(必須) 作業環境を作成するセカンダリサブネット ID。
* fsx_account_id *	文字列	(必須) FSX インスタンスに関連付けるネットアップアカウントの ID。指定しない場合、Cloud Manager は最初のアカountを使用します。アカウントが存在しない場合、Cloud Manager は新しいアカウントを作成します。アカウント ID は、Cloud Manager のアカウントタブにあります https://cloudmanager.netapp.com 。
* FSX_workspace_id *	文字列	(必須) 作業環境の Cloud Manager ワークスペースの ID。
* FSX_admin_password *	文字列	(必須) Cloud Volumes ONTAP の管理パスワード。
* FSX_Throughput_capacity *	文字列	(任意) スループットの容量。
* FSX_storage_capacity_size *	文字列	(オプション) 最初のデータアグリゲートの EBS ボリュームサイズGB の場合、単位は [100 または 500] です。TB の場合、単位は [1,2,4,8,16] です。デフォルトは「1」です。
* FSX_storage_capacity_size_unit *	文字列	(オプション) ['GB' または 'TB']。デフォルトは「TB」です。
* FSX_cloudmanager_aws_credential_name *	文字列	(必須) AWS クレデンシャルアカウント名。



ANF

AzureにANFボリュームを導入するための環境設定ファイルを用意しています

このセクションでは、Azure に ANF（Azure NetApp Files）ボリュームを導入 / 設定するためのさまざまな Terraform 設定ファイルを示します。

Terraform ドキュメント : <https://registry.terraform.io/providers/hashicorp/azurerm/latest/docs>

手順

テンプレートを実行するには、次の手順を実行します。

1. リポジトリをクローニングします。

```
git clone
https://github.com/NetApp/na_cloud_volumes_automation.git
```

2. 目的のフォルダに移動します

```
cd na_cloud_volumes_automation
```

3. Azure CLI にログインします（Azure CLI がインストールされている必要があります）。

```
az login
```

4. vars/azure_anf.tfvars の変数値を更新します



既存の VNet およびサブネットを使用して ANF ボリュームを導入することもできます。変数「vnet_creation_bool」と「subnet_creation_bool」の値を false に設定し、「subnet_id_for_anf_vol」を指定します。これらの値を true に設定して新しい VNet とサブネットを作成する場合にも、新しく作成したサブネットからサブネット ID が自動的に取得されます。

5. Terraform リポジトリを初期化して、すべての前提条件をインストールし、導入の準備をします。

```
terraform init
```

6. terraform validate コマンドを使用して、terraform ファイルを確認します。

```
terraform validate
```

7. 設定を事前に実行して、導入で想定されるすべての変更をプレビューします。


```
terraform plan -target="module.anf" -var  
-file="vars/azure_anf.tfvars"
```

8. 導入を実行

```
terraform apply -target="module.anf" -var  
-file="vars/azure_anf.tfvars"
```

展開を削除します

```
terraform destroy
```

降水量：

「シングルノードインスタンス」

単一のネットアップ ANF ボリュームに対応する Terraform 変数。

* 名前 *	* タイプ *	* 概要 *
* AZ_location*	文字列	(必須) リソースが存在する、サポートされている Azure の場所を指定します。これを変更すると、新しいリソースが強制的に作成されます。
* AZ_PREFIX *	文字列	(必須) ネットアップボリュームを作成するリソースグループの名前。これを変更すると、新しいリソースが強制的に作成されます。
* AZ_vnet_address_space *	文字列	(必須) ANF ボリューム導入用として新しく作成した VNet で使用するアドレススペースです。
* AZ_subnet_address_prefix *	文字列	(必須) ANF ボリューム導入用に新しく作成した VNet で使用するサブネットアドレスプレフィックスです。
* AZ_volume_path *	文字列	(必須) ボリュームの一意のファイルパス。マウントターゲットの作成時に使用します。これを変更すると、新しいリソースが強制的に作成されます。
* az_capacity_pool_size *	整数	(必須) 容量プールサイズ (TB) 。
* az_vnet_creation_bool *	ブール値	(必須) 新しい VNet を作成する場合は、このブール値を「true」に設定します。既存の VNet を使用するには、このパラメータを「false」に設定します。
* az_subnet_creation_bool *	ブール値	(必須) 新しいサブネットを作成するには、このブーリアンを「true」に設定します。既存のサブネットを使用する場合は「false」に設定します。

* 名前 *	* タイプ *	* 概要 *
* az_subnet_id_for_anf_vol *	文字列	(必須) 「subnet_creation_bool」を true に設定して既存のサブネットを使用する場合に、サブネット ID を指定します。false に設定する場合は、デフォルト値のままにします。
* AZ_NetApp_POOL_SERVICE_LEVEL *	文字列	(必須) ファイルシステムのターゲットパフォーマンス。有効な値は 'Premium' 'Standard' または Ultra です
* AZ_NetApp_vol_SERVICE_LEVEL *	文字列	(必須) ファイルシステムのターゲットパフォーマンス。有効な値は 'Premium' 'Standard' または Ultra です
* AZ_NetApp_vol_protocol *	文字列	(オプション) リストで表されるターゲットボリュームプロトコル。サポートされる単一の値には 'CIFS' 'nfsv3' または 'NFSv4.1' があります。引数が定義されていない場合、デフォルトは「nfsv3」です。これを変更すると、新しいリソースが強制的に作成され、データが失われます。
* AZ_NetApp_vol_security_style *	文字列	(任意) ボリュームセキュリティ形式。有効値は「Unix」または「NTFS」です。指定されない場合、単一プロトコル・ボリュームは 'nfsv3' または 'nfsv3' ボリュームの場合は 'UNIX' にデフォルトで作成されますが 'CIFS' の場合は 'NTFS' にデフォルト設定されます。デュアル・プロトコル・ボリュームでは、指定しない場合 'ntfs_' の値になります
* AZ_NetApp_vol_storage_quota *	文字列	(必須) ファイルシステムに許可される最大ストレージクォータ（ギガバイト単位）。



このスクリプトでは、推奨事項に従って、prevent_destroy 構成ファイル内の偶発的なデータ損失の可能性を軽減するためのLifecycle引数。詳細については、を参照してください prevent_destroy lifecycle引数はterraformのドキュメントを参照してください。 <https://developer.hashicorp.com/terraform/tutorials/state/resource-lifecycle#prevent-resource-deletion>。

ANF データ保護

Azure上にデータ保護機能を備えたANFボリュームを導入するための構成ファイルを用意しています
このセクションでは、Azure でデータ保護を使用して ANF（Azure NetApp Files）ボリュームを導入 / 設定するためのさまざまな Terraform 設定ファイルについて説明します。

Terraform ドキュメント : <https://registry.terraform.io/providers/hashicorp/azurerm/latest/docs>

手順

テンプレートを実行するには、次の手順を実行します。

1. リポジトリをクローニングします。

```
git clone
https://github.com/NetApp/na_cloud_volumes_automation.git
```

2. 目的のフォルダに移動します

```
cd na_cloud_volumes_automation
```

3. Azure CLI にログインします（Azure CLI がインストールされている必要があります）。

```
az login
```

4. vars/azure_anf_data_protection_tfvars の変数値を更新します。



既存の VNet およびサブネットを使用して ANF ボリュームを導入することもできます。変数「vnet_creation_bool」と「subnet_creation_bool」の値を false に設定し、「subnet_id_for_anf_vol」を指定します。これらの値を true に設定して新しい VNet とサブネットを作成する場合にも、新しく作成したサブネットからサブネット ID が自動的に取得されます。

5. Terraform リポジトリを初期化して、すべての前提条件をインストールし、導入の準備をします。

```
terraform init
```

6. terraform validate コマンドを使用して、terraform ファイルを確認します。

```
terraform validate
```

7. 設定を事前に実行して、導入で想定されるすべての変更をプレビューします。

```
terraform plan -target="module.anf_data_protection" -var
-file="vars/azure_anf_data_protection.tfvars"
```

8. 導入を実行

```
terraform apply -target="module.anf_data_protection" -var
-file="vars/azure_anf_data_protection.tfvars"
```

展開を削除します

降水量：

「ANF データ保護」

データ保護が有効になっている単一の ANF ボリューム用の変数を Terraform します。

* 名前 *	* タイプ *	* 概要 *
* AZ_location*	文字列	(必須) リソースが存在する、サポートされている Azure の場所を指定します。これを変更すると、新しいリソースが強制的に作成されます。
* AZ_alt_location *	文字列	(必須) セカンダリボリュームを作成する Azure の場所
* AZ_PREFIX *	文字列	(必須) ネットアップボリュームを作成するリソースグループの名前。これを変更すると、新しいリソースが強制的に作成されます。
* AZ_vnet_primary_address_space *	文字列	(必須) ANF プライマリボリューム導入用として新しく作成した VNet が使用するアドレススペース。
* AZ_vnet_secondary_address_space *	文字列	(必須) ANF セカンダリボリューム導入用として新しく作成した VNet が使用するアドレススペース。
* AZ_subnet_primary_address_prefix *	文字列	(必須) ANF プライマリボリューム導入用に新しく作成した VNet で使用するサブネットアドレスプレフィックスです。
* AZ_subnet_secondary_address_prefix *	文字列	(必須) ANF セカンダリボリューム導入用に新しく作成した VNet で使用するサブネットアドレスプレフィックスです。
* AZ_volume_path_primary *	文字列	(必須) プライマリボリュームの一意的ファイルパス。マウントターゲットの作成時に使用します。これを変更すると、新しいリソースが強制的に作成されます。
* AZ_volume_path_secondary *	文字列	(必須) セカンダリボリュームの一意的ファイルパス。マウントターゲットの作成時に使用します。これを変更すると、新しいリソースが強制的に作成されます。
* AZ_capacity_pool_size_primary *	整数	(必須) 容量プールサイズ (TB) 。
* AZ_capacity_pool_size_secondary *	整数	(必須) 容量プールサイズ (TB) 。
* az_vnet_primary_creation_bool *	ブール値	(必須) プライマリボリューム用の新しい VNet を作成する場合は、このブーリアンを「true」に設定します。既存の VNet を使用するには、このパラメータを「false」に設定します。

* 名前 *	* タイプ *	* 概要 *
* az_vnet_secondary_creation_bool *	ブール値	(必須) セカンダリボリューム用の新しい VNet を作成する場合は、このブーリアンを「true」に設定します。既存の VNet を使用するには、このパラメータを「false」に設定します。
* az_subnet_primary_creation_bool *	ブール値	(必須) このブール値を「true」に設定して、プライマリボリュームの新しいサブネットを作成します。既存のサブネットを使用する場合は 'false に設定します
* az_subnet_secondary_creation_bool *	ブール値	(必須) セカンダリボリュームの新しいサブネットを作成するには、このブーリアンを「true」に設定します。既存のサブネットを使用する場合は 'false に設定します
* az_primary_subnet_id_for_anf_vol *	文字列	(必須) 「subnet_primary_creation_bool」を true に設定して既存のサブネットを使用する場合に、サブネット ID を指定します。false に設定する場合は、デフォルト値のままにします。
* AZ_SECONDARY_subnet_id_on_anf_vol *	文字列	(必須) 「subnet_secondary_creation_bool」を true に設定して既存のサブネットを使用する場合に備えて、サブネット ID を指定します。false に設定する場合は、デフォルト値のままにします。
* AZ_NetApp_POOL_SERVICE_LEVEL_PRIMARY *	文字列	(必須) ファイルシステムのターゲットパフォーマンス。有効な値は 'Premium' 'Standard' または Ultra です
* AZ_NetApp_POOL_SERVICE_LEVEL_SECONDARY *	文字列	(必須) ファイルシステムのターゲットパフォーマンス。有効な値は 'Premium' 'Standard' または Ultra です
* AZ_NetApp_vol_SERVICE_LEVEL_PRIMARY *	文字列	(必須) ファイルシステムのターゲットパフォーマンス。有効な値は 'Premium' 'Standard' または Ultra です
* AZ_NetApp_vol_SERVICE_LEVEL_SECONDARY *	文字列	(必須) ファイルシステムのターゲットパフォーマンス。有効な値は 'Premium' 'Standard' または Ultra です
* AZ_NetApp_vol_protocol_primary *	文字列	(オプション) リストで表されるターゲットボリュームプロトコル。サポートされる単一の値には 'CIFS' 'nfsv3' または 'NFSv4.1' があります引数が定義されていない場合、デフォルトは「nfsv3」です。これを変更すると、新しいリソースが強制的に作成され、データが失われます。
* AZ_NetApp_vol_protocol_secondary *	文字列	(オプション) リストで表されるターゲットボリュームプロトコル。サポートされる単一の値には 'CIFS' 'nfsv3' または 'NFSv4.1' があります引数が定義されていない場合、デフォルトは「nfsv3」です。これを変更すると、新しいリソースが強制的に作成され、データが失われます。

* 名前 *	* タイプ *	* 概要 *
AZ_NetApp_vol_storage_quota_policy *	文字列	(必須) ファイルシステムに許可される最大ストレージクォータ (ギガバイト単位)。
AZ_NetApp_vol_storage_QUOTA_SECONDARY *	文字列	(必須) ファイルシステムに許可される最大ストレージクォータ (ギガバイト単位)。
AZ_DP_replication_frequency *	文字列	(必須) レプリケーション頻度。サポートされる値は「10 分」、「時間単位」、「日単位」です。値は大文字と小文字が区別されます。



このスクリプトでは、推奨事項に従って、prevent_destroy 構成ファイル内の偶発的なデータ損失の可能性を軽減するためのLifecycle引数。詳細については、を参照してください prevent_destroy lifecycle引数はterraformのドキュメントを参照してください。 <https://developer.hashicorp.com/terraform/tutorials/state/resource-lifecycle#prevent-resource-deletion>。

ANF デュアルプロトコル

Azure上にデュアルプロトコルを使用して**ANF**ボリュームを導入するための構成ファイルが用意されています

このセクションでは、Azure でデュアルプロトコルを有効にした ANF (Azure NetApp Files) ボリュームを導入 / 設定するためのさまざまな Terraform 設定ファイルについて説明します。

Terraform ドキュメント : <https://registry.terraform.io/providers/hashicorp/azurerm/latest/docs>

手順

テンプレートを実行するには、次の手順を実行します。

1. リポジトリをクローニングします。

```
git clone
https://github.com/NetApp/na_cloud_volumes_automation.git
```

2. 目的のフォルダに移動します

```
cd na_cloud_volumes_automation
```

3. Azure CLI にログインします (Azure CLI がインストールされている必要があります) 。

```
az login
```

- vars/azure_anf_dual_protocol.tfvars の変数値を更新します。



既存の VNet およびサブネットを使用して ANF ボリュームを導入することもできます。変数「vnet_creation_bool」と「subnet_creation_bool」の値を false に設定し、「subnet_id_for_anf_vol」を指定します。これらの値を true に設定して新しい VNet とサブネットを作成する場合にも、新しく作成したサブネットからサブネット ID が自動的に取得されます。

- Terraform リポジトリを初期化して、すべての前提条件をインストールし、導入の準備をします。

```
terraform init
```

- terraform validate コマンドを使用して、terraform ファイルを確認します。

```
terraform validate
```

- 設定を事前に実行して、導入で想定されるすべての変更をプレビューします。

```
terraform plan -target="module.anf_dual_protocol" -var  
-file="vars/azure_anf_dual_protocol.tfvars"
```

- 導入を実行

```
terraform apply -target="module.anf_dual_protocol" -var  
-file="vars/azure_anf_dual_protocol.tfvars"
```

展開を削除します

```
terraform destroy
```

降水量：

「シングルノードインスタンス」

デュアルプロトコルが有効な単一の ANF ボリューム用の Terraform 変数。

* 名前 *	* タイプ *	* 概要 *
* AZ_location*	文字列	(必須) リソースが存在する、サポートされている Azure の場所を指定します。これを変更すると、新しいリソースが強制的に作成されます。

* 名前 *	* タイプ *	* 概要 *
* AZ_PREFIX *	文字列	(必須) ネットアップボリュームを作成するリソースグループの名前。これを変更すると、新しいリソースが強制的に作成されます。
* AZ_vnet_address_space *	文字列	(必須) ANF ボリューム導入用として新しく作成した VNet で使用するアドレススペースです。
* AZ_subnet_address_prefix *	文字列	(必須) ANF ボリューム導入用に新しく作成した VNet で使用するサブネットアドレスプレフィックスです。
* AZ_volume_path *	文字列	(必須) ボリュームの一意のファイルパス。マウントターゲットの作成時に使用します。これを変更すると、新しいリソースが強制的に作成されます。
* az_capacity_pool_size *	整数	(必須) 容量プールサイズ (TB)。
* az_vnet_creation_bool *	ブール値	(必須) 新しい VNet を作成する場合は、このブール値を「true」に設定します。既存の VNet を使用するには、このパラメータを「false」に設定します。
* az_subnet_creation_bool *	ブール値	(必須) 新しいサブネットを作成するには、このブーリアンを「true」に設定します。既存のサブネットを使用する場合は「false」に設定します。
* az_subnet_id_for_anf_vol *	文字列	(必須) 「subnet_creation_bool」を true に設定して既存のサブネットを使用する場合に、サブネット ID を指定します。false に設定する場合は、デフォルト値のままにします。
* AZ_NetApp_POOL_SERVICE_LEVEL *	文字列	(必須) ファイルシステムのターゲットパフォーマンス。有効な値は 'Premium' 'Standard' または Ultra です。
* AZ_NetApp_vol_SERVICE_LEVEL *	文字列	(必須) ファイルシステムのターゲットパフォーマンス。有効な値は 'Premium' 'Standard' または Ultra です。
* AZ_NetApp_vol_protocol1 *	文字列	(必須) ターゲットボリュームプロトコル。リストで表されます。サポートされる単一の値には 'CIFS' 'nfsv3' または 'NFSv4.1' があります。引数が定義されていない場合、デフォルトは「nfsv3」です。これを変更すると、新しいリソースが強制的に作成され、データが失われます。
* AZ_NetApp_vol_protocol2 *	文字列	(必須) ターゲットボリュームプロトコル。リストで表されます。サポートされる単一の値には 'CIFS' 'nfsv3' または 'NFSv4.1' があります。引数が定義されていない場合、デフォルトは「nfsv3」です。これを変更すると、新しいリソースが強制的に作成され、データが失われます。
* AZ_NetApp_vol_storage_quota *	文字列	(必須) ファイルシステムに許可される最大ストレージクォータ (ギガバイト単位)。

* 名前 *	* タイプ *	* 概要 *
AZ_SMB_server_username *	文字列	(必須) ActiveDirectory オブジェクトを作成するユーザ名。
AZ_SMB_server_password *	文字列	(必須) ActiveDirectory オブジェクトを作成するためのユーザパスワード。
AZ_SMB_SERVER_NAME *	文字列	(必須) ActiveDirectory オブジェクトを作成するサーバ名。
AZ_SMB_DNS_servers *	文字列	(必須) ActiveDirectory オブジェクトを作成するための DNS サーバ IP。



このスクリプトでは、推奨事項に従って、prevent_destroy 構成ファイル内の偶発的なデータ損失の可能性を軽減するためのLifecycle引数。詳細については、を参照してください prevent_destroy lifecycle引数はterraformのドキュメントを参照してください。 <https://developer.hashicorp.com/terraform/tutorials/state/resource-lifecycle#prevent-resource-deletion>。

Snapshot からの ANF ボリューム

Azure上の**Snapshot**から**ANF**ボリュームを導入するための構成ファイルをテラフォームしてください

このセクションでは、Azure 上の Snapshot から ANF (Azure NetApp Files) ボリュームを導入 / 設定するためのさまざまな Terraform 設定ファイルを示します。

Terraform ドキュメント : <https://registry.terraform.io/providers/hashicorp/azurerm/latest/docs>

手順

テンプレートを実行するには、次の手順を実行します。

1. リポジトリをクローニングします。

```
git clone
https://github.com/NetApp/na_cloud_volumes_automation.git
```

2. 目的のフォルダに移動します

```
cd na_cloud_volumes_automation
```

3. Azure CLI にログインします (Azure CLI がインストールされている必要があります) 。

```
az login
```

4. vars/azure_anf_volume_from_snapshot.tfvars の変数値を更新します。



既存の VNet およびサブネットを使用して ANF ボリュームを導入することもできます。変数「vnet_creation_bool」と「subnet_creation_bool」の値を false に設定し、「subnet_id_for_anf_vol」を指定します。これらの値を true に設定して新しい VNet とサブネットを作成する場合にも、新しく作成したサブネットからサブネット ID が自動的に取得されます。

1. Terraform リポジトリを初期化して、すべての前提条件をインストールし、導入の準備をします。

```
terraform init
```

2. terraform validate コマンドを使用して、terraform ファイルを確認します。

```
terraform validate
```

3. 設定を事前に行って、導入で想定されるすべての変更をプレビューします。

```
terraform plan -target="module.anf_volume_from_snapshot"
-var-file="vars/azure_anf_volume_from_snapshot.tfvars"
```

4. 導入を実行

```
terraform apply -target="module.anf_volume_from_snapshot"
-var-file="vars/azure_anf_volume_from_snapshot.tfvars"
```

展開を削除します

```
terraform destroy
```

降水量：

「シングルノードインスタンス」

Snapshot を使用する単一の ANF ボリューム用の変数を Terraform します。

* 名前 *	* タイプ *	* 概要 *
* AZ_location*	文字列	(必須) リソースが存在する、サポートされている Azure の場所を指定します。これを変更すると、新しいリソースが強制的に作成されます。

* 名前 *	* タイプ *	* 概要 *
* AZ_PREFIX *	文字列	(必須) ネットアップボリュームを作成するリソースグループの名前。これを変更すると、新しいリソースが強制的に作成されます。
* AZ_vnet_address_space *	文字列	(必須) ANF ボリューム導入用として新しく作成した VNet で使用するアドレススペースです。
* AZ_subnet_address_prefix *	文字列	(必須) ANF ボリューム導入用に新しく作成した VNet で使用するサブネットアドレスプレフィックスです。
* AZ_volume_path *	文字列	(必須) ボリュームの一意のファイルパス。マウントターゲットの作成時に使用します。これを変更すると、新しいリソースが強制的に作成されます。
* az_capacity_pool_size *	整数	(必須) 容量プールサイズ (TB)。
* az_vnet_creation_bool *	ブール値	(必須) 新しい VNet を作成する場合は、このブール値を「true」に設定します。既存の VNet を使用するには、このパラメータを「false」に設定します。
* az_subnet_creation_bool *	ブール値	(必須) 新しいサブネットを作成するには、このブーリアンを「true」に設定します。既存のサブネットを使用する場合は「false」に設定します。
* az_subnet_id_for_anf_vol *	文字列	(必須) 「subnet_creation_bool」を true に設定して既存のサブネットを使用する場合に、サブネット ID を指定します。false に設定する場合は、デフォルト値のままにします。
* AZ_NetApp_POOL_SERVICE_LEVEL *	文字列	(必須) ファイルシステムのターゲットパフォーマンス。有効な値は 'Premium' 'Standard' または Ultra です。
* AZ_NetApp_vol_SERVICE_LEVEL *	文字列	(必須) ファイルシステムのターゲットパフォーマンス。有効な値は 'Premium' 'Standard' または Ultra です。
* AZ_NetApp_vol_protocol *	文字列	(オプション) リストで表されるターゲットボリュームプロトコル。サポートされる単一の値には 'CIFS' 'nfsv3' または 'NFSv4.1' があります。引数が定義されていない場合、デフォルトは「nfsv3」です。これを変更すると、新しいリソースが強制的に作成され、データが失われます。
* AZ_NetApp_vol_storage_quota *	文字列	(必須) ファイルシステムに許可される最大ストレージクォータ (ギガバイト単位)。
* AZ_SNAPSHOT_ID *	文字列	(必須) 新しい ANF ボリュームを作成する際に使用する Snapshot ID。



このスクリプトでは、推奨事項に従って、prevent_destroy 構成ファイル内の偶発的なデータ損失の可能性を軽減するためのLifecycle引数。詳細については、を参照してください prevent_destroy lifecycle引数はterraformのドキュメントを参照してください。 <https://developer.hashicorp.com/terraform/tutorials/state/resource-lifecycle#prevent-resource-deletion>。

CVO シングルノードの導入

Azureにシングルノードの**CVO**を導入するための構成ファイルが**Terraform**に用意されています

このセクションでは、Azure 上にシングルノード CVO （ Cloud Volumes ONTAP ）を導入 / 構成するための各種 Terraform 構成ファイルを紹介します。

Terraform ドキュメント : <https://registry.terraform.io/providers/NetApp/netapp-cloudmanager/latest/docs>

手順

テンプレートを実行するには、次の手順を実行します。

1. リポジトリをクローニングします。

```
git clone
https://github.com/NetApp/na_cloud_volumes_automation.git
```

2. 目的のフォルダに移動します

```
cd na_cloud_volumes_automation
```

3. Azure CLI にログインします（ Azure CLI がインストールされている必要があります）。

```
az login
```

4. 変数 \azure_CVO-SILE_NODE_deployment.tfvars の変数を更新します。
5. Terraform リポジトリを初期化して、すべての前提条件をインストールし、導入の準備をします。

```
terraform init
```

6. terraform validate コマンドを使用して、terraform ファイルを確認します。

```
terraform validate
```

7. 設定を事前に実行して、導入で想定されるすべての変更をプレビューします。

```
terraform plan
-target="module.az_cvo_single_node_deployment" -var
-file="vars\azure_cvo_single_node_deployment.tfvars"
```

8. 導入を実行

```
terraform apply
-target="module.az_cvo_single_node_deployment" -var
-file="vars\azure_cvo_single_node_deployment.tfvars"
```

展開を削除します

```
terraform destroy
```

降水量：

「シングルノードインスタンス」

単一ノードの Cloud Volumes ONTAP （ CVO ） 用の Terraform 変数。

* 名前 *	* タイプ *	* 概要 *
* refresh_token *	文字列	（必須） NetApp Cloud Manager の更新トークン。これは NetApp Cloud Central から生成できます。
* AZ_Connector_name *	文字列	（必須） Cloud Manager Connector の名前。
* AZ_Connector_location *	文字列	（必須） Cloud Manager Connector を作成する場所。
* AZ_Connector_subscription_id *	文字列	（必須） Azure サブスクリプションの ID 。
* AZ_Connector_company *	文字列	（必須） ユーザの会社名。
* AZ_Connector_resource_group *	整数	（必須） リソースが作成される Azure 内のリソースグループ。
* AZ_Connector_subnet_id *	文字列	（必須） 仮想マシンのサブネットの名前です。

* 名前 *	* タイプ *	* 概要 *
* AZ_Connector_vnet_id *	文字列	(必須) 仮想ネットワークの名前。
* AZ_Connector_network_security_group_name *	文字列	(必須) インスタンスのセキュリティグループの名前。
* AZ_Connector_associate_public_IP_address *	文字列	(必須) 仮想マシンにパブリック IP アドレスを関連付けるかどうかを指定します。
* AZ_Connector_account_id *	文字列	(必須) コネクタを関連付けるネットアップアカウントの ID。指定しない場合、Cloud Manager は最初のアカントを使用します。アカウントが存在しない場合、Cloud Manager は新しいアカウントを作成します。アカウント ID は、Cloud Manager のアカウントタブにあります https://cloudmanager.netapp.com 。
* AZ_Connector_admin_password *	文字列	(必須) コネクタのパスワード。
* AZ_Connector_admin_username *	文字列	(必須) コネクタのユーザ名。
* AZ_CVO-NAME *	文字列	(必須) Cloud Volumes ONTAP 作業環境の名前。
* AZ_CVF_location *	文字列	(必須) 作業環境を作成する場所。
* AZ_CVO-subnet_id *	文字列	(必須) Cloud Volumes ONTAP システムのサブネットの名前。
* AZ_CVO-vnet_id *	文字列	(必須) 仮想ネットワークの名前。
* AZ_CVO-vnet_resource_group *	文字列	(必須) 仮想ネットワークに関連付けられた Azure 内のリソースグループ。
* AZ_CVO-data_encryption_type *	文字列	(必須) 作業環境に使用する暗号化のタイプ: [Azure]、[none]。デフォルトは「azure」です。
* AZ_CVO-storage_type *	文字列	(必須) 最初のデータ・アグリゲートのストレージ・タイプ: ['Premium_LRS' 'Standard_LRS' 'StandardSSD_LRS'] デフォルトは 'Premium_LRS' です
* AZ_CVO-svm_svm_name *	文字列	(必須) Cloud Volumes ONTAP の管理パスワード。

* 名前 *	* タイプ *	* 概要 *
* AZ_CVO-workspace_id *	文字列	(必須) Cloud Volumes ONTAP を導入する Cloud Manager ワークスペースの ID。指定しない場合、Cloud Manager は最初のワークスペースを使用します。ID は、の [ワークスペース (Workspace)] タブで確認できます https://cloudmanager.netapp.com 。
* AZ_CVF_capacity_tier *	文字列	(必須) 最初のデータ・アグリゲートのデータ階層化を有効にするかどうかを指定します ([lob,'none']) デフォルトは「BLOB」です。
* AZ_CVF_Writing_speed_state *	文字列	(必須) Cloud Volumes ONTAP の書き込み速度設定: [normal,high]デフォルトは「normal」です。この引数は HA ペアには関係ありません。
* AZ_CVF_ONTAP_VERSION *	文字列	(必須) 必要な ONTAP のバージョン。「use_latest_version」が true に設定されている場合は無視されます。デフォルトでは最新バージョンが使用されます。
* AZ_CVF_INSTAN Y_TYPE *	文字列	(必須) 選択したライセンスタイプに応じて使用するインスタンスのタイプ。Explore : ['Standard_DS3_v2'Standard : ['Standard_DS4_v2'Standard_DS13_v2'Standard_L8s_v2'Premium : ['Standard_DS5_v2'Standard_DS14_v2'v2"Pay_DS3_v2""PAY'v2 インスタンスタイプごとに定義された BYOL : すべてのライセンスタイプサポートされるインスタンスタイプの詳細については、Cloud Volumes ONTAP リリースノートを参照してください。デフォルトは 'Standard_DS4_v2' です
* AZ_CVF_LICENSE_TYPE *	文字列	(必須) 使用するライセンスのタイプ。シングルノードの場合: [azure-CO-EXPLORT-paygo,azure-CO-standard-paygo,azure-CO-Premium-paygo',azure-paygo]HA の場合: [azure-HA-COT-standard-paygo, azure-HA-COT -Premium-paygo', azure-HA-COT -Premium-BYOL , HA-capacity-paygo]デフォルトは「azure-CO-standard-paygo」です。「Capacity-paygo」または「HA-Capacity-paygo」を使用して、「Bring Your Own License Type Capacity Based」または「Freemium」を選択します。「Bring Your Own License Type Node-Based」を選択した場合は、「azure-CO-Premium-BYOL」または「azure-HA-CO-Premium-BYOL」を使用します。
* AZ_CVF_NSS_ACCOUNT *	文字列	(必須) この Cloud Volumes ONTAP システムで使用するネットアップサポートサイトのアカウント ID。ライセンスタイプが BYOL で、NSS アカウントが指定されていない場合、Cloud Manager は最初の既存の NSS アカウントの使用を試みます。
* AZ_tenant_id *	文字列	(必須) Azure に登録されているアプリケーション / サービスプリンシパルのテナント ID。
* AZ_application_id *	文字列	(必須) Azure に登録されているアプリケーション / サービスプリンシパルのアプリケーション ID。
* AZ_application_key *	文字列	(必須) Azure に登録されているアプリケーション / サービスプリンシパルのアプリケーションキー。

CVO HA の導入

AzureにCVO HAを導入するための構成ファイルがTerraformに用意されています

このセクションでは、Azure上でCVO（Cloud Volumes ONTAP）HA（ハイアベイラビリティ）を導入/構成するためのさまざまなTerraform構成ファイルを取り上げます。

Terraform ドキュメント：<https://registry.terraform.io/providers/NetApp/netapp-cloudmanager/latest/docs>

手順

テンプレートを実行するには、次の手順を実行します。

1. リポジトリをクローニングします。

```
git clone
https://github.com/NetApp/na_cloud_volumes_automation.git
```

2. 目的のフォルダに移動します

```
cd na_cloud_volumes_automation
```

3. Azure CLI にログインします（Azure CLI がインストールされている必要があります）。

```
az login
```

4. 変数 \azure_CVF_HA_deployment.tfvars の変数を更新します。

5. Terraform リポジトリを初期化して、すべての前提条件をインストールし、導入の準備をします。

```
terraform init
```

6. terraform validate コマンドを使用して、terraform ファイルを確認します。

```
terraform validate
```

7. 設定を事前に実行して、導入で想定されるすべての変更をプレビューします。

```
terraform plan -target="module.az_cvo_ha_deployment" -var
-file="vars\azure_cvo_ha_deployment.tfvars"
```

8. 導入を実行


```
terraform apply -target="module.az_cvo_ha_deployment" -var  
-file="vars\azure_cvo_ha_deployment.tfvars"
```

展開を削除します

```
terraform destroy
```

降水量：

HA ペア・インスタンス

HA ペアの Cloud Volumes ONTAP （ CVO ） の変数は Terraform です。

* 名前 *	* タイプ *	* 概要 *
* refresh_token *	文字列	（必須） NetApp Cloud Manager の更新トークン。これは NetApp Cloud Central から生成できます。
* AZ_Connector_name *	文字列	（必須） Cloud Manager Connector の名前。
* AZ_Connector_location *	文字列	（必須） Cloud Manager Connector を作成する場所。
* AZ_Connector_subscription_id *	文字列	（必須） Azure サブスクリプションの ID 。
* AZ_Connector_company *	文字列	（必須） ユーザの会社名。
* AZ_Connector_resource_group *	整数	（必須） リソースが作成される Azure 内のリソースグループ。
* AZ_Connector_subnet_id *	文字列	（必須） 仮想マシンのサブネットの名前です。
* AZ_Connector_vnet_id *	文字列	（必須） 仮想ネットワークの名前。
* AZ_Connector_network_security_group_name *	文字列	（必須） インスタンスのセキュリティグループの名前。

* 名前 *	* タイプ *	* 概要 *
* AZ_Connector_as sociate_public_IP_ address *	文字列	(必須) 仮想マシンにパブリック IP アドレスを関連付けるかどうかを指定します。
* AZ_Connector_ac count_id *	文字列	(必須) コネクタを関連付けるネットアップアカウントの ID。指定しない場合、Cloud Manager は最初アカウントを使用します。アカウントが存在しない場合、Cloud Manager は新しいアカウントを作成します。アカウント ID は、Cloud Manager のアカウントタブにあります https://cloudmanager.netapp.com 。
* AZ_Connector_ad min_password *	文字列	(必須) コネクタのパスワード。
* AZ_Connector_ad min_username*	文字列	(必須) コネクタのユーザ名。
* AZ_CVO-NAME *	文字列	(必須) Cloud Volumes ONTAP 作業環境の名前。
* AZ_CVF_location*	文字列	(必須) 作業環境を作成する場所。
* AZ_CVO- subnet_id *	文字列	(必須) Cloud Volumes ONTAP システムのサブネットの名前。
* AZ_CVO-vnet_id *	文字列	(必須) 仮想ネットワークの名前。
* AZ_CVO- vnet_resource_gro up *	文字列	(必須) 仮想ネットワークに関連付けられた Azure 内のリソースグループ。
* AZ_CVO- data_encryption_t ype*	文字列	(必須) 作業環境に使用する暗号化のタイプ: [Azure]、[none]。デフォルトは「azure」です。
* AZ_CVO- storage_type *	文字列	(必須) 最初のデータ・アグリゲートのストレージ・タイプ: ['Premium_LRS','Standard_LRS','StandardSSD_LRS']デフォルトは 'Premium_LRS' です
* AZ_CVO- svm_svm_svm_na me * をクリックし ます	文字列	(必須) Cloud Volumes ONTAP の管理パスワード。
* AZ_CVO- workspace_id *	文字列	(必須) Cloud Volumes ONTAP を導入する Cloud Manager ワークスペースの ID。指定しない場合、Cloud Manager は最初のワークスペースを使用します。ID は、の [ワークスペース (Workspace)] タブで確認できます https://cloudmanager.netapp.com 。
* AZ_CVF_capacity _tier *	文字列	(必須) 最初のデータ・アグリゲートのデータ階層化を有効にするかどうかを指定します ([blob,'none']) デフォルトは「BLOB」です。

* 名前 *	* タイプ *	* 概要 *
* AZ_CVF_Writing_speed_state *	文字列	(必須) Cloud Volumes ONTAP の書き込み速度設定: [normal,high]デフォルトは「normal」です。この引数は HA ペアには関係ありません。
* AZ_CVF_ONTAP_VERSION *	文字列	(必須) 必要な ONTAP のバージョン。「use_latest_version」が true に設定されている場合は無視されます。デフォルトでは最新バージョンが使用されます。
* AZ_CVF_INSTANCE_TYPE *	文字列	(必須) 選択したライセンスタイプに応じて使用するインスタンスのタイプ。Explore : ['Standard_DS3_v2','Standard_DS4_v2','Standard_DS13_v2','Standard_DS14_v2','Standard_DS5_v2','Standard_DS1_v2']BYOL : PAYGO 用に定義されたすべてのインスタンス・タイプサポートされるインスタンスタイプの詳細については、Cloud Volumes ONTAP リリースノートを参照してください。デフォルトは 'Standard_DS4_v2' です
* AZ_CVF_LICENSE_TYPE *	文字列	(必須) 使用するライセンスのタイプ。シングルノードの場合: [azure-CO-EXPLOR-paygo, azure-CO-standard-paygo, azure-CO-Premium-pole-BYOL、capacity-paygo]HA の場合: [azure-HA-COT-standard-paygo, azure-HA-CO-Premium-paygo, azure-HA-CO-Premium-BYOL、HA-capacity-paygo]デフォルトは「azure-CO-standard-paygo」です。「Capacity-paygo」または「HA-Capacity-paygo」を使用して、「Bring Your Own License Type Capacity Based」または「Freemium」を選択します。「Bring Your Own License Type Node-Based」を選択した場合は、「azure-CO-Premium-BYOL」または「azure-HA-CO-Premium-BYOL」を使用します。
* AZ_CVF_NSS_ACCOUNT *	文字列	(必須) この Cloud Volumes ONTAP システムで使用するネットアップサポートサイトのアカウント ID。ライセンスタイプが BYOL で、NSS アカウントが指定されていない場合、Cloud Manager は最初の既存の NSS アカウントの使用を試みます。
* AZ_tenant_id *	文字列	(必須) Azure に登録されているアプリケーション / サービスプリンシパルのテナント ID。
* AZ_application_id *	文字列	(必須) Azure に登録されているアプリケーション / サービスプリンシパルのアプリケーション ID。
* AZ_application_key *	文字列	(必須) Azure に登録されているアプリケーション / サービスプリンシパルのアプリケーションキー。



CVO シングルノードの導入

GCPにNetApp CVO（シングルノードインスタンス）を導入するための**Terraform**構成ファイル

このセクションでは、GCP（Google Cloud Platform）でシングルノードの NetApp CVO（Cloud Volumes ONTAP）を導入 / 構成するための、さまざまな Terraform 構成ファイルについて説明します。

Terraform ドキュメント：<https://registry.terraform.io/providers/NetApp/netapp-cloudmanager/latest/docs>

手順

テンプレートを実行するには、次の手順を実行します。

1. リポジトリをクローニングします。

```
git clone
https://github.com/NetApp/na_cloud_volumes_automation.git
```

2. 目的のフォルダに移動します

```
cd na_cloud_volumes_automation/
```

3. GCP 認証キー JSON ファイルをディレクトリに保存します。
4. 変数 /GCP_CVP_SILE_Node_deployment.tfvar' の変数値を更新します



変数「gCP_Connector_deploy_bool」の値を「true」または「false」に設定することで、コネクタの配置を選択できます。

5. Terraform リポジトリを初期化して、すべての前提条件をインストールし、導入の準備をします。

```
terraform init
```

6. terraform validate コマンドを使用して、terraform ファイルを確認します。

```
terraform validate
```

7. 設定を事前に実行して、導入で想定されるすべての変更をプレビューします。

```
terraform plan -target="module.gco_single_node" -var
-file="vars/gcp_cvo_single_node_deployment.tfvars"
```

8. 導入を実行

```
terraform apply -target="module.gcp_single_node" -var  
-file="vars/gcp_cvo_single_node_deployment.tfvars"
```

展開を削除します

```
terraform destroy
```

降水量：

「コネクタ」

NetApp GCP Connector インスタンスの CVO 導入用の Terraform 変数。

* 名前 *	* タイプ *	* 概要 *
* gCP_Connector_deploy_bool *	ブール 値	(必須) コネクタの配置を確認します。
* GCP_Connector_name *	文字列	(必須) Cloud Manager Connector の名前。
* gCP_Connector_project_id *	文字列	(必須) コネクタを作成する GCP PROJECT_ID。
* gCP_Connector_zone *	文字列	(必須) コネクタを作成する GCP ゾーン。
* gp_connector_company *	文字列	(必須) ユーザの会社名。
* gCP_Connector_service_account_email *	文字列	(必須) Connector インスタンスの SERVICE_ACCOUNT の電子メール。このサービスアカウントは、コネクタによるクラウドボリューム ONTAP の作成を許可するために使用されます。
* gCP_Connector_service_account_path *	文字列	(必須) GCP 認証に使用する service_account JSON ファイルのローカルパス。このサービスアカウントは、GCP でコネクタを作成するために使用します。
* gCP_Connector_account_id *	文字列	(オプション) コネクタに関連付けるネットアップアカウントの ID。指定しない場合、Cloud Manager は最初のアカウントを使用します。アカウントが存在しない場合、Cloud Manager は新しいアカウントを作成します。アカウント ID は、Cloud Manager のアカウントタブにあります https://cloudmanager.netapp.com 。

「シングルノードインスタンス」

GCP 上の単一の NetApp CVO インスタンスの Terraform 変数。

* 名前 *	* タイプ *	* 概要 *
* GCP_CVO-NAME *	文字列	(必須) Cloud Volumes ONTAP 作業環境の名前。
* GCP_CVO-PROJECT_ID *	文字列	(必須) GCP プロジェクトの ID 。
* GCP_CVP_ZONE *	文字列	(必須) 作業環境を作成するリージョンのゾーン。
* GCP_CVP_GCP_SERVICE_ACCOUNT *	文字列	(必須) コールドデータを Google Cloud Storage に階層化できるようにするための、GCP_SERVICE_ACCOUNT 電子メール。
* GCP_CVO-svm_svm_password* をクリックします	文字列	(必須) Cloud Volumes ONTAP の管理パスワード。
* GCP_CVP_Workspace_id *	文字列	(オプション) Cloud Volumes ONTAP を導入する Cloud Manager ワークスペースの ID 。指定しない場合、Cloud Manager は最初のワークスペースを使用します。ID は、の [ワークスペース (Workspace)] タブで確認できます https://cloudmanager.netapp.com 。
* GCP_CVP_LICENSE_TYPE *	文字列	(任意) 使用するライセンスのタイプ。シングルノードの場合： [容量 - 給与]、[GCP - COT - EXPLORETe-paygo]、[GCP - COT - standard-paygo]、[GCP - COT - Premium-paygo]、[GCP - COT - Premium-BYOL]、HA の場合： [HA キャパシティ - ペイゴー]、[GCP - HA - ベッド - 探検 - ペイゴー]、[GCP - HA - ベビーベッド - スタンダード - ペイゴー]、[GCP - HA - ベビーベッド - プレミアム - ペイゴー]、[GCP - HA - ベビーベッド - プレミアム - BYOL]。デフォルトは、単一ノードの場合は「capacity-paygo」、HA の場合は「ha-capacity-paygo」です。
* GCP_CVP_capacity_package_name *	文字列	(オプション) 容量パッケージの名前： ['Essential','Professional','Freemium']。デフォルトは「Essential」です。

CVO HA の導入

NetApp CVO (HAペア) を **GCP** に導入するための構成ファイルを **Terraform** しています

このセクションでは、GCP (Google Cloud Platform) のハイアベイラビリティペアで NetApp CVO (Cloud Volumes ONTAP) を導入 / 構成するための、さまざまな Terraform 構成ファイルについて説明します。

Terraform ドキュメント : <https://registry.terraform.io/providers/NetApp/netapp-cloudmanager/latest/docs>

手順

テンプレートを実行するには、次の手順を実行します。

1. リポジトリをクローニングします。

```
git clone
https://github.com/NetApp/na_cloud_volumes_automation.git
```

2. 目的のフォルダに移動します

```
cd na_cloud_volumes_automation/
```

3. GCP 認証キー JSON ファイルをディレクトリに保存します。

4. 変数 /GCP_CVP_HA_deployment.tfvars の変数値を更新します。



変数「gCP_Connector_deploy_bool」の値を「true」または「false」に設定することで、コネクタの配置を選択できます。

5. Terraform リポジトリを初期化して、すべての前提条件をインストールし、導入の準備をします。

```
terraform init
```

6. terraform validate コマンドを使用して、terraform ファイルを確認します。

```
terraform validate
```

7. 設定を事前に実行して、導入で想定されるすべての変更をプレビューします。

```
terraform plan -target="module.gcp_ha" -var
-file="vars/gcp_cvo_ha_deployment.tfvars"
```

8. 導入を実行

```
terraform apply -target="module.gcp_ha" -var
-file="vars/gcp_cvo_ha_deployment.tfvars"
```

展開を削除します


```
terraform destroy
```

降水量：

「コネクタ」

NetApp GCP Connector インスタンスの CVO 導入用の Terraform 変数。

* 名前 *	* タイプ *	* 概要 *
* gCP_Connector_deploy_bool *	ブール値	(必須) コネクタの配置を確認します。
* GCP_Connector_name *	文字列	(必須) Cloud Manager Connector の名前。
* gCP_Connector_project_id *	文字列	(必須) コネクタを作成する GCP PROJECT_ID。
* gCP_Connector_zone *	文字列	(必須) コネクタを作成する GCP ゾーン。
* gp_connector_company *	文字列	(必須) ユーザの会社名。
* gCP_Connector_service_account_email *	文字列	(必須) Connector インスタンスの SERVICE_ACCOUNT の電子メール。このサービスアカウントは、コネクタによるクラウドボリューム ONTAP の作成を許可するために使用されます。
* gCP_Connector_service_account_path *	文字列	(必須) GCP 認証に使用する service_account JSON ファイルのローカルパス。このサービスアカウントは、GCP でコネクタを作成するために使用します。
* gCP_Connector_account_id *	文字列	(オプション) コネクタに関連付けるネットアップアカウントの ID。指定しない場合、Cloud Manager は最初のアカウントを使用します。アカウントが存在しない場合、Cloud Manager は新しいアカウントを作成します。アカウント ID は、Cloud Manager のアカウントタブにあります https://cloudmanager.netapp.com 。

HA ペア

GCP の HA ペアの NetApp CVO インスタンスの Terraform 変数。

* 名前 *	* タイプ *	* 概要 *
* GCP_CVP_is_HA *	ブール値	(オプション) 作業環境が HA ペアであるかどうかを示します ([true 、 false]) 。デフォルトは false です。
* GCP_CVO-NAME *	文字列	(必須) Cloud Volumes ONTAP 作業環境の名前。
* GCP_CVO-PROJECT_ID *	文字列	(必須) GCP プロジェクトの ID 。
* GCP_CVP_ZONE *	文字列	(必須) 作業環境を作成するリージョンのゾーン。
* GCP_CVP_node1_ZONE *	文字列	(オプション) ノード 1 のゾーン。
* GCP_CVP_node2_ZONE *	文字列	(オプション) ノード 2 のゾーン。
* GCP_CVP_mediator_zone *	文字列	(オプション) メディエーター用のゾーン。
* GCP_CVP_vPC_id *	文字列	(オプション) VPC の名前。
* GCP_CVP_subnet_id *	文字列	(オプション) Cloud Volumes ONTAP のサブネットの名前。デフォルトは「 default 」です。
* GCP_CVP_vpc0_Node_or_data_connectivity*	文字列	(オプション) NIC 1 の VPC パス。ノードとデータの接続に必要です。共有 VPC を使用する場合は、network_project_id を指定する必要があります。
* GCP_CVP_vpc1_cluster_connectivity*	文字列	(オプション) NIC 2 の VPC パス。クラスタ接続に必要です。
* GCP_CVP_vpc2_HA_Connectivity *	文字列	(オプション) HA 接続に必要な NIC 3 の VPC パス。
* GCP_CVP_vpc3_data_replication *	文字列	(オプション) データレプリケーションに必要な NIC4 の VPC パス。
* GCP_CVP_SUBnet0_Node_or_data_connectivity*	文字列	(任意) ノードおよびデータ接続に必要な NIC1 のサブネットパス。共有 VPC を使用する場合は、network_project_id を指定する必要があります。

* 名前 *	* タイプ *	* 概要 *
* GCP_CVP_SUBNet1_cluster_connectivity*	文字列	(オプション) クラスタ接続に必要な NIC 2 のサブネット・パス
* GCP_CVP_SUBNet2_HA_connectivity *	文字列	(任意) HA 接続に必要な NIC 3 のサブネットパス。
* GCP_CVP_SUBNet3_data_replication *	文字列	(任意) データ複製に必要な NIC4 のサブネット・パス
* GCP_CVP_GCP_SERVICE_ACCOUNT *	文字列	(必須) コールドデータを Google Cloud Storage に階層化できるようにするための、GCP_SERVICE_ACCOUNT 電子メール。
* GCP_CVO-svm_svm_password* をクリックします	文字列	(必須) Cloud Volumes ONTAP の管理パスワード。
* GCP_CVP_Workspace_id *	文字列	(オプション) Cloud Volumes ONTAP を導入する Cloud Manager ワークスペースの ID。指定しない場合、Cloud Manager は最初のワークスペースを使用します。ID は、の [ワークスペース (Workspace)] タブで確認できます https://cloudmanager.netapp.com 。
* GCP_CVP_LICENSE_TYPE *	文字列	(任意) 使用するライセンスのタイプ。シングルノードの場合: [容量 - 給与]、[GCP - COT - EXPLOREte-paygo]、[GCP - COT - standard-paygo]、[GCP - COT - Premium-paygo]、[GCP - COT - Premium-BYOL]、HA の場合: [HA キャパシティ - ペイゴ]、[GCP - HA - ベッド - 探検 - ペイゴ]、[GCP - HA - ベビーベッド - スタンダード - ペイゴ]、[GCP - HA - ベビーベッド - プレミアム - ペイゴ]、[GCP - HA - ベビーベッド - プレミアム - BYOL]。デフォルトは、単一ノードの場合は「capacity-paygo」、HA の場合は「ha-capacity-paygo」です。
* GCP_CVP_capacity_package_name *	文字列	(オプション) 容量パッケージの名前: ['Essential','Professional','Freemium']。デフォルトは「Essential」です。
* GCP_CVP_GCP_volume_size *	文字列	(オプション) 最初のデータアグリゲートの GCP ボリュームサイズ。GB の場合、単位は [100 または 500] です。TB の場合、単位は [1,2,4,8] です。デフォルトは「1」です。
* GCP_CVP_GCP_volume_size_unit *	文字列	(オプション) ['GB' または 'TB']。デフォルトは「TB」です。

CVS ボリューム

GCPにNetApp CVSボリュームを導入するためのTerraform構成ファイル

このセクションでは、GCP（Google Cloud Platform）で NetApp CVS（Cloud Volume サービス）ボリュームを導入 / 設定するためのさまざまな Terraform 設定ファイルについて説明します。

Terraform ドキュメント : <https://registry.terraform.io/providers/NetApp/netapp-gcp/latest/docs>

手順

テンプレートを実行するには、次の手順を実行します。

1. リポジトリをクローニングします。

```
git clone
https://github.com/NetApp/na_cloud_volumes_automation.git
```

2. 目的のフォルダに移動します

```
cd na_cloud_volumes_automation/
```

3. GCP 認証キー JSON ファイルをディレクトリに保存します。
4. 変数 /gcp_cvs_volume.tfvars の変数値を更新します
5. Terraform リポジトリを初期化して、すべての前提条件をインストールし、導入の準備をします。

```
terraform init
```

6. terraform validate コマンドを使用して、terraform ファイルを確認します。

```
terraform validate
```

7. 設定を事前に実行して、導入で想定されるすべての変更をプレビューします。

```
terraform plan -target="module.gcp_cvs_volume" -var
-file="vars/gcp_cvs_volume.tfvars"
```

8. 導入を実行

```
terraform apply -target="module.gcp_cvs_volume" -var
-file="vars/gcp_cvs_volume.tfvars"
```

展開を削除します

```
terraform destroy
```

降水量：

「CVS ボリューム」

NetApp GCP CVS ボリュームの Terraform 変数。

* 名前 *	* タイプ *	* 概要 *
* gcp_cvs_name *	文字列	(必須) NetApp CVS ボリュームの名前。
* gcp_cvs_project_id *	文字列	(必須) CVS ボリュームを作成する GCP project_id。
* gcp_cvs_gcp_service_account_path *	文字列	(必須) GCP 認証に使用する service_account JSON ファイルのローカルパス。このサービスアカウントは、GCP で CVS ボリュームを作成するために使用します。
* gcp_cvs_region *	文字列	(必須) CVS ボリュームを作成する GCP ゾーン。
* gcp_cvs_network *	文字列	(必須) ボリュームのネットワーク VPC。
* gcp_cvs_size *	整数	(必須) ボリュームのサイズは、1024~102400 で (GiB 単位)。
* gcp_cvs_volume_path *	文字列	(オプション) ボリュームのボリュームパスの名前。
* gcp_cvs_protocol_types *	文字列	(必須) ボリュームの protocol_type。NFS の場合は「NFSv3」または「NFSv4」を、SMB の場合は「CIFS」または「MB」を使用します。

AWS Lambda関数を使用したFSx for ONTAP の監視と自動サイズ変更

概要：AWS Lambda機能を使用したFSx for ONTAP の監視と自動サイズ変更

FSx for ONTAPは、AWSで利用できるファーストパーティのエンタープライズクラスのクラウドストレージサービスです。人気のあるNetApp ONTAPファイルシステムを基盤に、信頼性、拡張性、パフォーマンス、機能豊富なファイルストレージを提供します。

FSx for ONTAP は、シームレスな導入と管理のエクスペリエンスを提供します。ストレージの専門知識は必要ありません。監視を簡易化するために、AWSラムダ機能（しきい値に基づいて合計ストレージ容量、ボリュームサイズ、LUNサイズのサイズを自動変更）を使用できます。このドキュメントでは、FSx for ONTAP を定期的に監視し、ユーザ指定のしきい値を超えたときに通知とサイズ変更を行い、サイズ変更処理を管理者に通知する自動セットアップを作成するためのステップバイステップのガイドを提供します。

解決策 には次の機能があります。

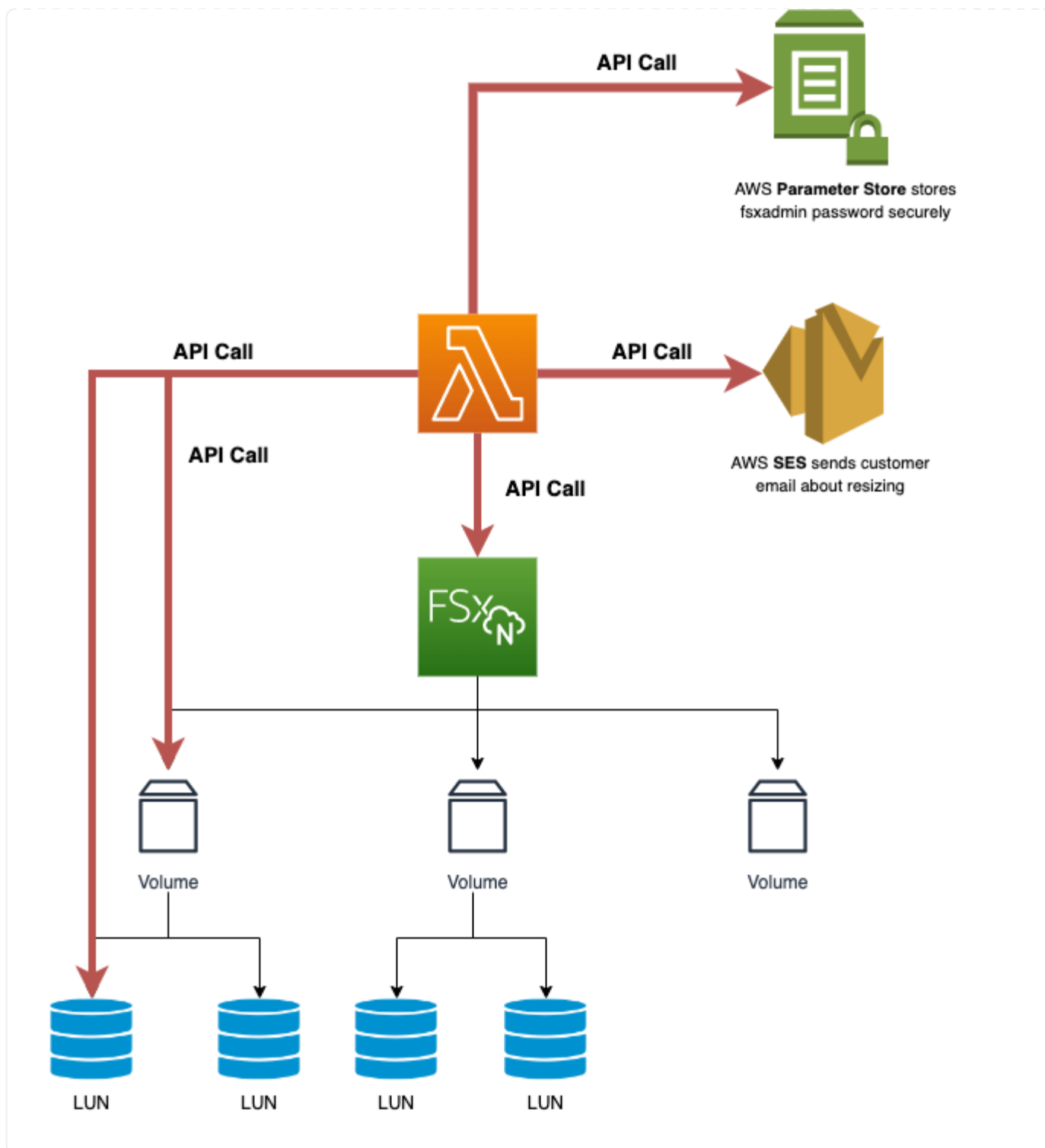
- 監視機能：
 - FSx for ONTAP の全体的なストレージ容量の使用状況
 - 各ボリュームの使用量（シンプロビジョニング/シックプロビジョニング）
 - 各LUNの使用量（シンプロビジョニング/シックプロビジョニング）
- ユーザ定義のしきい値を超えた場合に、上記のいずれかのサイズを変更できます
- 使用状況の警告やサイズ変更の通知をEメールで受信するアラートメカニズム
- ユーザ定義のしきい値を超過した古いSnapshotを削除する機能
- 関連付けられているFlexCloneボリュームとSnapshotのリストを取得する機能
- 定期的にチェックを監視する機能
- インターネットアクセスの有無にかかわらず、解決策 を使用できます
- 手動またはAWS CloudFormationテンプレートを使用してデプロイする機能
- 1つのVPCで複数のFSx for ONTAPファイルシステムを監視する機能

前提条件

作業を開始する前に、次の前提条件を満たしていることを確認してください。

- FSx for ONTAP が導入されました
- FSx for ONTAP への接続を備えたプライベートサブネット
- FSx for ONTAP には「fsxadmin」パスワードが設定されています

- AWS Lambda関数は、ストレージ容量、ボリューム、LUNのサイズを取得および更新するために、FSx for ONTAP へのAPI呼び出しを行います。
- 「fsxadmin」パスワードは、セキュリティレイヤを追加するためにAWS SSMパラメータストアにセキュアな文字列として保存されます。
- AWS SES（Simple Email Service）は、サイズ変更イベントが発生したときにエンドユーザーに通知するために使用します。
- インターネットアクセスのないVPCに解決策を導入する場合は、AWS SSM、FSx、SESのVPCエンドポイントが設定され、LambdaがAWS内部ネットワーク経由でこれらのサービスにアクセスできるようになります。



解決策 の導入

導入の自動化



単一のFSx for ONTAPファイルシステムに対応

一連の手順に従って、この解決策 の自動導入を完了します。

ステップ1: GitHubリポジトリのクローンを作成します

ローカルシステム上のGitHubリポジトリのクローンを作成します。

```
git clone https://github.com/NetApp/fsxn-monitoring-auto-resizing.git
```

ステップ2: AWS S3バケットをセットアップする

1. [AWS Console]>* S3 に移動し、[Create bucket]*をクリックします。デフォルトの設定でバケットを作成します。
2. バケット内で、* Upload > Add files をクリックし、システム上のクローンされた**GitHub**リポジトリから Utilities.zip *を選択します。

The screenshot shows the AWS S3 console interface. At the top, the navigation bar includes the AWS logo, 'Services', a search bar, and a keyboard shortcut '[Option+S]'. The breadcrumb trail indicates the path: Amazon S3 > Buckets > vedant-fsx-bucket > Upload. The main heading is 'Upload' with an 'Info' link. Below this, a message states: 'Add the files and folders you want to upload to S3. To upload a file larger than 160GB, use the AWS CLI, AWS SDK or Amazon S3 REST API. [Learn more](#)'. A dashed box contains the instruction: 'Drag and drop files and folders you want to upload here, or choose **Add files** or **Add folder**.' Below this, a section titled 'Files and folders (1 Total, 27.4 MB)' contains buttons for 'Remove', 'Add files', and 'Add folder'. A note says 'All files and folders in this table will be uploaded.' There is a search bar labeled 'Find by name' and a pagination control showing '< 1 >'. A table lists the files to be uploaded:

<input type="checkbox"/>	Name	Folder	Type	Size
<input type="checkbox"/>	Utilities.zip	-	application/zip	27.4 MB

Below the table, the 'Destination' section shows the upload path: 's3://vedant-fsx-bucket'. It also includes a 'Destination details' section with the text: 'Bucket settings that impact new objects stored in the specified destination.'

手順3：AWS SES SMTPのセットアップ（インターネットアクセスがない場合は必須）

インターネットアクセスなしで解決策を導入する場合は、次の手順を実行します（注：VPCエンドポイントのセットアップに関連するコストが発生します）。

1. AWSコンソール>* AWS Simple Email Service（SES）> SMTP Settings に移動し、Create SMTP credentials *をクリックします。
2. IAMユーザ名を入力するか、デフォルト値のままにして*をクリックします。さらに使用できるように、SMTPユーザ名*と* SMTPパスワード*を保存してください。



SES SMTPのセットアップがすでに設定されている場合は、この手順をスキップします。

Specify user details

Create user for SMTP

Create an IAM user with SMTP credentials for SMTP authentication with Amazon SES.

User name

ses-smtp-user.20230904-201255

The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and * , _ , @ , - (hyphen)

Permissions policy for user

This permissions policy gives the user permissions to access AWS SES.

```
1 {  
2   "Version": "2012-10-17",  
3   "Statement": [  
4     {  
5       "Effect": "Allow",  
6       "Action": "ses:SendRawEmail",  
7       "Resource": "*" }  
8   ]  
9 }  
10 }
```

Tags - optional

Tags are key-value pairs you can add to AWS resources to help identify, organize, or search for resources. Choose any tags you want to associate with this user.

No tags associated with the resource.

Add new tag

You can add up to 50 more tags.

Cancel

Create user

ウィンドウを示しています"]

ステップ4：AWS CloudFormationの導入

1. AWS Console > CloudFormation > Create stack > With New Resources (Standard) に移動します。

```
Prepare template: Template is ready
Specify template: Upload a template file
Choose file: Browse to the cloned GitHub repo and select fsxn-
monitoring-solution.yaml
```

The screenshot shows the 'Create stack' wizard in the AWS CloudFormation console. On the left, a sidebar lists the steps: Step 1: Create stack (active), Step 2: Specify stack details, Step 3: Configure stack options, and Step 4: Review. The main content area is titled 'Create stack' and has two sections. The 'Prerequisite - Prepare template' section has three radio buttons: 'Template is ready' (selected), 'Use a sample template', and 'Create template in Designer'. The 'Specify template' section has a 'Template source' section with two radio buttons: 'Amazon S3 URL' and 'Upload a template file' (selected). Below this, there is a 'Choose file' button and the filename 'fsxn-monitoring-solution.yaml'. At the bottom, the 'S3 URL' is shown as 'https://s3.us-west-1.amazonaws.com/cf-templates-90gl94i22sf-us-west-1/2023-03-29T065412.191Zrsr-fsxn-monitoring-solution.yaml'. There are 'Cancel' and 'Next' buttons at the bottom right.

ウィンドウを示しています"]

[次へ]をクリックします

2. スタックの詳細を入力します。[Next]をクリックし、[I acknowledge that AWS CloudFormation might create IAM resources]のチェックボックスをオンにして、[Submit]をクリックします。



「VPCにインターネットアクセスがありますか？」の場合 はFalseに設定されています。「AWS SESのSMTPユーザ名」と「AWS SESのSMTPパスワード」は必須です。それ以外の場合は、空のままにすることができます。

Specify stack details

Stack name

Stack name

DemoFSxMonitoringSolution

Stack name can include letters (A-Z and a-z), numbers (0-9), and dashes (-).

Parameters

Parameters are defined in your template and allow you to input custom values when you create or update a stack.

Network Configuration

VPC:

Choose the Id of the VPC.

vpc: [REDACTED]

Private Subnet 1:

Choose the Id of the private subnet 1 in Availability Zone 1.

subnet: [REDACTED]

Private Subnet 2:

Choose the Id of the private subnet 2 in Availability Zone 2.

subnet: [REDACTED]

FSx for ONTAP Configuration

Management IP address

Enter the "Management endpoint - IP address" from the FSx for ONTAP console on AWS.

10.10.10.10

File System ID

Enter the "File system ID" from the FSx for ONTAP console on AWS.

fs [REDACTED]

ONTAP Administrator Username

Enter the FSx for ONTAP "ONTAP administrator username" from FSx for ONTAP console on AWS.

fsxadmin

Password for ONTAP Administrator Account

Enter the password set for ONTAP Administrator user for FSx for ONTAP.

Resize Threshold (%)

Enter the threshold percentage from 0-100. This threshold will be used to measure Storage Capacity, Volume and LUN usage and when the % use of any increases above this threshold, resize activity will occur.

90

Enable Warning Notifications

Set this variable to True to receive notification when Storage Capacity/Volume/LUN usage exceeds 75% but is less than threshold.

True

Enable Snapshot Deletion

Set this variable to True to enable volume level snapshot deletion for snapshots older than the value specified in "Snapshot Age Threshold for Deletion (No. of Days)".

True

Snapshot Age Threshold for Deletion (No. of Days)

Enter the number of days of volume level snapshots you want to retain. Any snapshots older than the value provided will be deleted and the same will be notified via email.

30

General Configuration

S3 Bucket Name

Enter the name of the S3 Bucket where paramiko.zip and requests.zip is uploaded. Ensure S3 key for paramiko.zip is paramiko.zip and for requests.zip is requests.zip.

DemoFSxMonitoringBucket

Does VPC have Internet access?

Set this variable to True if the VPC used for deploying this solution has access to Internet. Set to False otherwise.

True

Does SSM VPC Endpoint already exist for the selected VPC?

If Internet access is not available, set this variable to True if the VPC Endpoint for SSM already exists in the VPC. Set to False otherwise.

False

Does FSx VPC Endpoint already exist for the selected VPC?

If Internet access is not available, set this variable to True if the VPC Endpoint for FSx already exists in the VPC. Set to False otherwise.

False

Does SES VPC Endpoint already exist for the selected VPC?

If Internet access is not available, set this variable to True if the VPC Endpoint for SES already exists in the VPC. Set to False otherwise.

False

SMTP Username for AWS SES

If Internet access is not available, enter the smtp username for AWS SES.

Enter String

SMTP Password for AWS SES

If Internet access is not available, enter the smtp password for AWS SES.

Enter String

Sender Email ID
Enter the email ID registered on SES that will be used by the lambda function to send notification alerts related to monitoring and resizing.

abc@xyz.com

Receiver Email ID
Enter the email ID on which you want to receive the alert notifications.

abc@xyz.com

Schedule Expression for frequency of running the solution
Self-trigger your target on an automated schedule using Cron or rate expressions. Cron expressions are in UTC. e.g. rate(1 day), cron(0 17 * * MON-FRI ?).

rate(1 day)

Cancel Previous Next

- CloudFormationデプロイメントが開始されると、「送信者メールID」に記載されているメールIDに、AWS SESでのメールアドレスの使用を承認するように求めるメールが送信されます。リンクをクリックしてメールアドレスを確認します。
- CloudFormationスタックのデプロイが完了すると、警告/通知がある場合は、通知の詳細が記載された電子メールが受信者の電子メールIDに送信されます。

FSx for ONTAP Monitoring

File System Storage Capacity Notification

Storage Capacity used is greater than 90%. File System Storage Capacity resized to: 1240 GB

Volume Notification

Volume Name	Use %	Notification Type	Updated Size
clonevol3	88.39%	Warning	
vol2	88.39%	Warning	
clonevol2	88.39%	Warning	
vol1	78.43%	Warning	

Snapshot Notification

Snapshot Name	Volume Name	Snapshot Age	Space Freed Up	Status
clone_clonevol2.2023-03-22_095434.0	vol2	1 day	296KB	Deleted
clone_clonevol3.2023-03-22_170720.0	clonevol2	1 day	392KB	Deleted

Clone Information

Volume Name	Parent Snapshot	Snapshot Size
clonevol2	clone_clonevol3.2023-03-22_170720.0	392.0KB
vol2	clone_clonevol2.2023-03-22_095434.0	296.0KB

手動での導入



1つのVPCで複数のFSx for ONTAPファイルシステムを監視できます。

一連の手順に従って、この解決策 の手動導入を完了します。

ステップ1: GitHubリポジトリのクローンを作成します

ローカルシステム上のGitHubリポジトリのクローンを作成します。

```
git clone https://github.com/NetApp/fsxn-monitoring-auto-resizing.git
```

手順2: AWS SES SMTPのセットアップ（インターネットアクセスがない場合は必須）

インターネットアクセスなしで解決策を導入する場合は、次の手順を実行します（注：VPCエンドポイントのセットアップに関連するコストが発生します）。

1. AWSコンソール> AWS Simple Email Service (SES) > **SMTP Settings**に移動し、**Create SMTP credentials** *をクリックします
2. IAMユーザ名を入力するか、デフォルト値のままにして、[Create]をクリックします。あとで使用するために、ユーザ名とパスワードを保存しておきます。

Specify user details

Create user for SMTP
Create an IAM user with SMTP credentials for SMTP authentication with Amazon SES.

User name

The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and * * _ - (hyphen)

Permissions policy for user
This permissions policy gives the user permissions to access AWS SES.

```
1 {  
2   "Version": "2012-10-17",  
3   "Statement": [  
4     {  
5       "Effect": "Allow",  
6       "Action": "ses:SendRawEmail",  
7       "Resource": "*" }  
8   ]  
9 }  
10 }
```

Tags - optional
Tags are key-value pairs you can add to AWS resources to help identify, organize, or search for resources. Choose any tags you want to associate with this user.

No tags associated with the resource.

Add new tag

You can add up to 50 more tags.

Cancel Create user

ウィンドウを示しています"]

手順3：fsxadminパスワードのSSMパラメータを作成します

[AWS Console]>*に移動し、[Create Parameter]*をクリックします。

```
Name: <Any name/path for storing fsxadmin password>
Tier: Standard
Type: SecureString
KMS key source: My current account
  KMS Key ID: <Use the default one selected>
Value: <Enter the password for "fsxadmin" user configured on FSx for
ONTAP>
```

[パラメーターを作成（Create Parameter）]*をクリックする。
監視するすべてのFSx for ONTAPファイルシステムについて、上記の手順を繰り返します。

The screenshot shows the AWS Management Console interface for creating a new parameter. The breadcrumb navigation indicates the path: AWS Systems Manager > Parameter Store > Create parameter. The 'Create parameter' page is displayed with the following details:

- Parameter details**
 - Name:** /fsxn/password/
 - Description — Optional:** (empty text box)
 - Tier:** Standard (selected). Parameter Store offers standard and advanced parameters.
 - Standard:** Limit of 10,000 parameters. Parameter value size up to 4 KB. Parameter policies are not available. No additional charge.
 - Advanced:** Can create more than 10,000 parameters. Parameter value size up to 8 KB. Parameter policies are available. Charges apply.
 - Type:** SecureString (selected).
 - String:** Any string value.
 - StringList:** Separate strings using commas.
 - SecureString:** Encrypt sensitive data using KMS keys from your account or another account.
 - KMS key source:** My current account (selected).
 - My current account:** Use the default KMS key for this account or specify a customer-managed key for this account. [Learn more](#)
 - Another account:** Use a KMS key from another account [Learn more](#)
 - KMS Key ID:** alias/aws/ssm
- Value:** (masked with asterisks)

A warning message is displayed: "You have selected the default AWS managed key. All users in the current AWS account and Region will have access to this parameter. To restrict access to the parameter, use a customer managed key (CMK) instead. [Learn more](#)".

インターネットにアクセスせずに解決策を展開する場合は、SMTPユーザ名とSMTPパスワードを保存する手順と同じ手順を実行します。それ以外の場合は、これら2つのパラメータの追加をスキップします

ステップ4:電子メールサービスをセットアップします

[AWS Console]>*に移動し、[Create Identity]*をクリックします。

Identity type: Email address

Email address: <Enter an email address to be used for sending resizing notifications>

[Create identity]*をクリックします

「送信者EメールID」に記載されているEメールIDには、AWS SESでのEメールアドレスの使用許可を所有者に求めるEメールが送信されます。リンクをクリックしてメールアドレスを確認します。

The screenshot shows the AWS Management Console interface. At the top, there's a navigation bar with the AWS logo, 'Services' menu, a search bar, and a '[Option+S]' button. Below this is a 'Resource Groups & Tag Editor' section. The main content area is titled 'Amazon SES > Configuration: Verified identities > Create identity'. The page has a heading 'Create identity' followed by a description: 'A verified identity is a domain, subdomain, or email address you use to send email through Amazon SES. Identity verification at the domain level extends to all email addresses under one verified domain identity.' Below this is a form section titled 'Identity details' with an 'Info' link. It contains two radio buttons for 'Identity type': 'Domain' (unselected) and 'Email address' (selected). The 'Email address' option is highlighted with a blue border and includes a note: 'To verify ownership of an email address, you must have access to its inbox to open the verification email.' Below the radio buttons is a text input field for 'Email address' containing 'abc@xyz.com', with a note below it stating: 'Email address can contain up to 320 characters, including plus signs (+), equals signs (=) and underscores (_).' There is also a checkbox for 'Assign a default configuration set' which is unchecked, with a note explaining that enabling it applies the configuration set to messages sent from this identity by default. At the bottom of the form is a section titled 'Tags - optional' with an 'Info' link, stating 'You can add one or more tags to help manage and organize your resources, including identities.' It shows 'No tags associated with the resource.' and an 'Add new tag' button, with a note 'You can add 50 more tags.' At the very bottom right of the page are two buttons: 'Cancel' and 'Create identity'.

aws Services Search [Option+S]
Resource Groups & Tag Editor

Amazon SES > Configuration: Verified identities > Create identity

Create identity

A verified identity is a domain, subdomain, or email address you use to send email through Amazon SES. Identity verification at the domain level extends to all email addresses under one verified domain identity.

Identity details Info

Identity type

☐ Domain
To verify ownership of a domain, you must have access to its DNS settings to add the necessary records.

☒ Email address
To verify ownership of an email address, you must have access to its inbox to open the verification email.

Email address

abc@xyz.com

Email address can contain up to 320 characters, including plus signs (+), equals signs (=) and underscores (_).

☐ Assign a default configuration set
Enabling this option ensures that the assigned configuration set is applied to messages sent from this identity by default whenever a configuration set isn't specified at the time of sending.

Tags - optional Info

You can add one or more tags to help manage and organize your resources, including identities.

No tags associated with the resource.

Add new tag

You can add 50 more tags.

Cancel Create identity

手順5：VPCエンドポイントをセットアップする（インターネットアクセスを使用できない場合は必須）



インターネットアクセスなしで展開されている場合にのみ必要です。VPCエンドポイントに関連する追加コストが発生します。

1. [AWS Console]>*>[Endpoints]に移動し、[Create Endpoint]*をクリックして、次の詳細を入力します。

```
Name: <Any name for the vpc endpoint>
Service category: AWS Services
Services: com.amazonaws.<region>.fsx
vpc: <select the vpc where lambda will be deployed>
subnets: <select the subnets where lambda will be deployed>
Security groups: <select the security group>
Policy: <Either choose Full access or set your own custom policy>
```

[Create endpoint]をクリックします。

[VPC](#) > [Endpoints](#) > Create endpoint

Create endpoint [Info](#)

There are three types of VPC endpoints – Interface endpoints, Gateway Load Balancer endpoints, and Gateway endpoints. Interface endpoints and Gateway Load Balancer endpoints are powered by AWS PrivateLink, and use an Elastic Network Interface (ENI) as an entry point for traffic destined to the service. Interface endpoints are typically accessed using the public or private DNS name associated with the service, while Gateway endpoints and Gateway Load Balancer endpoints serve as a target for a route in your route table for traffic destined for the service.

Endpoint settings

Name tag - optional

Creates a tag with a key of 'Name' and a value that you specify.

Service category

Select the service category



AWS services

Services provided by Amazon



PrivateLink Ready partner services

Services with an AWS Service Ready designation



AWS Marketplace services


Services that you've purchased through AWS Marketplace



Other endpoint services

Find services shared with you by service name

Services (1/1)

< 1 >  

Service Name	Owner	Type
 com.amazonaws.us-west-1.fsx	amazon	Interface

VPC
Select the VPC in which to create the endpoint

VPC
The VPC in which to create your endpoint.

vpc- [REDACTED] (DemoFSxN-vpc) [Refresh]

▶ Additional settings

Subnets (2/2) [Info](#)

<input checked="" type="checkbox"/>	Availability Zone	Subnet ID
<input checked="" type="checkbox"/>	us-west-1a (usw1-az3)	subnet-[REDACTED]
<input checked="" type="checkbox"/>	us-west-1b (usw1-az1)	subnet-[REDACTED]

subnet-[REDACTED] × DemoFSxN-subnet-private1-us-west-1a

subnet-[REDACTED] × DemoFSxN-subnet-private2-us-west-1b

IP address type

☒ IPv4

☐ IPv6

☐ Dualstack

Security groups (1/1) [Info](#)

Find resources by attribute or tag

<input checked="" type="checkbox"/>	Group ID	Group name	VPC ID
<input checked="" type="checkbox"/>	sg-[REDACTED]	default	vpc-[REDACTED]

sg-[REDACTED] ×

2. SESおよびSSM VPCエンドポイントの作成についても、同じ手順を実行します。すべてのパラメータは上記と同じですが、サービスは<region>それぞれ* com.amazonaws.smtp および com.amazonaws.smtp *に対応します<region>。

ステップ6: AWS Lambda関数を作成してセットアップします

1. AWSコンソール>* AWS Lambda > Functions に移動し、**FSx for ONTAP**と同じリージョンの Create Function *をクリックします。
2. デフォルトの*Author from scratch*を使用して、次のフィールドを更新します。

Function name: <Any name of your choice>
Runtime: Python 3.9
Architecture: x86_64
Permissions: Select "Create a new role with basic Lambda permissions"
Advanced Settings:
 Enable VPC: Checked
 VPC: <Choose either the same VPC as FSx for ONTAP or a VPC that can access both FSx for ONTAP and the internet via a private subnet>
 Subnets: <Choose 2 private subnets that have NAT gateway attached pointing to public subnets with internet gateway and subnets that have internet access>
 Security Group: <Choose a Security Group>

[機能の作成]*をクリックします。

Resource Groups & Tag Editor

▼ Advanced settings

☐ **Enable Code signing** [info](#)
Use code signing configurations to ensure that the code has been signed by an approved source and has not been altered since signing.

☐ **Enable function URL** [info](#)
Use function URLs to assign HTTPS endpoints to your Lambda function.

☐ **Enable tags** [info](#)
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources, track your AWS costs, and enforce attribute-based access control.

☒ **Enable VPC** [info](#)
Connect your function to a VPC to access private resources during invocation.

VPC
Choose a VPC for your function to access.
vpc: [redacted] [refresh]

Subnets
Select the VPC subnets for Lambda to use to set up your VPC configuration.
Choose subnets
subnet: [redacted] us-east-1a [x] subnet: [redacted] us-east-1d [x]

Security groups
Choose the VPC security groups for Lambda to use to set up your VPC configuration. The table below shows the inbound and outbound rules for the security groups that you choose.
Choose security groups
sg: [redacted] [x]
default VPC security group

- 新しく作成したLambda関数に移動し、* Layers セクションまでスクロールして Add a layer *をクリックします。

Layers [info](#) Edit Add a layer

Merge order	Name	Layer version	Compatible runtimes	Compatible architectures	Version ARN
There is no data to display.					

- の下にある[新しいレイヤーを作成する]*をクリックします
- レイヤーを作成し、* Utilities.zip ファイルをアップロードします。互換性のあるランタイムとして Python 3.9 を選択し、Create *をクリックします。

aws

Services

Search

[Option+S]

≡

Lambda > Layers > Create layer

Create layer

Layer configuration

Name

Utilities

Description - optional

Paramiko and Requests Libraries

☒ Upload a .zip file

☐ Upload a file from Amazon S3

Upload

Utilities.zip
28.70 MB

For files larger than 10 MB, consider uploading using Amazon S3.

Compatible architectures - optional Info

Choose the compatible instruction set architectures for your layer.

☐ x86_64

☐ arm64

Compatible runtimes - optional Info

Choose up to 15 runtimes.

Runtimes

Python 3.9

License - optional Info

ウィンドウを示しています。"]

6. AWS Lambda Function > * Add Layer > Custom Layers *に戻り、ユーティリティレイヤーを追加します。

Services
Search
[Option+S]

Lambda > Layers > Add layer

Add layer

Function runtime settings

Runtime Python 3.9	Architecture x86_64
-----------------------	------------------------

Choose a layer

Layer source [Info](#)
Choose from layers with a compatible runtime and instruction set architecture or specify the Amazon Resource Name (ARN) of a layer version. You can also [create a new layer](#).

☐ AWS layers
Choose a layer from a list of layers provided by AWS.
 ☒ Custom layers
Choose a layer from a list of layers created by your AWS account or organization.
 ☐ Specify an ARN
Specify a layer by providing the ARN.

Custom layers

Layers created by your AWS account or organization that are compatible with your function's runtime.

Utilities

Version
2

Cancel
Add

Layers [Info](#)

Edit
Add a layer

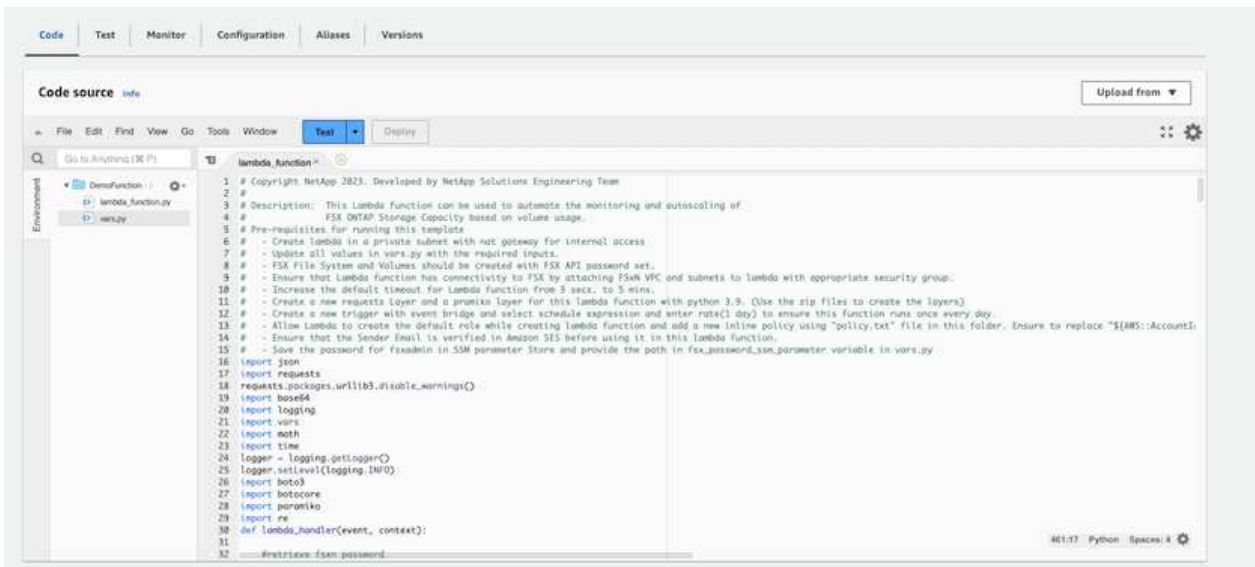
Merge order	Name	Layer version	Compatible runtimes	Compatible architectures	Version ARN
1	Utilities	2	python3.9	-	arn:aws:lambda:us-east-1:██████████:layer:Utilities:2

- Lambda関数の* Configuration タブに移動し、General Configuration の Edit をクリックします。[タイムアウト]を 5分*に変更し、*[保存]*をクリックします。
- Lambda関数の* Permissions タブに移動し、割り当てられたロールをクリックします。ロールの権限タブで、*権限の追加>*インラインポリシーの作成*をクリックします。
 - JSONタブをクリックし、GitHubリポジトリからpolicy.jsonファイルの内容を貼り付けます。
 - \$ {aws : : AccountId} はすべて自分のアカウントIDに置き換えて、*[Review Policy]*をクリックしてください
 - ポリシーの名前を指定し、*[ポリシーの作成]*をクリックします
- AWS Lambda Function Code Sourceセクションの* fsxn_monitoring_resizing_lambda.py の内容をgitリポジトリから lambda_function.py *にコピーします。
- lambda_function.pyと同じレベルに新しいファイルを作成し、vars.py*という名前を付けて、vars.py

の内容をgitリポジトリからlambda関数vars.pyファイルにコピーします。vars.pyの変数値を更新します。以下の変数定義を参照し、Deploy *をクリックします。

* 名前 *	* タイプ *	* 概要 *
* fsxList*	リスト	(必須) 監視するすべてのFSx for ONTAPファイルシステムのリスト。 監視および自動サイズ変更の対象として、すべてのファイルシステムをリストに含めます。
* fsxMgmtIp *	文字列	(必須) AWSのFSx for ONTAPコンソールから「管理エンドポイント- IPアドレス」を入力します。
* fsxId *	文字列	(必須) AWSのFSx for ONTAPコンソールから「ファイルシステムID」を入力します。
ユーザー名	文字列	(必須) AWSのFSx for ONTAPコンソールからFSx for ONTAPの「ONTAP administrator username」を入力します。
* resize_threshold *	整数	(必須) しきい値のパーセンテージを0~100の範囲で入力します。このしきい値は、ストレージ容量、ボリューム、LUNの使用率を測定するために使用され、使用率がこのしきい値を超えるとサイズ変更アクティビティが発生します。
* FSX_PASSWORD_SSM_PARAMETER *	文字列	(必須) 「fsxadmin」パスワードの保存にAWS Parameter Storeで使用するパス名を入力します。
* WARN_NOTIFICATION *	ブール値	(必須) この変数をTrueに設定すると、ストレージ容量/ボリューム/LUNの使用率が75%を超え、しきい値を下回ったときに通知が送信されます。
* enable_snapshot_deletion *	ブール値	(必須)) 「snapshot_age_threshold_in_days」で指定した値よりも古いSnapshotに対してボリュームレベルのSnapshotの削除を有効にするには、この変数をTrueに設定します。

* snapshot_age_threshold_in_days *	整数	(必須) ボリュームレベルのSnapshotを保持する日数を入力します。指定した値よりも古いSnapshotコピーは削除され、同じSnapshotコピーがEメールで通知されます。
* internet_access *	ブール値	(必須) このラムダが展開されているサブネットからインターネットアクセスが可能な場合は、この変数をTrueに設定します。それ以外の場合は、Falseに設定します。
* smtp_region *	文字列	(オプション) 「internet_access」変数がFalseに設定されている場合は、ラムダがデプロイされている領域を入力します。例: us-east-1 (この形式)
* SMTP_USERNAME_SSM_PARAMETER *	文字列	(オプション) 「internet_access」変数がFalseに設定されている場合は、SMTPユーザ名を格納するためにAWS Parameter Storeで使用するパス名を入力します。
* SMTP_PASSWORD_SSM_PARAMETER *	文字列	(オプション) 「internet_access」変数がFalseに設定されている場合は、SMTPパスワードの保存にAWS Parameter Storeで使用するパス名を入力します。
* sender_email *	文字列	(必須) SESに登録されている電子メールIDを入力します。このIDは、監視とサイズ変更に関する通知アラートを送信するためにlambda関数で使用されます。
* recipient_email *	文字列	(必須) アラート通知を受信するEメールIDを入力します。



11. をクリックし、空の**JSON**オブジェクトでテストイベントを作成し、[呼び出し]*をクリックしてテストを実行し、スクリプトが正しく実行されているかどうかを確認します。
12. テストに成功したら、* **Configuration > Triggers > Add Trigger ***に移動します。

Select a Source: EventBridge
 Rule: Create a new rule
 Rule name: <Enter any name>
 Rule type: Schedule expression
 Schedule expression: <Use "rate(1 day)" if you want the function to run daily or add your own cron expression>

[Add]をクリックします。



Add trigger

Trigger configuration [Info](#)



EventBridge (CloudWatch Events)

aws events management-tools



Rule

Pick an existing rule, or create a new one.

- ☒ Create a new rule
- ☐ Existing rules

Rule name

Enter a name to uniquely identify your rule.

DemoFSxNRule

Rule description

Provide an optional description for your rule.

Rule type

Trigger your target based on an event pattern, or based on an automated schedule.

- ☐ Event pattern
- ☒ Schedule expression

Schedule expression

Self-trigger your target on an automated schedule using [Cron or rate expressions](#). Cron expressions are in UTC.

rate(1 day)

e.g. rate(1 day), cron(0 17 ? * MON-FRI *)

Lambda will add the necessary permissions for Amazon EventBridge (CloudWatch Events) to invoke your Lambda function from this trigger. [Learn more](#) about the Lambda permissions model.

まとめ

提供された解決策を使用すると、FSx for ONTAPストレージを定期的に監視し、ユーザ指定のしきい値に基づいてサイズを変更し、アラートメカニズムを提供する監視解決策を簡単に設定できます。これにより、FSx for ONTAP を使用して監視するプロセスがシームレスになり、管理者はビジネスクリティカルな作業に集中できるようになり、必要に応じてストレージが自動的に拡張されます。

著作権に関する情報

Copyright © 2024 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。