



Amazon FSx for NetApp ONTAP管理

NetApp Automation

NetApp
November 18, 2025

目次

Amazon FSx for NetApp ONTAP管理	1
Amazon FSx for NetApp ONTAP管理 - クラウドへのバースト	1
手順1：Dockerをインストールして設定する	1
手順2：Docker Composeをインストールする	2
ステップ3：Dockerイメージを準備する	3
手順4：AWSクレデンシャル用の環境ファイルを作成する	4
手順5：外部ボリュームを作成する	4
ステップ 6: Amazon FSx for NetApp ONTAP管理とFlexCache用に Amazon FSx をプロビジョニングする	5
ステップ 7: Amazon FSx for NetApp ONTAP管理とFlexCache を破棄する	5
Amazon FSx for NetApp ONTAP管理 - 災害復旧	6
手順1：Dockerをインストールして設定する	6
手順2：Docker Composeをインストールする	7
ステップ3：Dockerイメージを準備する	8
手順4：AWSクレデンシャル用の環境ファイルを作成する	9
手順5：外部ボリュームを作成する	9
手順6：バックアップソリューションを導入する	10

Amazon FSx for NetApp ONTAP管理

Amazon FSx for NetApp ONTAP管理 - クラウドへのバースト

この自動化ソリューションを使用すると、ボリュームと関連するFlexCacheを使用してAmazon FSx for NetApp ONTAP管理をプロビジョニングできます。



Amazon FSx for NetApp ONTAP管理は、*FSx for ONTAP*とも呼ばれます。

このソリューションについて

このソリューションで提供される自動化コードでは、大まかに次の処理が実行されます。

- ・デスティネーションのFSx for ONTAPファイルシステムのプロビジョニング
- ・ファイルシステム用のStorage Virtual Machine (SVM) のプロビジョニング
- ・ソースシステムとデスティネーションシステム間にクラスタピア関係を作成する
- ・FlexCacheのソースシステムとデスティネーションシステム間にSVMピア関係を作成する
- ・必要に応じて、FSx for ONTAPを使用してFlexVolボリュームを作成
- ・FSx for ONTAPでFlexCacheボリュームを作成し、ソースをオンプレミスのストレージに設定

この自動化はDockerとDocker Composeに基づいており、以下で説明するようにLinux仮想マシンにインストールする必要があります。

開始する前に

プロビジョニングと設定を完了するには、次の情報が必要です。

- ・ダウンロードする必要があります "[Amazon FSx for NetApp ONTAP管理 - クラウドへのバースト](#)" NetApp ConsoleWeb UI を介した自動化ソリューション。ソリューションはファイルとしてパッケージ化されています AWS_FsxN_BTC.zip。
- ・ソースシステムとデスティネーションシステム間のネットワーク接続。
- ・次の特性を持つLinux VM：
 - DebianベースのLinuxディストリビューション
 - FSx for ONTAPのプロビジョニングと同じVPCサブセットに導入
- ・AWSアカウント。

手順1：Dockerをインストールして設定する

DebianベースのLinux仮想マシンにDockerをインストールして設定します。

手順

1. 環境を準備

```
sudo apt-get update
sudo apt-get install apt-transport-https ca-certificates curl gnupg-
agent software-properties-common
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key
add -
sudo add-apt-repository "deb [arch=amd64]
https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"
sudo apt-get update
```

2. Dockerをインストールし、インストールを確認します。

```
sudo apt-get install docker-ce docker-ce-cli containerd.io
docker --version
```

3. 必要なLinuxグループをユーザと関連付けて追加します。

最初に、グループ* Docker *がLinuxシステムに存在するかどうかを確認します。表示されない場合は、グループを作成してユーザを追加します。デフォルトでは、現在のシェルユーザがグループに追加されます。

```
sudo groupadd docker
sudo usermod -aG docker $(whoami)
```

4. 新しいグループとユーザ定義をアクティビ化する

ユーザを使用して新しいグループを作成した場合は、定義をアクティビ化する必要があります。これを行うには、Linuxからログアウトしてから再度ログインします。または、次のコマンドを実行します。

```
newgrp docker
```

手順2：Docker Composeをインストールする

DebianベースのLinux仮想マシンにDocker Composeをインストールします。

手順

1. Docker Composeをインストールします。

```
sudo curl -L
"https://github.com/docker/compose/releases/latest/download/docker-
compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
sudo chmod +x /usr/local/bin/docker-compose
```

- インストールが正常に完了したことを確認します。

```
docker-compose --version
```

ステップ3：Dockerイメージを準備する

自動化ソリューションに付属のDockerイメージを抽出してロードする必要があります。

手順

- ソリューションファイルを、自動化コードを実行する仮想マシンにコピーし `AWS_FSxN_BTC.zip` ます。

```
scp -i ~/private-key.pem -r AWS_FSxN_BTC.zip user@IP_ADDRESS_OF_VM
```

入力パラメータ `private-key.pem` は、AWS仮想マシン認証（EC2インスタンス）に使用する秘密鍵ファイルです。

- ソリューションファイルを含む適切なフォルダに移動し、ファイルを解凍します。

```
unzip AWS_FSxN_BTC.zip
```

- 解凍操作で作成された新しいフォルダに移動し `aws_fsxn_flexcache_image_latest.tar.gz` を一覧表示します。ファイルが表示されます。

```
ls -la
```

- Dockerイメージファイルをロードします。ロード操作は通常数秒で完了します。

```
docker load -i aws_fsxn_flexcache_image_latest.tar.gz
```

- Dockerイメージがロードされたことを確認します。

```
docker images
```

タグが付いた `latest` Dockerイメージが表示されます `aws_fsxn_flexcache_image`。

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
aws_fsxn_flexcahce_image	latest	ay98y7853769	2 weeks ago	1.19GB

手順4：AWSクレデンシャル用の環境ファイルを作成する

アクセスキーとシークレットキーを使用して認証用のローカル変数ファイルを作成する必要があります。次に、ファイルをファイルに追加し `env` ます。

手順

1. 次の場所にファイルを作成し `awsauth.env` ます。

```
path/to/env-file/awsauth.env
```

2. ファイルに次の内容を追加します。

```
access_key=<>
secret_key=<>
```

形式*は、上記のとの `value` 間にスペースを入れずに正確に指定する必要があります `key`。

3. 変数を使用して、ファイル `AWS_CREDS` への絶対ファイルパスを追加し `env` ます。例：

```
AWS_CREDS=path/to/env-file/awsauth.env
```

手順5：外部ボリュームを作成する

Terraform状態ファイルやその他の重要なファイルが永続的であることを確認するには、外部ボリュームが必要です。ワークフローとデプロイメントを実行するには、Terraformでこれらのファイルが使用可能である必要があります。

手順

1. Docker Composeの外部に外部ボリュームを作成します。

コマンドを実行する前に、ボリューム名（最後のパラメータ）を適切な値に更新してください。

```
docker volume create aws_fsxn_volume
```

2. コマンドを使用して、外部ボリュームへのパスを環境ファイルに追加し `env` ます。

```
PERSISTENT_VOL=path/to/external/volume:/volume_name
```

既存のファイルの内容とコロンの書式を維持することを忘れないでください。例：

```
PERSISTENT_VOL=aws_fsxn_volume:/aws_fsxn_flexcache
```

NFS共有を外部ボリュームとして追加するには、次のようなコマンドを使用します。

```
PERSISTENT_VOL=nfs/mnt/document:/aws_fsx_flexcache
```

3. Terraform変数を更新します。

- フォルダに移動し `aws_fsn_variables` ます。
- との variables.tf` 2つのファイルが存在することを確認します `terraform.tfvars`。
- 環境に応じて、の値を更新します terraform.tfvars。

詳細については、を参照してください ["Terraformリソース : AWS_FSX_APS_FILE_SYSTEM ONTAP"](#)。

ステップ 6: Amazon FSx for NetApp ONTAP管理とFlexCache用に Amazon FSx をプロビジョニングする

Amazon FSx for NetApp ONTAP管理およびFlexCache用にプロビジョニングできます。

手順

- rootフォルダ (AWS_FSN_BTC) に移動し、provisioningコマンドを実行します。

```
docker-compose -f docker-compose-provision.yml up
```

このコマンドは、2つのコンテナを作成します。1つ目のコンテナでFSx for ONTAPが導入され、2つ目のコンテナでクラスタビアリング、SVMピアリング、デスティネーションボリューム、FlexCacheが作成されます。

- プロビジョニングプロセスを監視します。

```
docker-compose -f docker-compose-provision.yml logs -f
```

このコマンドは出力をリアルタイムで表示しますが、ファイルを介してログをキャプチャするように設定されて `deployment.log` います。これらのログファイルの名前を変更するには、ファイルを編集し `.env` で変数を更新し `DEPLOYMENT_LOGS` ます。

ステップ 7: Amazon FSx for NetApp ONTAP管理とFlexCache を破棄する

オプションで、Amazon FSx for NetApp ONTAP管理とFlexCacheを削除および除去できます。

- ファイル内の `terraform.tfvars` 変数を「destroy」に設定し `flexcache_operation` ます。
- rootフォルダ (AWS_FSN_BTC) に移動し、次のコマンドを実行します。

```
docker-compose -f docker-compose-destroy.yml up
```

このコマンドは、2つのコンテナを作成します。最初のコンテナはFlexCacheを削除し、2番目のコンテナはFSx for ONTAPを削除します。

- プロビジョニングプロセスを監視します。

```
docker-compose -f docker-compose-destroy.yml logs -f
```

Amazon FSx for NetApp ONTAP管理 - 災害復旧

この自動化ソリューションを使用すると、Amazon FSx for NetApp ONTAP管理を使用してソースシステムの災害復旧バックアップを取得できます。



Amazon FSx for NetApp ONTAP管理は、*FSx for ONTAP*とも呼ばれます。

このソリューションについて

このソリューションで提供される自動化コードでは、大まかに次の処理が実行されます。

- ・デスティネーションのFSx for ONTAPファイルシステムのプロビジョニング
- ・ファイルシステム用のStorage Virtual Machine (SVM) のプロビジョニング
- ・ソースシステムとデスティネーションシステム間にクラスタピア関係を作成する
- ・SnapMirrorのソースシステムとデスティネーションシステム間にSVMピア関係を作成する
- ・デスティネーションボリュームを作成
- ・ソースボリュームとデスティネーションボリューム間にSnapMirror関係を作成する
- ・ソースボリュームとデスティネーションボリューム間のSnapMirror転送を開始する

この自動化はDockerとDocker Composeに基づいており、以下で説明するようにLinux仮想マシンにインストールする必要があります。

開始する前に

プロビジョニングと設定を完了するには、次の情報が必要です。

- ・ダウンロードする必要があります "[Amazon FSx for NetApp ONTAP管理 - 災害復旧](#)" NetApp ConsoleWeb UI を介した自動化ソリューション。このソリューションは次のようにパッケージ化されています。FSxN_DR.zip。このzipには、AWS_FsxN_Bck_Prov.zip このドキュメントで説明されているソリューションを開くために使用するファイルです。
- ・ソースシステムとデスティネーションシステム間のネットワーク接続。
- ・次の特性を持つLinux VM：
 - DebianベースのLinuxディストリビューション
 - FSx for ONTAPのプロビジョニングと同じVPCサブセットに導入
- ・AWSアカウント。

手順1：Dockerをインストールして設定する

DebianベースのLinux仮想マシンにDockerをインストールして設定します。

手順

1. 環境を準備

```
sudo apt-get update
sudo apt-get install apt-transport-https ca-certificates curl gnupg-
agent software-properties-common
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key
add -
sudo add-apt-repository "deb [arch=amd64]
https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"
sudo apt-get update
```

2. Dockerをインストールし、インストールを確認します。

```
sudo apt-get install docker-ce docker-ce-cli containerd.io
docker --version
```

3. 必要なLinuxグループをユーザと関連付けて追加します。

最初に、グループ* Docker *がLinuxシステムに存在するかどうかを確認します。存在しない場合は、グループを作成してユーザを追加します。デフォルトでは、現在のシェルユーザがグループに追加されます。

```
sudo groupadd docker
sudo usermod -aG docker $(whoami)
```

4. 新しいグループとユーザ定義をアクティブ化する

ユーザを使用して新しいグループを作成した場合は、定義をアクティブ化する必要があります。これを行うには、Linuxからログアウトしてから再度ログインします。または、次のコマンドを実行します。

```
newgrp docker
```

手順2：Docker Composeをインストールする

DebianベースのLinux仮想マシンにDocker Composeをインストールします。

手順

1. Docker Composeをインストールします。

```
sudo curl -L  
"https://github.com/docker/compose/releases/latest/download/docker-  
compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose  
sudo chmod +x /usr/local/bin/docker-compose
```

- インストールが正常に完了したことを確認します。

```
docker-compose --version
```

ステップ3：Dockerイメージを準備する

自動化ソリューションに付属のDockerイメージを抽出してロードする必要があります。

手順

- ソリューションファイルを、自動化コードを実行する仮想マシンにコピーし `AWS_FSxN_Bck_Prov.zip` ます。

```
scp -i ~/private-key.pem -r AWS_FSxN_Bck_Prov.zip  
user@<IP_ADDRESS_OF_VM>
```

入力パラメータ `private-key.pem` は、AWS仮想マシン認証（EC2インスタンス）に使用する秘密鍵ファイルです。

- ソリューションファイルを含む適切なフォルダに移動し、ファイルを解凍します。

```
unzip AWS_FSxN_Bck_Prov.zip
```

- 解凍操作で作成された新しいフォルダに移動し `AWS_FSxN_Bck_Prov`、ファイルを一覧表示します。ファイルが表示されます `aws_fsxn_bck_image_latest.tar.gz`。

```
ls -la
```

- Dockerイメージファイルをロードします。ロード操作は通常数秒で完了します。

```
docker load -i aws_fsxn_bck_image_latest.tar.gz
```

- Dockerイメージがロードされたことを確認します。

```
docker images
```

タグが付いた latest `Dockerイメージが表示されます `aws_fsxn_bck_image`。

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
aws_fsxn_bck_image	latest	da87d4974306	2 weeks ago	1.19GB

手順4：AWSクレデンシャル用の環境ファイルを作成する

アクセスキーとシークレットキーを使用して認証用のローカル変数ファイルを作成する必要があります。次に、ファイルをファイルに追加し ` `.env` ます。

手順

1. 次の場所にファイルを作成し `awsauth.env` ます。

```
path/to/env-file/awsauth.env
```

2. ファイルに次の内容を追加します。

```
access_key=<>
secret_key=<>
```

形式*は、上記のとの `value` 間にスペースを入れずに正確に指定する必要があります `key`。

3. 変数を使用して、ファイル `AWS_CREDS` への絶対ファイルパスを追加し ` `.env` ます。例：

```
AWS_CREDS=path/to/env-file/awsauth.env
```

手順5：外部ボリュームを作成する

Terraform状態ファイルやその他の重要なファイルが永続的であることを確認するには、外部ボリュームが必要です。ワークフローとデプロイメントを実行するには、Terraformでこれらのファイルが使用可能である必要があります。

手順

1. Docker Composeの外部に外部ボリュームを作成します。

コマンドを実行する前に、ボリューム名（最後のパラメータ）を適切な値に更新してください。

```
docker volume create aws_fsxn_volume
```

2. コマンドを使用して、外部ボリュームへのパスを環境ファイルに追加し ` `.env` ます。

```
PERSISTENT_VOL=path/to/external/volume:/volume_name
```

既存のファイルの内容とコロンの書式を維持することを忘れないでください。例：

```
PERSISTENT_VOL=aws_fsxn_volume:/aws_fsxn_bck
```

NFS共有を外部ボリュームとして追加するには、次のようなコマンドを使用します。

```
PERSISTENT_VOL=nfs/mnt/document:/aws_fsx_bck
```

3. Terraform変数を更新します。

- a. フォルダに移動し `aws_fsxn_variables` ます。
- b. との `variables.tf` 2つのファイルが存在することを確認します `terraform.tfvars`。
- c. 環境に応じて、の値を更新します `terraform.tfvars`。

詳細については、を参照してください ["Terraformリソース : AWS_FSX_APS_FILE_SYSTEM ONTAP"](#)。

手順6：バックアップソリューションを導入する

ディザスタリカバリバックアップソリューションを導入してプロビジョニングできます。

手順

1. rootフォルダ (AWS_FSxN_BCK_Provisioning) に移動し、provisioningコマンドを実行します。

```
docker-compose up -d
```

このコマンドは、3つのコンテナを作成します。1つ目のコンテナはFSx for ONTAPを導入します。2つ目のコンテナでは、クラスタピアリング、SVMピアリング、およびデスティネーションボリュームが作成されます。3番目のコンテナでSnapMirror関係が作成され、SnapMirror転送が開始されます。

2. プロビジョニングプロセスを監視します。

```
docker-compose logs -f
```

このコマンドは出力をリアルタイムで表示しますが、ファイルを介してログをキャプチャするように設定されて `deployment.log` います。これらのログファイルの名前を変更するには、ファイルを編集し `.env` で変数を更新し `DEPLOYMENT_LOGS` ます。

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。