



# **Amazon FSx for NetApp ONTAP**

## **NetApp Automation**

NetApp  
November 18, 2025

This PDF was generated from <https://docs.netapp.com/ko-kr/netapp-automation/solutions/fsxn-burst-to-cloud.html> on November 18, 2025. Always check docs.netapp.com for the latest.

# 목차

Amazon FSx for NetApp ONTAP .....	1
Amazon FSx for NetApp ONTAP - 클라우드로의 버스트 .....	1
1단계: Docker 설치 및 구성 .....	1
2단계: Docker 구성 설치 .....	2
3단계: Docker 이미지 준비 .....	3
4단계: AWS 자격 증명에 대한 환경 파일을 생성합니다 .....	4
5단계: 외부 볼륨을 만듭니다 .....	4
6단계: NetApp ONTAP 관리 및 FlexCache Amazon FSx for NetApp ONTAP 프로비저닝 .....	5
7단계: NetApp ONTAP 관리 및 FlexCache Amazon FSx for NetApp ONTAP 삭제 .....	5
Amazon FSx for NetApp ONTAP 관리 - 재해 복구 .....	6
1단계: Docker 설치 및 구성 .....	6
2단계: Docker 구성 설치 .....	7
3단계: Docker 이미지 준비 .....	8
4단계: AWS 자격 증명에 대한 환경 파일을 생성합니다 .....	9
5단계: 외부 볼륨을 만듭니다 .....	9
6단계: 백업 솔루션을 배포합니다 .....	10

# Amazon FSx for NetApp ONTAP

## Amazon FSx for NetApp ONTAP - 클라우드로의 버스트

이 자동화 솔루션을 사용하면 볼륨과 관련 FlexCache 를 사용하여 Amazon FSx for NetApp ONTAP 프로비저닝할 수 있습니다.



Amazon FSx for NetApp ONTAP 관리 기능은 \*FSx for ONTAP\*라고도 합니다.

살펴봅니다

이 솔루션과 함께 제공되는 자동화 코드는 개략적으로 다음과 같은 작업을 수행합니다.

- 대상 FSx for ONTAP 파일 시스템을 프로비저닝합니다
- 파일 시스템에 대한 SVM(Storage Virtual Machine)을 프로비저닝합니다
- 소스 시스템과 타겟 시스템 간에 클러스터 피어링 관계를 생성합니다
- 소스 시스템과 FlexCache용 타겟 시스템 간에 SVM 피어링 관계를 생성합니다
- 선택적으로 FSx for ONTAP를 사용하여 FlexVol 볼륨을 생성합니다
- 온프레미스 스토리지를 가리키는 FlexCache 볼륨을 FSx for ONTAP에서 생성합니다

자동화는 아래 설명과 같이 Linux 가상 머신에 설치해야 하는 Docker 및 Docker Compose를 기반으로 합니다.

시작하기 전에

프로비저닝 및 구성을 완료하려면 다음이 있어야 합니다.

- 다운로드가 필요합니다 "[Amazon FSx for NetApp ONTAP - 클라우드로의 버스트](#)" NetApp Console 웹 UI를 통한 자동화 솔루션. 솔루션은 파일로 패키징됩니다. `AWS_FSxN_BTC.zip`.
- 소스 시스템과 대상 시스템 간의 네트워크 연결
- 다음과 같은 특성을 가진 Linux VM:
  - 데비안 기반 리눅스 배포판
  - FSx for ONTAP 프로비저닝에 사용되는 동일한 VPC 서브넷에 구축됩니다
- AWS 계정.

### 1단계: Docker 설치 및 구성

데비안 기반 Linux 가상 머신에 Docker를 설치하고 구성합니다.

단계

1. 환경을 준비합니다.

```
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. Docker를 설치하고 설치를 확인합니다.

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

## 3. 필요한 Linux 그룹을 관련 사용자와 추가합니다.

먼저 Linux 시스템에 \* docker \* 그룹이 있는지 확인하십시오. 그렇지 않으면 그룹을 만들고 사용자를 추가합니다. 기본적으로 현재 셸 사용자가 그룹에 추가됩니다.

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

## 4. 새 그룹 및 사용자 정의를 활성화합니다

사용자와 함께 새 그룹을 만든 경우 정의를 활성화해야 합니다. 이렇게 하려면 Linux에서 로그아웃했다가 다시 로그인하거나 다음 명령을 실행할 수 있습니다.

```
newgrp docker
```

## 2단계: Docker 구성 설치

Docker Compose를 데비안 기반 Linux 가상 머신에 설치합니다.

단계

### 1. Docker Compose를 설치합니다.

```
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
```

2. 설치가 성공적으로 완료되었는지 확인합니다.

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

### 3단계: Docker 이미지 준비

자동화 솔루션과 함께 제공된 Docker 이미지를 추출하여 로드해야 합니다.

단계

1. 자동화 코드가 실행될 가상 머신에 솔루션 파일을 AWS\_FSxN\_BTC.zip 복사합니다.

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

input 매개 변수는 private-key.pem AWS 가상 머신 인증(EC2 인스턴스)에 사용되는 개인 키 파일입니다.

2. 솔루션 파일이 있는 올바른 폴더로 이동하여 파일의 압축을 풉니다.

```
unzip AWS_FSxN_BTC.zip
```

3. 압축 풀기 작업으로 생성된 새 폴더로 AWS\_FSxN\_BTC 이동하여 파일을 나열합니다. 파일을 볼 수 aws\_fsxn\_flexcache\_image\_latest.tar.gz 있습니다.

```
ls -la
```

4. Docker 이미지 파일을 로드합니다. 로드 작업은 일반적으로 몇 초 내에 완료됩니다.

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

5. Docker 이미지가 로드되었는지 확인합니다.

```
docker images
```

Docker aws\_fsxn\_flexcache\_image 이미지와 태그가 'latest' 표시됩니다.

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

## 4단계: AWS 자격 증명에 대한 환경 파일을 생성합니다

액세스 및 비밀 키를 사용하여 인증할 로컬 변수 파일을 만들어야 합니다. 그런 다음 파일을 `.env` 파일에 추가합니다.

단계

1. 다음 위치에 파일을 만듭니다 `awsauth.env`.

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

2. 파일에 다음 내용을 추가합니다.

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

형식은 \* 와(과 value) 사이의 공백을 제외하고 위에 표시된 것과 정확히 key 일치해야 합니다.

3. 변수를 사용하여 절대 파일 경로를 파일에 `AWS_CREDS` 추가합니다 `.env`. 예를 들면 다음과 같습니다.

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

## 5단계: 외부 볼륨을 만듭니다

Terraform 상태 파일과 기타 중요한 파일이 영구적으로 유지되도록 하려면 외부 볼륨이 필요합니다. Terraform에서 워크플로우 및 배포를 실행하려면 이러한 파일을 사용할 수 있어야 합니다.

단계

1. Docker Compose 외부에서 외부 볼륨을 생성합니다.

명령을 실행하기 전에 볼륨 이름(마지막 매개 변수)을 적절한 값으로 업데이트해야 합니다.

```
docker volume create aws_fsxn_volume
```

2. 다음 명령을 사용하여 외부 볼륨의 경로를 환경 파일에 추가합니다 `.env`.

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

기존 파일 내용과 콜론 서식을 유지해야 합니다. 예를 들면 다음과 같습니다.

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

대신 다음과 같은 명령을 사용하여 NFS 공유를 외부 볼륨으로 추가할 수 있습니다.

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

3. Terraform 변수를 업데이트합니다.

- a. 폴더로 `aws\_fsxn\_variables` 이동합니다.
- b. 다음 두 파일이 있는지 terraform.tfvars 확인합니다. 및 variables.tf.
- c. 사용자 환경에 필요한 대로 의 값을 terraform.tfvars 업데이트합니다.

자세한 내용은 을 ["Terraform 리소스: AWS\\_FSX\\_ONTAP\\_FILE\\_SYSTEM"](#) 참조하십시오.

## 6단계: NetApp ONTAP 관리 및 FlexCache Amazon FSx for NetApp ONTAP 프로비저닝

NetApp ONTAP 관리 및 FlexCache 를 Amazon FSx for NetApp ONTAP 프로비저닝할 수 있습니다.

### 단계

1. 폴더 루트(AWS\_FSXN\_BTC)로 이동하여 provisioning 명령을 실행합니다.

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

이 명령은 두 개의 컨테이너를 만듭니다. 첫 번째 컨테이너에서 FSx for ONTAP를 구축하고 두 번째 컨테이너에서 클러스터 피어링, SVM 피어링, 타겟 볼륨 및 FlexCache를 생성합니다.

2. 프로비저닝 프로세스를 모니터링합니다.

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

이 명령은 실시간으로 출력을 제공하지만 파일을 통해 로그를 캡처하도록 deployment.log 구성되었습니다. 파일을 편집하고 변수를 업데이트하여 DEPLOYMENT\_LOGS 이러한 로그 파일의 이름을 변경할 수 .env 있습니다.

## 7단계: NetApp ONTAP 관리 및 FlexCache Amazon FSx for NetApp ONTAP 삭제

선택적으로 Amazon FSx for NetApp ONTAP 관리 및 FlexCache 삭제하고 제거할 수 있습니다.

1. terraform.tfvars `파일의 변수를 `flexcache\_operation "destroy"로 설정합니다.
2. 루트 폴더(AWS\_FSXN\_BTC)로 이동하여 다음 명령을 실행합니다.

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

이 명령은 두 개의 컨테이너를 만듭니다. 첫 번째 컨테이너에서 FlexCache을 삭제하고 두 번째 컨테이너에서 FSx for ONTAP을 삭제합니다.

3. 프로비저닝 프로세스를 모니터링합니다.

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

# Amazon FSx for NetApp ONTAP 관리 - 재해 복구

Amazon FSx for NetApp ONTAP 관리를 사용하여 소스 시스템의 재해 복구 백업을 수행하는 데 이 자동화 솔루션을 사용할 수 있습니다.



Amazon FSx for NetApp ONTAP 관리 기능은 \*FSx for ONTAP\*라고도 합니다.

살펴봅니다

이 솔루션과 함께 제공되는 자동화 코드는 개략적으로 다음과 같은 작업을 수행합니다.

- 대상 FSx for ONTAP 파일 시스템을 프로비저닝합니다
- 파일 시스템에 대한 SVM(Storage Virtual Machine)을 프로비저닝합니다
- 소스 시스템과 타겟 시스템 간에 클러스터 피어링 관계를 생성합니다
- 소스 시스템과 SnapMirror용 타겟 시스템 간에 SVM 피어링 관계를 생성합니다
- 대상 볼륨을 생성합니다
- 소스 볼륨 및 타겟 볼륨 간에 SnapMirror 관계를 생성합니다
- 소스 볼륨과 타겟 볼륨 간에 SnapMirror 전송을 시작한다

자동화는 아래 설명과 같이 Linux 가상 머신에 설치해야 하는 Docker 및 Docker Compose를 기반으로 합니다.

시작하기 전에

프로비저닝 및 구성을 완료하려면 다음이 있어야 합니다.

- 다운로드가 필요합니다 "[Amazon FSx for NetApp ONTAP 관리 - 재해 복구](#)" NetApp Console 웹 UI를 통한 자동화 솔루션. 솔루션은 다음과 같이 패키징됩니다. FSxN\_DR.zip. 이 zip에는 다음이 포함되어 있습니다. AWS\_FSxN\_Bck\_Prov.zip 이 문서에 설명된 솔루션을 배포하는 데 사용할 파일입니다.
- 소스 시스템과 대상 시스템 간의 네트워크 연결
- 다음과 같은 특성을 가진 Linux VM:
  - 데비안 기반 리눅스 배포판
  - FSx for ONTAP 프로비저닝에 사용되는 동일한 VPC 서브넷에 구축됩니다
- AWS 계정.

## 1단계: Docker 설치 및 구성

데비안 기반 Linux 가상 머신에 Docker를 설치하고 구성합니다.

단계

1. 환경을 준비합니다.



```
sudo apt-get update
sudo apt-get install apt-transport-https ca-certificates curl gnupg-
agent softwareproperties-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. Docker를 설치하고 설치를 확인합니다.

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

## 3. 필요한 Linux 그룹을 관련 사용자와 추가합니다.

먼저 Linux 시스템에 \* docker \* 그룹이 있는지 확인하십시오. 그룹이 없는 경우 그룹을 만들고 사용자를 추가합니다. 기본적으로 현재 셸 사용자가 그룹에 추가됩니다.

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

## 4. 새 그룹 및 사용자 정의를 활성화합니다

사용자와 함께 새 그룹을 만든 경우 정의를 활성화해야 합니다. 이렇게 하려면 Linux에서 로그아웃했다가 다시 로그인하거나 다음 명령을 실행할 수 있습니다.

```
newgrp docker
```

## 2단계: Docker 구성 설치

Docker Compose를 데비안 기반 Linux 가상 머신에 설치합니다.

단계

### 1. Docker Compose를 설치합니다.

```
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
```

2. 설치가 성공적으로 완료되었는지 확인합니다.

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

### 3단계: Docker 이미지 준비

자동화 솔루션과 함께 제공된 Docker 이미지를 추출하여 로드해야 합니다.

단계

1. 자동화 코드가 실행될 가상 머신에 솔루션 파일을 AWS\_FSxN\_Bck\_Prov.zip 복사합니다.

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

input 매개 변수는 private-key.pem AWS 가상 머신 인증(EC2 인스턴스)에 사용되는 개인 키 파일입니다.

2. 솔루션 파일이 있는 올바른 폴더로 이동하여 파일의 압축을 풉니다.

```
unzip AWS_FSxN_Bck_Prov.zip
```

3. 압축 풀기 작업으로 생성된 새 폴더로 AWS\_FSxN\_Bck\_Prov 이동하여 파일을 나열합니다. 파일을 볼 수 aws\_fsxn\_bck\_image\_latest.tar.gz 있습니다.

```
ls -la
```

4. Docker 이미지 파일을 로드합니다. 로드 작업은 일반적으로 몇 초 내에 완료됩니다.

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

5. Docker 이미지가 로드되었는지 확인합니다.

```
docker images
```

Docker aws\_fsxn\_bck\_image 이미지와 태그가 'latest' 표시됩니다.

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

## 4단계: AWS 자격 증명에 대한 환경 파일을 생성합니다

액세스 및 비밀 키를 사용하여 인증할 로컬 변수 파일을 만들어야 합니다. 그런 다음 파일을 `.env` 파일에 추가합니다.

단계

1. 다음 위치에 파일을 만듭니다 `awsauth.env`.

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

2. 파일에 다음 내용을 추가합니다.

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

형식은 \* 와(과 value) 사이의 공백을 제외하고 위에 표시된 것과 정확히 key 일치해야 합니다.

3. 변수를 사용하여 절대 파일 경로를 파일에 `AWS_CREDS` 추가합니다 `.env`. 예를 들면 다음과 같습니다.

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

## 5단계: 외부 볼륨을 만듭니다

Terraform 상태 파일과 기타 중요한 파일이 영구적으로 유지되도록 하려면 외부 볼륨이 필요합니다. Terraform에서 워크플로우 및 배포를 실행하려면 이러한 파일을 사용할 수 있어야 합니다.

단계

1. Docker Compose 외부에서 외부 볼륨을 생성합니다.

명령을 실행하기 전에 볼륨 이름(마지막 매개 변수)을 적절한 값으로 업데이트해야 합니다.

```
docker volume create aws_fsxn_volume
```

2. 다음 명령을 사용하여 외부 볼륨의 경로를 환경 파일에 추가합니다 `.env`.

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

기존 파일 내용과 콜론 서식을 유지해야 합니다. 예를 들면 다음과 같습니다.

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

대신 다음과 같은 명령을 사용하여 NFS 공유를 외부 볼륨으로 추가할 수 있습니다.

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

3. Terraform 변수를 업데이트합니다.

- a. 폴더로 `aws\_fsxn\_variables` 이동합니다.
- b. 다음 두 파일이 있는지 terraform.tfvars 확인합니다. 및 variables.tf.
- c. 사용자 환경에 필요한 대로 의 값을 terraform.tfvars 업데이트합니다.

자세한 내용은 을 "[Terraform 리소스: AWS\\_FSX\\_ONTAP\\_FILE\\_SYSTEM](#)" 참조하십시오.

## 6단계: 백업 솔루션을 배포합니다

재해 복구 백업 솔루션을 배포하고 프로비저닝할 수 있습니다.

단계

1. 루트 폴더(AWS\_FSxN\_Bck\_Prov)로 이동하여 provisioning 명령을 실행합니다.

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

이 명령은 세 개의 컨테이너를 만듭니다. 첫 번째 컨테이너에서 FSx for ONTAP을 구축합니다. 두 번째 컨테이너는 클러스터 피어링, SVM 피어링 및 대상 볼륨을 생성합니다. 세 번째 컨테이너에서 SnapMirror 관계를 생성하고 SnapMirror 전송을 시작합니다.

2. 프로비저닝 프로세스를 모니터링합니다.

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

이 명령은 실시간으로 출력을 제공하지만 파일을 통해 로그를 캡처하도록 deployment.log 구성되었습니다. 파일을 편집하고 변수를 업데이트하여 DEPLOYMENT\_LOGS 이러한 로그 파일의 이름을 변경할 수 .env 있습니다.

## 저작권 정보

Copyright © 2025 NetApp, Inc. All Rights Reserved. 미국에서 인쇄된 본 문서의 어떠한 부분도 저작권 소유자의 사전 서면 승인 없이는 어떠한 형식이나 수단(복사, 녹음, 녹화 또는 전자 검색 시스템에 저장하는 것을 비롯한 그래픽, 전자적 또는 기계적 방법)으로도 복제될 수 없습니다.

NetApp이 저작권을 가진 자료에 있는 소프트웨어에는 아래의 라이선스와 고지사항이 적용됩니다.

본 소프트웨어는 NetApp에 의해 '있는 그대로' 제공되며 상품성 및 특정 목적에의 적합성에 대한 명시적 또는 묵시적 보증을 포함하여(이에 제한되지 않음) 어떠한 보증도 하지 않습니다. NetApp은 대체품 또는 대체 서비스의 조달, 사용 불능, 데이터 손실, 이익 손실, 영업 중단을 포함하여(이에 국한되지 않음), 이 소프트웨어의 사용으로 인해 발생하는 모든 직접 및 간접 손해, 우발적 손해, 특별 손해, 징벌적 손해, 결과적 손해의 발생에 대하여 그 발생 이유, 책임론, 계약 여부, 엄격한 책임, 불법 행위(과실 또는 그렇지 않은 경우)와 관계없이 어떠한 책임도 지지 않으며, 이와 같은 손실의 발생 가능성이 통지되었다 하더라도 마찬가지입니다.

NetApp은 본 문서에 설명된 제품을 언제든지 예고 없이 변경할 권리를 보유합니다. NetApp은 NetApp의 명시적인 서면 동의를 받은 경우를 제외하고 본 문서에 설명된 제품을 사용하여 발생하는 어떠한 문제에도 책임을 지지 않습니다. 본 제품의 사용 또는 구매의 경우 NetApp에서는 어떠한 특허권, 상표권 또는 기타 지적 재산권이 적용되는 라이선스도 제공하지 않습니다.

본 설명서에 설명된 제품은 하나 이상의 미국 특허, 해외 특허 또는 출원 중인 특허로 보호됩니다.

제한적 권리 표시: 정부에 의한 사용, 복제 또는 공개에는 DFARS 252.227-7013(2014년 2월) 및 FAR 52.227-19(2007년 12월)의 기술 데이터-비상업적 품목에 대한 권리(Rights in Technical Data -Noncommercial Items) 조항의 하위 조항 (b)(3)에 설명된 제한사항이 적용됩니다.

여기에 포함된 데이터는 상업용 제품 및/또는 상업용 서비스(FAR 2.101에 정의)에 해당하며 NetApp, Inc.의 독점 자산입니다. 본 계약에 따라 제공되는 모든 NetApp 기술 데이터 및 컴퓨터 소프트웨어는 본질적으로 상업용이며 개인 비용만으로 개발되었습니다. 미국 정부는 데이터가 제공된 미국 계약과 관련하여 해당 계약을 지원하는 데에만 데이터에 대한 전 세계적으로 비독점적이고 양도할 수 없으며 재사용이 불가능하며 취소 불가능한 라이선스를 제한적으로 가집니다. 여기에 제공된 경우를 제외하고 NetApp, Inc.의 사전 서면 승인 없이는 이 데이터를 사용, 공개, 재생산, 수정, 수행 또는 표시할 수 없습니다. 미국 국방부에 대한 정부 라이선스는 DFARS 조항 252.227-7015(b)(2014년 2월)에 명시된 권한으로 제한됩니다.

## 상표 정보

NETAPP, NETAPP 로고 및 <http://www.netapp.com/TM>에 나열된 마크는 NetApp, Inc.의 상표입니다. 기타 회사 및 제품 이름은 해당 소유자의 상표일 수 있습니다.