



# **AWS용 Cloud Volumes ONTAP**

## **NetApp Automation**

NetApp  
November 18, 2025

# 목차

AWS용 Cloud Volumes ONTAP .....	1
AWS용 Cloud Volumes ONTAP - 클라우드 사용 급증 .....	1
1단계: Docker 및 Docker Compose를 설치합니다.....	1
2단계: Docker 이미지 준비.....	2
3단계: 환경 변수 파일을 만듭니다.....	4
4단계: NetApp Intelligent Services에 가입하세요 .....	4
5단계: 외부 볼륨을 만듭니다.....	5
6단계: AWS 환경을 위한 Cloud Volumes ONTAP를 구축합니다.....	6

# AWS용 Cloud Volumes ONTAP

## AWS용 Cloud Volumes ONTAP - 클라우드 사용 급증

이 문서에서는 NetApp 고객이 NetApp Console 자동화 허브에서 사용할 수 있는 NetApp Cloud Volumes ONTAP for AWS 자동화 솔루션을 지원합니다.

AWS 환경을 위한 Cloud Volumes ONTAP 자동화 솔루션은 Terraform을 사용하여 Cloud Volumes ONTAP for AWS 환경을 컨테이너화된 구축을 자동화하므로 수동 개입 없이 AWS용 Cloud Volumes ONTAP를 빠르게 구축할 수 있습니다.

시작하기 전에

- 다운로드해야 합니다 ["Cloud Volumes ONTAP AWS - 클라우드 사용 급증"](#) 콘솔 웹 UI를 통한 자동화 솔루션. 솔루션은 다음과 같이 패키징됩니다. `cvo_aws_flexcache.zip`.
- Cloud Volumes ONTAP와 동일한 네트워크에 Linux VM을 설치해야 합니다.
- Linux VM을 설치한 후에는 이 솔루션의 단계에 따라 필요한 종속성을 설치해야 합니다.

### 1단계: Docker 및 Docker Compose를 설치합니다

Docker를 설치합니다

다음 단계에서는 Ubuntu 20.04 Debian Linux 배포 소프트웨어를 예로 사용합니다. 실행하는 명령은 사용 중인 Linux 배포 소프트웨어에 따라 다릅니다. 해당 구성에 대한 자세한 내용은 해당 Linux 배포 소프트웨어 설명서를 참조하십시오.

단계

1. 다음 명령을 실행하여 Docker를 `sudo` 설치합니다.

```
sudo apt-get update
sudo apt-get install apt-transport-https cacertificates 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
sudo apt-get install docker-ce docker-ce-cli containerd.io
```

2. 설치를 확인합니다.

```
docker -version
```

3. Linux 시스템에 "docker"라는 그룹이 생성되었는지 확인합니다. 필요한 경우 그룹을 생성합니다.

```
sudo groupadd docker
```

4. Docker에 액세스해야 하는 사용자를 그룹에 추가합니다.

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

5. 변경 사항은 로그아웃한 후 터미널에 다시 로그인하면 적용됩니다. 또는 변경 사항을 즉시 적용할 수 있습니다.

```
newgrp docker
```

## Docker Compose를 설치합니다

### 단계

1. 다음 명령을 실행하여 Docker Compose를 sudo 설치합니다.

```
sudo curl -L
"https://github.com/docker/compose/releases/download/1.29.2/docker-
compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose

sudo chmod +x /usr/local/bin/docker-compose
```

2. 설치를 확인합니다.

```
docker-compose -version
```

## 2단계: Docker 이미지 준비

### 단계

1. `cvo\_aws\_flexcache.zip` Cloud Volumes ONTAP를 배포하는 데 사용할 Linux VM에 폴더를 복사합니다.

```
scp -i ~/<private-key>.pem -r cvo_aws_flexcache.zip
<awsuser>@<IP_ADDRESS_OF_VM>:<LOCATION_TO_BE_COPIED>
```

- private-key.pem 은(는) 암호 없이 로그인할 수 있는 개인 키 파일입니다.
- awsuser VM 사용자 이름입니다.
- IP\_ADDRESS\_OF\_VM VM IP 주소입니다.
- LOCATION\_TO\_BE\_COPIED 폴더가 복사될 위치입니다.

- 폴더 압축을 `cvo_aws_flexcache.zip` 품니다. 현재 디렉터리나 사용자 지정 위치에서 폴더의 압축을 풀 수 있습니다.

현재 디렉터리에서 폴더를 추출하려면 다음을 실행합니다.

```
unzip cvo_aws_flexcache.zip
```

사용자 지정 위치에서 폴더를 추출하려면 다음을 실행합니다.

```
unzip cvo_aws_flexcache.zip -d ~/<your_folder_name>
```

- 콘텐츠의 압축을 푼 후 폴더로 `CVO_Aws_Deployment` 이동하고 다음 명령을 실행하여 파일을 봅니다.

```
ls -la
```

다음 예제와 유사한 파일 목록이 표시됩니다.

```
total 32
drwxr-xr-x  8 user1  staff   256 Mar 23 12:26 .
drwxr-xr-x  6 user1  staff   192 Mar 22 08:04 ..
-rw-r--r--  1 user1  staff   324 Apr 12 21:37 .env
-rw-r--r--  1 user1  staff  1449 Mar 23 13:19 Dockerfile
drwxr-xr-x 15 user1  staff   480 Mar 23 13:19 cvo_Aws_source_code
drwxr-xr-x  4 user1  staff   128 Apr 27 13:43 cvo_Aws_variables
-rw-r--r--  1 user1  staff   996 Mar 24 04:06 docker-compose-
deploy.yml
-rw-r--r--  1 user1  staff  1041 Mar 24 04:06 docker-compose-
destroy.yml
```

- 파일을 찾습니다 `cvo_aws_flexcache_ubuntu_image.tar`. Cloud Volumes ONTAP for AWS를 구축하는 데 필요한 Docker 이미지가 포함되어 있습니다.
- 파일의 압축을 품니다.

```
docker load -i cvo_aws_flexcache_ubuntu_image.tar
```

- Docker 이미지가 로드될 때까지 몇 분 기다린 다음 Docker 이미지가 성공적으로 로드되었는지 확인합니다.

```
docker images
```

다음 예와 같이 태그와 함께 `latest` 이름이 지정된 Docker 이미지가 표시됩니다

cvo\_aws\_flexcache\_ubuntu\_image.

REPOSITORY SIZE	TAG	IMAGE ID	CREATED
cvo_aws_flexcache_ubuntu_image 1.14GB	latest	18db15a4d59c	2 weeks ago



필요한 경우 Docker 이미지 이름을 변경할 수 있습니다. Docker 이미지 이름을 변경하는 경우 및 `docker-compose-destroy` 파일에서 Docker 이미지 이름을 업데이트해야 `docker-compose-deploy` 합니다.

### 3단계: 환경 변수 파일을 만듭니다

이 단계에서는 두 개의 환경 변수 파일을 만들어야 합니다. 한 파일은 AWS 액세스 키와 비밀 키를 사용하여 AWS Resource Manager API를 인증하기 위한 것입니다. 두 번째 파일은 콘솔 Terraform 모듈이 AWS API를 찾아 인증할 수 있도록 환경 변수를 설정하는 데 사용됩니다.

단계

1. 다음 위치에 파일을 만듭니다 `awsauth.env`.

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

- a. 파일에 다음 내용을 `awsauth.env` 추가합니다.

`access_key=<> secret_key=<>`

형식은 \* 위에 표시된 것과 정확히 일치해야 합니다.

2. 절대 파일 경로를 `.env` 파일에 추가합니다.

환경 변수에 해당하는 환경 파일의 `AWS_CREDS` 절대 경로를 `awsauth.env` 입력합니다.

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

3. ``cvo_aws_variable``폴더로 이동하여 자격 증명 파일에서 액세스 및 비밀 키를 업데이트합니다.

파일에 다음 내용을 추가합니다.

`AWS_ACCESS_KEY_ID = <> AWS_SYSTEM_ACCESS_KEY=<>`

형식은 \* 위에 표시된 것과 정확히 일치해야 합니다.

### 4단계: NetApp Intelligent Services에 가입하세요

클라우드 공급업체를 통해 NetApp Intelligent Services에 가입하여 시간당 요금(PAYGO)을 지불하거나 연간 계약을 맺으세요. NetApp Intelligent Services에는 NetApp 백업 및 복구, Cloud Volumes ONTAP, NetApp 클라우드 계층화, NetApp 랜섬웨어 복원력 및 NetApp 재해 복구가 포함됩니다. NetApp 데이터 분류는 추가 비용 없이 구독에 포함되어 있습니다.

## 단계

1. Amazon Web Services(AWS) 포털에서 \*SaaS\*로 이동하여 \* NetApp Intelligent Services 구독\*을 선택합니다.

Cloud Volumes ONTAP와 동일한 리소스 그룹을 사용하거나 다른 리소스 그룹을 사용할 수 있습니다.

2. NetApp 콘솔 포털을 구성하여 SaaS 구독을 콘솔로 가져옵니다.

이 구성은 AWS 포털에서 직접 구성할 수 있습니다.

구성을 확인하기 위해 콘솔 포털로 리디렉션됩니다.

3. 콘솔 포털에서 \*저장\*을 선택하여 구성을 확인합니다.

## 5단계: 외부 볼륨을 만듭니다

Terraform 상태 파일과 기타 중요한 파일을 영구적으로 유지하려면 외부 볼륨을 생성해야 합니다. Terraform에서 워크플로 및 배포를 실행하려면 파일을 사용할 수 있는지 확인해야 합니다.

## 단계

1. Docker 외부에서 외부 볼륨 생성 Compose:

```
docker volume create <volume_name>
```

예:

```
docker volume create cvo_aws_volume_dst
```

2. 다음 옵션 중 하나를 사용합니다.

- a. 환경 파일에 외부 볼륨 경로를 추가합니다 .env.

아래 표시된 형식을 정확히 따라야 합니다.

형식:

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

예:

```
PERSISTENT_VOL=cvo_aws_volume_dst:/cvo_aws
```

- b. NFS 공유를 외부 볼륨으로 추가합니다.

Docker 컨테이너가 NFS 공유와 통신할 수 있으며 읽기/쓰기와 같은 올바른 권한이 구성되어 있는지 확인합니다.

- i. 다음과 같이 NFS 공유 경로를 Docker Compose 파일의 외부 볼륨에 대한 경로로 추가합니다. 형식:

```
PERSISTENT_VOL=path/to/nfs/volume:/cvo_aws
```

예:

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

3. `cvo\_aws\_variables`폴더로 이동합니다.

폴더에 다음 변수 파일이 표시됩니다.

- terraform.tfvars
- variables.tf

4. 요구 사항에 따라 파일 내의 값을 terraform.tfvars 변경합니다.

파일의 변수 값을 수정할 때는 특정 지원 문서를 읽어야 terraform.tfvars 합니다. 값은 지역, 가용 영역 및 Cloud Volumes ONTAP for AWS에서 지원하는 기타 요인에 따라 달라질 수 있습니다. 여기에는 단일 노드에 대한 라이선스, 디스크 크기, VM 크기 및 고가용성(HA) 쌍이 포함됩니다.

Console 에이전트 및 Cloud Volumes ONTAP Terraform 모듈에 대한 모든 지원 변수는 이미 정의되어 있습니다. variables.tf 파일. 변수 이름을 참조해야 합니다. variables.tf 추가하기 전에 파일 terraform.tfvars 파일.

5. 요구 사항에 따라 다음 옵션을 또는 false 로 설정하여 FlexCache 및 FlexClone를 활성화하거나 비활성화할 수 true 있습니다.

다음 예에서는 FlexCache 및 FlexClone를 사용합니다.

- is\_flexcache\_required = true
- is\_flexclone\_required = true

## 6단계: AWS 환경을 위한 Cloud Volumes ONTAP를 구축합니다

다음 단계에 따라 Cloud Volumes ONTAP for AWS를 구축하십시오.

단계

1. 루트 폴더에서 다음 명령을 실행하여 배포를 트리거합니다.

```
docker-compose -f docker-compose-deploy.yml up -d
```

두 개의 컨테이너가 트리거되고 첫 번째 컨테이너가 Cloud Volumes ONTAP를 배포하며 두 번째 컨테이너가 원격 측정 데이터를 AutoSupport로 전송합니다.

두 번째 컨테이너는 첫 번째 컨테이너가 모든 단계를 성공적으로 완료할 때까지 대기합니다.

2. 로그 파일을 사용하여 배포 프로세스의 진행 상황을 모니터링합니다.

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

이 명령은 실시간으로 출력을 제공하고 다음 로그 파일에 데이터를 캡처합니다.  
deployment.log

```
telemetry_asup.log
```

다음 환경 변수를 사용하여 파일을 편집하여 이러한 로그 파일의 이름을 변경할 수 .env 있습니다.

```
DEPLOYMENT_LOGS
```

```
TELEMETRY_ASUP_LOGS
```

다음 예제에서는 로그 파일 이름을 변경하는 방법을 보여 줍니다.

```
DEPLOYMENT_LOGS=<your_deployment_log_filename>.log
```

```
TELEMETRY_ASUP_LOGS=<your_telemetry_asup_log_filename>.log
```

작업을 마친 후

다음 단계에 따라 임시 환경을 제거하고 배포 프로세스 중에 만든 항목을 정리할 수 있습니다.

단계

1. FlexCache를 배포한 경우 변수 파일에서 다음 옵션을 terraform.tfvars 설정하면 FlexCache 볼륨이 정리되고 이전에 생성된 임시 환경이 제거됩니다.

```
flexcache_operation = "destroy"
```



가능한 옵션은 `deploy` 및 `destroy`입니다.

2. FlexClone를 배포한 경우 변수 파일에서 다음 옵션을 terraform.tfvars 설정하면 FlexClone 볼륨이 정리되고 이전에 생성된 임시 환경이 제거됩니다.

```
flexclone_operation = "destroy"
```



가능한 옵션은 deploy 및 `destroy`입니다.

## 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.